



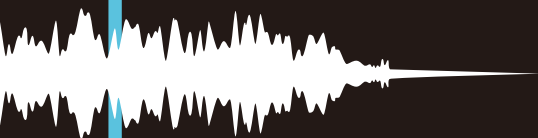
# CMMR

## 2023 TOKYO

The 16th International Symposium on  
Computer Music Multidisciplinary Research

Music: Bridge after the turmoil

**13 - 17 November 2023**





Proceedings of the

**16th International Symposium on  
Computer Music Multidisciplinary Research**

13 – 17th November, 2023  
Tokyo, Japan

CMMR 2023 Organizing Committee, Japan

in collaboration with

The Laboratory PRISM  
“Perception, Representations, Image, Sound, Music”  
Marseille, France





Published by

CMMR 2023 Organizing Committee, Japan

in collaboration with

The Laboratory PRISM

“Perception, Representations, Image, Sound, Music”

Marseille, France

November, 2023

All copyrights remain with the authors.

Proceedings Editors: T. Kitahara, M. Aramaki,

R. Kronland-Martinet, S. Ystad

ISBN 979-10-97498-04-7

Les éditions de PRISM



## **Welcome Message from General Chair**

We are pleased to welcome you to the 16th edition of CMMR which will be held as a face-to-face event. CMMR 2023 will be held in Tokyo in November 2023 jointly organized by the CMMR 2023 executive committee and the CNRS - Laboratoire de Mécanique et d'Acoustique, Marseille, France, and will be the first face-to-face social lively activity for CMMR in four years. All presentations and conference events will take place on-site. We hope that by participating in CMMR 2023 and actively interacting with each other, you will be able to intensively exchange ideas, gain rich inspiration, and make great progress in your research.

Since there has been a long and winding road to the in-person CMMR 2023, I would like to take a moment to reflect on it here. Originally, the CMMR 2020 Committee was formed in the spring of 2019 to prepare for CMMR 2020 to be held in November in Tokyo. However, CMMR 2020 was cancelled and postponed for one year because the corona pandemic showed no sign of abating at that time. The call for papers for CMMR 2021 was issued in January 2021 without it having been decided whether to hold the conference in-person or online, but in April of the same year the decision was made to hold CMMR 2021 completely online after all. In CMMR 2021 held in November, we had 33 technical papers, 13 musical works, and 264 participants from around the world. However, the experience of organizing and managing CMMR 2021, which was a completely online international conference, provided an opportunity to realize the irreplaceable value of holding international conferences in-person.

We had been struggling with the global corona disaster for almost three years. At last, I thought we came to the point where we could foresee the end of COVID-19 infection. We considered that if CMMR were to be held in November, 2023, by that time the world would be steadily recovering and bustling with activity. Therefore, the CMMR 2023 Committee was formed in the spring of 2022, with the aim of holding CMMR 2023 in-person, in Tokyo. Most of the CMMR 2021 Committee members agreed with the intentions and so joined the CMMR 2023 Committee. In this way, the CMMR 2023 began its activities. Incidentally, three years of constrained online activity has conversely led us to notice a new format of participation in international conferences with wider coverage and greater flexibility. Thus, CMMR 2023 aims at realizing a CMMR that combines a face-to-face international conference with online broadcasting service to the audience around the world.

The conference theme established for CMMR 2023 is "Music: Bridge after the turmoil." Here, the bridge has several meanings. Firstly, it is a bridge that connects researchers who have been divided by the Corona disaster. Also, it is a bridge connecting the multidisciplinary research fields that CMMR is aiming for, and a bridge between the CMMR's of the past and the CMMR's of the future. Then, music, which is the subject of CMMR, is also a bridge to bring different things together, a bridge to unite things that are far apart, and a bridge to overcome academic challenges. By considering music as a bridge in this way, we can certainly deepen the multidisciplinary research centered

on music and informatics. I look forward to discussing music as a bridge with you all at CMMR 2023.

CMMR 2023 is delighted to include three keynote speakers who will deliver speeches based on the conference theme “Music: Bridge after the turmoil”; Dr. Shigeki Sagayama (Visiting Professor, Univ. of Electro-Communications, Japan), Dr. Yi-Hsuan Yang (Full Professor, the College of Electrical Engineering and Computer Science, National Taiwan University), and Dr. Tatsuya Daikoku (Project Assistant Professor, International Research Center for Neurointelligence, The University of Tokyo, Japan). These three keynote talks will surely open the door to new multidisciplinary research for individual participants. In addition, we are honored to offer three special sessions that are timely and in keeping with the conference theme: Singing information processing organized by Dr. Tomoyasu Nakano (National Institute of Advanced Industrial Science and Technology (AIST)), Music and Sound Generation: Emerging Approaches and Diverse Applications organized by Dr. Taketo Akama (Sony Computer Science Laboratories, Inc.), and Computational Research on Music Evolution organized by Dr. Eita Nakamura (Kyoto University).

CMMR 2023 is grateful to the following association and companies for their financial support:

- Distinguished Sponsor: Special Interest Group on Music and Computer (SIGMUS, IPSJ),
- Gold Sponsor: Piano Teacher’s National Association of Japan (PTNA),
- Silver Sponsor: Yamaha Corporation, and
- Commercial Sponsor: Crypton Future Media, Inc.

Thanks to their support, we are able to hold the productive, impressive, and well-organized international conference. We would like to express our deepest gratitude to all of them.

On behalf of the CMMR 2023 Committee, I hope that many of you will be reunited with colleagues and that newcomers will also join us, sparking lively discussions and embarking on new research journeys.

Keiji Hirata  
General Chair of CMMR 2023

## **Message from Scientific Program Chairs**

We would like to thank you for attending the 16th International Symposium on Computer Music and Multidisciplinary Research (CMMR 2023).

When we decided to hold CMMR 2020 in Japan, we set the following three goals:

- To let worldwide researchers, engineers, and musicians engaged in the computer music field gather and communicate with each other in the face-to-face manner.
- To let participants enjoy staying in Japan.
- To let participants understand the high activity of the Japanese computer music research community.

Because of COVID-19, however, we had to postpone the CMMR 2020 by one year, and ended up holding CMMR 2021 online. Therefore, we could not achieve any goals mentioned above. On this occasion, we are honored to hold CMMR, for the first time in the Far East, in person.

To rebirth CMMR as a face-to-face conference, we set the conference theme to “Music: Bridge after the Turmoil” (The general chair will explain the meaning of this theme in Welcome Message from General Chair). To encourage face-to-face communication, we have prepared three presentation formats (oral, poster, and demo) and have asked all presenters to participate onsite, avoiding a hybrid form. We received more than 80 long/short papers and 30 demo papers from Asian, European, and American countries. Accepted long and short papers were allocated to oral or poster sessions according to the authors’ preferences and the reviewers’ recommendations. As a result, 12 oral sessions (including special sessions), 3 poster sessions, and 3 demo sessions will be organized at the conference.

We hope that all of you stay safely in Tokyo and enjoy participating in CMMR 2023.

On behalf of the CMMR 2023 Scientific Program Chairs,  
Tetsuro Kitahara

## **Organization**

### **General Chair**

Keiji Hirata (Future University Hakodate, Japan)

### **General Co-Chair**

Satoshi Tojo (Asia University, Japan)

### **Scientific Program Chairs**

Tetsuro Kitahara (Nihon University, Japan)

Mitsuko Aramaki (AMU-CNRS-PRISM, France)

Richard Kronland-Martinet (AMU-CNRS-PRISM, France)

Sølvi Ystad (AMU-CNRS-PRISM, France)

### **Music Program Chair**

Takuro Shibayama (Tokyo Denki University, Japan)

### **Demo Program Chair**

Masatoshi Hamanaka (RIKEN, Japan)

### **Public Relation Chairs**

Masatoshi Hamanaka (RIKEN, Japan)

Shun Sawada (Tokyo University of Science, Japan)

### **Treasurer**

Masaki Matsubara (University of Tsukuba, Japan)

### **Proceedings Chair/Registration Chair**

Aiko Uemura (Nihon University, Japan)

### **Sponsor Chair**

Masaki Matsubara (University of Tsukuba, Japan)

Yui Uehara (Kanagawa University, Japan)

### **Local Organizer**

Hidefumi Ohmura (Tokyo University of Science, Japan)

Ryo Hatano (Tokyo University of Science, Japan)

Shun Sawada (Tokyo University of Science, Japan)

### **Secretary**

Yui Uehara (Kanagawa University, Japan)

**Demo Program Committee**

Masatoshi Hamanaka (RIKEN, Japan)  
Stefano Kalonaris (RIKEN, Japan)  
Hiroya Miura (RIKEN, Japan)  
Nami Iino (NII, Japan)

**Scientific Program Committee**

Tetsuro Kitahara (Nihon University, Japan)  
Mitsuko Aramaki (AMU-CNRS-PRISM, France)  
Richard Kronland-Martinet (AMU-CNRS-PRISM, France)  
Sølvi Ystad (AMU-CNRS-PRISM, France)  
Taketo Akama (Sony Computer Science Laboratories, Inc., Japan)  
Daichi Ando (Tokyo Metropolitan University, Japan)  
Mathieu Barthet (Queen Mary University of London, UK)  
Corentin Bernard (AMU-CNRS-PRISM, France)  
Gilberto Bernardes (University of Porto, Portugal)  
Tifanie Bouchara (CNAM)  
Marcelo Caetano (AMU-CNRS-PRISM, France)  
F. Amílcar Cardoso (DEI, CISUC, University of Coimbra)  
Roger Dannenberg (Carnegie Mellon University, US)  
Georg Essl (University of Wisconsin, Milwaukee)  
Ichiro Fujinaga (McGill University, Canada)  
Mylène Gioffredo (AMU-CNRS-PRISM, France)  
Masataka Goto (National Institute of Advanced Industrial Science and Technology (AIST), Japan)  
Mitsuyo Hashida (The University of Fukuchiyama, Japan)  
Rumi Hiraga (Tsukuba University of Technology, Japan)  
Tatsunori Hirai (Komazawa University, Japan)  
Shigeyuki Hirai (Kyoto Sangyo University, Japan)  
Keiji Hirata (Future University Hakodate, Japan)  
Akinori Ito (Tohoku University, Japan)  
Takayuki Itoh (Ochanomizu University, Japan)  
Katsutoshi Itoyama (Tokyo Institute of Technology, Japan)  
Haruhiro Katayose (Kwansei Gakuin University, Japan)  
Luca Andrea Ludovico (University of Milan, Italy)  
Akira Maezawa (Yamaha Corporation, Japan)  
Sylvain Marchand (University of La Rochelle, France)  
Masaki Matsubara (University of Tsukuba, Japan)  
Andrew McPherson (Imperial College London)  
Tomohiko Nakamura (National Institute of Advanced Industrial Science and Technology (AIST), Japan)  
Eita Nakamura (Kyoto University, Japan)  
Tomoyasu Nakano (National Institute of Advanced Industrial Science and Technology (AIST), Japan)  
Juhan Nam (Korea Advanced Institute of Science and Technology, Korea)

Marco Buongiorno Nardelli (University of North Texas)  
Hidefumi Ohmura (Tokyo University of Science, Japan)  
Noriko Otani (Tokyo City University, Japan)  
Samuel Porot (AMU-CNRS-PRISM, France)  
Marcelo Queiroz (Computer Science Department – University of São Paulo)  
Jocelyn Roze (AMU-CNRS-PRISM, France)  
Shinji Sako (Nagoya Institute of Technology, Japan)  
Charles de Paiva Santana (AMU-CNRS-PRISM, France)  
Ariane Stolfi (Universidade Federal do Sul da Bahia, Brazil)  
Bob Sturm (KTH Stockholm, Sweden)  
Shinnosuke Takamichi (The University of Tokyo, Japan)  
Satoshi Tojo (Asia University, Japan)  
Yudai Tsujino (Meiji University, Japan)  
Adrien Vidal (Aix-Marseille University, AMU-CNRS-PRISM, France)  
Marcelo Wanderley (McGill University, Canada)  
Anna Xambo (De Montfort University – NIMEUK)  
Ryosuke Yamanishi (Kansai University, Japan)  
Nozomiko Yasui (National Institute of Technology, Japan)  
Kazuyoshi Yoshii (Kyoto University, Japan)

**Music Program Committee**

Takuro Shibayama (Tokyo Denki University, Japan)  
Kiyoshi Furukawa (Tokyo University of The Arts, Japan)  
Kazuko Narita (Doshisha Women’s College of Liberal Arts, Japan)  
Haruka Hirayama (Hokkaido Information University, Japan)  
Masatsune Yoshio (Showa University of Music, Japan)

**Steering Chairs**

Mitsuko Aramaki (PRISM, AMU-CNRS, France)  
Mathieu Barthet (QMUL, United Kingdom)  
Matthew Davies (INESC TEC, Portugal)  
Richard Kronland-Martinet (PRISM, AMU-CNRS, France)  
Sølvi Ystad (PRISM, AMU-CNRS, France)

## Table of Contents

### Keynote Talks

- Deep Learning-based Automatic Music Generation: An Overview.....2  
*Yi-Hsuan Yang*
- 17 Years with Automatic Music Composition System “Orpheus” .....3  
*Shigeki Sagayama*
- Exploring the Neural and Computational Basis of Statistical Learning in the Brain  
to Unravel Musical Creativity and Cognitive Individuality .....4  
*Tatsuya Daikoku*

### Long / Short Papers

#### Creative Music Systems

- Controllable Automatic Melody Composition Model across Pitch/Stress-accent  
Languages .....6  
*Takuya Takahashi, Shigeki Sagayama and Toru Nakashika*
- Design of a music recognition, encoding, and transcription online tool ..... 18  
*David Rizo, Jorge Calvo-Zaragoza, Juan C. Martínez-Sevilla, Adrián Roselló  
and Eliseo Fuentes-Martínez*
- Verse Generation by Reverse Generation Considering Rhyme and Answer in  
Japanese Rap Battles..... 30  
*Ryota Mibayashi, Takehiro Yamamoto, Kosetsu Tsukuda, Kento Watanabe,  
Tomoyasu Nakano, Masataka Goto and Hiroaki Ohshima*

#### Cognitive Science and Skill Science for Music

- Combining Vision and EMG-Based Hand Tracking for Extended Reality Musical  
Instruments.....42  
*Max Graf and Mathieu Barthelet*
- Emotional Impact of Source Localization in Music Using Machine Learning and  
EEG: a proof-of-concept study..... 54  
*Timothy Schmele, Eleonora De Filippi, Arijit Nandi, Alexandre Pereda Baños  
and Adan Garriga*
- Exploring Patterns of Skill Gain and Loss on Long-term Training and Non-training  
in Rhythm Game..... 66



*Ayane Sasaki, Mio Matsuura, Masaki Matsubara, Yoshinari Takegawa and Keiji Hirata*

**Special Session - Music and Sound Generation: Emerging Approaches and Diverse Applications 1**

Benzaiten: A Non-expert-friendly Event of Automatic Melody Generation Contest ..... 78

*Yoshitaka Tomiyama, Tetsuro Kitahara, Taro Masuda, Koki Kitaya, Yuya Matsumura, Ayari Takezawa, Tsuyoshi Odaira and Kanako Baba*

Pitch Class and Octave-Based Pitch Embedding Training Strategies for Symbolic Music Generation ..... 86

*Yuqiang Li, Shengchen Li and George Fazekas*

VaryNote: A Method to Automatically Vary the Number of Notes in Symbolic Music ..... 98

*Juan M. Huerta, Bo Liu and Peter Stone*

**Special Session - Music and Sound Generation: Emerging Approaches and Diverse Applications 2**

ShredGP: Guitarist Style-Conditioned Tablature Generation with Transformers ..... 110

*Pedro Sarmiento, Adarsh Kumar, Dekun Xie, CJ Carr, Zack Zukowski and Mathieu Barthelet*

ProgGP: From GuitarPro Tablature Neural Generation To Progressive Metal Production ..... 122

*Jackson Loth, Pedro Sarmiento, CJ Carr, Zack Zukowski and Mathieu Barthelet*

Reconstructing Human Expressiveness in Piano Performances with a Transformer Network ..... 134

*Jingjing Tang, Geraint Wiggins and György Fazekas*

**Poster Session 1**

Effective Textual Feedback in Musical Performance Education: A Quantitative Analysis Across Oboe, Piano, and Guitar ..... 146

*Rina Kagawa, Nami Iino, Hideaki Takeda and Masaki Matsubara*

A Melody Input Support Interface by Presenting Subsequent Candidates based on a Connection Cost ..... 158

*Tatsunori Hirai*

Phoneme-inspired playing technique representation and its alignment method for electric bass database ..... 170

*Junya Koguchi and Masanori Morise*

An Audio-to-Audio Approach to Generate Bass Lines from Guitar's Chord Backing ..... 178

*Tomoo Kouzai and Tetsuro Kitahara*

Teaching Chorale Generation Model to Avoid Parallel Motions ..... 186

*Eunjin Choi, Hyerin Kim, Juhan Nam and Dasaem Jeong*

DiffVel: Note-Level MIDI Velocity Estimation for Piano Performance by A Double Conditioned Diffusion Model ..... 197

*Hyon Kim and Xavier Serra*

8+8=4: Formalizing Time Units to Handle Symbolic Music Durations ..... 209

*Emmanouil Karystinaios, Francesco Foscarin, Florent Jacquemard, Masahiko Sakai, Satoshi Tojo and Gerhard Widmer*

Soundscape4DEI as a Model for Multilayered Sonifications ..... 221

*João Neves, Pedro Martins, F. Amílcar Cardoso, Jónatas Manzolli, Mariana Seïça and M. Zenha Rela*

### **Computational Musicology 1**

Interpretable Rule Learning and Evaluation of Early Twentieth-century Music Styles ..... 233

*Christofer Julio, Feng-Hsu Lee and Li Su*

Toward empirical analysis for stylistic expression in piano performance ..... 245

*Yu-Fen Huang and Li Su*

SANGEET: A XML based Open Dataset for Research in Hindustani Sangeet... 257

*Chandan Misra and Swarup Chattopadhyay*

### **Special Session - Music and Sound Generation: Emerging Approaches and**

#### **Diverse Applications 3**

JAZZVAR: A Dataset of Variations found within Solo Piano Performances of Jazz Standards for Music Overpainting ..... 265

*Eleanor Row, Jingjing Tang and György Fazekas*

A Live Performance Rule System informed by Irish Traditional Dance Music .. 277

*Marco Amerotti, Steve Benford, Bob L. T. Sturm and Craig Vear*

VERSNIZ - Audiovisual Worldbuilding through Live Coding as a Performance Practice in the Metaverse ..... 289

*Damian Dziwis*

Spatial Sampling in Mixed Reality - An Overview of Ten Years of Research and Creation..... 301

*Grégory Beller, Jacob Sello, Georg Hajdu and Thomas Görne*

### **HCI in Music**

Networked performance as a space for collective creation and student engagement ..... 313

*Hans Kretz*

eLabOrate(D): An Exploration of Human/Machine Collaboration in a Telematic Deep Listening Context ..... 323

*Rory Hoy and Doug Van Nort*

Estimating Interaction Time in Music Notation Editors..... 335

*Matthias Nowakowski and Aristotelis Hadjakos*

### **Poster Session 2**

Human-Swarm Interactive Music Systems: Design, Algorithms, Technologies, and Evaluation ..... 347

*Pedro Lucas and Kyrre Glette*

Improving Instrumentality of Sound Collage Using CNMF Constraint Model... 359

*Sora Miyaguchi, Naotoshi Osaka and Yusuke Ikeda*

Quantum Circuit Design using Genetic Algorithm for Melody Generation with Quantum Computing..... 367

*Tatsunori Hirai*

Automated Arrangements of Multi-Part Music for Sets of Monophonic Instruments..... 379

*Matthew McCloskey, Gabrielle Curcio, Amulya Badineni, Kevin McGrath, Georgios Papamichail and Dimitris Papamichail*

Automatic Orchestration of Piano Scores for Wind Bands with User-Specified Instrumentation..... 387

*Takuto Nabeoka, Eita Nakamura and Kazuyoshi Yoshii*

A quantitative evaluation of a musical performance support system utilizing a musical sophistication test battery ..... 395

*Yasumasa Yamaguchi, Taku Kawada, Toru Nagahama and Tatsuya Horita*

SBERT-based Chord Progression Estimation from Lyrics Trained with Imbalanced Data..... 403

*Mastuti Puspitasari, Takuya Takahashi, Gen Hori, Shigeki Sagayama and Toru Nakashika*

PolyDDSP: A Lightweight and Polyphonic Differentiable Digital Signal Processing Library..... 411

*Tom Baker, Ricardo Climent and Ke Chen*

The Unfinder: Finding and reminding in electronic music..... 419

*Rikard Lindell and Henrik Frisk*

**Special Session - Singing Information Processing**

Towards Potential Applications of Machine Learning in Computer-Assisted Vocal Training..... 430

*Antonia Stadler, Emilia Parada-Cabaleiro and Markus Schedl*

Effects of Convolutional Autoencoder Bottleneck Width on StarGAN-based Singing Technique Conversion..... 442

*Tung-Cheng Su, Yung-Chuan Chang and Yi-Wen Liu*

**Special Session - Computational Research on Music Evolution**

Historical Changes of Modes and their Substructure Modeled as Pitch Distributions in Plainchant from the 1100s to the 1500s..... 450

*Eita Nakamura, Tim Eipert and Fabian C. Moss*

Computational Analysis of Selection and Mutation Probabilities in the Evolution of Chord Progressions..... 462

*Eita Nakamura*

A network approach to harmonic evolution and complexity in western classical music..... 474

*Marco Buongiorno Nardelli*

On the Analysis of Voicing Novelty in Classical Piano Music ..... 484

*Halla Kim and Juyong Park*

Bipartite network analysis of the stylistic evolution of sample-based music..... 492

*Dongju Park and Juyong Park*

**Audio Signal Processing**

Algorithms for Roughness Control Using Frequency Shifting and Attenuation of Partial in Audio ..... 500

*Jeremy Hyrkas*

Bridging the Rhythmic Gap: A User-Centric Approach to Beat Tracking in Challenging Music Signals..... 512

*António Sá Pinto and Gilberto Bernardes*

**Poster Session 3**

Creating a New Lullaby Using an Automatic Music Composition System in  
Collaboration with a Musician ..... 524

*So Hirawata, Noriko Otani, Daisuke Okabe and Masayuki Numao*

Automatic Phrasing System for Expressive Performance Based on The Generative  
Theory of Tonal Music ..... 536

*Madoka Goto, Masahiko Sakai and Satoshi Tojo*

NUFluteDB: Flute Sound Dataset with Appropriate and Inappropriate Blowing  
Styles ..... 547

*Sai Oshita and Tetsuro Kitahara*

Melody Blending: A Review and an Experiment ..... 555

*Stefano Kalonaris and Omer Gold*

Balancing Musical Co-Creativity: The Case Study of Mixboard, a Mashup  
Application for Novices ..... 567

*Thomas Ottolin, Raghavasimhan Sankaranarayanan, Qinying Lei, Nitin Hugar  
and Gil Weinberg*

Global Prediction of Time-span Tree by Fill-in-the-blank Task..... 575

*Riku Takahashi, Risa Izu, Yoshinari Takegawa and Keiji Hirata*

Music Emotions in Solo Piano: Bridging the Gap Between Human Perception and  
Machine Learning..... 587

*Emilia Parada-Cabaleiro, Anton Batliner, Maximilian Schmitt, Björn Schuller  
and Markus Schedl*

Listeners' Perceived Emotions in Human vs. Synthetic Performance of  
Rhythmically Complex Musical Excerpts..... 599

*Ève Poudrier, Bryan Jacob Bell, Jason Yin Hei Lee and Craig Stuart Sapp*

**Computational Musicology 2**

deepGTTM-IV: Deep Learning Based Time-span Tree Analyzer of GTTM..... 611

*Masatoshi Hamanaka, Keiji Hirata and Satoshi Tojo*

Music and Logic: a connection between two worlds ..... 619

*Matteo Bizzarri*

**Music Information Retrieval**

A Novel Local Alignment-Based Approach to Motif Extraction in Polyphonic  
Music ..... 631

*Tiange Zhu, Danny Diamond, James McDermott, Raphaël Fournier-S'niehotta,  
Mathieu Daquin and Philippe Rigaux*

Predicting Audio Features of Background Music from Game Scenes .....	643
<i>Ryusei Hayashi and Tetsuro Kitahara</i>	
A Music Exploration Interface Based on Vocal Timbre and Pitch in Popular Music .....	655
<i>Tomoyasu Nakano, Momoka Sasaki, Mayuko Kishi, Masahiro Hamasaki, Masataka Goto and Yoshinori Hijikata</i>	
Exploring Diverse Sounds: Identifying Outliers in a Music Corpus .....	667
<i>Le Cai, Sam Ferguson, Gengfa Fang, and Hani Alshamrani</i>	

## **Demo Papers**

AR-based Guitar Strumming Learning Support System that Provides Audio Feedback by Hand Tracking.....	680
<i>Kaito Abiki, Saizo Aoyagi, Akira Hattori, Ken Honda and Tatsunori Hirai</i>	
The Demonstration of MVP Support System as an AR Realtime Pitch Feedback System .....	684
<i>Yasumasa Yamaguchi, Taku Kawada, Toru Nagahama and Tatsuya Horita</i>	
Melody Reduction for Beginners' Guitar Practice .....	688
<i>Hinata Segawa, Shunsuke Sakai and Tetsuro Kitahara</i>	
Structural Analysis of Utterances during Guitar Instruction .....	692
<i>Nami Iino, Hiroya Miura, Hideaki Takeda, Masatoshi Hamanaka and Takuichi Nishimura</i>	
Music in the Air: Creating Music from Practically Inaudible Ambient Sound....	696
<i>Ji Won Yoon and Woon Seung Yeo</i>	
Creating an interactive and accessible remote performance system with the Piano Machine .....	700
<i>Patricia Alessandrini, Constantin Basica and Prateek Verma</i>	
A Singing Toolkit: Gestural Control of Voice Synthesis, Voice Samples and Live Voice.....	704
<i>D. H. Molina Villota, C. D'Alessandro, G. Locqueville, and T. Lucas</i>	
Sonifying Players' Positional Relation in Football.....	708
<i>Masaki Okuta and Tetsuro Kitahara</i>	
Talking with Fish: an OpenCV Musical Installation.....	712
<i>Gabriel Zalles Ballivian</i>	
The Sound Morphing Toolbox: Musical Instrument Sound Modeling and Transformation Techniques.....	716
<i>Marcelo Caetano and Richard Kronland-Martinet</i>	

Morphing of Drum Loop Sound Sources Using CNN-VAE .....	720
<i>Mizuki Kawahara, Tomoo Kouzai and Tetsuro Kitahara</i>	
Generating Tablature of Polyphony Consisting of Melody and Bass Line .....	724
<i>Shunsuke Sakai, Hinata Segawa and Tetsuro Kitahara</i>	
Development of an easily-usable smartphone application for recording instrumental sounds .....	728
<i>Takanori Horibe and Masanori Morise</i>	
A Research on Music Generation by Deep-Learning including ornaments - A case study of world harp instruments- .....	732
<i>Arturo Alejandro Arzamendia Lopez, Akinori Ito and Koji Mikami</i>	
Automatic Music Composition System to Enjoy Brewing Delicious Coffee .....	736
<i>Noriko Otani, So Hirawata and Daisuke Okabe</i>	
Expressor: A Transformer Model for Expressive MIDI Performance .....	740
<i>Tolly Collins and Mathieu Barthet</i>	
Real-Time Piano Accompaniment Using Kuramoto Model for Human-Like Synchronization.....	744
<i>Kit Armstrong, Ji-Xuan Huang, Tzu-Ching Hung, Jing-Heng Huang and Yi- Wen Liu</i>	
Intuitive Control of Scraping and Rubbing Through Audio-tactile Synthesis.....	748
<i>Mitsuko Aramaki, Corentin Bernard, Richard Kronland-Martinet, Samuel Poirot and Sølvi Ystad</i>	
From jSymbolic 2 to 3: More Musical Features .....	752
<i>Cory McKay</i>	
Comparing vocoders for automatic vocal tuning .....	756
<i>D. H. Molina Villota and C. D'Alessandro</i>	
Music recognition, encoding, and transcription (MuRET) online tool demonstration .....	760
<i>David Rizo, Jorge Calvo-Zaragoza, Juan C. Martínez-Sevilla, Adrián Roselló, and Eliseo Fuentes-Martínez</i>	
Microtonal Music Dataset v1 .....	764
<i>Tatsunori Hirai, Lamo Nagasaka and Takuya Kato</i>	
Lighting Control based on Colors Associated with Lyrics at Bar Positions .....	768
<i>Shoyu Shinjo and Aiko Uemura</i>	
Melody Changing Interfaces for Melodic Morphing .....	772
<i>Masatoshi Hamanaka</i>	
Relative Representation of Time-Span Tree.....	776

<i>Risa Izu, Yoshinari Takegawa and Keiji Hirata</i>	
Zero-Shot Music Retrieval For Japanese Manga .....	780
<i>Megha Sharma and Yoshimasa Tsuruoka</i>	
Visualizing Musical Structure of House Music .....	784
<i>Justin Tomoya Wulf and Tetsuro Kitahara</i>	
<b>Author Index</b> .....	789



# **Keynote Talks**

## **Deep Learning-based Automatic Music Generation: An Overview**

Yi-Hsuan Yang

College of Electrical Engineering and Computer Science,  
National Taiwan University

This talk aims to provide a tutorial-like overview of the recent advances in deep generative models for automatic music generation. The talk has four parts. In the first part, I will briefly mention data representations for symbolic-domain and audio-domain music that have been employed by deep generative models. In the second part, I will use MIDI music generation to demonstrate the use of sequence models such as the Transformers to build the language model (LM) for symbolic-domain music, with a special focus on the modeling of the long-range temporal dependency of musical events. In the third part, moving forward to the audio domain, I will review advances in timbre synthesis, generative source separation, Mel-vocoders, and audio codec models, to demonstrate the development of audio encoders and decoders for music, capable of generating short audio excerpts of music with high fidelity and perceptual quality. In the final part, I will talk about how we can build upon technologies developed in the previous parts to create LMs for audio-domain music, and their applications to singing voice generation, accompaniment generation, as well as text-to-music in general. I will conclude the talk with a few open challenges in the field.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

## **17 Years with Automatic Music Composition System “Orpheus”**

Shigeki Sagayama

Professor Emeritus, Graduate School of Information Science and Technology, The University of Tokyo  
Visiting Researcher, Graduate School of Informatics and Engineering, The University of Electro-Communications

Our research on automatic music composition was started at the University of Tokyo in 2006 after a long experience in speech recognition and synthesis, music processing, and other related areas. Naturally, we took the probabilistic model approach toward computational melody generation to directly reflect the music-theoretic and linguistic knowledge and requirements rather than the machine learning approach to avoid collecting a enormous training data for imitating existing music pieces. To simulate well-trained human composers hardly violating music rules, we formulated melody generation as finding the safest path of state transitions in a Hidden Markov Model of time and pitch of notes so that resulted melody along the path best satisfies musical and linguistical probabilistic constraints and user’s preference while wide variety of outcome is guaranteed within correctness in academic music criteria. Model probability is empirically defined as the appropriateness mainly based on music theory of harmony and pitch-accent prosodic rules of Japanese language, and the optimal (i.e., least problematic) path is efficiently derived by a modified Viterbi algorithm similarly to speech recognition. The current web-based version (“Orpheus” ver 3) for Japanese lyrics (<https://www.orpheus-music.org/>) was launched in 2012 along with the duet generation and voice and accompaniment sounding functions and became one of most popular music composition services that created 0.7M music pieces and received 19M access count through the internet during recent 4 years and has been often introduced by media (TV, radio, net news, newspapers, books, etc.) to the public as an example of generative AI. Our 11-year experience of web-based service led us to further issues. To think of the universal melody generation model across both stress/pitch-accent languages (and hopefully tone languages in addition), 2-dimensional HMM is discussed instead of our current rhythm-tree approach. To truly assist the user’s creativity and to enhance their composition skills, we discuss the user interface for automatic composition as a composer’s workbench. Also, automatic music interpolation is discussed for mid-skilled users in music composition to interactively complete the music piece from fragments of melody, sub-melody and harmonies provided by the user. Music (particularly, highly theoretically and academically sophisticated European classical music) can be positioned anywhere between the pair of extremes of artificial intelligence such as correctness-first math formula processing and eloquence-first natural language processing to consider the future research direction (possibly, a mixture of both).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

## **Exploring the Neural and Computational Basis of Statistical Learning in the Brain to Unravel Musical Creativity and Cognitive Individuality**

Tatsuya Daikoku

Graduate School of Information Science and Technology  
The University of Tokyo

Music is ubiquitous in human culture. The interaction between music and the human brain engages various neural mechanisms that underlie learning, action, and creativity. Recent studies have suggested that “statistical learning” plays a significant role in musical creativity as well as musical acquisition. Statistical learning is an innate and implicit function of the human brain that is closely linked to brain development and the emergence of individuality. It begins early in life and plays a critical role in the creation and comprehension of music. Over time, the brain updates and constructs statistical models, with the model’s individuality changing based on the type and degree of stimulation received. However, the detailed mechanisms underlying this process are unknown.

In this talk, I will present a series of my “neural” and “computational” studies on how creativity emerges within the framework of statistical learning in the brain. Based on these interdisciplinary findings, I propose two core factors of musical creativity, including the critical insight into cognitive individuality through “reliability” of prediction and the construction of information “hierarchy” through chunking. Then, I will also introduce a neuro-inspired Hierarchical Bayesian Statistical Learning model (HBSL) that takes into account both reliability and hierarchy, mimicking the statistical learning processes of the brain. Using this model, I will demonstrate a newly devised system that visualizes the individuality of musical creativity. This study has the potential to shed light on the underlying factors that contribute to the heterogeneous nature of the innate ability of statistical learning, as well as the paradoxical phenomenon in which individuals with certain cognitive traits that impede specific types of perceptual abilities exhibit superior performance in creative contexts.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

# **Long / Short Papers**

# Controllable Automatic Melody Composition Model across Pitch/Stress-accent Languages

Takuya Takahashi<sup>1</sup>, Shigeki Sagayama<sup>1</sup> and Toru Nakashika<sup>1</sup> \*

The University of Electro-Communications, Tokyo, Japan  
{takahashi, sagayama, nakashika}@uec.ac.jp

**Abstract.** This study proposes a model for automatically composing linguistically and musically natural song melodies reflecting the linguistic characteristics of both pitch-accent (e.g., Japanese) and stress-accent (e.g., English) languages as well as user’s intentions. We have designed and provided publically, for more than 10 years, an automatic composition system (called “Orpheus”) for Japanese lyrics. Extending the principle for lyrics written in stress-accent languages, a new compositional model was constructed by introducing a melodic rhythm generator formulated by a probabilistic model considering the relationship between stress of lyrics and rhythm intensity (linguistic naturalness and music theory) and the rhythm style chosen by the user (controllability). The parameters of the proposed model can be learnt from domain knowledge without large amounts of data. In our experimental evaluation, the proposed system achieved ratings equal to or better than state-of-the-art deep learning approaches in terms of musical coherence, singability and listenability.

**Keywords:** Automatic music composition, Lyric to melody, Music theory, Linguistic naturalness for melody, Controllability

## 1 Introduction

Automatic music composition is one of the most interesting and challenging tasks in generative AI (such as ChatGPT<sup>1</sup>), as interest in generative AI has grown in recent years. How would users want to use automatic composition technologies? We believe automatic composition technologies should be an assistive tool that users can use for their creative activities so that beginners can compose music with only their intention without knowledge of composition theory which takes time to learn, and experts can gain new inspiration from the generation from AI composers. We are particularly interested in building a universal model for generating song for singing automatically based on user-given lyrics, that follows Western musical norms.

\* This work was supported by Grant-in-Aid for Scientific Research (B) No. 21H03462 from Japan Society for the Promotion of Science (JSPS).

<sup>1</sup> <https://openai.com/blog/chatgpt>

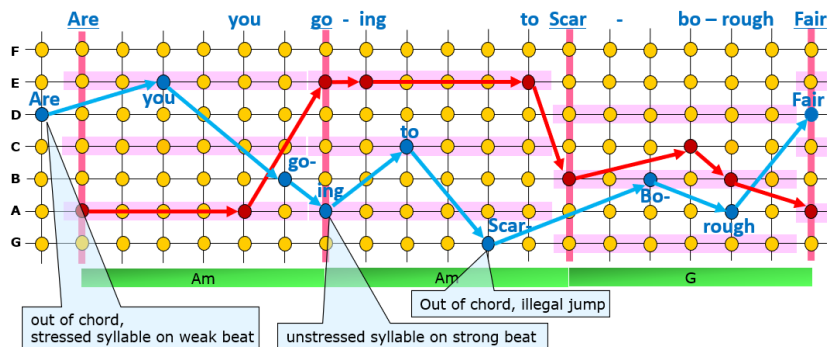


This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

State-of-the-art data-driven methods of deep learning from large amounts of data [1–3] have been proposed for automatically generating music from lyrics. Although such methods can cleverly learn patterns in the training data and generate coherent music that is close to the training data, difficulties in collecting large amounts of paired data, controllability to reflect user intent, diversity of generation (avoiding plagiarism) and musical accuracy (adherence to music theory) are often discussed. For example, Zheng et al. [4] reported owing to their subjective experiments that melodies generated by the deep learning models proposed by Sheng et al. [2] and Ju et al. [3] are difficult to sing and listen to the lyrics. Besides, DeepBach trained on Bach chorales using deep learning cannot generate pieces that adhere to music theory such as musical prohibition as Fang et al. [5] and Karatsu et al. [6] argued. Such reports may suggest that it is difficult for state-of-the-art deep learning approaches to learn singability, listenability and music theory.

Can we then rule and model the composition process in the automatic composition of songs, in addition to learning patterns from data like deep learning methods? For example, Oliveira et al. [7] investigated the relationship between stress syllables and melodic rhythm in 42 Portuguese songs and reported a correlation between stress and melodic rhythm (referred to as the stress-rhythm constraint). In an attempt to generate melodies from English lyrics, a method based on the stress-rhythm constraint and  $n$ -gram models was proposed by Monteith et al. [8]. Zhang et al. [4] report improving melodies generated by the latest deep learning methods (Sheng et al. [2], Ju et al. [3]) adjusting melody generation process based on linguistic naturalness constraints for tone languages and stress-accent languages similar to Monteith et al. [8] However, the approach of Zhang et al. [4] requires large amounts of data to train base deep models and leaves issues in terms of diversity, adherence to music theory and user controllability. We (Fukayama et al.) [9] previously proposed Orpheus, which generates the pitch of a melody based on the Japanese lyrics, music theory and user intentions (melodic rhythm, chord progression, register, etc.). As statistically demonstrated by Watanabe et al.[10], for lyrics in pitch-accent languages such as Japanese, the correlated nature of lyric prosody and the vertical movement of melodic pitch (called prosody-pitch constraints) is incorporated as a linguistic naturalness constraint in the melodic pitch generation model of Orpheus [9]. Orpheus [9] has been operating as a Web service (Orpheus v3) for more than 10 years, has over 15,000 subscribers, and has composed more than 500,000 songs. For simplicity, this approach is referred to as “Orpheus v3”.

In addition to the principle of melody generation from lyrics in pitch-accented languages in Orpheus v3, melody generation from lyrics in stress-accented languages was also expected. In this study, a model that can automatically generate a natural melody based on a given lyric written in pitch/stress-accent languages and user’s intention was realized by a combination of pitch generator from Orpheus v3 considering prosody-pitch constraints and music theory and a newly proposed rhythm generation model considering stress-rhythm constraints and music theory. Since each generator in the proposed model is formulated in probabilistic models, it can learn the probabilistic parameters from domain knowledge with explicit consideration of linguistic naturalness for both aspects of pitch and rhythm, music theory and user’s intention without a large amount of data as in deep learning. Moreover, as Orpheus v3 users had commented that



**Fig. 1.** Conceptual diagram of a singing song generation model based on path-finding. The red path represents acceptable melodies and the blue represents unacceptable melodies.

they found it difficult to sing due to the fact that the sentence breaks did not match the bar line, it was also hypothesised that placing sentence beginnings on stronger beats would improve singability and listenability in pitch-accented languages. The compositional principles of the proposed model were evaluated objectively in terms of linguistic naturalness, as well as subjectively by the audience.

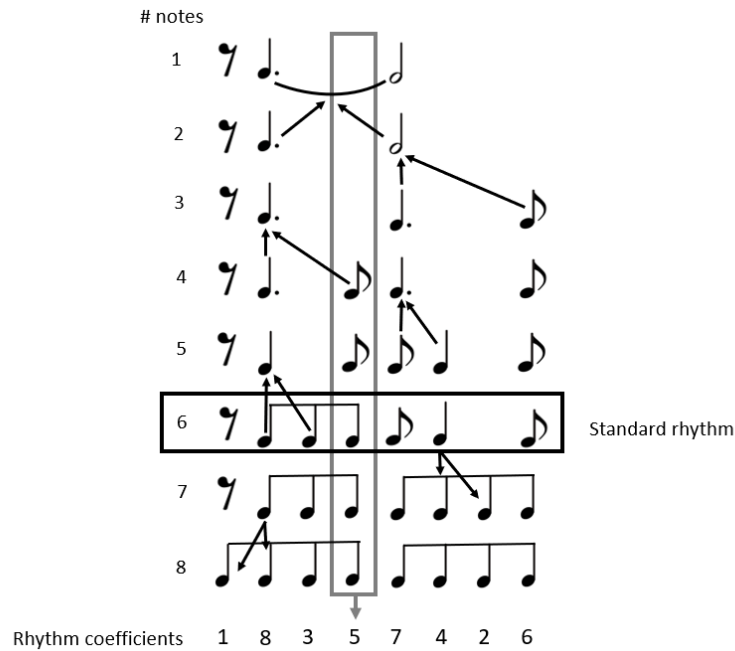
## 2 Melody Generation Model

Since melodic composition from lyrics is the problem of assigning notes to lyric syllables, it can be understood as the problem of finding a path on a grid of points in a two-dimensional plane of time (e.g. 16th note resolution) and pitch (e.g. semitones) to each syllable, as shown in Figure 1. Although pathways, i.e. pitch and rhythm combinations, are vast, the pathway cost of finding a valid melody with respect to domain knowledge of music theory, linguistic naturalness and user intent can be defined mathematically. Such a model for simultaneously generating the rhythm (onset time and duration) sequence ( $\hat{r}_{1:N}$ ) and the pitch sequence ( $\hat{p}_{1:N}$ ) optimised to the compositional conditions including lyrics ( $\mathbf{z}$ ) can be formulated as the maximisation of a probabilistic model:

$$\hat{r}_{1:N}\hat{p}_{1:N} = \arg \max_{r_{1:N}, p_{1:N}} p(r_{1:N}, p_{1:N} | \mathbf{z}) \quad (1)$$

where  $r_{1:N}$  and  $p_{1:N}$  are random variables. However, since Equation 1 is too computationally complex, we assumed in Orpheus v3 that pitch and rhythm are independent and rhythm is given in advance, and the melodic pitch generator  $\arg \max_{p_{1:N}} p(p_{1:N} | \mathbf{z})$  was realised by applying the Viterbi algorithm in a probabilistic model that follows Markov processes. In this paper, by introducing a rhythm generator ( $\arg \max_{r_{1:N}} p(r_{1:N} | \mathbf{z})$ ), which considers the musical domain knowledge, to the pitch generator of Orpheus v3, we propose a new melody generation model with high controllability that optimised to all aspects of user intention, music theory and linguistic naturalness in terms of both





**Fig. 2.** Example of a rhythm tree for rhythm generation from each rhythm family. The rhythm coefficients at the bottom represent the importance of each onset event to the rhythm family.

rhythm and pitch. Note that this is an approximate solution of Equation 1 due to computational complexity, i.e. the vertical (pitch) and horizontal (rhythm) axes of Figure 1 are optimised separately. The next section describes the proposed rhythm generators.

## 2.1 Melodic rhythm generator

**Overview** What are the requirements for melodic rhythm generation models in creative automatic composition systems for users? It is not easy to create rhythm from scraps without studying the composition techniques. However, it is natural to want to compose the same section, e.g. the first and second choruses, with similar melodic-rhythmic patterns as suggested by Fukayama et al. [9]. This means that the rhythmic pattern should be controllable from section to section.

**Melodic rhythm generation in Orpheus v3** In Orpheus v3, the rhythm generator is represented by “rhythm tree structure.” As shown in Figure 2, a rhythm tree has a rhythm called a “standard rhythm,” which is a good representation of its rhythm generator, and it is expanded and integrated so that they are perceived as similar to standard rhythms, even if the number of notes changes. By providing the rhythm tree that can generate similar rhythm patterns for each number of notes, users can control the melodic

rhythm by setting a rhythm family for each section. Rhythm trees in Orpheus v3 do not take into account the linguistic naturalness of the lyrics, as they are defined before the lyrics are input.

**Melodic rhythm generation model considering rhythmic style, accent position and duration** We aimed at an automatic generation model of melodic rhythms that preserves the rhythm family selected by the user as before, while also ensuring the linguistic naturalness of the lyrics by considering the relationship between syllable stress intensity and rhythmic intensity. Given a syllable feature vector sequence  $\mathbf{s}_{1:N}$ , such a model that generates a rhythmic sequence  $r_{1:N}$  containing  $N$  onset events can be modelled by dynamic Bayesian networks (DBNs) as follows:

$$\begin{aligned} p(r_{1:N}|\mathbf{s}_{1:N}) &\propto p(\mathbf{s}_{1:N}|r_{1:N})p(r_{1:N}) \\ &\approx p(r_1)p(\mathbf{s}_1|r_1)\prod_{i=2}^N p(\mathbf{s}_i|r_i)p(r_i|r_{i-1}) \end{aligned} \quad (2)$$

where rhythm sequence generation probabilities are approximated by Markov process and the  $i$  represents the consecutive numbers of syllables and not the rhythmic time.

Figure 3 shows the trellis of the proposed DBNs, in which horizontal nodes representing the onset events (16th-note resolution) and nodes are developed for the number of syllables in the vertical direction. Horizontal transition jumps are permitted to represent the duration of rhythmic events, while vertical jumps are not permitted as the syllables should be in sequence. However, the position of the rests has to be provided by the user same as the syllables.

The rhythm sequence with the highest likelihood generated from the proposed DBNs can be efficiently obtained using the Viterbi approach [11] and the rhythm sequence is assumed to respect rhythm family and linguistically naturalness. Finally, a melody is generated by combining the most likely rhythm generated by the proposed rhythm generator and the most likely pitch sequence generated by the Orpheus v3 pitch generator. The following sections describe how each probability parameter is learnt.

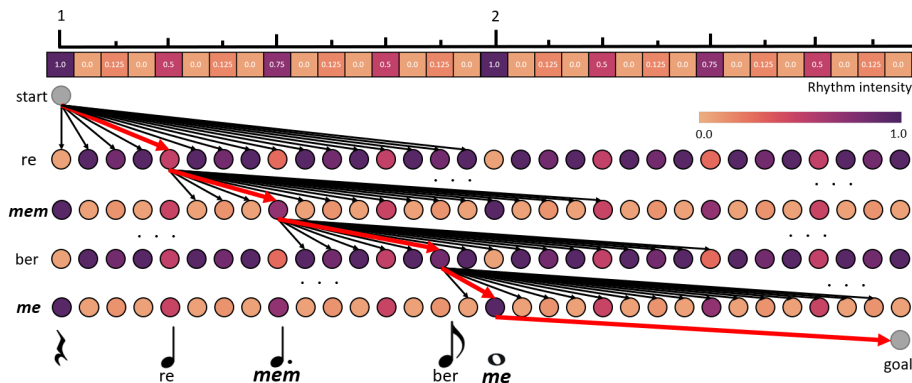
**Linguistic naturalness constraints** The  $p(\mathbf{s}_i|r_i)$  serves as a term to guarantee the linguistic naturalness of the lyrics. The  $s_i$  is the feature vector of the syllable, which includes the syllable stress intensity  $s_{i,\text{stress}}$  and the syllable length  $s_{i,\text{length}}$ . Assuming the independence of each of them,

$$p(\mathbf{s}_i|r_i) = p(s_{i,\text{stress}}|r_i)p(s_{i,\text{length}}|r_i) \quad (3)$$

$p(s_{i,\text{stress}}|r_i)$  can be formulated based on the findings of Oliveira et al. [7]. Assuming that  $s_{i,\text{stress}}$  follows a normal distribution with mean as the rhythmic intensity  $r_{i,\text{intensity}}$  and standard deviation as 1 empirically,

$$p(s_{i,\text{stress}}|r_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(s_{i,\text{stress}} - r_{i,\text{intensity}})^2} \quad (4)$$

where  $r_{i,\text{intensity}}$  can be determined by music-theoretic knowledge, for example, the rhythmic intensity can be set heuristically as shown in Figure 3 based on music-theoretical



**Fig. 3.** An example of the path-finding trellis of the rhythm generation DBNs at 16th note resolution in 4/4 time when not syncopated. The rhythmic intensity is determined heuristically based on music theory knowledge and the state likelihood of each node is calculated based on the difference between the rhythmic intensity and syllable stress intensity as an example.

knowledge. In terms of  $p(s_{i,\text{length}})$ , it would be possible to formulate syllable length ( $s_{i,\text{length}}$ ) following a normal distribution with mean as rhythmic duration ( $r_{i,\text{duration}}$ ) same as Equation 4. However, in this study, only rhythm events with stress-syllable and too short duration (sixteenth notes) were penalised from the point of view of singability, and all other values were given the same probability. In this way, the model can consider both music theory and linguistic naturalness.

**Rhythm family constraints** The  $p(r_i|r_{i-1})$  is the transition probability of a rhythmic event and can be trained on the basis of the rhythm trees defined in Orpheus v3 to generate rhythm based on user-specified rhythm families. We hypothesise that in a rhythm tree, the smaller the number of pitch accents, the more characteristic rhythmic patterns remain, and the higher the number of pitch accents, the more redundant patterns are injected. Thus, by calculating how many times an onset event appeared vertically for the entire rhythm tree, as shown in Figure 2, rhythm coefficients, which indicate how important each onset event was for that rhythm family, can be calculated. This rhythm coefficient was normalised by dividing it by the sum of the rhythm coefficients within the rhythm family and was the likelihood for onset event. The parameters in the rhythm generation models trained with such likelihoods can generate essential patterns within the rhythm family with high priority. The inclusion of such likelihoods in the model is expected to generate melodic rhythms that respect the user’s chosen rhythmic family.

### 3 Experimental Evaluation

In this experiment, the naturalness and coherence of the melodies generated by the proposed automatic composition system from Japanese and English lyrics were assessed objectively and subjectively.

### 3.1 Input data and proposed model setup

The stress accent intensity of lyrics was determined as follows.

- Japanese: Strong accent (1.0) placed at the beginning of a phrase. Accent values for other syllables were set to 0.0.
- English: 1.0 was placed on the primary accent and 0.75 on the secondary accent, and if the accented word was a weak form (e.g. preposition), the accent value was multiplied by 0.25. Accent values for all other syllables were set to 0.0.

The prosodies of lyrics were determined as follows.

- Japanese: Morphological analysis results from Mecab<sup>2</sup> are used.
- English: Only intonation within words was restricted, with reference to the F0 of the speech sound of the lyrics. Other pitch changes were allowed freely.

All rhythm families used in this experiment were non-syncopated 4/4 time rhythm patterns, and the intensity values for each onset event in one bar with 16th note resolution were set to the same values as in Figure 3, based on music-theoretical knowledge.

### 3.2 Objective evaluation

**Experimental condition** Objective evaluation experiments investigated the reflection of linguistic naturalness constraints in the melodies generated by the proposed model and Orpheus v3. A total of 12 songs, which were generated by each of Orpheus v3 and the proposed model based on a combination of 4 randomly selected rhythm families and their accompanying composition conditions (such as chord progression, accompaniment, drums, etc.), and 3 nursery rhyme lyrics (London bridge, Amazing grace, Scarborough Fair; opening eight bars of lyrics), were evaluated.

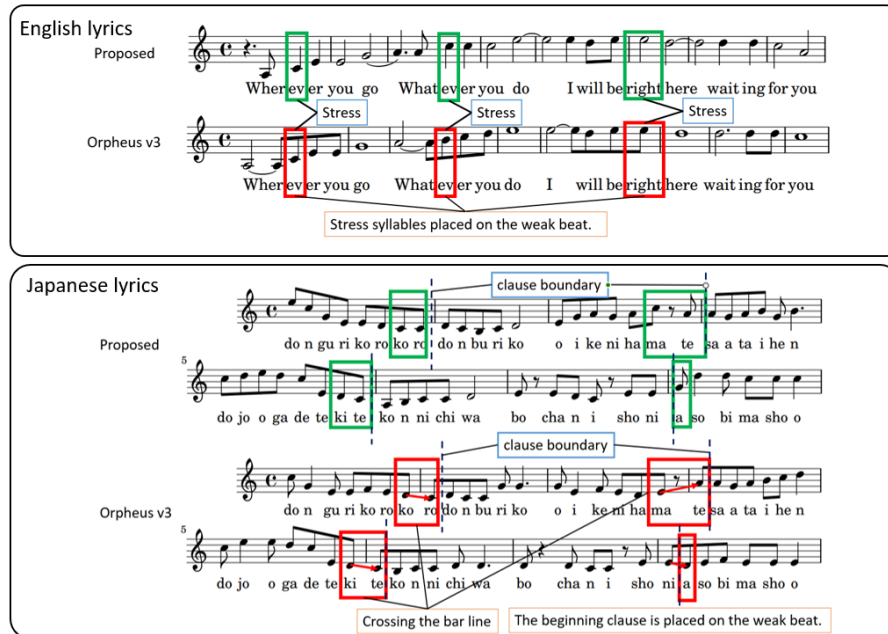
**Results** Rhythmic naturalness was assessed by the mean square error between the rhythmic intensity of each rhythmic event in the generated melody ( $E_r(r_i)$  from Equation 4) and the stress intensity of the phoneme corresponding to each rhythmic event in the lyrics as defined in section 3.1. Rhythm naturalness was about 0.296 for Orpheus v3 and **0.195** for the proposed method. In this way, the melodies generated by the proposed method are more consistent regarding the relationship between stress and rhythm.

Pitch naturalness was similarly evaluated for all syllables in the lyrics by the percentage of match between the transition direction of the melodic pitch (up or down) and the prosody of the syllable (up or down). As a result, Orpheus v3 and the proposed method achieved almost the same values, 0.991 and **0.994** respectively. Therefore, even if the proposed rhythm generator is introduced, the existing pitch generators are still functioning adequately.

Figure 4 shows an example of the song actually generated from the proposed model and Orpheus v3 respectively, based on the same lyrics and compositional conditions. As can be seen from these figures, the knowledge obtained in the objective evaluation experiment can be found concretely. Further examples of generated scores and sound sources can be found on the URL<sup>3</sup>.

<sup>2</sup> <https://taku910.github.io/mecab/>

<sup>3</sup> <https://coconuts-palm-lab.com/cmmr2023>



**Fig. 4.** Examples of generated melodies for the comparative and proposed methods when English and Japanese lyrics are used as input.

### 3.3 Subjective evaluation (Rhythm family)

**Assessment conditions** This experiment evaluated the similarity of melodic rhythms generated by the proposed model or Orpheus v3 for lyrics based on the same rhythm family but with different numbers of syllables and different accents. In this experiment, participants were asked to subjectively rate the rhythmic similarity of melodies generated by a combination of three nursery rhymes and three randomly selected rhythm families, similar to the objective assessment experiment. The evaluation is conducted with the XAB test, where X and A or, B are melodies generated from the same rhythm family. Users were instructed to listen to X first, then A and B, and to choose from A or B whichever they felt was closer to X in terms of rhythmic pattern.

**Stimuli** In addition to the melody, accompaniments and drum patterns generated from pre-prepared compositional conditions were assigned to each of the three rhythmic families to facilitate the capture of the beat and chord progression. The MIDI data generated from the models were synthesised using FluidSynth [12] at 44100 Hz, using the Fluid R3 sound font. Note, however, that as this is an experiment to assess the similarity of rhythmic patterns, the melody is played on a saxophone to make it easier to distinguish from the others, and the singing voice is not included in the sound source.

**Participants and procedure** The experiment was conducted online. The 25 participants in the experiment were all Japanese, with an average age of about 25. 80% of the participants had not trained musically for more than two years. The experiment began with an investigation of the participants' backgrounds, which included a survey of their age, country of residence and musical background based on Goldsmith-MSI<sup>4</sup>. In the main part of the experiment, participants were given just one question with answers in a similar format to the actual XAB test for a tutorial on the XAB test, and after gaining an understanding of the XAB test, they answered 10 XAB tests per person.

**Results** The results of the XAB test on rhythmic similarity showed that the proportion of selecting sound sources containing melodies generated from the same rhythmic family as sound source X was about 75% in both the proposed model and Orpheus v3. The results suggest that the proposed model can generate melodic rhythms with comparable rhythmic control performance to Orpheus v3 while it respects user-selected rhythmic families as well as linguistic naturalness constraints. There was concern that the constraints of linguistic naturalness might break the rhythmic patterns of the rhythm family, but this may suggest that probabilistic parameter learning with rhythm coefficients is a reasonable representation of the original rhythmic patterns.

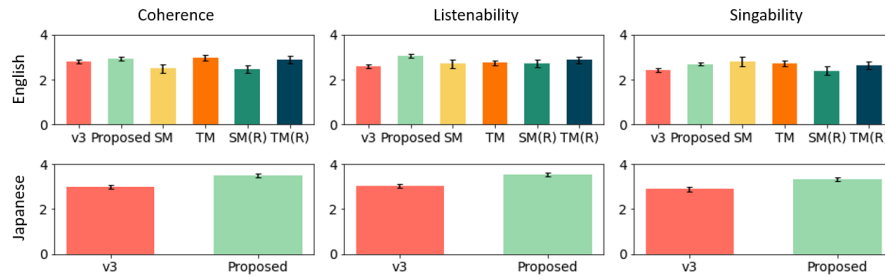
### 3.4 Subjective evaluation (Generated melody assessment)

**Assessment conditions** This experiment aims to subjectively assess the consistency, singability and listenability of the melodies generated by the proposed model and the comparative method. For comparison, we used Orpheus v3 as a baseline for English lyrics, and also compared it with SongMASS[2], TeleMelody[2], SongMASS + Relyme [4], and TeleMelody + Relyme [4], respectively. In the case of melody generation with Japanese lyrics, since it was not possible to prepare Japanese lyrics/melody pair data for training the latest deep learning methods and Relyme [4] principles were targeted at Chinese and English lyrics, only the proposed method was compared with Orpheus v3.

**Stimuli** The English lyrics were picked from the three lyrics used by Zhang et al. [4]<sup>5</sup> as test data in order to conduct fair test. For the Japanese lyrics, three well-known Japanese children's songs (Donguri Korokoro, Okina Noppo No Furudoke and Urashima Taro) were selected. The singing voice based on the lyrics was synthesised by the Maki Tsurumaki sound source on Synthesizer V. The deep learning method under comparison does not have the support for generating accompaniment or drums, so for fair evaluation, the English lyrics experiment used only the singing voice and melody guide (played on a saxophone), excluding the accompaniment and drums. In the Japanese comparison experiment, the sound sources were synthesised using FluidSynth (same as section 3.3) according to the MIDI of the melody, accompaniment and drums generated from each model and combined with the singing voice.

<sup>4</sup> <https://www.gold.ac.uk/music-mind-brain/gold-msi/>

<sup>5</sup> <https://ai-music.github.io/relyme/>



**Fig. 5.** Results of subjective experimental evaluation. v3 stands for the Orpheus v3 baseline method [9], SM for Songmass [2], TL for Telemelody [3] and R for Relyme [4] in combination with the methods of [2] and [3], respectively.

**Participants and procedure** The experiment was conducted online. The participants were 25 people who also took part in the experiment in section 3.3. Participants answered five-point Likert scale questions on three aspects of musical coherence, singability and listenability for songs generated from English (25) and Japanese (15) lyrics. However, to clarify which syllables were assigned to which notes and to minimize the effect of the singing synthesis, participants were presented with an image of the melodic score as well as the sound source simultaneously. In addition, we reminded participants that singing synthesis and playing instruments are not the scopes of our study.

**Results** The mean and standard error statistics of the experimental results are summarised in Figure 5. For English lyrics, the proposed method was rated significantly higher than the baseline method, Orpheus v3, on all items of coherence, listenability and singability. When comparing state-of-the-art deep learning methods with the proposed method, the proposed method obtained significantly higher ratings for coherence than some deep learning models (SM, SM(+R)) and for listenability than most deep learning models, respectively. Although there were no significant differences between the deep learning method and the proposed method, most of the participants in this experiment were not music experts and therefore had difficulty in assessing singability. Since there are no items where the proposed method is significantly inferior to the state-of-the-art deep learning methods, it suggests that the proposed method may be equal to or superior to the state-of-the-art deep learning methods in English lyrics. In addition, focusing on the standard errors, the fact that the latest deep learning methods have large standard errors while the proposed method has small standard errors seems to indicate the robustness of the proposed method.

For Japanese lyrics, all items were rated significantly higher for the proposed method than for the baseline method Orpheus v3. This may be because the placement of the initial syllable of a passage on a stronger beat might sharpen the semantic break in pitch-accent language. The results suggest that there is a clear relationship between semantic delimitation and melodic rhythm in the pitch-accent language, and that the proposed rhythm generator works effectively in the pitch-accent language.

However, as this experiment was conducted with 25 Japanese subjects, there is a bias, and therefore a similar experiment needs to be conducted with more participants and not only with native speakers of Japanese, but also with native speakers of other languages, in order to make a more generalised assessment.

## 4 Discussion

The results of the evaluation experiments show that the proposed method can generate melodies that respect the rhythmic pattern of the user-selected rhythm family, while taking into account linguistic constraints such as stress-rhythm constraints and prosody-pitch constraints. It has also been shown that the consideration of linguistic naturalness, as incorporated in the proposed method, improves singability and listenability compared to the baseline method. The proposed method, trained only on domain knowledge without training on large amounts of training data, was rated as good as or better than state-of-the-art deep learning methods.

Since pitch and rhythm are optimised separately in the proposed model, it is difficult to add constraints considering pitch and rhythm simultaneously. For example, when singing in the high register, a series of notes with short duration makes singing difficult and non-chord tones, such as neighbour and passing tones, are known to have weak beats and short duration. In order to apply such knowledge as a constraint for melody generation, it is necessary to consider path-finding on a 2D plane, as shown in Figure 1, and thus to study how to solve the problem of the computational cost of Figure 1.

In order to realise a universal compositional principle, it is expected to support lyrics in tonal languages in addition to stress and pitch-accented languages. For tonal languages, this can be resolved by DBNs with states that take into account the possibility of pitch transitions occurring within a single syllable.

The proposed model is formulated by a hidden Markov model, which means that the computational complexity increases exponentially when trying to consider long contexts. From the results of the experimental evaluation, it seemed possible to generate melodies that could convince the audience by considering local music theory and linguistic naturalness. However, a longer context might allow the model to take into account song styles (e.g. genre, artist) in the training data and add user-selectable compositional styles to the compositional conditions.

## 5 Conclusion

This study proposed a probabilistic model that targets the generation of the most linguistically and musically natural song melody based on the user's input lyrics and compositional conditions. In addition to the melodic pitch generator considering relationship between prosody and melodic pitch of lyrics that have been considered in our previous research (Orpheus), the proposed system introduced a melodic rhythm generator in which the probability parameters are learned so that the stress of lyrics and the melodic rhythm intensity are consistent while respecting the rhythm style selected by the user. The results of the experimental evaluation showed that it is possible to generate melodies that are reasonable in terms of linguistic naturalness and music theory,



while maintaining the same level of controllability as the previous Orpheus v3. Subjective evaluation experiments showed that the melodies generated by the proposed model were equal to or better than state-of-the-art deep learning methods in terms of musical coherence, singability and listenability.

In the future, by solving the problem of computational complexity, the aim is to build a model that simultaneously considers pitch and rhythm, which can make use of vocal and other music-theoretical knowledge such as non-chord tones. We will also explore how knowledge from deep learning methods that can consider longer-term context can be used in the proposed model.

## References

1. Hangbo Bao, Shaohan Huang, Furu Wei, Lei Cui, Yu Wu, Chuanqi Tan, Songhao Piao, and Ming Zhou. Neural melody composition from lyrics. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*, pages 499–511. Springer, 2019.
2. Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. Song-mass: Automatic song writing with pre-training and alignment constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13798–13805, 2021.
3. Zeqian Ju, Peiling Lu, Xu Tan, Rui Wang, Chen Zhang, Songruoyao Wu, Kejun Zhang, Xiangyang Li, Tao Qin, and Tie-Yan Liu. Telemelody: Lyric-to-melody generation with a template-based two-stage method. *arXiv preprint arXiv:2109.09617*, 2021.
4. Chen Zhang, Luchin Chang, Songruoyao Wu, Xu Tan, Tao Qin, Tie-Yan Liu, and Kejun Zhang. Relyme: Improving lyric-to-melody generation by incorporating lyric-melody relationships. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1047–1056, 2022.
5. Alexander Fang, Alisa Liu, Prem Seetharaman, and Bryan Pardo. Bach or mock? a grading function for chorales in the style of js bach. *arXiv preprint arXiv:2006.13329*, 2020.
6. Yuki Karatsu and Satoru Fukayama. A comparative study of data augmentation methods for automatic harmony generation with a low number of forbidden violations. *Proc. Spring Meet. Acoust. Soc. Jpn.*, 2023.
7. Hugo R Gonalo Oliveira, F Amilcar Cardoso, and Francisco C Pereira. Tra-la-lyrics: An approach to generate text based on rhythm. In *Proceedings of the 4th. International Joint Workshop on Computational Creativity*. A. Cardoso and G. Wiggins, 2007.
8. Kristine Monteith, Tony R Martinez, and Dan Ventura. Automatic generation of melodic accompaniments for lyrics. In *ICCC*, pages 87–94, 2012.
9. Satoru Fukayama, Kei Nakatsuma, Shinji Sako, Yuichiro Yonebayashi, Tae Hun Kim, Si Wei Qin, Takuho Nakano, Takuya Nishimoto, and Shigeki Sagayama. Orpheus: Automatic composition system considering prosody of japanese lyrics. In *Entertainment Computing–ICEC 2009: 8th International Conference, Paris, France, September 3-5, 2009. Proceedings 8*, pages 309–310. Springer, 2009.
10. Kento Watanabe, Yuichiro Matsubayashi, Fukayama Satoru, Nakano Michiyasu, Goto Masataka, and Inui Kentaro. Automatic lyric generation based on correlation between melody and lyrics. *IPSJ SIG Technical Report*, 2017(16):1–12, 2017.
11. Randal J Leistikow. *Bayesian modeling of musical expectations via maximum entropy stochastic grammars*. Stanford University, 2006.
12. Jan Newmarch and Jan Newmarch. Fluidsynth. *Linux Sound Programming*, pages 351–353, 2017.

# Design of a music recognition, encoding, and transcription online tool

David Rizo<sup>1,2</sup>, Jorge Calvo-Zaragoza<sup>1</sup>, Juan C. Martínez-Sevilla<sup>1</sup>, Adrián Roselló<sup>1</sup>,  
and Eliseo Fuentes-Martínez<sup>1</sup> \*

<sup>1</sup> Universidad de Alicante

<sup>2</sup> Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana (ISEA.CV)  
drizo@dlsi.ua.es

**Abstract.** In recent years, Optical Music Recognition (OMR) technologies have experienced a notable boost thanks mainly to the use of new pipelines based on machine learning, specially on deep neural networks. These methods are usually studied just from the point of view of the accuracy of the output of the networks. However, from a practical perspective in a real-world context, this is not enough. In this paper we present a design of a tool devised for allowing the scientific study of the complete OMR workflow in different scenarios and notations, including both the possibility of analyzing the real impact of improvements in automatic recognition models and how they are integrated for practical purposes in the work of the transcriber.

**Keywords:** Optical music recognition, encoding, transcription, user experience

## 1 Introduction

Digitizing sheet music and other music-related documents can provide several benefits, including easier access for researchers, music practitioners, musicologists, and the general public, as well as preservation of musical heritage. Digitized sheet music can be searched, played, analyzed, and annotated using specialized software tools, allowing for new discoveries and insights into musical history and culture.

One notable example of an effort to digitize music collections into digital images is the International Music Score Library Project<sup>3</sup> (IMSLP), which aims to create a virtual library of public domain sheet music. The IMSLP has digitized thousands of scores from various composers and genres, making them freely available for download and use. Other organizations and institutions, such as libraries, museums, and universities, are also scanning their music collections into image files to increase access and preserve musical heritage.

\* This work has been supported by the Spanish Ministerio de Ciencia e Innovación through project MultiScore (No. PID2020-118447RA-I00), supported by UE FEDER funds.

<sup>3</sup> <https://www.imslp.org> (accessed April 19th, 2023).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

As a matter of fact, having the musical content information, i.e., the audio and the scans, not digitally encoded ends up being a waste of resources given that current music information retrieval pipelines require it. By this means, we cannot consider a score digitization process as finished until the digital score version encoding is produced.

Alfaro-Contreras et al. [3] showed that the most efficient way to obtain these digital scores is to resort to an automated reading of documents by using the so-called Optical Music Recognition (OMR) [5]. This technology has achieved different levels of recognition accuracy depending on the type of documents, the quality of the medium, and the type and complexity of the notation. But still, in most cases, the OMR does not yield perfect results. The need of post-editing depends on the tasks to be performed on the recognized content. Some projects such as F-Tempo<sup>4</sup> directly use—possibly with errors—OMR output to approximate perform searches. When requiring a curated transcription, a manual correction process has to be done. For instance, this pipeline was used to encode a large number of files of the KernScores repository.<sup>5</sup>

The low accuracy of OMR is not the only obstacle in real use-cases. There is no OMR system yet able to process the whole set of symbols found in early notations. The processing of orchestral scores of varying layout, the presence of *ossias*, or dealing with works where the different parts are written in separate sheets, make it even more challenging. This is why, in many real projects, the encoding process is eventually performed by transcribers using computerized notation tools such as Finale<sup>6</sup>, Dorico<sup>7</sup>, MuseScore<sup>8</sup>, or Sibelius<sup>9</sup>. In projects such as Didone [26], around 4 000 Eighteenth-Century Italian Opera arias are being manually copied in Finale to be later stored into MusicXML files [13]. The same approach was used to obtain the encodings of the modern version of the renaissance works from the “Josquin Research Project” (JRP).<sup>10</sup>

In this scenario, the main advances in OMR are achieved by modern artificial intelligence techniques based on machine learning, namely deep learning [5]. Improvements are attained through the correct selection of neural network architectures and the availability of training data in sufficient quantity and quality for those architectures. The research-oriented OMR tool “Music Recognition, Encoding, and Transcription” (MuRET) [19] was introduced to push the development of both OMR techniques and the creation and curation of datasets. This tool was developed with JavaFX<sup>11</sup> as a desktop application. It included our first OMR models [8], which allowed for the curation of a number of training sets and the development of new OMR approaches [24, 9].

Once the usefulness of MuRET became clear, we decided to port it to a web application for two main reasons. The first is that the technology is continuously evolving, which made it difficult to deploy the application on a daily basis. The second, and more important, is to naturally maintain a growing repository of both ongoing transcription documents and trained OMR models shared among all users. Having evaluated this re-

<sup>4</sup> <https://f-tempo.org> (accessed April 19th, 2023).

<sup>5</sup> <http://kern.ccarh.org/> (accessed April 19th, 2023).

<sup>6</sup> <https://www.finalemusic.com> (accessed April 19th, 2023).

<sup>7</sup> <https://www.steinberg.net/dorico> (accessed April 19th, 2023).

<sup>8</sup> <https://musescore.org> (accessed April 19th, 2023).

<sup>9</sup> <https://www.avid.com/sibelius> (accessed April 19th, 2023).

<sup>10</sup> <https://josquin.stanford.edu> (accessed April 19th, 2023).

<sup>11</sup> <https://www.oracle.com/es/java/technologies/javase/javafx-overview.html> (accessed April 19th, 2023).

search oriented OMR online tool in real scenarios, in this paper the main decisions to build it are described with which we expect to contribute to the improvement of ongoing and future OMR investigations.

The paper is organized as follows. Section 2 details other alternatives to MuRET that, due to the fact that this tool is eminently research-oriented, may be most suitable to be used in transcription projects. Next, the requirements taken into account in Section 3 and decisions made during the development of the current online version are discussed in Section 4 that may be useful for other similar projects. Finally, conclusions will be drawn and future works outlined in Section 5.

## 2 State of the art

There are several Optical Music Recognition (OMR) tools available to transcribe Common Western Modern Notation (CWMN). Only one open-source Audiveris<sup>12</sup> is available, and a number of commercial packages such as SmartScore<sup>13</sup>, PhotoScore<sup>14</sup>, or PlayScore 2<sup>15</sup>. The effectiveness of each tool can vary depending on the complexity and quality of the sheet music being analyzed. A brief analysis of them for recognizing music theory books can be found in [14], that shows how far they are from retrieving successful results on complex scenarios.

The transcription of notations other than CWMN is very restricted to very few applications. In the context of the SIMSSA project [11], two applications in the past years have been used to automatically extract musical information from images, although they are no longer maintained: Gamut and Aruspix [17]. In addition, within this project, an OMR meta-workflow called Rodan was built in which users can create their own systems using predefined image processing and machine learning blocks [12]. Although not designed for any specific notation, most of the existing blocks are intended for neume recognition. More recently, another approach based on convolutional neural networks was developed specifically for mensural notation, which reported high accuracies [25].

Given this context, to the best of our knowledge, no tool ready to deal both with handwritten and printed sources of several kinds of notation is available other than our proposal MuRET.

## 3 Requirements

The ultimate goal of MuRET is to facilitate OMR research from a holistic perspective. This means that the tool must support the research of all individual steps of the workflow to obtain a final digital score from the different images, considering both the automatic processes and user manual interactions. Being this a research tool, it must be prepared to be scaled to any possible scenario in terms of notation type, parts arrangement, document layout, calligraphies and fonts, and transcription purposes.

<sup>12</sup> <https://github.com/Audiveris/audiveris> (accessed April 19th, 2023).

<sup>13</sup> <https://www.musitek.com> (accessed April 19th, 2023).

<sup>14</sup> <https://www.neuratron.com/photoscore.htm> (accessed April 19th, 2023).

<sup>15</sup> <https://www.playscore.co> (accessed April 19th, 2023).

From an end-user point of view, the user should be first allowed to manage collections of works made of digital images of any format and resolution. For transcribing a new work, the constituent elements in each image such as pages, staves, and lyrics must be identified. Then, in the case of being a polyphonic work, they must be assigned to the different instruments, voices, or parts. The contents in staves and text regions must be recognized using a variety of approaches that, after being combined, will make up a final digital score that will be exported to standard encoding formats. All possible aids that the machine can compute, such as displaying early notations in modern forms, or hints in the final scoring-up, should be provided. Additionally, the tool must incorporate process-oriented functionalities, as an aid to account for the current status of work on several simultaneous works, or the inclusion of comments both for the whole work or elements inside it.

Ultimately, the system must be able to perform all processes in an assisted manner, so that the output of the various automatic classifiers is corrected when necessary by the user as quickly as possible.

From the OMR process research perspective, the tool must be ready to accommodate different approaches to convert a set of input images into a digital score. For each of those approaches, in the case of being based on machine learning, the extraction of new training sets from the already processed works and the posterior training, upload, and use of new models, must be supported. For analyzing the actual behavior of any paradigm and model besides the usual model performance metrics, all actions of the user must be recorded and categorized for being later analyzed.

Several non-functional requirements arise that can influence the design of the tool. It must allow the simultaneous transcription of the same work by different, possibly remote, users. To allow the accommodation of new repertoires with a minimum amount of effort, and to make the user unconcerned about formats and resolutions, the use of IIIF framework<sup>16</sup> for exchanging the processed works to other external tools is recommended. Finally, the system must be usable with a standard computer setup.

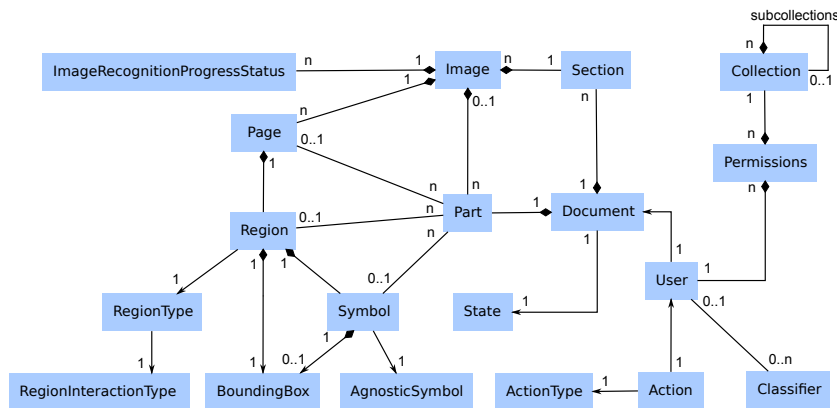


Fig. 1: Domain model. Some classes and relations, and all attributes have been omitted for easier reading.

<sup>16</sup> <https://iiif.io> (accessed April 19th, 2023).

## 4 Solution design

The current online version of MuRET<sup>17</sup> presents a possible approach to solve all the requirements detailed above. In order to implement it, several decisions have been made that will be described in this section.

The system has been structured using a three-tier web architecture style. The *presentation layer* has been solved using Angular with the Redux pattern<sup>18</sup>, the *application layer* has been implemented using Spring Boot<sup>19</sup>, and the *data layer* served by a MariaDB<sup>20</sup> relational database.

All the system data is eventually stored in records of the database that is converted to the object-oriented hierarchy shown in Fig. 1 through the Hibernate Object-Relation Mapping (ORM)<sup>21</sup>. The names of the classes can be easily understood from the explanation in the following lines.

### 4.1 User collections and images

Users must be registered by a system administrator to work on the tool. No self-registering process is offered. All works are organized into collections and sub-collections, whose access is granted by the administrator.

A document is the core entity of a transcription project. After being uploaded as individual image files or inside a PDF document, the images of a document to be transcribed are grouped, at least, into one default section. This is useful for dealing with compound works such as masses and operas. Images and sections can be deleted, edited, and re-ordered. Images that contain cover sheets, or non-musical content, can be hidden for subsequent automatic recognition processes. Users are allowed to assign to each work metadata such as the notation type, manuscript type (printed or handwritten), composer and printer.

For the purpose of helping the user in daily work tasks, the work in progress and image recognition annotations stages can be marked up to the final transcription (see buttons to mark this progress below in the bottom-right of Fig. 5c).

### 4.2 Document analysis

After organizing the images, the first step to transcribe a work, known as *document analysis*, is to segment each image into separate components, a series of regions of different types such as staves and lyrics are identified (Fig. 2). Usually, each image contains just one page, but it is also usual to receive scans of books where images contain several pages as in the case of the image in that figure. This process can be performed either manually by drawing bounding boxes on top of the image and assigning a region type to each drawn box, or by using an automatic classifier that identifies the different segments in the image. Across MuRET, when an operation can be performed automatically, the

<sup>17</sup> <https://muret.dlsi.ua.es/muret>

<sup>18</sup> <https://angular.io> and <https://redux.js.org> (accessed April 19th, 2023).

<sup>19</sup> <https://spring.io/projects/spring-boot> (accessed April 19th, 2023).

<sup>20</sup> <https://mariadb.org> (accessed April 19th, 2023).

<sup>21</sup> <https://hibernate.org/orm/> (accessed April 19th, 2023).

user can select and apply a classifier (see the two available models of the drop-down control at the top-right of the Fig. 2), and correct the output if necessary. Classifiers are run currently in the user browser using TensorFlow.js.<sup>22</sup> This decision has the advantage that it allows avoiding collapsing the server machine when several users are using MuRET at the same time running different models that have to be loaded in memory. The main drawback is that the used models have had to be tuned to keep their size at the minimum for being used in standard computers.

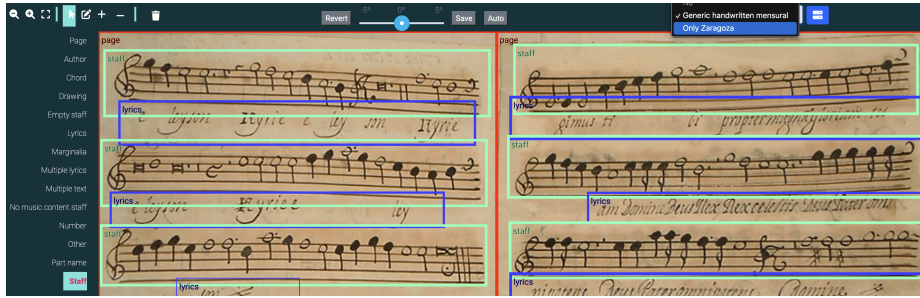


Fig. 2: Document analysis screenshot. In this example, only the staves and lyrics regions are segmented. The snapshot shows two possible classifiers to perform a document analysis (top-right), and controls to rotate, manually or automatically, the image (top-center). The current catalog of region types shown at the left of the image can be easily modified.

### 4.3 Part management

Most of the sources to be processed are polyphonic, consisting of several voices, instruments, or parts. There is a variety of arrangements, such as works made of parts distributed across pages (the image in Fig. 2 corresponds just to a part), choir-books where the same page contains two voices (Fig. 3b), ensembles or orchestral scores (Fig. 3a). The kind of book to transcribe could be of a totally different nature. For instance, it can be a compilation of songs not related to any instrument, such as a jazz *Real Book*, or be a catalog containing lists of incipits. In some cases, the volume to be transcribed describes music theory as it is the case of music treatises [14], where most of the content is textual with some illustrative music examples. The process of dealing with parts is performed currently manually. In all cases, the internal implementation of all those situations is reduced to the case where the whole image or page is linked to a part, or when each region must be assigned to a part. The system is also prepared to deal with chorale layouts with two staves for four voices. In that case, each individual symbol inside the region must be linked to each part. For reducing user effort, the system offers aids to manage the set of instruments and to reuse the different layouts between pages.

### 4.4 Region contents recognition

Once the different staves are identified and assigned to the part they belong to, the musical contents inside the image crop that corresponds to each region must be recognized and

<sup>22</sup> <https://www.tensorflow.org/js> (accessed April 19th, 2023).

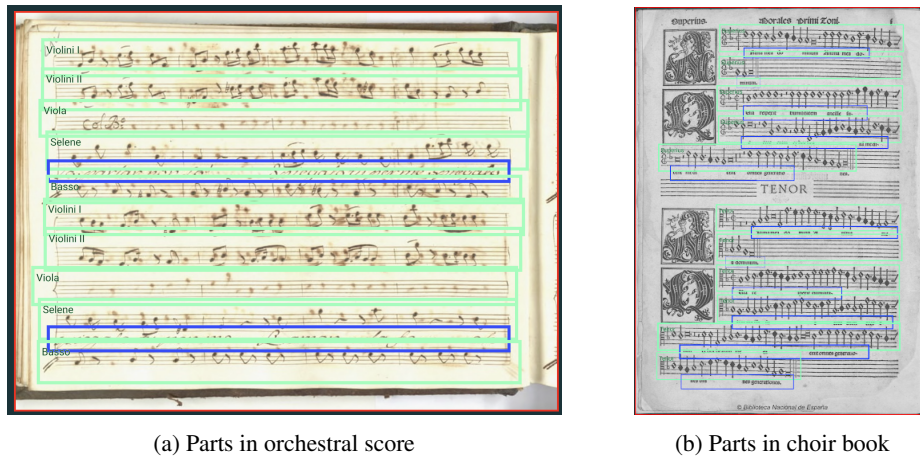


Fig. 3: Different parts and arrangements. All regions must be attributed to a part.

encoded. Currently, lyrics can be encoded as text but they are just stored in the database without any further treatment (Fig. 4a). New region types can be easily incorporated in the future. There are several approaches that can be followed to obtain an encoding from the image, either manually or by applying an automatic classifier. The first consists of manually tracing the graphic symbols so a classifier [8], by using both the stroke (Fig. 4b) and the image obtained from the bounding box that encloses the stroke, identifies each symbol among a set of possible *agnostic symbols* [7] (i.e. graphical symbols without an attributed musical meaning yet), and the vertical position in the staff as an absolute value regardless the clef. Although we use this *symbol-agnostic* concept where we identify complete glyphs, we could easily adapt it to recognize primitives (note head, stem, etc.) as proposed in [16].

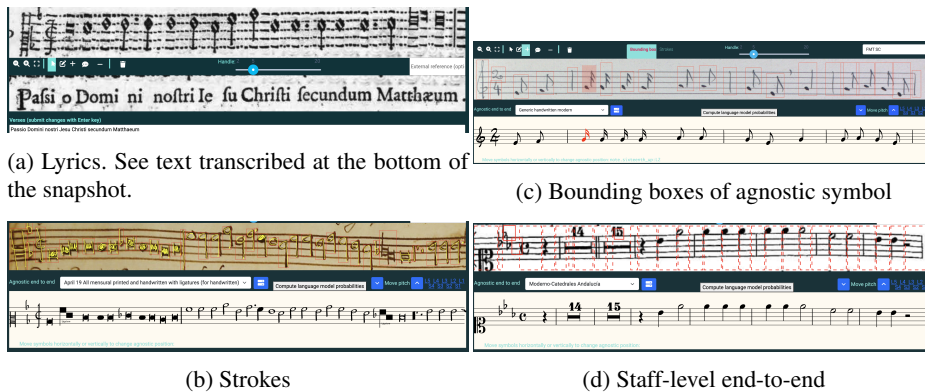


Fig. 4: Transcription of regions.



An alternative is just to draw the bounding box surrounding each symbol to use just the image clipping as input information for the classifier [15] to identify the agnostic symbol type (Fig. 4c). In either of the two options, an agnostic symbol sequence is obtained, i.e., a sequence of symbols ordered from left to right, top to bottom. For instance, the beginning of the agnostic sequence in Fig. 4c is: `clef.G2:L2 digit.2:L4 digit.4:L2 note.8th:L1 note.8th:S1 verticalLine:L1`. The rendering of the bottom staff is performed by using fonts that have a glyph for each agnostic symbol that are just placed in the position of the symbol in the image. For white mensural notation, the Capitan [20] font is used that was developed on-purpose, and for modern notation, Petaluma font<sup>23</sup> has been utilized.

The next possibility is to use a staff-level end-to-end classifier that identifies the agnostic symbols in the image in such a way that the sequence order is respected but the bounding boxes of each symbol are not detected but their approximate horizontal position [7] (dashed lines in Fig. 4d show those approximate positions). The user can move and correct any of the symbols. In that case, to take advantage of the interaction for obtaining a new training sample, the bounding box is drawn (see the second flat in the key signature in Fig. 4d).

#### 4.5 Music encoding of individual staves

The agnostic sequence must be converted to a meaningful music encoding denoted as *semantic encoding* [7]. For example, the sequence of three flats at the beginning of the agnostic sequence in Fig. 4d must be converted to a E♭ major (or its relative minor) key. When encoding the pitch of the notes, this key signature must be taken into account for correctly assigning if necessary the right accidental. This translation from *agnostic* to *semantic* (Fig 5a) can be performed either using a rules-based automaton transducer [20], or translation technologies based on machine learning approaches [1]<sup>24</sup>. For early notations, a valid conversion into modern notation is performed with any consideration of transposition or metric change (Fig. 5c). The rendering of the bottom staff is delegated to Verovio [18], through a previous conversion of our internal format introduced below into MEI (Fig. 5b). If the staff to convert is not the top staff, the contextual information from previous staves, such as the previous time signature, is propagated. The conversion to Plaine and Easie Code [4] for cataloging in RISM<sup>25</sup> is performed by that library as well.

It is important to note that MEI or Verovio do not always support all required features. For instance, bar-lines crossing a note in late mensural notations or the rendering of *signum congruentiae*. In those cases, our principle has been to internally store a specific tag for each unsupported feature and print text marks to visualize them.

A key decision taken to design MuRET was the method to store those semantic sequences. We have not chosen any standard format as the internal encoding, but an ad-hoc representation extended from the Humdrum [23] formats `**kern` and `**mens` [21] in what we name `**skm`. Later, when exporting the final encoding, standard formats are

<sup>23</sup> <https://www.smuf1.org/fonts/> (accessed April 19th, 2023).

<sup>24</sup> Note that there is an agnostic representation for dealing with more complex situations such as the presence of chords [2]. In any case, the workflow is the same regardless of the agnostic encoding.

<sup>25</sup> <https://rism.info/> (accessed April 19th, 2023).

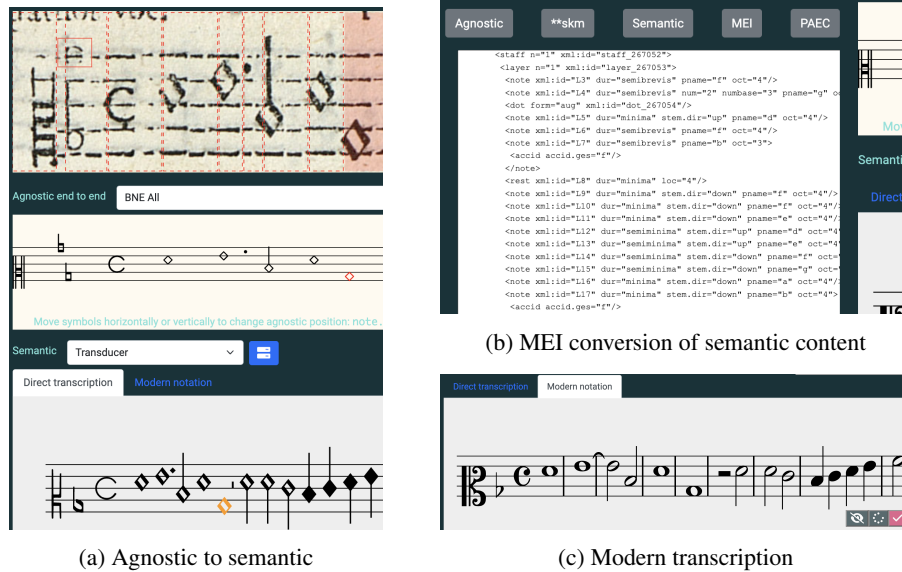


Fig. 5: Semantic contents recognized from the image.

used. This has allowed us to adapt the format to encode specific situations that were not possible with standard `**kern/**mens` when we required it (e.g. *custos* symbol, the position of rests, canceling accidentals in mensural notation, or dots after a barline). This encoding contains also information about the agnostic symbol each semantic element is related to, which allows for later exporting this kind of graphical information to formats such as the *facsimile* element in MEI. The choice of extending Humdrum formats and not other more comprehensive ones such as SCORE [23], MusicXML [13] or MEI [22], is the ease with which users can fully manually encode or correct the output of the classifiers. In any case, the translation process generates a sequence of objects of a musical object-oriented hierarchy that are just converted into `**skm` in order to serialized them allowing its presentation in the interface and storage.

MuRET does not include yet the direct recognition from the image to the `**skm` encoding because we have experimented to be faster and more accurate to use this intermediate representation as shown in [6]. If a classifier was proven to yield better results, both the transcription and correction processes, it could be easily introduced in the application.

#### 4.6 Scoring up and exporting

Finally, when all previous steps have been finished, the user can select the images (Fig. 6a) that want to be used to generate a final score (Fig. 6b). This operation is accomplished by concatenating all the staves in the selected images grouped by the parts they belong to, exporting them from our internal format to MEI, and letting Verovio engrave the score. In order to share the transcription with external services, the previous MEI can be ex-

ported as it is rendered by Verovio. MuRET also is able to convert to a parts-based MEI format including graphical information in the facsimile element (Fig. 6c). This functionality was included for exchanging information with specialized tools such as MP-Editor to perform scoring-up processing in mensural notation [10].

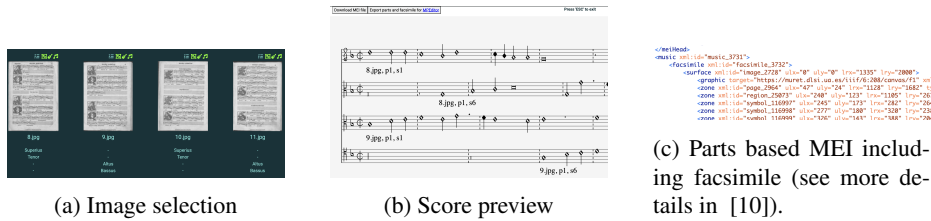


Fig. 6: Previsualizing and exporting

#### 4.7 Model training

As mentioned above, the classifiers that support the automatic processes use machine learning models, that need training data for being built. The system allows downloading different training sets from selected collections and works. This training data is just a set of JSON files containing an export of the objects in our internal data model (Fig. 1). After being trained offline, these models can be uploaded again to the system. This incremental workflow, i.e., fixing the output of the different classifiers, downloading datasets with corrected data, building new models, and uploading them to improve the performance of the system, is being shown in our transcription projects to be a proper way to proceed. Using this approach for transcribing a work by Jacob van Eyck, printed by Paulus Matthijsz in 1649, the recognition and post-editing effort were reduced by a factor of 10, allowing the end user to obtain a complete and correct encoding of a standard book page in under one minute per page.

#### 4.8 User action logging and user experience

Some models perform better than others in a theoretical way, but the corrections required to fix their outputs lead to a higher effort by the user. This can be measured by analyzing the actions each individual user performs on the tool that is logged and conveniently categorized into meaningful operations, such as the editing of regions, symbols, semantic content, part management or classifier use. The timestamps of all operations is also registered, as well as the document, region or symbol involved in each operation. Currently, we have stored more than 300 000 actions from different users.

These action logs have helped us to improve the user experience of the system by evincing many operations that are frequent and repetitive, decreasing the final throughput. These issues, such as those related to the feedback of the system in error messages, long actions, or the graphical design of interaction controls, have been gradually corrected or taken into account to include new capabilities to the system.

## 5 Conclusions and future work

This paper has depicted the main blocks required to integrate an OMR system scalable to work with any kind of notation and scenario that besides being useful for real transcription projects can help in the improvement of OMR research.

This tool is being continuously improved as new features in projects arise. Most of the effort is performed on improving the OMR models by using the increasing quantity of already transcribed works of different kinds. We are working towards addressing the current weaknesses of the system, namely: adding new front-end deep learning frameworks and formats such as ONXX<sup>26</sup>, the direct use of the IIIF manifest from servers without the need of uploading any image to the system, the possibility of performing an automatic classification of a whole work to be later corrected to complement the current totally interactive workflow, the possibility of directly training models online, and the endless task of improving the user experience of the tool.

## References

1. Ríos-Vila; A., M. Esplà-Gomis, D. Rizo, P.J. Ponce de León, and J.M. Iñesta. Applying automatic translation for optical music recognition's encoding step. *Applied Sciences*, 11(9), april 2021.
2. M. Alfaro-Contreras, J. Calvo-Zaragoza, and J.M. Iñesta. Approaching end-to-end optical music recognition for homophonic scores. In *Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA 2019*, pages 147–158, Madrid, August 2019. Springer.
3. M. Alfaro-Contreras, D. Rizo, J.M. Iñesta, and J. Calvo-Zaragoza. OMR-assisted transcription: a case study with early prints. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 35–41, Online, November 2021. ISMIR.
4. B.S. Brook. The Simplified Plaine and Easie Code System for Notating Music: A Proposal for International Adoption. *Fontes Artis Musicae*, 12(2-3):156–160, jan 1965.
5. J. Calvo-Zaragoza, J. Hajič, and A. Pacha. Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, 53(4):1–35, 2020.
6. J. Calvo-Zaragoza and D. Rizo. Camera-PrIMuS: Neural end-to-end optical music recognition on realistic monophonic scores. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 248–255, 2018.
7. J. Calvo-Zaragoza and D. Rizo. End-to-end neural optical music recognition of monophonic scores. *Applied Sciences*, 8(4), 2018.
8. J. Calvo-Zaragoza, D. Rizo, and J.M. Iñesta. Two (note) heads are better than one: Pen-based multimodal interaction with music scores. In *International Society for Music Information Retrieval Conference*, 2016.
9. 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.*, 22(4):1573–1584, 2019.
10. K. Desmond, L. Pugin, J. Regimbal, D. Rizo, C. Sapp, and M. E. Thomae. Encoding polyphony from medieval manuscripts notated in mensural notation. In *Music Encoding Conference Proceedings 2021*, page 197–219. Humanities Commons, May 2022.

<sup>26</sup> <https://onnx.ai/> (accessed April 19th, 2023).

11. I. Fujinaga, A. Hankinson, and J.E. Cumming. Introduction to SIMSSA (single interface for music score searching and analysis). In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, DLfM '14, page 1–3, New York, NY, USA, 2014. Association for Computing Machinery.
12. I. Fujinaga and G. Vigiensoni. The art of teaching computers: The SIMSSA optical music recognition workflow system. In *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*, pages 1–5. IEEE, 2019.
13. M. Good and G. Actor. Using MusicXML for File Interchange. *Web Delivering of Music, International Conference on*, 0:153, 2003.
14. F. Moss, N. Nápoles-López, M. Köster, and D. Rizo. Challenging sources: a new dataset for omr of diverse 19th-century music theory examples. In *Proceedings of the 4th International Workshop on Reading Music Systems (WoRMS 2022)*, November 2022.
15. A. Nuñez-Alcover, P.J. Ponce de León, and J. Calvo-Zaragoza. Glyph and position classification of music symbols in early music manuscripts. In *Proc. of the 9th Iberian Conference on Pattern Recognition and Image Analysis, LNCS vol. 11867*, pages 159–168, Madrid, Spain, July 2019. AERFAI, APRP, Springer.
16. A. Pacha, K. Choi, B. Cotiasnon, Y. Riquebourg, R. Zanibbi, and HorsH. Eidenberger. Handwritten music object detection: Open issues and baseline results. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 163–168. IEEE, 2018.
17. L. Pugin, J. Hockman, J.A. Burgoyne, and I. Fujinaga. Gamera Versus Aruspix: Two optical music recognition approaches. In *9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, USA, September, 2008*, pages 419–424, 2008.
18. L. Pugin, R. Zitellini, and P. Roland. Verovio - A library for Engraving MEI Music Notation into SVG. In *International Society for Music Information Retrieval*, jan 2014.
19. D. Rizo, J. Calvo-Zaragoza, and J.M. Iñesta. MuRET: A music recognition, encoding, and transcription tool. In *Proceedings of the 5th International Conference on Digital Libraries for Musicology*, DLfM '18, page 52–56, New York, NY, USA, 2018. Association for Computing Machinery.
20. D. Rizo, B. Pascual, J.M. Iñesta, A. Ezquerro, and L.A. González. Towards the digital encoding of hispanic white mensural notation. *Anuario Musical*, (72):293–304, 2017.
21. D. Rizo, N. Pascual-León, and C. S. Sapp. White mensural manual encoding: from humdrum to mei. *Cuadernos de Investigación Musical*, (6):373–393, 2018.
22. P. Roland. The music encoding initiative (MEI). In *Proceedings of the First International Conference on Musical Applications Using XML*, pages 55–59, jan 2002.
23. E. Selfridge-Field, editor. *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, Cambridge, MA, USA, 1997.
24. J. Sober-Mira, J. Calvo-Zaragoza, D. Rizo, and J.M. Iñesta. Pen-based music document transcription with convolutional neural networks. In *Graphics Recognition, Current Trends and Evolutions - 12th IAPR International Workshop, GREC 2017, Kyoto, Japan, November 9-10, 2017, Revised Selected Papers*, volume 11009 of *Lecture Notes in Computer Science*, pages 71–80. Springer, 2017.
25. J. Stoessel, D. Collins, and S. Bolland. Using optical music recognition to encode 17th-century music prints: The canonic works of paolo agostini (c.1583–1629) as a test case. In *7th International Conference on Digital Libraries for Musicology*, DLfM 2020, page 1–9, New York, NY, USA, 2020. Association for Computing Machinery.
26. A. Torrente and A. Llorens. The Musicology Lab: Teamwork and the Musicological Toolbox. In *Music Encoding Conference Proceedings 2021*, pages 9–20. Humanities Commons, 2022.

# Verse Generation by Reverse Generation Considering Rhyme and Answer in Japanese Rap Battles

Ryota Mibayashi<sup>1</sup>, Takehiro Yamamoto<sup>1</sup>, Kosetsu Tsukuda<sup>2</sup>, Kento Watanabe<sup>2</sup>,  
Tomoyasu Nakano<sup>2</sup>, Masataka Goto<sup>2</sup>, and Hiroaki Ohshima<sup>1</sup> \*

<sup>1</sup> Graduate School of Information Science, University of Hyogo

<sup>2</sup> National Institute of Advanced Industrial Science and Technology (AIST)

af22h007@guh.u-hyogo.ac.jp; t.yamamoto@gsis.u-hyogo.ac.jp;

k.tsukuda@aist.go.jp; kento.watanabe@aist.go.jp;

t.nakano@aist.go.jp; m.goto@aist.go.jp; ohshima@ai.u-hyogo.ac.jp

**Abstract.** Rap battle is a competition in which two rappers improvise rap verses alternately, and a verse is composed of multiple sentences uttered in one turn by a rapper. In this paper, we propose a method for generating response verses that are semantically related and *rhyme* with the opponent’s verse in rap battles. Our approach uses a language generation model BERT2BERT to generate rap sentences and constructs a verse by appropriately arranging them using a BERT model. When generating rap sentences, it is important to include words that *rhyme* with a specific word in the opponent’s verse, but it is difficult to include such words using a conventional sentence generation model that generates sentences in a forward direction from the beginning of the sentence. To address this issue, our proposed method trains the model to generate sentences in a reverse direction from the end of the sentence, which enables the model to generate rap sentences that highly likely have *rhymes* at the end. To train the model, we constructed our own rap battle corpus consisting of 6,791 verses. Our experimental results demonstrate that our proposed method outperforms a method that uses a conventional model generating sentences in a forward direction.

**Keywords:** Rap battle, Verse generation, BERT2BERT, Rhyme, Answer

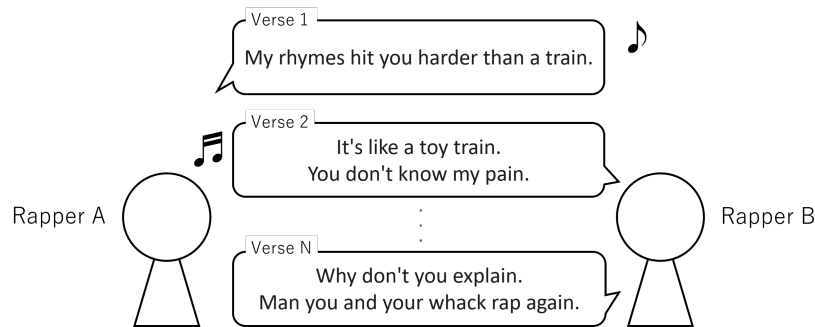
## 1 Introduction

A rap battle is a competition where two rappers perform improvised rap alternately. The rap that is delivered in one turn is called a verse, and a single verse is generally composed of several rap sentences. Figure 1 shows an example of a rap battle. Rapper *A* delivers the verse “*My rhymes hit you harder than a train.*” Rapper *B* responds to that verse with “*It’s like a toy train. You don’t know my pain.*” The rap battle ends

\* This work was supported by JSPS KAKENHI Grant Numbers JP21H03775, JP21H03774, JP21H03554, JP22H03905.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



**Fig. 1.** An example of a rap battle.

after repeating these responses several times. In rap battles, the winner is determined by the audience or judges. The quality of the verse is one of the important factors in determining the winner.

The quality of the verse is typically determined by *rhymes*, *answers*, and *flows* [1]. *Flow* refers to the expression style, such as rhythm and accentuation, of the rapping (singing) voice by a rapper. On the other hand, *rhymes* and *answers* depend on the text of the verse, as shown below.

- *Rhyme* refers to a pair of words with the same vowel sequence. For example, the pair of “train” in verse 1 and “pain” in verse 2 of Figure 1 is a *rhyme* because both of them have the same vowel sequence “ein.”
- *Answer* refers to a response verse that is related to the opponent’s verse. For example, the verse “It’s like a toy train.” in verse 2 of Figure 1 is an *answer* because it is semantically related to the verse 1.

High-quality verses need to reflect both *rhyme* and *answer*, and in rap battles, it is necessary to respond immediately with such verses to the opponent’s verses.

Rap battle competitions are held in many countries. For example, there is the “*True Freestyle Rap Battle*” in the U.S. and the “*UNPRETTY RAPSTAR*” rap battle TV program in Korea. In Japan, rap battle competitions such as “*Gaisen MCBATTLE*” and “*Sengoku MCBATTLE*” are also popular. In addition to such competitions in which professional rappers participate, there are also many competitions in which amateur rappers participate. In amateur rap battles, it is also important to respond to a verse with *rhymes* and *answers*, but it is not easy, especially for beginners, to come up with such a verse immediately in response to an opponent’s verse. If there were a system that could automatically generate high-quality verses for any given verse, such people would be able to practice rapping by referring to the generated verses.

In this paper, as a first step toward realizing such a system, we propose a method for generating a verse with both *rhymes* and *answers* in a rap battle. We do not deal with *flows* since we focus on the text of the verse. In the proposed method, a language generation model, BERT2BERT, is used to generate the verse. More specifically, the method is composed of two steps: generating rap sentences and ordering rap sentences. In the first step, given a sentence in the opponent’s verse as input, we generate rap sentences

considering the *rhyme* based on the input sentence. While sentence generation models usually generate a sentence in the forward direction from the beginning of the sentence, we generate a rap sentence in the reverse direction from the end of the sentence so that it can contain a *rhyme* at the end. Although there are a large number of words that satisfy a particular *rhyme*, our method searches for appropriate words based on the opponent's verse so that the *answer* can also be taken into account. Since multiple rap sentences are generated in this step, the second step aims to construct an appropriate verse by ordering the generated rap sentences by predicting next sentences using BERT [2].

Our contributions can be summarized as follows.

- We propose a verse generation method that takes into account *rhymes* and *answers* in rap battles and generates sentences from the end of the sentence.
- To train the model of the proposed method, we develop a training corpus consisting of 6,791 rap battle verses.
- We conduct evaluation experiments and show that the proposed method outperforms a method that generates sentences from the beginning of the sentence.

## 2 Related Work

### 2.1 Text Generation

In recent text generation methods, deep learning is commonly used. For example, in the early stages, seq2seq [3] models using autoregressive structures such as RNN [4] and LSTM [5] were employed. The seq2seq model has an Encoder that aggregates text information and a Decoder that generates texts from the aggregated information, and is also known as an Encoder-Decoder model. Subsequently, a mechanism called Attention that generates texts by selecting text information effectively was proposed. More recently, a model called Transformer [6], which uses Attention, has been proposed. Transformer is an Encoder-Decoder model with a large number of parameters and Attention.

BERT and GPT-2 are well-known models that use Transformer. BERT [2] is a general-purpose feature extractor for natural language texts and is a multi-layered Encoder-only model of Transformer. It can consider the context of the text because it takes account of the information of the text from both directions. GPT-2 [7] is a multi-layered Decoder-only model that is commonly used in text generation tasks. Different from BERT, GPT-2 considers the information of the text from one direction only. These models are generally used after having acquired general-purpose linguistic knowledge by training on a large corpus in advance. By fine-tuning a pre-trained model for a specific task, a model specialized for that task can be learned.

BERT2BERT [8] is a generative model that transfers the pre-trained parameters of BERT to the Encoder and Decoder of Transformer. It has been shown to be effective in the generation tasks to use the pre-training weights obtained by the BERT model [2]. Based on the usefulness of the BERT model, we leverage the BERT model to generate the text of rap verses.



**Table 1.** Differences between existing studies and ours.

	Containing <i>rhymes</i>	Generating text for rap battles	Using rap battle data
Rapformer [9]	✓	-	-
GhostWriter [10]	✓	-	-
DopeLearning [11]	✓	-	-
Manjavacas et al. [12]	✓	-	-
DeepRapper [13]	✓	-	-
Wu et al. [14]	✓	✓	-
Shimon [15]	✓	✓	-
Ours	✓	✓	✓

## 2.2 Rap Generation

In research aimed at generating rap lyrics, some rhyme-aware methods have been proposed that replace words in the generated lyrics with other words that *rhyme* [9], or generate rhyming lyrics by learning from lyrics that contain *rhymes* [10–13]. Although those previous methods are shown to be effective in rap generation, they are not sufficient for our purpose of rap battles because of a limited word choice. In rap battles, since the opponent’s verse would contain a wide range of topics, it is important to generate a response verse using a wide range of words containing a specified *rhyme*.

Rapformer [9], for example, is a method that first uses the Transformer model [6] to generate a lyric sentence and then replaces the last word of the generated sentence with another rhyming word. Since this replacement cannot change the other words in the generated sentence, the replaced rhyming word must be semantically consistent with the other words, thus limiting word choice.

The same limitation exists in other rap generation methods that use deep learning models to learn *rhymes* in lyrics [10–12]. Potash et al. [10] proposed a method called Ghostwriter that uses LSTM models to generate new rap lyrics from existing rap lyrics data. DopeLearning proposed by Malmi et al. [11] selects the most appropriate rap sentence by comparing the similarity between the generated rap sentence and the previous one. Manjavacas et al. [12] proposed a method for generating rap lyrics by using LSTM as well. Those methods generate words in order starting from the first word to reach the last word. The choice of the last word is limited since it should be semantically consistent with the previous words. On the other hand, Xue et al. [13] proposed a method called DeepRapper that uses a Transformer-Decoder to consider the beat when generating rap verses. DeepRapper generates a sentence in the reverse direction and the above limitation is mitigated. However, since DeepRapper generates multiple lyric sentences at once, the choice of rhyming words in the second and subsequent sentences is still limited by the context of the previous sentence.

With a focus on rap battles, there have been a few studies on generating rap battle verses. Wu et al. [14] developed a chatbot system for rap battles that uses a modified version of RAAM (Recursive Auto-Associative Memory) called TRAAM (Transduction Recursive Auto-Associative Memory) for generating verses. Shimon proposed by Savery et al. [15] generates rap sentences by using LSTM based on existing rap lyrics

data, and then rearranges them to create a verse based on keywords in the input verse. These studies used their own hip-hop lyric corpora, but did not use a rap battle corpus since there was no existing corpus for rap battles.

Table 1 summarizes the studies introduced above. Our study differs from them in that our method can generate a response verse using a wide range of rhyming words because it generates a rap sentence in the reverse direction starting from a rhyming word at the end of the sentence. Moreover, we develop our own corpus specialized for rap battles and use it for training our sentence generation model.

### 3 Verse Generation Method for Rap Battles

An overview of the proposed verse generation method specialized for rap battles is shown in Figure 2. Since rhyming is indispensable in rap battles, our method takes into account the following process through which rappers typically create a rhyming response verse to the opponent’s verse. First, based on the last word in the opponent’s verse, rappers decide on a word to use as a *rhyme*. It is also important to choose a word that can serve as an *answer* to the opponent’s verse. Then, they create a rap verse that includes that word.

With reference to this process, our method first finds candidate words that have *rhyme* vowels based on the opponent’s verse and selects a set of semantically related words from them (section 3.2). It then generates a rap sentence by using each of the selected words, resulting in a set of the rap sentences (section 3.3). Finally, following the previous approach of arranging the sentences to generate rap lyrics [11], the method constructs a response verse by appropriately arranging (deciding the order of) the generated rap sentences (section 3.4).

#### 3.1 Rap Battle Corpus

**Rap Battle Corpus** Our rap battle corpus was created by transcribing videos of rap battles in Japanese by using the crowdsourcing service Lancers<sup>3</sup>. The videos of rap battles were selected from the following three popular YouTube channels with over 100,000 subscribers: *UMB*<sup>4</sup>, *Gaisen MCBATTLE*<sup>5</sup>, and *Sengoku MCBATTLE*<sup>6</sup>. Workers hired through Lancers were native Japanese speakers and were not required to be familiar with rap battles. They accessed a web page that we created for the task and transcribed all the verses in the rap battles while watching the specified videos on YouTube. In total, 691 rap battles were transcribed by 194 workers.

To expand the size of the corpus, we also transcribed rap battles that were broadcasted on two TV shows: *High School Rap Championship* and *Freestyle Dungeon*. Although these videos were publicly available on the web, viewing them required payment. Hence, instead of using crowdsourcing, we had one university student transcribe 596 rap battles.

<sup>3</sup> <https://www.lancers.jp/>

<sup>4</sup> <https://www.youtube.com/user/umbofficial>

<sup>5</sup> [https://www.youtube.com/channel/UCe\\_EvY8GrvYgx8PbwRBc75g](https://www.youtube.com/channel/UCe_EvY8GrvYgx8PbwRBc75g)

<sup>6</sup> <https://www.youtube.com/user/senritumc>

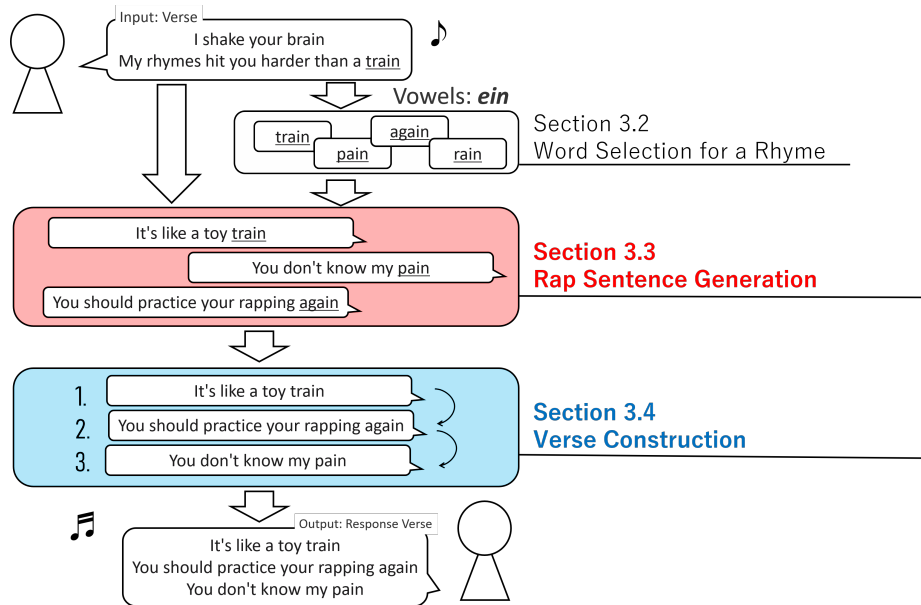


Fig. 2. Overview of the proposed verse generation method.

The final statistics for the corpus are as follows: 1,287 rap battles, 6,791 verses, 52,422 sentences in verses, an average of 7.71 sentences per verse, and an average of 15.40 tokens (words) per sentence. All the raps in the videos are performed in Japanese.

**Rap Sentence Pair Data** To train a model for generating rap sentences, we need pairs of input and output rap sentences. Therefore, we created pairs of rap sentences by taking two consecutive sentences in a verse. For example, if a verse consists of three sentences, two pairs of rap sentences are created (a pair of the first and second sentences and a pair of the second and third sentences). In total, we created 22,125 pairs of rap sentences from the 6,791 verses in the corpus (hereafter we refer to the pair data “rap sentence pair data”).

### 3.2 Word Selection for a Rhyme

As described in section 3.1, the proposed method first selects words to use as *rhymes* based on the opponent’s verse. This process consists of the following three steps.

First, a word that *rhymes* is extracted from the opponent’s verse. Rhyming with the final word in the opponent’s verse is highly valued in rap battles because it requires a rapper’s high ability to respond quickly right after listening to it. Therefore, in this study, the Japanese morphological analysis module MeCab<sup>7</sup> is used to divide the opponent’s

<sup>7</sup> <https://taku910.github.io/mecab/>

verse into morphemes, and the last noun in the verse is extracted as the target word to *rhyme* with.

Next, words that have the same vowel sounds as the target word are searched as candidate words that *rhyme*. We use the words included in the AWD-J dictionary<sup>8</sup> for this search. All the words in the dictionary are converted to vowel sequences in advance, and words with the same vowel sequence as the target word are searched. If the number of vowels in the target word is less than four, a word that has the same vowel at the end is searched. If the number of vowels is four or more, at least four vowels matching from the end are searched to relax the search constraints.

Finally, words that are semantically related to the target word are selected from the searched words. To compute semantic relationships, Japanese word embeddings learned with fastText<sup>9</sup> are used, and the top seven words (i.e., the most semantically related seven words) in terms of the cosine similarity with the target word are selected. In addition, to ensure that one of the generated rap sentences is a definite *answer* to the opponent's verse, the target word itself is also added, resulting in a total of eight words.

### 3.3 Rap Sentence Generation

After selecting eight words to use as *rhymes*, the next step is to generate a rap sentence that includes each of these words. In general, when generating sentences using a Decoder (forward generation model), words are output from the beginning to the end of the sentence (Figure 3 (a)). However, it is difficult to generate a sentence that must end with the word selected as a *rhyme*. We therefore propose a method for generating a sentence by outputting words in reverse order from the end to the beginning of the sentence. For this method, we first train a rap sentence generation model (reverse generation model) using the reversed sentences (Figure 3 (b)). The method then uses the trained model to generate a rap sentence using the selected word as the starting token (Figure 3 (c)). The details are described below.

For rap sentence generation, we used a generation model BERT2BERT [8], which transfers the pre-trained parameters of the BERT model to the Encoder and Decoder of the Transformer. As the pre-trained BERT model for both the Encoder and Decoder, we used the model publicly available from Tohoku University<sup>10</sup>, which were pre-trained on Japanese Wikipedia text data.

To fine-tune BERT2BERT, we used the rap sentence pair data created in section 3.1. First, as shown in Figure 3 (b), each sentence in the pair is divided into tokens using a tokenizer<sup>11</sup>. Next, the first sentence in the pair is used as the input sentence for the Encoder by arranging the tokens in forward order, and the second sentence is used as the output sentence for the Decoder by arranging the tokens in reverse order. This enables

<sup>8</sup> <https://sociocom.naist.jp/awd-j/>

<sup>9</sup> <https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ja.300.vec.gz>

<sup>10</sup> <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

<sup>11</sup> <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

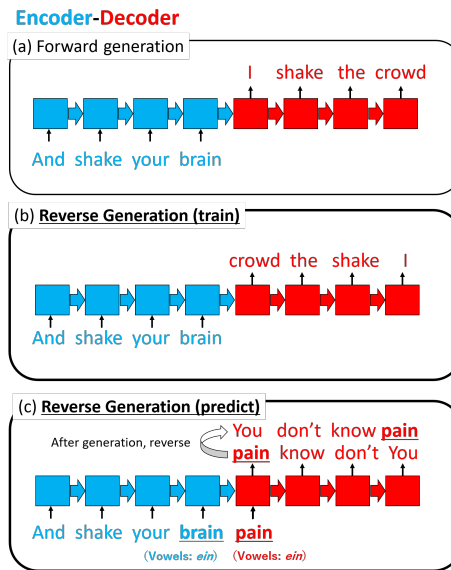


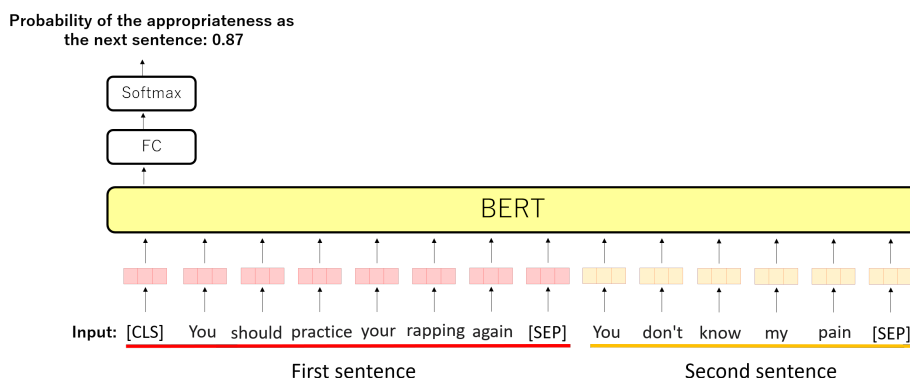
Fig. 3. Differences between forward and reverse generation models.

the model to learn to generate sentences in reverse order. The following hyperparameters were used to train the Encoder and Decoder: a batch size of 32, Adam optimizer, Cross Entropy Loss function, a learning rate of  $2e-7$ , a dropout rate of 0.1, max length of 128, and early stopping with a patience of 10. To train the model, we divided the rap sentence pair data into train, validation, and test data with an 8:1:1 ratio. The training was completed in 224 epochs.

Finally, we used the trained model to generate a token sequence by inputting the last sentence of the opponent's verse to the Encoder and specifying one of the words selected in section 3.2 as the initial token for the Decoder. Then, by reversing and concatenating the generated token sequence, a rap sentence is generated for each of the eight selected words. The parameters used for generation were as follows: max length of 30, top k of 10, top p of 0.95, and no repeat n-gram size of 2.

### 3.4 Verse Construction

In section 3.3, because eight rap sentences are independently generated, we need to arrange them in an appropriate order to construct a verse. To do this, we first train a BERT model through the next sentence prediction task in which the model predicts whether two given sentences are appropriate as consecutive sentences (Figure 4). The sentence pairs in the rap sentence pair data was used as positive examples, while negative examples were created by replacing the second sentence of each sentence pair in the rap sentence pair data with a randomly selected second sentence of another pair. We created 10,748 pairs of sentences (5,374 positive examples and 5,374 negative examples) and divided them into train, validation, and test data at a ratio of 8:1:1. We again used the



**Fig. 4.** The next sentence prediction task.

Japanese pre-trained BERT model released by Tohoku University<sup>12</sup>. The hyperparameters during training were as follows: a batch size of 8, Adam optimizer, Cross Entropy Loss function, a learning rate of  $2e-7$ , a dropout rate of 0.1, max length of 128, and early stopping with a patience of 10. The training was completed in 30 epochs, and the accuracy on the test data was 0.60.

Using the trained BERT model, we construct the response verse. Our method first selects the rap sentence containing the target word used at the end of the opponent's verse as the first sentence of the response verse. It then uses the trained BERT to select the most appropriate sentence (i.e., the sentence with the highest estimated probability of being suitable) among the remaining seven sentences as the second sentence of the verse. It repeats this selection process: it selects the most appropriate remaining sentence as the  $n + 1$ th sentence next to the  $n$ th sentence ( $2 \leq n \leq 7$ ). It thus constructs the response verse consisting of the eight sentences.

## 4 Evaluation

Quantitative and qualitative evaluations were conducted to present the effectiveness of the proposed method. The quantitative evaluation was based on three aspects: the naturalness of rap, the quality of *rhyme*, and the quality of *answer*. In the qualitative evaluation, we compared the forward and reverse generation results.

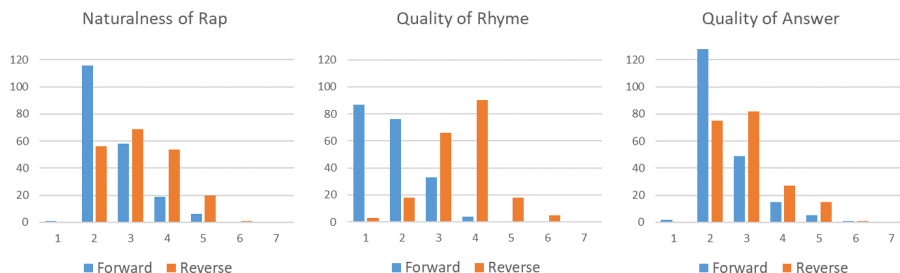
### 4.1 Quantitative Evaluation

To verify the usefulness of generating sentences in reverse order from the end of the sentence, a comparison was made with the method of generating sentences in forward order from the beginning of the sentence. In the comparison method, all processes except for the direction of sentence generation were the same as the proposed method. The

<sup>12</sup> <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

**Table 2.** The average scores of the three aspects for the proposed and comparison methods.

method	Naturalness of rap	Quality of <i>rhyme</i>	Quality of <i>answer</i>
Forward generation (comparison method)	2.56	1.77	2.48
Reverse generation (proposed method)	<b>3.20</b>	<b>3.58</b>	<b>2.92</b>

**Fig. 5.** The score distribution of the three aspects for the proposed (“Reverse”) and comparison (“Forward”) methods.

hyperparameters used for training the comparison method were also the same. First, we randomly selected 100 of the 1,287 rap battles in the rap battle corpus, and used the last sentence of each rap battle as the input sentence (i.e., the last sentence of the verse). Then, for each input sentence, one verse was generated using the proposed method and the comparison method, resulting in a total of 200 verses generated for the 100 input sentences. Note that the last sentence of each rap battle was not input to the Encoder during the training, so this experiment treated the sampled 100 sentences as unknown data. Two evaluators with over five years of experience in watching rap battles evaluated each of the 200 verses based on a 7-point scale (1: very poor to 7: very good) for each of the three aspects: the naturalness of rap, the quality of *rhyme*, and the quality of *answer*.

Table 2 shows the average score of the three aspects for the proposed (reverse generation) and comparison (forward generation) methods. As shown in the table, the proposed method outperformed the comparison method in all the aspects. To analyze the differences in the results between the methods in more detail, the score distribution of the three aspects is shown in Figure 5. As the proposed method considerably outperformed the comparison method in the quality of *rhyme* in Table 2, Figure 5 also shows that the proposed method generated many verses with scores of four or higher, and that the comparison method rarely generated verses with scores of four or higher. Since all processes except for the generation direction suitable for *rhymes* were the same in the proposed and comparison methods, as expected, their score differences were smaller in the naturalness of rap and the quality of *answer* than in the quality of *rhyme* in Table 2. However, as shown in Figure 5, the comparison method generated a large number of

<p>Opponent's verse:</p> <p>リズム倍で乗ってはめて (Riding on the rhythm, fitting in.) 遊ぶだけは俺もできる (I can play just for fun too.) こんな奴のスキルならば更に上に行く (My skills go beyond you.) ニガリを越すため3連覇 (To surpass Nigari, I'll aim for a triple crown.) お前全然足りない鍛錬が (You haven't trained nearly enough.) おいおい俺にアンサーがない (Hey, you say I don't have an answer.) ってアンサーがないのは (But the one without an answer is you.) あんたじゃない勝ちたいなら願 (If you want to win, use rhyme.) 沢山踏もうコイツは f a k e 拡散希望 (KIBOU) (Let's use a lot of rhymes, this guy is fake, I hope it spreads.)</p>	<p>Response verse generated by the proposed method:</p> <p>俺のネタにしてもマジで腐って俺はここに拡散希望 (KIBOU) (My jokes are seriously rotten, and I want them to spread here.) そうじゃない俺はイケてるマジで政治志望 (SIBOU) (Not like that, I'm seriously interested in politics.) 何度も言って俺を否定してるんじゃないよみんなが見てきた理想 (RISOU) (I'm not saying it over and over again, everyone has seen the ideal.) 口だけは教えてくれると思っただけで俺は否定して見てるだけの未来志向 (SIKOU) (I don't expect anyone to teach me with just words, I'm just looking towards the future.) 言っただけじゃあ俺が持っているだけの魅力 (MIRYOKU) (I didn't just say it, I have the charm.) そんなことは絶対に言わないって言われても興味はない俺の利用 (RIYOU) (Even if you say you'll never say something like that, I'm not interested in using it.) ラップして確かに無限の意欲 (IYOKU) (Rapping with infinite motivation for sure.) お前が負けるのは俺は自分の意向 (IKOU) (You'll lose because I have my own intentions.)</p>
	<p>Response verse generated by the comparison method:</p> <p>俺が勝ち上がることはない (I won't be able to climb to the top.) 俺が勝つのを見たぜ (But I saw myself win.) 俺は俺が優勝した (I won the championship.) 俺も俺のやり方ではない (I'm not doing it my way either.) 俺はそうやってるのは (I'm doing it like this.) この場でやってるぜ (I'm doing it here.) 俺がラップはこのままでやってきてるんだぜ (I'll keep rapping like this.) 俺はラップは上手いんだ (I'm good at rapping.)</p>

Fig. 6. Examples of response verses generated by the proposed and comparison methods. Since verses are generated in Japanese, the English translation is shown in parentheses.

verses with a score of two, while the proposed method generated fewer verses with a score of two, indicating that the proposed method has the advantage that low-quality verses are less likely to be generated. From these results, it was demonstrated that generating verse sentences in reverse order from the end of the sentence is indeed effective.

## 4.2 Qualitative Evaluation

The upper and lower rows of Figure 6 show examples of response verses generated by the proposed and comparison methods, respectively. In the example in the upper row, the vowels of the words that need to *rhyme* with the opponent's verse are “*iou*” and highlighted with red color. The proposed method reflects this *rhyme* by ensuring that each sentence in the response verse ends with a word that has the vowels of “*iou*.” Despite imposing the constraint of rhyming in each sentence, meaningful sentences can be generated and are suitable as *answers*. In contrast, none of the sentences in the comparison method's verse end with a word that has the vowels of “*iou*.” These results also clearly demonstrate the usefulness of learning and generating sentences in reverse order.

## 5 Conclusion

In this paper, we proposed a verse generation method that takes into account *rhymes* and *answers* in rap battles and verified its effectiveness. Although we used a Japanese



rap battle corpus, the proposed method itself is language-independent, and we would like to verify its usefulness in other languages such as English in the future. To construct response verses more flexibly, future work will also include the extension of our method to find a rhymed word that is different from the last noun in the verse and is strongly related to the opponent's verse, or to take consonant similarity [16] into consideration in the case of Japanese rap [17]. Finally, as mentioned in section 1, since our future goal is to support people who are unfamiliar with rap to practice it, we would like to develop an interactive verse generation system equipped with the proposed method and verify its usefulness in training support.

## References

1. Venla, S. Interactive Oral Composition: Resources, Strategies, and the Construction of Improved Utterances in a Finnish Freestyle Rap Battle. *The Journal of American Folklore*, vol.132, no.523, pp. 3–35 (2019).
2. Devlin, J., Chang, M., Lee, K., and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL'19*, pp. 4171–4186 (2019).
3. Sutskever, I., Vinyals, O., and Le, Q.: Sequence to Sequence Learning with Neural Networks. *Proceedings of NIPS'14*, pp. 1–9 (2014).
4. Rumelhart, D., Hinton, G., and Williams, R.: Learning Internal Representations by Error Propagation. tech. rep., University of California, San Diego (1985).
5. Gers, F., Schmidhuber, J., and Cummins, F.: Learning to Forget: Continual Prediction with LSTM. *The Journal of Neural computation*, vol.12, no.10, pp. 2451–2471 (2000).
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, U., and Polosukhin, I.: Attention Is All You Need. *Proceedings of NIPS'17*, pp. 6000–6010 (2017).
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I.: Language Models Are Unsupervised Multitask Learners. tech. rep., OpenAI (2019).
8. Sascha, R., Shashi, N., and Aliaksei, S.: Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Proceedings of TACL'20*, pp. 264–280 (2020).
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality. arXiv preprint arXiv:1310.4546 (2013).
10. Potash, P., Romanov, A., and Rumshisky, A.: Ghostwriter: Using an LSTM for Automatic Rap Lyric Generation. *Proceedings of EMNLP'15*, pp. 1919–1924 (2015).
11. Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A.: DopeLearning: A Computational Approach to Rap Lyrics Generation. *Proceedings of SIGKDD'16*, pp. 195–204 (2016).
12. Manjavacas, E., K., , and Karsdorp, F.: Generation of Hip-Hop Lyrics with Hierarchical Modeling and Conditional Templates. *Proceedings of INLG'19*, pp. 301–310 (2019).
13. Xue, L., Song, K., Wu, D., Tan, X., Zhang, N., Qin, T., Zhang, W., and Liu, T.: DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling. *Proceedings of ACL'21*, pp. 69–81 (2021).
14. Wu, D. and Addanki, K.: Learning to Rap Battle with Bilingual Recursive Neural Networks. *Proceedings of IJCAI'15*, pp. 2524–2530 (2015).
15. Savery, R., Zahray, L., and Weinberg, G.: Shimon the Rapper: A Real-Time System for Human-Robot Interactive Rap Battles. *Proceedings of ICCV'20*, pp. 212–219 (2020).
16. Kawahara, S. Half rhymes in Japanese rap lyrics and knowledge of similarity. *The Journal of East Asian Linguistics*, vol.16, no.2, pp. 113–144 (2007).
17. Manabe, N. Globalization and Japanese creativity: Adaptations of Japanese language to rap. *The Journal of Ethnomusicology*, vol.50, no.1, pp. 1–36 (2006).

# Combining Vision and EMG-Based Hand Tracking for Extended Reality Musical Instruments\*

Max Graf<sup>1</sup> and Mathieu Barthe<sup>1</sup>

Centre for Digital Music, Queen Mary University of London  
{max.graf, m.barthe}@qmul.ac.uk

**Abstract.** Hand tracking is a critical component of natural user interactions in extended reality (XR) environments, including extended reality musical instruments (XRMI). However, self-occlusion remains a significant challenge for vision-based hand tracking systems, leading to inaccurate results and degraded user experiences. In this paper, we propose a multimodal hand tracking system that combines vision-based hand tracking with surface electromyography (sEMG) data for finger joint angle estimation. We validate the effectiveness of our system through a series of hand pose tasks designed to cover a wide range of gestures, including those prone to self-occlusion. By comparing the performance of our multimodal system to a baseline vision-based tracking method, we demonstrate that our multimodal approach significantly improves tracking accuracy for several finger joints prone to self-occlusion. These findings suggest that our system has the potential to enhance XR experiences by providing more accurate and robust hand tracking, even in the presence of self-occlusion.

**Keywords:** Extended reality, extended reality musical instruments, hand tracking, surface electromyography, deep learning

## 1 Introduction

Extended reality (XR) is an umbrella term encompassing virtual, augmented and mixed reality (VR/AR/MR). In recent years, the increased popularity of XR technology has seen the establishment of extended reality musical instruments (XRMI) as a research field [23]. Milgram et al. described the reality-virtuality continuum [21], along which digital applications can be placed. It stretches from real-world environments to fully virtual environments. Head-mounted XR devices bridge this continuum. They are capable of rendering three-dimensional imagery onto screens, removing the necessity for separate monitors or mobile displays, blending the real and virtual worlds together. The rapid development of XR technologies has opened up new possibilities for musical creation, performance, and interaction, with the emergence of various XR-based musical instruments and applications. Many XRMI follow an embodied interaction paradigm. These instruments offer novel opportunities for artists to experiment with embodied interaction techniques, spatial sound design, and immersive performances, thus expanding the boundaries of traditional music making. XRMI fall within the larger category

\* This work was supported by the UKRI Centre for Doctoral Training in AI & Music [grant number EP/S022694/1].



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

of digital musical instruments (DMIs). [27] suggest that the control of DMIs can be made intimate (personal and familiar) by using appropriate control metaphors, low latency action-to-sound, and continuous gesture recognition. This study is part of a larger project that aims to support control intimacy in XRMIs.

Based on the current state of XR technology and prior work in XRMIs [3, 9], we highlight that gesture sensing errors on XR devices are a bottleneck for intimate musical control. Head-mounted XR devices (HMDs) rely on a set of sensors to record data and provide embodied control interfaces for users, e.g., head-tracking, hand tracking, and body pose detection. The transduction of these real-world sensor data to digital representations depends on computational methods. In this work we focus on the problem of hand tracking, more specifically, accurate tracking of finger joints. Hand-tracking algorithms often use visual information from camera sensors in conjunction with machine learning techniques, for example, in the Oculus Quest 2 device [11]. The accuracy of vision-based hand-tracking algorithms may be high [26], but current recognition rates do not reach 100%. Self-occlusion - the occlusion of finger joints by other parts of the hand - as well as challenging lighting situations lead to failure cases in vision-based tracking systems. Such error cases may produce instances of jitter, tracking loss, or glitches in the virtual representation of the hands, which can have detrimental effects on the usability and user experience in XRMIs, as shown in a previous study [9].

This work aims to address such sensing-related issues through the use of surface electromyography (sEMG) sensors. EMG sensors measure the electrical potential produced during muscle contractions in the body. Surface electromyograms can be obtained through electrodes that are positioned on the surface of the skin, above muscle tissue regions. We present an investigation into the potential of sEMG sensors and deep learning models to enhance hand-tracking accuracy in XRMIs. Our approach combines sEMG data and vision-based tracking methods to address sensing-related issues commonly encountered in XRMI performance. Thereby, we aim to improve the tracking accuracy and responsiveness of XR musical instruments, especially in situations where vision-based tracking falls short.

The scope of this paper is limited to the exploration of sEMG and deep learning techniques for hand-tracking in XR musical instruments. While our findings may have broader applications in other areas of XR interaction, the primary focus is on the improvement of XRMI design and user experience. Through our work, we aim to contribute to the ongoing development of more accurate, intuitive, and expressive extended reality musical instruments.

## **2 Background**

The development of XRMIs has attracted growing interest as VR, AR, and MR technologies continue to advance. Early studies in this domain focused on the design and evaluation of virtual interfaces for musical performance and interaction [19, 23, 8, 22]. Several works investigated user experience [6, 5], interaction techniques [2] and collaborative music making [20, 10]. More recent studies have explored the creation of novel instruments and control schemes [5, 3, 4, 9]. While there is no gold standard for XRMI design, many XRMIs rely on hand-tracking to facilitate embodied interaction with the

instrument [22, 8, 4, 9]. Various vision-based tracking methods are employed, including depth-sensing cameras [22, 8, 4], and machine learning-based approaches [11]. Despite the progress in hand tracking research, limitations such as occlusion, lighting issues, and computational complexity continue to pose challenges for hand-controlled XRMI applications.

Surface electromyography (sEMG) has emerged as a promising alternative to vision-based tracking methods for capturing user input in various applications, including XR. Several studies have explored the use of sEMG data and deep learning architectures for hand gesture recognition [18, 17, 1, 16]. These works have reported promising results, highlighting the potential of employing sEMG and deep learning models for precise finger movement estimation. However, some of these works depend on complex tracking setups [1] or leverage low-resolution tracking data for training [16].

A notable limitation of these studies is the lack of shared training data and code, hindering the reproducibility and comparability of the results across different research efforts. Several datasets on the topic of finger joint angle estimation through sEMG data have been published. However, they either require specialised sEMG measuring equipment [13, 15, 14], making the reproduction of results an expensive endeavour, or introduce temporal biases into the dataset due to lack of synchronisation during the recording procedure [12]. The absence of implementation details ([18, 17, 1]) makes it difficult for other researchers to build upon these works, potentially slowing down the progress in the field.

The potential benefits of integrating sEMG data and deep learning models with vision-based tracking methods has not been thoroughly investigated in the context of XRMIs. While machine learning has found its way into the NIME community [24], the use of machine learning approaches to improve XRMI control remains an under-explored area. This study aims to develop a multimodal hand tracking approach that leverages sEMG data, deep learning models, and vision-based tracking techniques. We share the training data and code<sup>1</sup>, fostering further research and innovation in the domain of sEMG-based neural interfaces.

### **3 Deep Learning Model for Finger Joint Angle Estimation**

We have developed a software pipeline for data collection, feature extraction and modelling of sEMG data. We focus on eight finger joints that are prone to self-occlusion: the metacarpophalangeal and proximal interphalangeal joints of the index, middle, ring and pinky fingers. Specifically, we want to model the rotations of the finger bones connected to these joints relative to the hand. The thumb is excluded from our investigation. Modelling thumb rotations with sEMG data is a hard problem, since the majority of muscles related to thumb movements are located in the hand, rather than the forearm. With that in mind, the goal of the model is to estimate the eight finger joint angles from a window of sEMG data.

---

<sup>1</sup> <https://github.com/maxgraf96/sEMG-myoe-unity>

### 3.1 Data Collection

We collect surface EMG signal measurements using the Thalmic Labs Myo armband<sup>2</sup> and vision-based hand tracking data from the Oculus Quest 2 XR headset. Both devices are employed simultaneously to capture the muscle activity and finger joint rotations, respectively. This approach allows users to capture data without the need for external tracking devices.

The Myo armband is a non-invasive wearable device that features eight sEMG sensors. The armband is worn on the forearm, with the sensors evenly distributed around the circumference of the arm, allowing it to capture the activity of the forearm muscles during finger movements. We obtain sEMG data from the Myo armband using the *Pyomyo* Python framework [25], extracting rectified and smoothed signals at a sampling frequency of 50Hz. The Oculus Quest 2 XR headset is equipped with four monochrome cameras that provide a wide field of view, enabling it to capture hand positions and movements. The built-in hand tracking algorithm [11] processes the camera data and estimates the 3D rotations of the user's hand joints in real time. In our system, we sample the hand joint rotations from the XR device at 50 Hz and synchronize them with the sEMG data from the Myo armband.

One researcher recorded hand gestures and movements in a controlled environment, with a focus on gestures relevant to XRMI interaction. This study should be seen as a proof-of-concept for our methodology. Hence, for this study, we focused on data from the right hand only. The gestures included various finger flexions and extensions, as well as combinations of multiple finger movements. They were performed at different speeds, forces, and orientations. We conducted three data collection sessions across three days to account for the natural variability in sEMG readings, ensuring a more robust dataset. The armband was fitted on the right forearm, covering the right flexor carpi radialis, flexor digitorum superficialis and the right extensor carpi radialis longus, as described in [7]. During the data collection process, the XR headset was strategically positioned in diverse locations and orientations to minimise self-occlusion of the hand. Data collection sessions lasted between ten and fifteen minutes, resulting in a substantial amount of synchronized sEMG and hand tracking data.

### 3.2 Feature Extraction

Figure 1 shows the data flow in our pipeline. The selection of features was informed by both the literature and a series of experiments. We use Python to extract both time domain and frequency domain features from the sEMG data. The pipeline takes 2D windows of sEMG samples, with  $N$  number of samples and  $C$  channels. We then compute the following features per channel: in the time domain, mean absolute value (MAV), root mean square (RMS), and variance (VAR); in the frequency domain, median frequency bin (MDF), mean frequency bin (MNF), and peak frequency bin (PF). Additionally, wavelet coefficients at the fourth level are extracted to provide further information about the signal's characteristics, as reported in [1]. Wavelet analysis provides a multi-resolution representation of the sEMG signal, capturing both the time and frequency

<sup>2</sup> <https://xinreality.com/wiki/Myo>

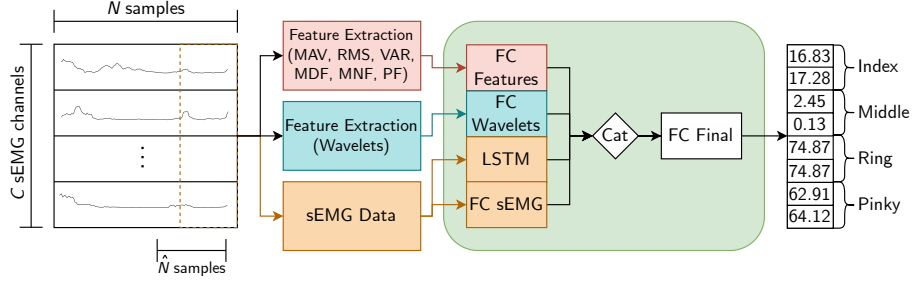


Fig. 1: Data preprocessing, feature extraction, and model pipeline

characteristics of the data. This comprehensive feature representation aims to capture essential information from the sEMG signals both locally and globally, enabling holistic representation of the data.

### 3.3 Model Architecture

The model architecture is a combination of a Long Short-Term Memory (LSTM) network and multiple separate sets of fully connected (FC) layers, aiming to capture both general trends in the sEMG signal data provided by the features and high-frequency characteristics of the signal. Formally, our model learns a mapping

$$F : \mathbb{R}^{N \times C} \rightarrow \mathbb{R}^M \quad (1)$$

where  $N$  denotes the number of sEMG samples,  $C$  represents the number of sEMG channels and  $M$  denotes the number of predicted joint angle values at every time step. More accurately, we learn a mapping

$$F(\mathbf{s}) = \phi_{\text{final}} \left( \phi_{\text{lstm}}(\mathbf{s}_{\hat{N}}) \oplus \phi_{\text{feat}}(\psi_{\text{time-freq}}(\mathbf{s}_N)) \oplus \phi_{\text{wav}}(\psi_{\text{wavelet}}(\mathbf{s}_N)) \oplus \phi_{\text{filt}}(\mathbf{s}_{\hat{N}}) \right) \quad (2)$$

where

- $F(\mathbf{s})$  represents the mapping function that takes  $N \times C$  sEMG samples ( $\mathbf{s}$ ) and outputs  $M$  finger joint angles.
- $\mathbf{s}_N$  denotes all  $N \times C$  sEMG samples.
- $\mathbf{s}_{\hat{N}}$  denotes the last  $\hat{N} \times C$  sEMG samples in the data point.
- $\phi_{\text{lstm}}$  represent the LSTM layers, and  $\phi_{\text{feat}}$ ,  $\phi_{\text{wav}}$  and  $\phi_{\text{filt}}$  represent the fully connected layers processing time/frequency domain features, wavelet features and filtered EMG data respectively.
- $\phi_{\text{final}}$  denotes the final set of fully connected layers.
- $\psi_{\text{time-freq}}$  and  $\psi_{\text{wavelet}}$  denote the feature extraction functions for time-frequency domain features and wavelet features, respectively.
- $\oplus$  represents the tensor concatenation operation.

The LSTM network, a type of recurrent neural network, operates on the subset  $s_{\hat{N}}$ , which contains the last  $\hat{N}$  samples of the EMG data, capturing the temporal dependencies within the most recent portion of the sEMG signal. LSTMs can effectively learn medium-to-long-range dependencies in time-series data and retain information across multiple time steps. Additionally,  $s_{\hat{N}}$  is fed into a separate fully connected layer, which was empirically found to improve the model's performance. The time and frequency domain features are computed over all  $N$  samples and processed by a set of fully connected layers. These layers are designed to extract higher-level representations from the sEMG features, capturing general patterns and trends in the data. The wavelet features are fed into a separate set of fully connected layers, allowing the model to learn distinct patterns associated with the wavelet coefficients. This additional information can help the model to better discriminate between different types of hand movements and gestures.

The outputs of the LSTM layers and the three sets of fully connected layers (time-frequency domain features, wavelet features, and  $\hat{N}$  sEMG samples) are concatenated and passed to a final set of fully connected layers. This combination of network components aims to capture a comprehensive representation of the sEMG signal, taking into account both general trends and high-frequency changes. The final output of the model is an estimation of the eight finger joint angles described above.

### 3.4 Model Implementation and Training

We selected  $N = 150$ , which gives a sampling window size of three seconds. Related works use window sizes of five seconds [18, 17]. During our experiments with the model architecture, we found that a smaller window size of 150 samples did not reduce the quality of the predicted finger joint angles, while yielding a performance gain in the data processing pipeline.  $\hat{N} = 50$  was selected as a trade-off between LSTM accuracy and performance. Higher values for  $\hat{N}$  produced slightly better results, but incurred a performance degradation, slowing down the model at inference time.

The collected data comprises approximately 80000 sEMG samples with corresponding finger joint angle measurements. The sEMG samples were separated into training and validation sets using a 90/10 ratio. Our deep learning model is built using the PyTorch framework. The model was trained on a single NVidia RTX 2080 Ti GPU for approximately 500000 steps with a batch size of 256. The mean squared error function was employed as a loss metric. We applied an exponentially decreasing learning rate, starting at 0.003 and reducing to 0.0003 over the first 10000 steps. During training, we tracked the mean average difference between the predicted joint angles and the ground truth angles for both training and validation sets. Our criterion for stopping the training procedure was the moment of obtaining a mean joint angle difference value of less than  $1^\circ$  across the validation set.

## 4 Multimodal XR Hand Tracking with sEMG and Vision-Based Tracking

In this section, we present our approach to multimodal XR hand tracking by combining sEMG and vision-based tracking techniques. In our system, the vision-based tracking

data provides information about the overall hand position and orientation in 3D space, while the sEMG-based model produces granular information about individual finger joint rotations. The sEMG data are continuously sampled, preprocessed, and passed to the trained deep learning model to estimate the eight finger joint angles. The hand position and orientation data are combined with the estimated finger joint angles to generate a complete hand pose representation.

The deep learning model is optimized for real-time performance, ensuring that the sEMG data can be processed with minimal latency. Our system operates at 50Hz, which incurs a latency of 20ms. For this work, data processing and model inference took place on a consumer notebook. The notebook concurrently runs a python server, responsible for sEMG data aggregation, preprocessing and model inference, and a 3D XR environment on the Unity platform. At every time step, the estimated finger joint angles are transferred from Python to Unity through the low-latency ZeroMQ framework<sup>3</sup>. A video demonstration of our system is available online<sup>4</sup>. It shows a side-by-side comparison of the vision-based tracking system and our multimodal approach.

#### 4.1 Evaluation

To validate the effectiveness of our multimodal hand tracking system, we conducted an experimental evaluation, which compared the multimodal tracking to the baseline vision-based hand tracking system provided by the XR device. We simultaneously collected tracking data from the vision-based tracking system and the multimodal tracking system. A Leap Motion sensor was used to acquire ground truth labels for finger joint angles. The Leap Motion is a high-precision vision-based hand tracking device that captures finger joint angles and hand position in 3D space. The ground truth labels were then compared to the results from both the vision-based and the multimodal tracking system.

The experimental setup included a series of hand pose tasks, designed to cover a wide range of hand movements, including gestures prone to occlusion. The tasks were selected with regard to their utility in playing a keyboard-inspired XRMI. The tasks were performed while wearing the Oculus Quest 2 headset and the Myo armband. The Leap Motion sensor was placed on a table to record ground truth data. We recorded tasks under two conditions: 1) Full view of the hand - here, the XR headset was positioned at a 50cm distance, 45° above the hand to ensure optimal visual tracking conditions. 2) Self-occlusion of the hand - in this condition, the distance was kept identical, but the angle of the XR headset was lowered, such that the back of the hand occluded the fingers. Figure 2 illustrates the six hand pose tasks devised: (i) extending all fingers and making a fist; individual flexion and extension of the (ii) index, (iii) middle, (iv) ring and (v) pinky fingers; (vi) sequential flexion and extension of pinky, ring, middle and index fingers (similar to the gesture of drumming on a table while waiting for something). Each task involves the execution of the gesture at three different speeds: slow, over approximately two seconds, moderate (one second), and fast (half a second). To account for variability

<sup>3</sup> <https://zeromq.org/>

<sup>4</sup> <https://www.youtube.com/watch?v=iv12g2t2oaI>



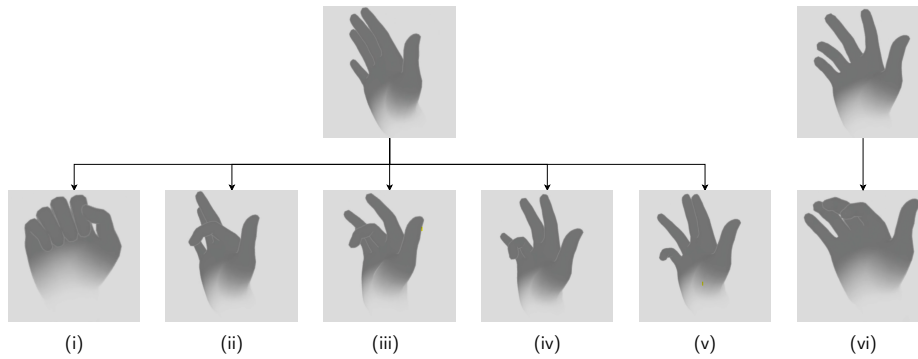


Fig. 2: Finger movements in the six hand pose tasks

in the sEMG measurements, all tasks were executed three times, over two days, under identical lighting conditions.

To measure the degree of finger occlusion, we integrated a ray-casting system with the XR application. We cast rays from the XR headset's 3D position to the eight finger bones whose rotations we measured every time a sample was taken. Rays intersecting other parts of the hand, e.g., the back of the hand, were used to mark the respective finger bones as occluded. This allowed us to quantify the level of occlusion per finger for every recording.

#### 4.2 Analysis Methods & Results

To evaluate the performance of the vision-based and multimodal tracking systems across every task, we obtained matrices of difference values between the estimated joint angles and the ground truth angles for both systems at every time step. The matrices were aggregated across the three sessions. We assessed the normality of the difference matrices using the Shapiro-Wilk test for each of the six hand pose tasks. We then applied the Wilcoxon signed-rank test to see whether there was a significant difference between the results produced by the vision-based and multimodal tracking systems.

Across tasks, the results of the Shapiro-Wilk test showed p-values  $< .001$ , indicating that the data in the difference value matrices did not fit a normal distribution. Therefore, we proceeded with the non-parametric Wilcoxon signed-rank test for further analysis. Figure 3 shows the results obtained from the mean joint angle differences across all finger joints per task, for both occlusion conditions (full view and occluded). Under the occluded condition, our model produces significantly lower deviations from the ground truth data across all tasks, compared to the vision-based tracking system. On average, it improves the finger joint angle tracking accuracy by five to 15 degrees across all fingers.

Table 1 shows the results obtained from the Wilcoxon signed-rank tests, aggregated across all eight tracked joints per task for both conditions. The p-values indicate significant differences between the difference value matrices. Additionally, the table lists the average finger occlusion results obtained through the raycasting occlusion measure

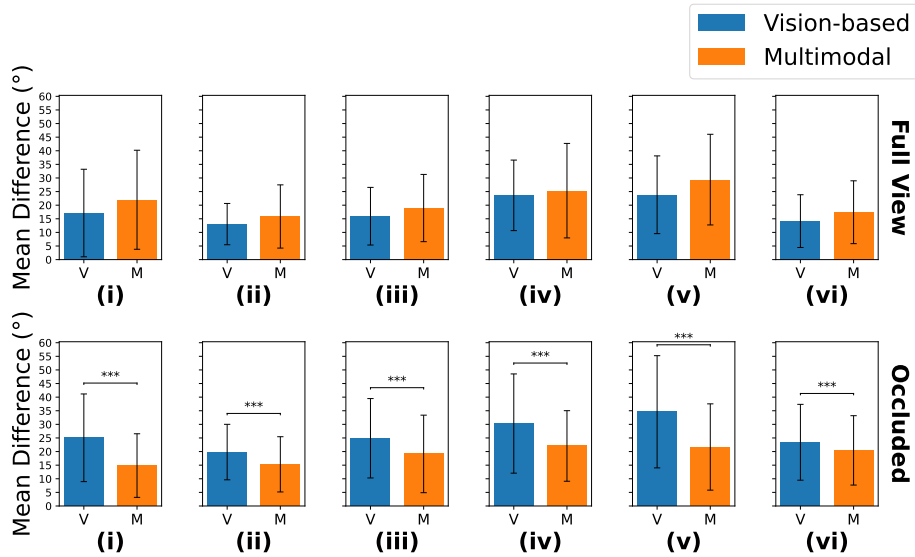


Fig. 3: Average deviation in degrees between the finger joint angles generated by the vision-based (V) and multimodal (M) tracking systems and the ground truth data for each task. Error bars show standard deviation; three asterisks indicate a significant difference between the V and M values ( $p < .001$ )

described above. The occlusion results describe the mean portion per task, in which the fingers were occluded by another part of the hand.

## 5 Discussion

The results of the evaluation showed that the multimodal hand tracking system outperformed the pure vision-based hand tracking system across all tasks under the occluded condition. These findings support our hypothesis that the integration of sEMG-based finger joint angle estimation can help overcome occlusion-related limitations in vision-based hand tracking, resulting in more accurate and reliable XRMI interactions. Under the full view condition, the vision-based hand tracking produced fewer errors in all tasks. This was expected, as the vision-based hand tracking system operates optimally under full view of the hand.

Despite the promising results, our study has several limitations. Due to the nature of sEMG data, the tracking performance of the multimodal approach is unlikely to extend to other users without fine-tuning the deep learning model. Surface EMG signals differ substantially between individuals and can be influenced by factors such as muscle fatigue, electrode placement, and individual anatomical differences. It will be valuable to investigate the system's performance across different users and under varying conditions. The identification of sEMG data representations that allow for generalisation under consideration of these factors without requiring extensive amounts of data is still

Table 1: P-values and average occlusion measurement results across tasks under both conditions

Tasks	Full view		Occluded	
	P-value	Occlusion (%)	P-value	Occlusion (%)
(i)	1.0	16.78	<.001	93.00
(ii)	1.0	2.65	<.001	70.58
(iii)	1.0	15.25	<.001	63.11
(iv)	1.0	16.78	<.001	53.02
(v)	1.0	27.94	<.001	78.82
(vi)	1.0	10.36	<.001	59.48

an ongoing research topic. However, our work allows XR users with access to sEMG devices to train their own models using our pipeline and code.

The performance of our multimodal hand tracking system was evaluated using a single type of XR headset and sEMG armband. Future research should explore more complex occlusion scenarios, as well as test the system’s performance across different hardware setups and sEMG devices, to better understand the generalisability of our findings.

With that in mind, we see numerous avenues for further research. The integration of additional tracking modalities, such as depth sensing or inertial measurement units (IMUs), could further enhance the robustness and accuracy of the multimodal hand tracking system by enabling stronger representations of the underlying data. A future study will explore the impact of our multimodal hand tracking system on usability, user experience and task performance in XRMI interactions, and provide insights into the practical implications of our findings. By conducting user studies with tasks that require precise hand movements and are susceptible to occlusion, the benefits of our system for real-world applications could be better understood.

Our study provides evidence that the combination of vision-based tracking and sEMG-based finger joint angle estimation can effectively address occlusion issues in hand tracking for XRMI interactions. The findings suggest that the multimodal hand tracking system has the potential to enhance user experiences and enable more immersive and natural interactions in virtual environments.

## 6 Conclusion

In this paper, we introduced a multimodal hand tracking system designed to address occlusion issues in XRMI interactions by combining vision-based tracking with sEMG-based finger joint angle estimation. The goal of this study was to demonstrate the potential of our proposed system to improve hand tracking accuracy and robustness, even when the hand is partially occluded.

While our results show promise, the experimental setup was relatively simple, and further research should explore more complex scenarios and investigate the system’s performance across different hardware and user conditions. Future work could also in-

tegrate additional tracking modalities and machine learning techniques to enhance the robustness and accuracy of the system.

Our multimodal hand tracking system demonstrates the potential to improve XRMI interactions by addressing occlusion issues in vision-based hand tracking. As XR technologies continue to evolve, the integration of complementary tracking modalities, such as sEMG and vision-based tracking, will likely play a crucial role in enhancing user experiences and enabling more immersive and natural interactions in virtual environments.

## References

- [1] C. Avian et al. “Estimating Finger Joint Angles on Surface EMG Using Manifold Learning and Long Short-Term Memory with Attention Mechanism”. In: *Biomedical Signal Processing and Control* 71 (Jan. 1, 2022), p. 103099.
- [2] F. Berthaut. “3D Interaction Techniques for Musical Expression”. In: *Journal of New Music Research* 49.1 (Jan. 1, 2020), pp. 60–72.
- [3] S. Bilbow. “Developing Multisensory Augmented Reality As A Medium For Computational Artists”. In: *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 72. New York, NY, USA: Association for Computing Machinery, Feb. 14, 2021, pp. 1–7.
- [4] S. Bilbow. “Evaluating Polaris~ - An Audiovisual Augmented Reality Experience Built on Open-Source Hardware and Software”. In: *NIME 2022*. June 16, 2022.
- [5] A. Çamcı, M. Vilaplana, and R. Wang. “Exploring the Affordances of VR for Musical Interaction Design with VIMes”. In: *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME. Birmingham, UK, June 1, 2020, pp. 121–126.
- [6] T. Deacon, T. Stockman, and M. Barthelet. “User Experience in an Interactive Music Virtual Reality System: An Exploratory Study”. In: *Bridging People and Sound*. Ed. by M. Aramaki, R. Kronland-Martinet, and S. Ystad. Vol. 10525. Cham: Springer International Publishing, 2017, pp. 192–216.
- [7] *Delsys EMG Sensor Placement Technical Note 101*. Accessed on 15.08.2023.
- [8] J. Fillwalk. “ChromaChord: A Virtual Musical Instrument”. In: *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 2015 IEEE Symposium on 3D User Interfaces (3DUI). Mar. 2015, pp. 201–202.
- [9] M. Graf and M. Barthelet. “Mixed Reality Musical Interface: Exploring Ergonomics and Adaptive Hand Pose Recognition for Gestural Control”. In: *International Conference on New Interfaces for Musical Expression*. NIME 2022. June 28, 2022.
- [10] R. Hamilton and C. Platz. “Gesture-Based Collaborative Virtual Reality Performance in Carillon”. In: *Proceedings of the 2016 International Computer Music Conference*. International Computer Music Conference. Utrecht, Netherlands, 2016, p. 5.
- [11] S. Han et al. “MEgATrack: Monochrome Egocentric Articulated Hand-Tracking for Virtual Reality”. In: *ACM Transactions on Graphics* 39.4 (Aug. 12, 2020), 87:87:1–87:87:13.

- [12] X. Hu et al. “Finger Movement Recognition via High-Density Electromyography of Intrinsic and Extrinsic Hand Muscles”. In: *Scientific Data* 9.1 (1 June 29, 2022), p. 373.
- [13] N. J. Jarque-Bou et al. “A Calibrated Database of Kinematics and EMG of the Forearm and Hand during Activities of Daily Living”. In: *Scientific Data* 6.1 (1 Nov. 11, 2019), p. 270.
- [14] N. Jiang. *Gesture Recognition and Biometrics Electromyography (GRABMyo) Dataset*. Jan. 4, 2022.
- [15] P. Kaczmarek, T. Mańkowski, and J. Tomczyński. “putEMG—A Surface Electromyography Hand Gesture Recognition Dataset”. In: *Sensors* 19.16 (16 Jan. 2019), p. 3548.
- [16] H. Lee, D. Kim, and Y.-L. Park. “Explainable Deep Learning Model for EMG-Based Finger Angle Estimation Using Attention”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30 (2022), pp. 1877–1886.
- [17] Y. Liu, C. Lin, and Z. Li. “WR-Hand: Wearable Armband Can Track User’s Hand”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (Sept. 14, 2021), 118:1–118:27.
- [18] Y. Liu, S. Zhang, and M. Gowda. “NeuroPose: 3D Hand Pose Tracking Using EMG Wearables”. In: *Proceedings of the Web Conference 2021. WWW ’21*. New York, NY, USA: Association for Computing Machinery, June 3, 2021, pp. 1471–1482.
- [19] T. Mäki-Patola et al. “Experiments with Virtual Reality Instruments”. In: *Proceedings of the 2005 Conference on New Interfaces for Musical Expression. NIME ’05*. SGP: National University of Singapore, May 1, 2005, pp. 11–16.
- [20] L. Men and N. Bryan-Kinns. “LeMo: Supporting Collaborative Music Making in Virtual Reality”. In: *IEEE 4th VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*. Apr. 5, 2018.
- [21] P. Milgram et al. “Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum”. In: *Telem manipulator and Telepresence Technologies* 2351 (Jan. 1, 1994).
- [22] A. G. Moore et al. “Wedge: A Musical Interface for Building and Playing Composition-Appropriate Immersive Environments”. In: *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. 2015 IEEE Symposium on 3D User Interfaces (3DUI). Arles, France: IEEE, Mar. 2015, pp. 205–206.
- [23] S. Serafin et al. “Virtual Reality Musical Instruments: Guidelines for Multisensory Interaction Design”. In: *Proceedings of the Audio Mostly 2016. AM ’16: Audio Mostly 2016*. Norrköping Sweden: ACM, Oct. 4, 2016, pp. 266–271.
- [24] Théo Jourdan and Baptiste Caramiaux. “Machine Learning for Musical Expression: A Systematic Literature Review”. In: NIME. 2023.
- [25] P. Walkington. *PyoMyo*. Version 0.0.5. Nov. 2021.
- [26] F. Weichert et al. “Analysis of the Accuracy and Robustness of the Leap Motion Controller”. In: *Sensors* 13.5 (5 May 2013), pp. 6380–6393.
- [27] D. Wessel and M. Wright. “Problems and Prospects for Intimate Musical Control of Computers”. In: *Computer Music Journal* 26.3 (Sept. 2002), pp. 11–22.

# Emotional Impact of Source Localization in Music Using Machine Learning and EEG: a proof-of-concept study

Timothy Schmele<sup>\*1,3</sup>, Eleonora De Filippi<sup>\*2</sup>, Arijit Nandi<sup>2</sup>,  
Alexandre Pereda Baños<sup>2</sup>, and Adan Garriga<sup>3</sup>

<sup>1</sup> Institute for Music Informatics and Musicology (IMWI), University of Music,  
76131 Karlsruhe, Germany

<sup>2</sup> Department of Big Data at Eurecat, Centre Tecnologic, 08005 Barcelona, Spain

<sup>3</sup> Department of Multimedia at Eurecat, Center Tecnologic, 08005 Barcelona, Spain.  
tim.schmele@eurecat.org; eleonora.defilippi@eurecat.org

**Abstract.** Little is currently known about how varied source locations affect a listener's emotional reaction to music. Here, using spectral features extracted from electrophysiology (EEG) data, we tested through machine learning whether four music source positions (front, back, left, and right) could be accurately distinguished according to the type of valence in a subject-wise manner. The findings demonstrate that distinct EEG correlates can reliably classify the four source locations and that the effect is stronger when music with a negative emotional valence is played outside of the listener's visual field. This proof-of-concept study may pave the way for advanced spatial audio analysis approaches in music information retrieval by considering the listener's emotional impact depending on the source direction of incidence.

**Keywords:** Spatial Music, Emotion Recognition, Affective Computing, EEG, Machine Learning, SVM, Source Localization

## 1 Introduction

Music's ability to modulate cognitive and emotional processes has been widely documented over the years [1–4], making it a relevant tool to investigate the brain correlates of emotional processes [5]. However, little is known about the impact of different sound-source locations on the emotional response to music, as only a few studies have addressed this issue [6–8]. In particular, the study conducted by Asutay et. al. [7] provided evidence that the effects of spatial source location on attentional processes are mediated by the emotional information conveyed by the sound [7]. The authors also demonstrated that a sound source behind the participant led to a more robust affective response in the listeners [7]. In another work, Tajadura-Jiménez et. al. concluded that

---

\* = these authors equally contributed to this work.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

sound sources outside the visual field produce emotional states of increased arousal [9], though these effects were more pronounced for natural sounds. Similarly, the work of Ekman and Kajastila showed that sounds are judged by the listener as scarier when they come from the back as compared to the front [6], although context has been found to be an important factor in eliciting the desired effect [8]. In a more detailed study using everyday sound events, Drossos et. al [10] showed that lateral positions do increase the listeners' affective state significantly, although this is dependent on the content of the audio sample used.

Moreover, there is a strong connection between space and music, as music can, in turn, evoke sensations of space and movement as a sense of *intrinsic space*, i.e. a metaphorical space, created by musical features in melody, harmony or rhythm, as opposed to the *literal, physical space* a sound source may occupy [11]. The most common effect is that of associating the perceived pitch with a sense of spatial height [12], although alternative spatial representations for the same also exist latently [13]. Furthermore, a correlation between the absolute pitch of a musical piece and emotional affect has been shown [14]. In a study conducted by Eitan et. al. [15], in which participants were asked to associate music with imagined, spatial motions of a human character, it was shown that most musical parameters significantly affect the imaginary motion, indicating a strong correlation between music and space perception.

Here we investigated whether four different music source spatial locations (i.e., front, back, left, and right) are reflected in a distinct pattern of electrophysiological activity that can be captured by a machine-learning approach. Moreover, we explored the interaction between distinct music source locations and the emotional salience of the musical excerpts played. The dimensional model supports the idea that emotions can be modeled as combinations of a few fundamental and basic dimensions. Valence and arousal, sometimes known as the "circumplex model," are two fundamental qualities that researchers unanimously concur are necessary to understand emotions [16]. The valence level varies from unpleasant (negative) to pleasant (positive), while the arousal level, specifically, ranges from not aroused (low arousal) to thrilled (high arousal).

We recorded electrophysiological (EEG) data while participants were listening to musical excerpts characterized by either positive or negative valence, both with middle values of arousal, and occurring from different spatial source locations. To take into account individual differences, we performed subject-based classification between each pair of spatial locations, according to the type of valence. We hypothesized that when the music source was located outside the listener's visual field (i.e., back, right, left) it would lead to a different electrophysiological pattern and impact on the affective state as compared to frontal source localization.

## **2 Materials and methods**

### **2.1 Stimuli**

The music excerpts used in this study were taken from the Database for Emotional Evaluation of Music (DEAM) [17]. It features a wide range of musical genres from popular Western styles to spoken word. All excerpts come from royalty-free music sources and

are thus very likely to be unknown to the average participant. In order to choose which samples to use for this study, we categorized the musical excerpts into 3 groups of high, mid, and low values along each emotional dimension (valence and arousal), based on the static evaluation metric. To evaluate the effect of valence in this study, those samples that fell into the mid-arousal category were first selected and then separated into two categories of positive or negative valence. The error between the average dynamic and static rating served as an ordering mechanism, along which the final samples could be selected.

## **2.2 Experimental design**

The sequence of audio samples was arranged into blocks of 3 randomly selected musical excerpts without repetition. Random selection was done anew for each participant. First, from each category, i.e. positive or negative valence, the samples were shuffled and grouped into blocks of 3 musical excerpts. Each block was assigned a spatial position (front, back, left, or right) at random, representing each position equally. Then, the blocks from each category were combined in random order into a single sequence of  $36 + 36 = 72$  samples.

The participant was first shown an introduction, explaining the experiment and the SAM questionnaire, followed by a short test if the participant has understood what the SAM represents. Then, a baseline rest period of 120s is recorded. After that, the main experiment started. In each block, 3 music samples from the same spatial position were played. Before each sample, a rest period of 5s was first presented to mentally reset the participant. At the end of each musical sample, the participant had to fill out the SAM questionnaire rating their emotional response to that particular musical stimulus. At the end of each block, another additional questionnaire was shown, where the participant answered how exhausted and attentive they felt.

The experiment ended after all blocks and their respective musical samples had been played to the participant. The EEG data were recorded using a 19-electrode Neuroelectronics® Instrument Controller (NIC2) at a sampling rate of 250 Hz. The subjects were informed about the experimental protocol, its approximate duration, and the meaning of the SAM scales. We instructed the participants not to move, particularly during playback of the stimulus, to reduce muscular artifacts in the EEG data. The experiment was conducted according to the Helsinki Declaration and all subjects signed the consent form.

## **2.3 Participants**

We recruited a total of 20 healthy participants (10 males and 10 females) with a mean age of 28.66 years (SD = 5.53), no history of psychiatric or neurological disorders, and normal hearing. Furthermore, subjects had no prior experience or formal music training. Nearly all participants were right-handed, with only one participant being left-handed. After preprocessing the EEG signal, 3 participants were removed from the analysis due to excessive artifacts. Only 17 participants were included in the analysis, comprising 9 females and 8 males with a median age of 28 (20-38).



## 2.4 Experimental Setup

The room in which the experiment took place was acoustically treated, with acoustic diffusion panels on the walls and absorption panels on the ceiling, as well as acoustically isolated from the outside. The reverberation time was relatively short, with an  $RT_{60}$  of 0.398s at 125Hz to 0.253s at 8kHz. The average  $RT_{60}$  between 500Hz and 1000Hz is around 0.293s. The audio stimuli were played from four positions: front, back, left, and right. A loudspeaker was placed in each position. The front and back loudspeakers were positioned at 3.2m from the listener, while the side speakers were at 2.3m.

To correct for the differences in distance between the loudspeakers, each loudspeaker was calibrated to 75dB SPL using pink noise at  $-20$ dBFS at the center listening position. The loudspeakers used were of the type Genelec 8040, fed by a Focusrite Scarlet 18i20 soundcard. The audio playback was done with the *sounddevice* module for Python, running on a Windows laptop computer.<sup>4</sup>

The interface was built for a web browser using HTML and CSS, with the functional elements programmed in Javascript and JQuery. The interface was designed with touch controls in mind, filtering accidental double taps and guiding the participant through the experiment. Whenever a button to continue was shown, the participant was also able to change their mind before sending off the result to be recorded. The Python script and the web front-end communicated using a simple socket connection over *localhost*. All activity on the touch screen was recorded over the socket connection.

## 2.5 EEG preprocessing

EEG data were analyzed in an offline manner using the EEGLAB toolbox on Matlab R2019b (The Mathworks, Inc.). The preprocessing steps included downsampling of the signal to 130 Hz and the application of a bandpass Butterworth filter ranging from 0.01 up to 40 Hz. To correct eye blinks and muscular artifacts, we used the Independent Component Analysis (ICA) algorithm. For each subject, we manually removed all components capturing artifacts. Afterward, we epoched the EEG data and created eight distinct datasets for each subject according to the experimental condition (i.e, spatial position and type of valence). Finally, we applied a spatial filter to reduce the volume conduction effect, using the surface Laplacian transform inspired by the spherical spline method described by [18–20].

## 2.6 Feature extraction

To preserve information about the temporal dynamics, we transformed the EEG data into the time-frequency domain using Complex Morlet Wavelet convolution (CMW) [21]. We chose CMW instead of alternative approaches like the Short-time Fourier Transform or the Hilbert Transform, because CMW is a Gaussian-shaped wavelet in the frequency domain.

Since we were interested in all frequency bands, we selected a range of frequencies going from 1 Hz up to 40 Hz. Following the use of CMW convolution, we retrieved the

<sup>4</sup> <https://github.com/multimedia-eurecat/Neuromuse>

power from the coefficients and then used a decibel-baseline normalization, utilizing all neutral trials as a baseline. We used a sliding-window strategy to reduce the time-frequency data for every trial in order to increase the sample size. There were a total of 39 windows in every trial, each lasting 1 second and overlapping by half a second.

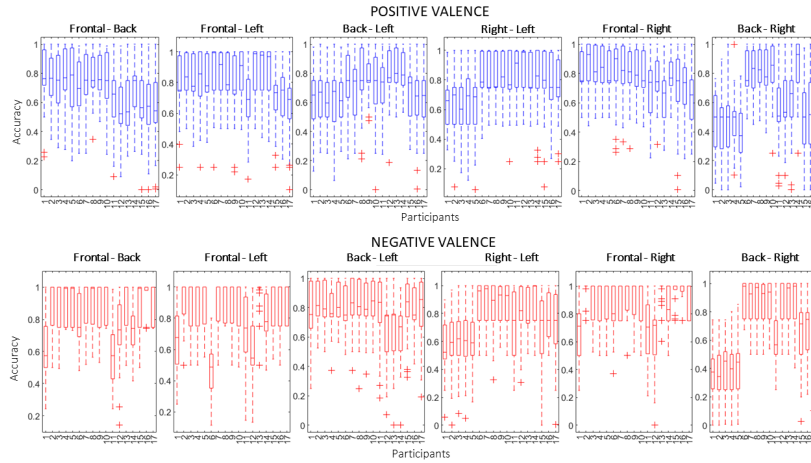
Then, we calculated the average change in power compared to the neutral baseline for seven frequency bands (delta 1–4 Hz, theta 4–8 Hz, low alpha 8–10 Hz, high alpha 10–12 Hz, low beta 13–18 Hz, high beta 18–30 Hz, and gamma 31–40 Hz), which constitute the spectral features. Within each window and for all the 19 channels and the seven frequency bands, the features extracted were the mean power, the standard deviation of the mean, and the frontal alpha asymmetry (FAA). The FAA coefficients were calculated for the channel pairs Fp1-Fp2 and F3-F4 in both low-alpha (8–10 Hz) and high-alpha (10–12 Hz) bands. The resulting feature array consisted of 351 samples for each class with a total of 270 features.

## 2.7 Classification and feature selection

For data classification, we utilized MATLAB R2022a Statistics and the Machine Learning Toolbox. As a base classifier, the linear Support Vector Machine (SVM) supervised learning approach was chosen, which uses a hyperplane as a decision boundary to optimize the margin of separation between two classes. Herewith, SVMs give a metric that permits scaling the certainty with which a window sample is allocated to one of the two classes: the sample's distance from the separation hyperplane. To evaluate the classifier's performance robustly, we used 6-fold cross-validation to train and test the classifier, allocating all windows in one trial to the same fold. Having said that, we also ran the 6-fold cross-validation fifty times and averaged the results across different classification runs.

It is well known that feature extraction and selection strategies assist to reduce computing complexity and develop models with greater generalization capabilities, in addition to enhancing predictive power [22]. That is, we used the Bioinformatics toolbox of MATLAB R2022a to do feature selection due to the large dimensionality of our dataset. The goal was to improve the classifier's learning performance and find the most common discriminative characteristics shared by all participants. We rated the characteristics based on their importance between the classes, using the t-test as an independent criterion for binary classification. For each feature, the built-in function in MATLAB calculates the absolute value of the two-sample t-test with pooled variance estimate. Finally, we identified the top 20 characteristics for each topic and combined them to determine which features were shared by all participants.

**Statistical comparisons** We used the Wilcoxon rank-sum method to investigate if the SVM performances were significantly above chance, thus we statistically compared accuracy distributions of real-labeled data with surrogate data (i.e., randomly shuffled labels). Furthermore, data from the self-assessment SAM questionnaire were analyzed using a general linear model, the multivariate analysis of variance (MANOVA), on IBM SPSS Statistics.



**Fig. 1.** Within-subject classification results of each binary classification between music position sources according to the type of valence. We performed 6-fold cross-validation 50 times, such that the boxplots depict the results of 50 classification runs for each participant.

### 3 Results

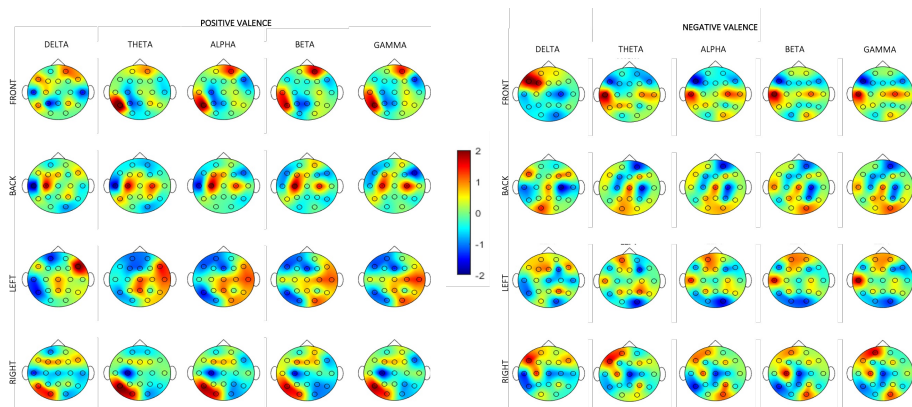
We investigated through machine learning whether the four music position sources could be accurately differentiated according to the type of valence in a subject-wise manner using spectral features extracted from EEG data cut into 1-second windows. The results of all cross-validation run for each participant are presented in Figure 1. The corresponding accuracy averaged across subjects for each binary classification run is summarized in Table 1. For both positive and negative valence, we showed that the highest average accuracy was reached when classifying the frontal localization versus each of the three sources located outside the visual field. In particular, this effect was stronger when classifying pairs of source locations using events characterized by negative valence.

**Table 1.** Classification results averaged across participants for each pair of binary classification runs presented according to the valence type. An asterisk indicates that the average accuracy is significantly above the chance level ( $p < 0.05$ ).

	Pair of music source locations	Mean accuracy across subjects		Pair of music source locations	Mean accuracy across subjects
Positive valence	Frontal - Back	70%*	Negative valence	Frontal - Back	85%*
	Frontal - Left	80%*		Frontal - Left	82%*
	Back - Left	69%*		Back - Left	78%*
	Right - Left	76%*		Right - Left	74%*
	Frontal - Right	77%*		Frontal - Right	87%*
	Back - Right	61%*		Back - Right	67%*

### 3.1 Highest-ranked features

To understand which channels and frequency bands were the most discriminative between the four location sources depending on the type of valence, we applied a feature selection algorithm and merged together the top twenty features for each subject. As represented in Figure 2, the results of the feature selection procedure showed that the electrophysiological correlates of the difference between frontal location and the three sources located outside the visual field (i.e., back, right, left) rely on different activities of channels mainly located in frontal and central areas, especially in the highest frequencies. In particular, we found that when using musical excerpts with negative valence, the difference between frontal location and each of the three sources out of the field of view was based on activity in beta (low and high) and gamma bands in channels Fp1, F3, F4, Fz, Cz, and T8, and in FAA measures for pairs of channels Fp1-Fp2 and F3-F4. On the other side, when comparing the source locations using positive valence, we showed that also brain activity in the alpha band, together with beta and gamma was important for differentiating frontal position from each of the other three sources. Moreover, in the case of positive valence, channels from posterior sites, especially Pz, P7, and P8, as well as central, frontal sites and FAA, were relevant for the classification of source locations, indicating a more widespread involvement of different brain areas.



**Fig. 2.** Topoplots indicating brain activity for each of the main frequency bands according to the source locations (i.e., front, back, left, and right) and the type of valence (i.e., negative and positive). Colors in brain plots indicate the power in that specific channel and frequency band, with red showing the highest power and blue the lowest.

### 3.2 Self-assessment SAM questionnaire

Results of self-reported ratings showed that there was a significant difference in arousal and valence ratings based on the type of event (source location and type of valence of musical excerpts),  $F(14, 12428) = 28.133$ ,  $p = 0.000$ ,  $Wilk's\lambda = 0.740$ ,

partial eta squared = 0.14. The source locations depending on the type of valence had a significant effect both on reported levels of perceived arousal,  $F(7, 1215) = 18.477$ ,  $p = 0.000$ , partial eta squared = 0.096, and valence,  $F(7, 1215) = 47.886$ ,  $p = 0.000$ , partial eta squared = 0.216. Averaged reported levels of perceived arousal and valence are presented in Table 2.

Post-hoc analysis revealed that there were significant differences between trials with positive valence and negative valence within each source location ( $p = 0.000$ , Bonferroni corrected). In particular, musical excerpts characterized by positive valence elicited higher reported levels of arousal ( $p = 0.000$ ) and valence ( $p = 0.000$ ) for each of the four source locations. Differences between sources were not significant ( $p > 0.05$ ) when comparing the same type of valence (i.e., either positive or negative), except the levels of reported arousal between the back and right when music with positive valence was played ( $p = 0.03$ , Bonferroni corrected).

**Table 2.** Average reported levels of arousal and valence by means of the SAM questionnaire on a Likert scale from 1 to 5.

Type of valence	Source location	Average SAM rating for arousal	Average SAM rating for valence
Positive valence	Frontal	3,41	3,25
	Back	3,57	3,21
	Left	3,22	3,17
	Right	3,45	3,36
Negative valence	Frontal	2,63	2,03
	Back	2,90	2,18
	Left	2,88	2,28
	Right	3,45	2,15

## 4 Discussion

In this work, we analyzed the impact of different source locations of music depending on the type of valence on the listener's affective brain processing by employing machine learning tools. The results demonstrate that frontal location can be accurately distinguished from each of the three sources (back, right, and left) located outside the listener's visual field. In particular, our results suggested that the emotional connotation of music (i.e., positive and negative valence) mediated the impact of the different source locations on the brain's electrophysiological signal, as reflected by music characterized with negative valence yielding higher classification performances in differentiating between the spatial sources as compared to musical excerpts characterized by positive valence.

Furthermore, by applying a feature selection procedure we showed that playing music from different source locations led to different electrophysiological brain responses in the highest frequencies (alpha, beta, and gamma) and in channels belonging to the frontal, central, and also parietal areas in the case of positive valence. The importance of

beta and gamma bands that we found here is consistent with earlier research showing the significance of these bands for differentiating between various emotional states [23,24]. Moreover, a previous study has found that the alpha band in parietal channels was associated with the processing of auditory stimuli, while the gamma band activity was related to music awareness [25]. Interestingly, we found the FAA between pairs of channels Fp1-Fp2 and F3-F4 to be an important measure for distinguishing between different locations, both for positive and negative valence conditions. Alpha activity in the frontal site has been largely used as an index of emotional processing, reflecting motivation and dominance of perceived emotion. Indeed, in the literature, positive emotional stimuli have been related to a relative increase in left hemisphere activity, whereas negative emotional stimuli have been associated with a larger right hemisphere activity [26,27]. For example, a previous study has found that musical excerpts characterized by positive valence induced lower frontal alpha power in the left hemisphere [28]. In addition to valence and arousal, frontal asymmetry was also linked to other factors, such as self-reported dominance [29].

Music has generally been used in research as a tool to elicit emotional responses in participants and study emotional processes in the brain [30, 31]. Recent applications of Brain-Computer Interfaces have used music as a way to convey information and/or feedback in a real-time manner to the subjects based on their own brain activity [32–34]. However, the difficulty of participants in engaging and sustaining genuine emotional states in an experimental context, particularly when trying to elicit complex emotions, has generally been a significant hurdle for neuroimaging and BCI studies based on affective processes. In this regard, the results of this study may pave the way for more effective use of music as a stimulus in experimental settings.

Moreover, the analysis of tones and their spatial orientation in relation to the listener is relevant in the context of *spatial music*. Here, spatial music refers to musical composition practices that specifically target spatial aspects of sound as a compositional parameter, such as the sound position or specific aspects of room acoustics [35, 36]. Indeed, emotion elicited through spatial music listening is not an aspect that is not often considered in this context. The discussion around space in music tends to be often of philosophical [35], or conceptual nature [36, 37] and often centers around aspects in electronics or hardware [37, 38], technology [39, 40] or taxonomy [41]. In a survey conducted in [42], composers are more often than not concerned with those spatial aspects in music that can be parameterized on a technical level and lesser with the emotional impact space can have on the listener. The results of this study indicate that an analysis of spatial music would need to take into account the correlation between the extracted emotional impact of the more traditional musical parameters, such as melody and rhythm, with the impact made by spatial features.

However, this study does present some limitations. Most importantly, the relatively small sample size may limit the generalization of our results. Also, despite having found a different brain pattern of activity in the EEG signal related to affective processing of music depending on the various source locations, this effect was not reflected in the analysis of subjective ratings of both arousal and valence. This may have been due to overthinking, rather than an immediate and intuitive reaction. When being asked how one would evaluate a piece of music, one would have to execute the said task by rec-

ollecting what was just heard. This evaluation will thus be skewed by the importance a subject might place on different musical aspects. Therefore, if a subject has little to no experience in associating spatial position with musical significance, then the recollection of the heard excerpt will most likely be focused on other aspects, filtering out the spatial direction from which the excerpt was heard. This means that mentally, i.e. in the inner ear, the music may have been heard *aspatially*. Future work may investigate the potential effect that musical training could have on the results in this context.

## 5 Conclusion

The present study has shown that machine learning methods are able to discern a listener's affective brain processing between different spatial positions of sound sources as a function of positive or negative affect. Annotated musical excerpts were classified into two groups of both median arousal and low or high valence values respectively. These samples were presented to the listener from the front, the lateral left or right positions, or the back, in random order. Our results showed that frontal location, compared to each of the other three sources located outside the visual field, is associated with different brain electrophysiological patterns related to emotional processing. In fact, we found a significant involvement of alpha, beta, and gamma frequency bands in frontal and central sites, together with FAA measures, in distinguishing between such source locations. These findings were not reflected in the subjective rating analysis, hinting that the subjects may have excluded the spatial aspect of the music when consciously evaluating the heard excerpts. While more analysis is necessary, these first results prove promising. Further analysis is necessary to understand how the source location is able to influence the emotional impact of music, particularly focusing on arousal. Also, it would be interesting assessing whether different types of music show divergence in the emotional impact depending on source location. Lastly, future work will also have to include the median plane to get a more comprehensive view of the effects of spatial source locations on the listener's affective state.

## References

1. J. A. Sloboda, "Music structure and emotional response: Some empirical findings," *Psychology of music*, vol. 19, no. 2, pp. 110–120, 1991.
2. Y. Hou and S. Chen, "Distinguishing different emotions evoked by music via electroencephalographic signals," *Computational intelligence and neuroscience*, vol. 2019, 2019.
3. 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.
4. P. Vuust, O. A. Heggli, K. J. Friston, and M. L. Kringelbach, "Music in the brain," *Nature Reviews Neuroscience*, vol. 23, no. 5, pp. 287–305, 2022.
5. S. Koelsch, "Brain correlates of music-evoked emotions," *Nature Reviews Neuroscience*, vol. 15, no. 3, pp. 170–180, 2014.
6. I. Ekman and R. Kajastila, "Localization cues affect emotional judgments—results from a user study on scary sound," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, Audio Engineering Society, 2009.

7. E. Asutay and D. Västfjäll, "Attentional and emotional prioritization of the sounds occurring outside the visual field.," *Emotion*, vol. 15, no. 3, p. 281, 2015.
8. S. Hughes and G. Kearney, "Fear and localisation: Emotional fine-tuning utilising multiple source directions," in *Audio Engineering Society Conference: 56th International Conference: Audio for Games*, Audio Engineering Society, 2015.
9. A. Tajadura-Jiménez, P. Larsson, A. Väljamäe, D. Västfjäll, and M. Kleiner, "When room size matters: acoustic influences on emotional responses to sounds.," *Emotion*, vol. 10, no. 3, p. 416, 2010.
10. K. Drossos, A. Floros, A. Giannakouloupoulos, and N. Kanellopoulos, "Investigating the impact of sound angular position on the listener affective state," *IEEE Transactions on Affective Computing*, vol. 6, no. 1, pp. 27–42, 2015.
11. E. Di Bona, "Listening to the space of music," *Rivista di estetica*, no. 66, pp. 93–105, 2017.
12. E. Rusconi, B. Kwan, B. L. Giordano, C. Umiltà, and B. Butterworth, "Spatial representation of pitch height: the smarc effect," *Cognition*, vol. 99, no. 2, pp. 113–129, 2006.
13. Z. Eitan and R. Timmers, "Beethoven's last piano sonata and those who follow crocodiles: Cross-domain mappings of auditory pitch in a musical context," *Cognition*, vol. 114, no. 3, pp. 405–422, 2010.
14. L. Jaquet, B. Danuser, and P. Gomez, "Music and felt emotions: How systematic pitch level variations affect the experience of pleasantness and arousal," *Psychology of Music*, vol. 42, no. 1, pp. 51–70, 2014.
15. Z. Eitan and R. Y. Granot, "How music moves:: Musical parameters and listeners images of motion," *Music perception*, vol. 23, no. 3, pp. 221–248, 2006.
16. P. J. Lang, M. K. Greenwald, M. M. Bradley, and A. O. Hamm, "Looking at pictures: Affective, facial, visceral, and behavioral reactions," *Psychophysiology*, vol. 30, no. 3, pp. 261–273, 1993.
17. A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLOS ONE*, vol. 12, March 2017.
18. F. Perrin, O. Bertrand, and J. Pernier, "Scalp current density mapping: value and estimation from potential data," *IEEE Transactions on Biomedical Engineering*, no. 4, pp. 283–288, 1987.
19. F. Perrin, J. Pernier, O. Bertnard, M.-H. Giard, and J. Echallier, "Mapping of scalp potentials by surface spline interpolation," *Electroencephalography and clinical neurophysiology*, vol. 66, no. 1, pp. 75–81, 1987.
20. F. Perrin, J. Pernier, O. Bertrand, and J. Echallier, "Spherical splines for scalp potential and current density mapping," *Electroencephalography and clinical neurophysiology*, vol. 72, no. 2, pp. 184–187, 1989.
21. M. X. Cohen, "A better way to define and describe morlet wavelets for time-frequency analysis," *NeuroImage*, vol. 199, pp. 81–86, 2019.
22. G. Anthony and H. Ruther, "Comparison of feature selection techniques for svm classification," in *10th International Symposium on Physical Measurements and Signatures in Remote Sensing*, pp. 1–6, 2007.
23. M. Li and B.-L. Lu, "Emotion classification based on gamma-band eeg," in *2009 Annual International Conference of the IEEE Engineering in medicine and biology society*, pp. 1223–1226, IEEE, 2009.
24. P. Li, H. Liu, Y. Si, C. Li, F. Li, X. Zhu, X. Huang, Y. Zeng, D. Yao, Y. Zhang, *et al.*, "Eeg based emotion recognition by combining functional connectivity network and local activations," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 10, pp. 2869–2881, 2019.
25. M. J. Mollakazemi, D. Biswal, and A. Patwardhan, "Target frequency band of cognition and tempo of music: Cardiac synchronous eeg," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 696–700, IEEE, 2018.



26. R. J. Davidson and J. Henriques, "Regional brain function in sadness and depression," *The neuropsychology of emotion*, pp. 269–297, 2000.
27. B. D. Poole and P. A. Gable, "Affective motivational direction drives asymmetric frontal hemisphere activation," *Experimental brain research*, vol. 232, no. 7, pp. 2121–2130, 2014.
28. L. A. Schmidt and L. J. Trainor, "Frontal brain electrical activity (eeg) distinguishes valence and intensity of musical emotions," *Cognition & Emotion*, vol. 15, no. 4, pp. 487–500, 2001.
29. B. Reuderink, C. Mühl, and M. Poel, "Valence, arousal and dominance in the eeg during game play," *International journal of autonomous and adaptive communications systems*, vol. 6, no. 1, pp. 45–62, 2013.
30. S. Brown, M. J. Martinez, and L. M. Parsons, "Passive music listening spontaneously engages limbic and paralimbic systems," *Neuroreport*, vol. 15, no. 13, pp. 2033–2037, 2004.
31. 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.
32. I. Daly, D. Williams, A. Kirke, J. Weaver, A. Malik, F. Hwang, E. Miranda, and S. J. Nasuto, "Affective brain–computer music interfacing," *Journal of Neural Engineering*, vol. 13, no. 4, p. 046022, 2016.
33. J. Heo, H. J. Baek, S. Hong, M. H. Chang, J. S. Lee, and K. S. Park, "Music and natural sounds in an auditory steady-state response based brain–computer interface to increase user acceptance," *Computers in Biology and Medicine*, vol. 84, pp. 45–52, 2017.
34. S. K. Ehrlich, K. R. Agres, C. Guan, and G. Cheng, "A closed-loop, music-based brain-computer interface for emotion mediation," *PloS one*, vol. 14, no. 3, p. e0213516, 2019.
35. M. A. Harley, *Space and spatialization in contemporary music: History and analysis, ideas and implementations*. PhD thesis, McGill University, 1994.
36. P. Miller, *Stockhausen and the Serial shaping of Space*. University of Rochester, Eastman School of Music, 2009.
37. M. Brech and H. von Coler, "Aspects of space in luigi nono's prometeo and the use of the halaphon," in *Kompositionen für hörbaren Raum/Compositions for Audible Space*, pp. 193–204, transcript Verlag, 2015.
38. E. Bates, *The Composition and Performance of Spatial Music*. PhD thesis, Trinity College Dublin, 2009.
39. N. Barrett, "Spatio-musical composition strategies," *Organised sound*, vol. 7, no. 3, pp. 313–323, 2002.
40. M. A. Baalman, "Spatial composition techniques and sound spatialisation technologies," *Organised Sound*, vol. 15, no. 3, pp. 209–218, 2010.
41. D. Smalley, "Space-form and the acousmatic image," *Organised sound*, vol. 12, no. 1, pp. 35–58, 2007.
42. F. Otondo, "Contemporary trends in the use of space in electroacoustic music," *Organised Sound*, vol. 13, no. 1, pp. 77–81, 2008.

## Exploring Patterns of Skill Gain and Loss on Long-term Training and Non-training in Rhythm Game

Ayane Sasaki<sup>1</sup>, Mio Matsuura<sup>1</sup>, Masaki Matsubara<sup>2</sup>,  
Yoshinari Takegawa<sup>1</sup>, and Keiji Hirata<sup>1</sup>

<sup>1</sup> Future University Hakodate

<sup>2</sup> University of Tsukuba

**Abstract.** The objective of this study is to categorize patterns of skill loss following skill gain, in order to develop a predictive model for skill retention in music games. The experiment was conducted using songs from the web-based music game “Sparebeat.” Participants were instructed to train daily on a piece of music slightly more challenging than their current skill level until they achieved a specified level of proficiency. Following this, participants took a break from training for at least one week, and their scores were recorded when they played the music immediately after the non-training phase. By analyzing the changes in scores during both the skill gain and loss phases, we identified three distinct patterns of skill loss.

**Keywords:** Educational Technology, Human Computer Interaction

### 1 Introduction

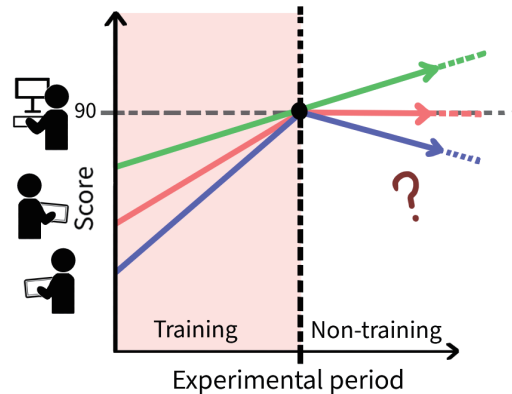
In recent years, advances in HCI technology have enabled the proposal of numerous learning support systems to assist learners in acquiring skills involving physical movements, such as tennis [1], golf [2–4], calligraphy [5, 6], playing musical instruments [7], and singing [8].

A common learning support framework involves adjusting the difficulty of skill gain based on the learner’s level, providing a sense of accomplishment during training and fostering motivation. For instance, bicycles equipped with training wheels enable inexperienced riders to train and eventually ride without assistance. The main challenges lie in determining how to modify the target skill’s difficulty and provide learners with environments that promote continued motivation.

While research has explored the cognitive aspects of skill gain, it is important to consider “skill loss,” the decline in acquired skills that occurs once a person stops training. Factors like individual differences in skill loss suggest that cognitive aspects of learners are involved in this process. By examining both skill gain and loss, we can develop a more accurate cognitive model of skill gain.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



**Fig. 1.** Overview: our study conducts long-term experiment to explore skill gain and loss patterns through training and non-training periods

Previous studies have investigated memorization and forgetting in tasks that do not involve physical actions, such as memorizing words [9–13]. In sports science, on the assumption that a sportsperson trains daily, research has focused on the relationship between sleep and memory consolidation through motor skill learning [14–16].

Our study aims to clarify the relationship between skill gain and loss in music games for beginners, by observing score changes and timing judgment classifications during both the skill gain and loss phases (Fig. 1). We also seek to discover and classify patterns of skill loss.

A unique feature of our study is its long-term experimental design, as the gain and loss of skills involving physical movements require a certain length of time. For example, learning to ride a bicycle typically takes several days to weeks. In sports like tennis and golf, there is virtually no upper limit to the time required for skill gain. Acquired skills are not forgotten until a certain length of time has passed<sup>3</sup>. In our experiment, participants trained until they reached a specific music game score, with some requiring up to 50 days of training. After the training phase, subjects entered a skill loss phase, with some continuing the experiment for nearly 90 days. Observing gain and loss over such a long period is expected to yield essential data and findings.

Music games are excellent targets for experiments involving skill gain and physical movement. As games, they inherently motivate players to continue practicing, and players can engage in music games for extended periods without boredom. Music games require a certain level of skill and training to achieve a high score and offer a mechanism to consistently and stably assess a player’s level of skill acquisition.

Towards identifying the relationship between skill gain and loss in music games, we conduct an exploratory study to find an appropriate hypothesis as the first step.

<sup>3</sup> Once a person is able to ride a bicycle, he or she will not completely forget the skill, although the skill level may deteriorate.

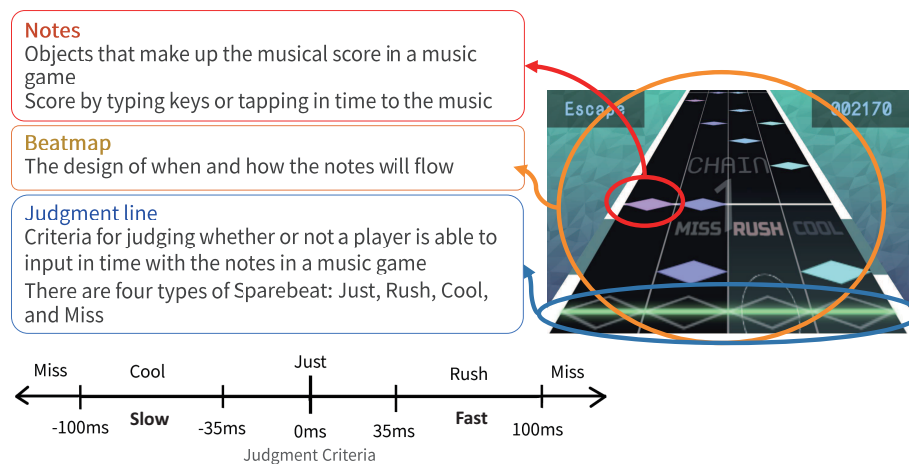


Fig. 2. Screenshot of music game 'Sparebeat' and criteria of judgement

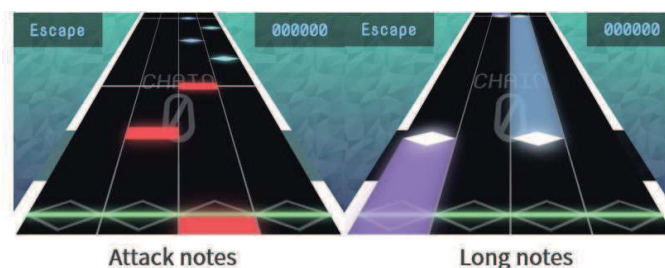


Fig. 3. Attack notes and Long notes

## 2 Materials and Methods

### 2.1 Music game Sparebeat

We used Sparebeat (Fig. 2), a music game simulator, for our experiments. It runs on web browsers and is playable on PCs, smartphones, and tablets. The playing screen consists of four black lanes with diamond-shaped notes in different colors moving towards a green line. Players must press the corresponding key as the notes cross the green line to earn points. Sparebeat has three types of notes with varying difficulty levels and display formats.

Music pieces for the experiment were selected based on each subject's skill level from a preliminary assessment. We chose pieces with a score of 650,000 to 700,000 points to ensure they were neither too easy nor too difficult, allowing us to measure skill gain effectively.

Sparebeat has four types of timing judgments for key presses: *Just*, *Rush*, *Cool*, and *Miss*. Each judgment depends on the accuracy of the player's timing when pressing the keys. As indicated by the criteria arrows at the bottom of Fig. 2, *Just* is correct, *Rush* is

**Table 1.** Judgment and score criteria

	Just	Rush	Cool	Miss
Normal notes	100%	50%	50%	0%
Long notes	100%	50%	50%	0%
Attack notes	200%	100%	100%	0%

**Table 2.** Experimental period

	Training phase	Non-training phase
Subject 1	16 days	90 days
Subject 2	10 days	58 days
Subject 3	10 days	58 days
Subject 4	34 days	63 days
Subject 5	50 days	50 days

fast, Cool is slow, and Miss is anything that does not fall into any of these categories. In addition to the “Normal notes” shown in Fig. 2, there are also “Long notes” and “Attack notes” (Fig. 3). The scoring system of Attack notes is different from that of Normal notes, and as shown in Table 1, the scoring is twice that of Normal notes.

The maximum score for any piece in Sparebeat is 1,000,000 points. The score per note varies depending on the piece and is calculated by dividing the full score by the total number of notes in the piece.

## 2.2 Participants

Five subjects participated, ranging from beginner to intermediate university and graduate students who had played music games as a hobby. None of the subjects had played Sparebeat before. They played the game on personal devices throughout the experiment.

## 2.3 Instructions for subjects

Subjects were asked to play their assigned piece once, train for 10 to 20 minutes, and then play it again. They trained daily, following a training set format. Once subjects consistently scored over 900,000 points, they entered a non-training phase during which they did not play the game. After this phase, they played their assigned piece once more, and their scores were recorded. This non-training set was repeated as necessary.

The threshold for suspending practice was determined to be 900,000 points due to experience and the results of when the Just, Rush, Cool, and Miss percentages exceed 900,000 points, which are discussed later in Section 3.1.

# 3 Results

## 3.1 Training phase

The duration of the training phase and the duration of the non-training phase for each subject are shown in Table 2. The length of the training phase and non-training phase

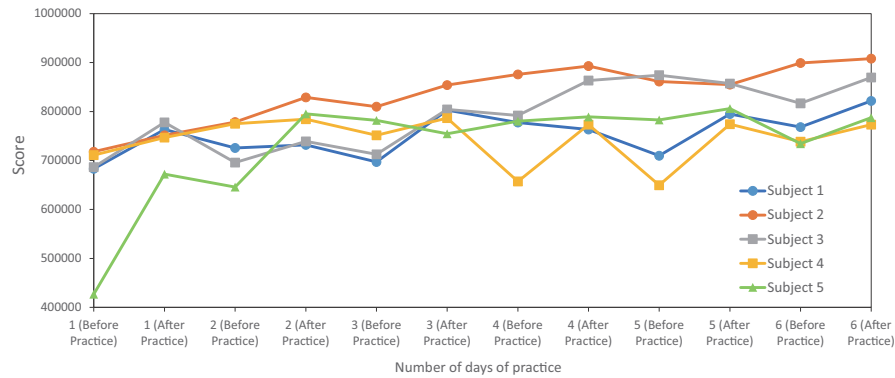


Fig. 4. Score transitions from Day 1 to Day 6 of training phase.

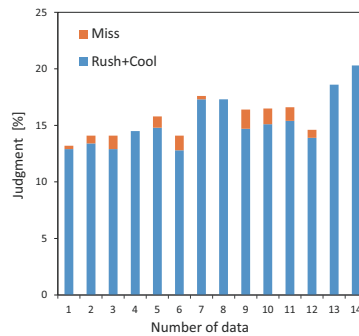
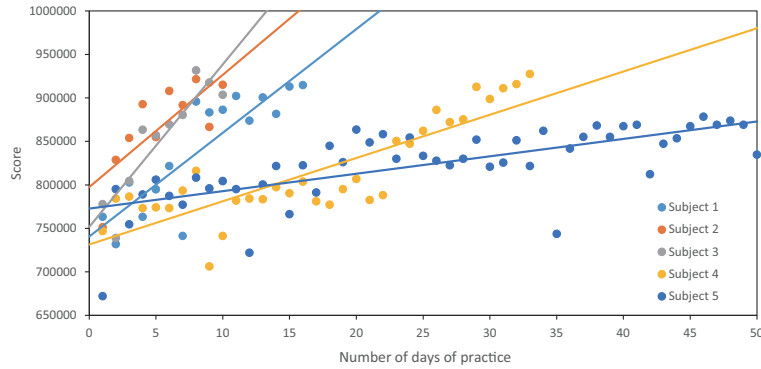


Fig. 5. Breakdown of Rush, Cool, and Miss ratio over 900,000 points among all subjects' performance

was different for each subject. The pre-training and post-training scores for each of the five subjects from the first day to the sixth day are shown in Fig. 4.

The graph in Fig. 4 shows that the post-training score is higher than the score on the first day of training on all training days. In addition, the pre-training score tends to be lower than the post-training score on a given training day. It can be said that the player generally improves with training. Although there were individual differences, the pre-training score was lower than the previous day's post-training score on the seventh day and beyond as well, but the score gradually increases with each training session.

During the training phase, the scores exceeding 900,000 are listed in descending order among the results that include all subjects before and after training, and the breakdown of Rush, Cool, and Miss in that data is shown in Fig. 5. Since the distribution of Rush and Cool scores is the same, the distribution of the percentage of the sum of Rush and Cool scores and the percentage of Miss scores is shown. From this figure, it



**Fig. 6.** Scatter plot of post-training scores in training phase

**Table 3.** Linear approximation of scores in training phase

	During training			Volatility	
	Slope	Intercept	Decision coefficient	Slope	Intercept
Subject 1	11,925	740,615	0.74	$-4.9 \times 10^{-4}$	0.018
Subject 2	12,902	797,598	0.59	$-7.6 \times 10^{-3}$	0.062
Subject 3	18,681	751,760	0.82	$-3.0 \times 10^{-3}$	0.033
Subject 4	4,970	731,551	0.72	$6.9 \times 10^{-4}$	-0.007
Subject 5	2,003	772,762	0.48	$-5.2 \times 10^{-4}$	0.040

can be seen that the ratio of Rush and Cool must be approximately 15% or less to exceed 900,000 points. However, even when the ratio is larger than 15%, the score exceeds 900,000 points as long as the Miss ratio is approximately less than 1%.

During the training phase, we focus only on the post-training scores in order to investigate the evolution of scores until the skill is mastered. Fig. 6 shows a scatter plot of the post-training scores only. Table 3 shows the slope, coefficient of determination, etc. when a linear approximation is applied.

Subjects 1 to 3, who had trained for 10 to 16 days, had a slope of more than 10,000, indicating relatively rapid progress. Subjects 4 and 5 had a slope of less than 5,000. The training phases were 34 days, 50 days, and more than one month, respectively, meaning that progress was gradual, as it took time for these subjects to reach a certain skill level. The rate of change indicates how much score had changed when the score on a given day of the training phase was compared to that of the previous day. Then, the rate of change for each of the days up to the time when a subject stopped practicing is made into a regression line, and the slopes are shown in Table 3. From this, we can see that the slope is negative for all subjects except subject 4, indicating that the fluctuation of the score becomes smaller as training is repeated. It is possible that after a certain level of progress, the growth of the score nearly levelled off, and the score stabilized.

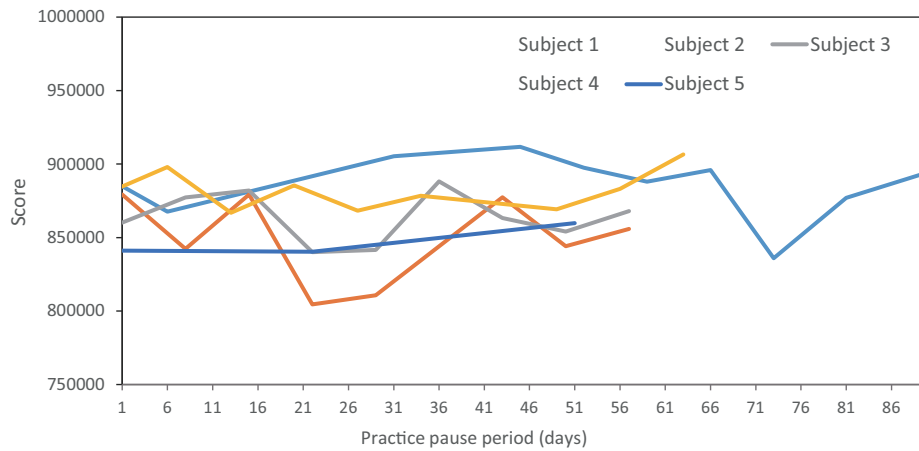


Fig. 7. Score transitions in non-training phase

Table 4. Linear approximation of scores in non-training phase

	Times	Slope	Intercept	Decision coefficient
Subject 1	10 times	-1,529	894,217	0.045
Subject 2	8 times	-1,834	857,464	0.023
Subject 3	9 times	-466	866,201	0.0057
Subject 4	9 times	675	878,937	0.02
Subject 5	3 times	9,346	828,400	0.72

### 3.2 Non-training phase

Fig. 7 shows the score transition for each of the trials in the non-training set. Table 4 shows the results of applying a linear approximation to each trial and score for the non-training set.

In Fig. 7, it was expected that scores would gradually increase as training continued, and then gradually decrease as training was paused, but this was not the case. It was found that there were variations in scores, such as a decrease in the first training session but an increase in the second training session. In addition, scores in the 700,000 range were seen at the beginning of training, but during this phase, all subjects scored above 800,000 and did not drop below that level.

In the slope of the linear approximation equation, Subject 1 to Subject 3 tended to drop slightly. Subject 5 is not included in the analysis at this time because the number of trials is still small (3) and it is necessary to increase the number of trials in order to compare the data. Subjects 2 and 3 had the same training phase, but subject 2's score decreased more, and the absolute value of the slope was larger than that of subject 1. The scatter of scores is also larger for subject 2.



### 3.3 Relationship between Rush and Cool and Score during the training phase

Fig. 8 show the score transition and the breakdown of judgment (percentage of Rush, Cool, and Miss) for each subject during the training phase. First, we compare three of the five subjects, Subject 1 to Subject 3, whose training phases were short and whose slopes in Table 4 were negative.

Subject 1's score did not increase until the seventh day, but increased after the eighth day, approaching 900,000 points. During this phase, Rush and Cool were reduced and the Miss rate, in particular, was reduced to 1.7%. Since then, the Miss rate remained low, and was 0% in four instances. When timing is judged as Miss, the score distribution is 0%, resulting in an increase in the Just rate and a significant increase in the score.

Subject 2 continued to increase his score steadily from the second day, reaching a score of 900,000 points on the sixth day. Both Rush and Cool were gradually decreasing, but the Miss rate was unstable, causing the score to decrease over several days.

Subject 3 had a high Cool rate until the third day, but it decreased after the fourth day, and exceeded 900,000 points on the eighth day. Compared to Subject 2, the Miss rate was stable and remained below 1% after the seventh day.

Something that these three subjects have in common is that the rate of Cool is higher than that of Rush on the first day. During the course of the increase in score, there were days when the ratio of Cool to Rush was reversed. This may be due to the fact that the players are not accustomed to playing music games on the first day, so their recognition of the notes flowing from above is not up to par, and their timing may fall a little behind that of Just. Then, it is thought that the sense of rhythm acquired from training experience when the player has become somewhat accustomed to the game will be out of sync with the sense of recognition of the notes, resulting in more Rushes.

Subjects 2 and 3 had the same training phase of 10 days, but the slope in Table 4 is more negative for Subject 2, and there is more variability in the scores. One possible reason for this is the instability of the Miss rate. Subject 2, whose Miss rate was unstable during the training phase, had an average Miss rate of 3.3%, and no Miss rate lower than 1%, even during the non-training phase. Subject 3 maintained a low Miss rate, averaging 0.9%. The percentages of Rush and Cool were lower in Subject 2, but the difference in Miss rate was larger than that, and the score was judged to be low.

Fig. 8-Sub.4 shows the scores and breakdown of judgments for subject 4. Subject 4's score did not increase and remained stagnant until the 23rd day. However, after that, Rush, Cool, and Miss gradually decreased, and the score reached 900,000 points on the 30th day. In Fig. 7, the score of subject 4 is the most stable, and the values of Rush, Cool, and Miss for subject 4 are also stable with respect to the score just before the end of the training phase.

Subject 5 had the longest training phase among all the subjects, but his score stopped growing around 860,000 points. In Fig. 8-Sub.5, the number of Misses is gradually decreasing, but Rush and Cool are quite unstable. The sum of Rush and Cool averages 25%, only once falling below 20%, and it does not decrease significantly, through to the end of the training phase. In this case, the length of the training phase is not proportional to the increase in score. Rather, the length of the training phase may have decreased the motivation to train, leading to stagnation and instability in scores. The possibility of such a causal relationship is a subject for further investigation.

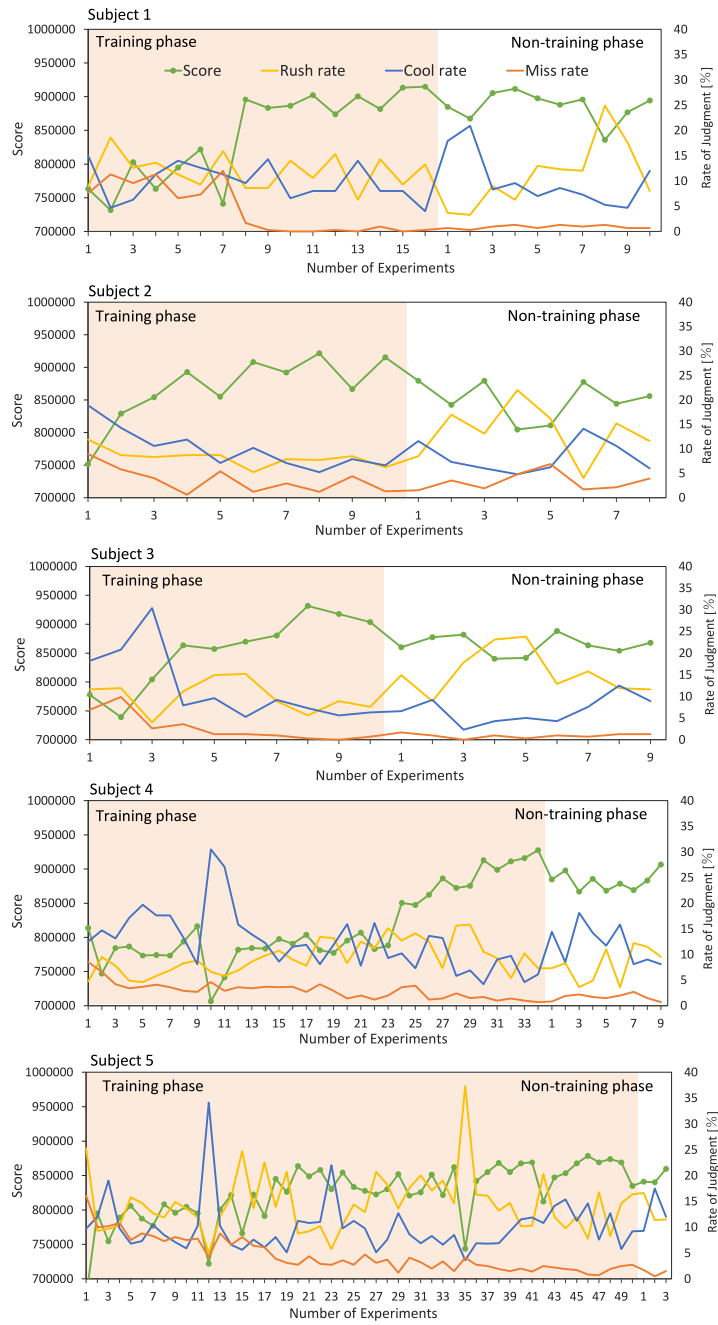


Fig. 8. Transitions of score and rate of judgments

## 4 Analysis

### 4.1 Pattern classification for skill loss

Based on the changes in the scores of the five subjects and the breakdown of their judgments, we classified the patterns of skill loss into the following three major categories: a pattern in which the subject forgets gradually after maintaining the score for a while, a pattern in which the score fluctuates wildly, and a pattern in which the subject does not easily forget. For Subject 5, the interval between experiments was irregular, and the frequency of experiments was low, so it was not possible to observe the daily fluctuation of the score. Therefore, we did not classify it as any of the patterns in this study.

**Pattern of maintaining for a while and then gradually losing the skill:** Subject 1 maintained a high score of around 900,000 points with a slight steady increase until the seventh experiment after entering the phase of training suspension. From the eighth experiment onward, the score exhibited a gradual downward trend (Fig. 7). During this phase, the score slightly decreased during the second experiment, slightly increased during the ninth experiment, and significantly decreased during the tenth experiment, but this is considered to be within the range where it can be called an exceptional phenomenon. We will hereafter continue to examine the trends and correlations in the breakdown of score judgments (ratio of Rush, Cool, and Miss).

**Pattern of wildly fluctuating scores:** The scores of Subjects 2 and 3 showed relatively large and repeated ups and downs (Fig. 7). The reason for the larger range of fluctuation in Subject 2's score than in Subject 3's score may be due to the instability of the Miss rate during the training phase (Section 3.3). Note that both Subjects 2 and 3 trained for a relatively short phase of time (10 days).

**Pattern of not easily losing the skill:** Fig. 7 shows that subject 4's score was the most stable and therefore that this subject exhibits a pattern of not easily losing the skill. Subject 4's score and percentage of Rush, Cool, and Miss grew steadily in the second half of the training phase. Empirically, we feel that skills that accumulate steadily during the training phase are less likely to be forgotten during phases of inactivity, and Subject 4 seems to fall into this pattern.

Subjects 3 and 4 have similar training phases, but different skill-loss patterns. First, let us examine the similarities between these subjects. Subjects 3 and 4 are similar in that Cool increases rapidly in the first half of the training phase, after which the ratio of Rush and Cool repeatedly reverses. Another thing these subjects have in common is that their scores and the values of Rush, Cool, and Miss are relatively stable just before the end of the training phase. On the other hand, in terms of the pattern of skill loss, Subject 3's score fluctuates between 800,000 and 900,000, while Subject 4's score remains stable above 850,000, and even exhibits an upward trend after 50 days. If the learning of a skill involves a cognitive process of retention, then the fact that Subject 3 had a short training phase of 10 days may mean that there was insufficient time for the acquired skill to take root.

#### **4.2 Relation between subjects' introspection and scores**

Open-ended interviews were conducted with subjects about their play during the training phase inactivity, and subjects were asked to talk about the relationship between their introspection and their scores, as well as their feelings about their play. The overall trend was that subjects who felt they were losing their skill did not experience a decrease in score, while those who did not feel that they were losing the skill did experience this. As for individual comments, these included: "My fingers remember the movements, and my score does not increase at all even if I stop practicing, but when I play after a long time, I find that I cannot complete the parts that used to be easy"; "My score has started to drop because I play only once a week"; "I do not really feel that I am forgetting. Once they are able to do well on that piece, they may be able to maintain a certain score on an easy piece with ease."

### **5 Conclusion**

In this study, we analyzed and classified skill-loss patterns as a preparatory step for constructing a model for predicting the loss of acquired skills in music games. We examined the extent to which subjects forgot, after stopping training, the acquired skill of attaining a certain score for an assigned piece in a music game, then we investigated the relationship between subjects' skill-loss patterns and training patterns. As a result, three types of skill-loss patterns were extracted.

Future work includes investigating whether these skill-loss patterns are applicable to other people and whether they can be generalized. For this purpose, we will increase the number of subjects and continue the experiment to confirm what kind of skill-loss patterns exist.

Subjects 3 and 4 had similar numbers of Rush, Cool, and Miss during the training phase, but their skill-loss patterns were classified differently. To clarify this difference, it is necessary to investigate the relationship between the length of the training phase and the skill-loss patterns. We will also investigate the relationship between Subject 5's training phase duration and score stagnation, as well as the relationship between length of training phase and decrease in motivation. As a future prospect, we would like to improve the content of experiments, for example by altering the time of the experiment, and investigating whether a subject's condition on that day affects the score, and where and how the subject made mistakes during the play.

### **Acknowledgements**

This work was supported by JSPS KAKENHI Grant number 21K18518 and 22H01047.

### **References**

1. Sadao Kawamura, Mizuto Ida, Takahiro Wada, and Jing-Long Wu. Development of a virtual sports machine using a wire drive system—a trial of virtual tennis. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 1, pages 111–116 vol.1, 1995.

2. Takuto Nakamura and Hideki Koike. Golf Club-Type Device with Force Feedback for Modifying Club Posture. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–7, New York, NY, USA, 2020. Association for Computing Machinery.
3. Chen-Chieh Liao, Hideki Koike, and Takuto Nakamura. Realtime center of mass adjustment via weight switching device inside a golf putter. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–8, New York, NY, USA, 2020. Association for Computing Machinery.
4. Takuto Nakamura, Daichi Saito, Erwin Wu, and Hideki Koike. Actuated club: Modification of golf-club posture with force feedback and motion prediction in vr environment. In *ACM SIGGRAPH 2020 Emerging Technologies*, SIGGRAPH '20, New York, NY, USA, 2020. Association for Computing Machinery.
5. Kazuyuki Henmi and Tsuneo Yoshikawa. Virtual lesson and its application to virtual calligraphy system. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, volume 2, pages 1275–1280 vol.2, 1998.
6. Saeka Tanaka, Yoshinari Takegawa, and Keiji Hirata. Proposal of a support system for learning brushstrokes in transcription for beginners. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, IUI '20, page 352–356, New York, NY, USA, 2020. Association for Computing Machinery.
7. Yoshinari Takegawa, Masahiko Tsukamoto, and Tsutomu Terada. Design and implementation of a piano practice support system using a real-time fingering recognition technique. In *Proceedings. 2011 International Computer Music Conference*, pages 1–8, 01 2011.
8. Tomoyasu Nakano, Masataka Goto, and Yuzuru Hiraga. Mirusinger: A singing skill visualization interface using real-time feedback and music cd recordings as referential data. In *Ninth IEEE International Symposium on Multimedia Workshops (ISMW 2007)*, pages 75–76, 2007.
9. Hermann Ebbinghaus. *Memory : a contribution to experimental psychology*. Dover Publications, 1987.
10. Michael W. Eysenck. *Attention and Arousal : Cognition and Performance*. Springer Berlin, Heidelberg, 2012.
11. Charan Ranganath and Robert S. Blumenfeld. Doubts about double dissociations between short- and long-term memory. *Trends in cognitive sciences*, 9(8):374–380, 2005.
12. Cristina M. Alberini. Mechanisms of memory stabilization: are consolidation and reconsolidation similar or distinct processes? *Trends in Neurosciences*, 28(1):51–56, 2005.
13. George Mandler. Organization and memory. *Psychology of Learning and Motivation*, 1:327–372, 1967.
14. Yuko Morita, Keiko Ogawa, and Sunao Uchida. Napping after complex motor learning enhances juggling performance. *Sleep Science*, 9(2):112–116, April 2016. Funding Information: This work was aided by the Japan Society for the Promotion of Science (JSPS) through the Grant-in-Aid for JSPS Fellows. The authors thank Dr. Thomas Maloney for kind and detailed editing of the manuscript. Publisher Copyright: © 2016 Brazilian Association of Sleep.
15. Masako Tamaki, Tatsuya Matsuoka, Hiroshi Nittono, and Tadao Hori. Fast Sleep Spindle (13-15 Hz) Activity Correlates with Sleep-Dependent Improvement in Visuomotor Performance. *Sleep*, 31(2):204–211, 02 2008.
16. Matthew P. Walker, Tiffany Brakefield, Alexandra Morgan, J.Allan Hobson, and Robert Stickgold. Practice with sleep makes perfect: Sleep-dependent motor skill learning. *Neuron*, 35(1):205–211, 2002.

## Benzaiten: A Non-expert-friendly Event of Automatic Melody Generation Contest

Yoshitaka Tomiyama<sup>1,2</sup>, Tetsuro Kitahara<sup>3</sup>, Taro Masuda<sup>1,4</sup>, Koki Kitaya<sup>1</sup>,  
Yuya Matsumura<sup>1,5</sup>, Ayari Takezawa<sup>1,6</sup>, Tsuyoshi Odaira<sup>7</sup>, and Kanako Baba<sup>8\*</sup>

<sup>1</sup>Music×Analytics Meetup Steering Committee, <sup>2</sup>NABLAS Inc., <sup>3</sup>Nihon University, <sup>4</sup>Nikkei Inc.  
<sup>5</sup>NOVASELL Inc., <sup>6</sup>Kyushu University, <sup>7</sup>Freelance, <sup>8</sup>Advanced Institute of Industrial Technology  
musicanalyticsmeetup@gmail.com

**Abstract.** This paper presents a contest-style music evaluation event called *Benzaiten*. There have been some attempts to evaluate different music generation systems with a unified criterion and/or platform, but it was not an event that non-experts could easily enjoy. At *Benzaiten*, we encouraged non-researcher people to join it as entrants by providing starter kits and communication channels. As well, we exercised ideas towards a high-quality entertainment event for laypeople to enjoy it. As a result, 15 people joined this event as entrants (eight of which moved to the main round), and more than 100 people participated as the audience.

**Keywords:** Melody generation, evaluation, contest

### 1 Introduction

Whereas the research on automatic melody generation has a long history, the recent development of machine learning (ML) technologies has been rapidly increasing the number of attempts at automatic melody generation [1].

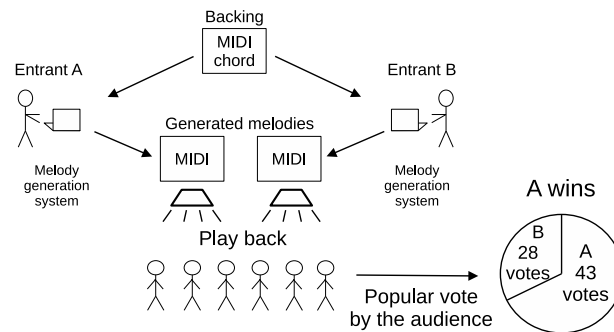
Evaluating melody generation systems/methods is still an important open problem in this field. Unlike speech recognition and image classification, the *correct* output (e.g., a melody) for a certain input (e.g., a chord progression) cannot be uniquely or objectively determined. We, therefore, have to conduct subjective quality tests on generated melodies employing music experts, but its methodology has not been necessarily established.

To provide a platform for evaluating different systems/methods on a unified criterion, some researchers made attempts to organize contest-based evaluation. Sturm et al. [2] organized the AI Music Generation Challenge 2000, in which they collected Irish double-jig pieces from entrants and hired experts to evaluate them. They also organized the 2021 edition focusing on Swedish traditional dance music [3]. Katayose et al. [4] held Performance Rendering Contests (Rencon) to provide a subjective evaluation platform for researchers developing expressive music performance rendering systems. Yeh

\* This work was supported by JSPS Kakenhi JP22H03711.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



**Fig. 1.** Basic scheme of Benzaiten. Two entrants are given a backing track and generate melodies during the event. Then, the winner is determined according to popular votes by the audience.

et al. [5] attempted on collaborative comparisons of harmonization systems developed by researchers from different institutes. However, these attempts have the following two problems:

- Because entrants are implicitly assumed to have skills or experiences in developing music generation systems/methods, there is no scheme for encouraging non-researcher people who have not tried such development.
- Because the primary purpose is to provide a unified platform for evaluation, they are not necessarily fun for laypeople as entertainment shows.

In this paper, we propose a novel contest-based music generation evaluation event, called *Benzaiten*. The most important policies in *Benzaiten* are *openness* and *fun* for non-experts. To develop the melody generation field furthermore, it is important to let a wide range of people have an interest in it. We, therefore, aim at an event that non-experts can enjoy as entrants and/or the audience. *Benzaiten* has the following features:

- To make it easy for novices to join it as entrants, we provided starter kits for developing melody generation systems and communication channels on Slack for sharing issues and ideas among potential entrants.
- To make it possible for laypeople to enjoy it as the audience, we exercised some ideas to make its quality as an entertainment show higher, including a popular vote and a one-on-one battle scheme.

## 2 Basic policy and event design

*Benzaiten* (Figure 1) is a contest-based melody generation evaluation event. Every entrant brings their melody generation system and generates a melody that fits a given backing track. The generated melody is played back within the event, then the winner is determined based on voting. As discussed in the Introduction, the basic policies in this event are *openness* and *fun* for non-experts as follows:

- *Openness to novices*: It is easy for various people to join this event as entrants, even if they are not ML and/or music experts.

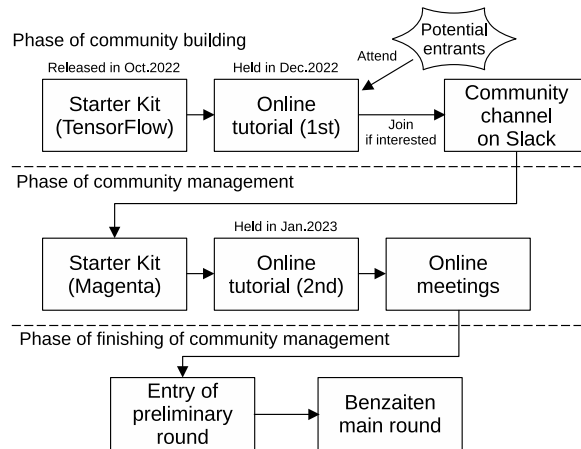


Fig. 2. Benzaiten's community management

- *Fun for laypeople*: Participants can enjoy this event as the audience even if they do not have music- or ML-related knowledge.

To achieve *openness to novices*, we make the following attempts:

- **Starter kits**: We developed two kinds of starter kits for this event, with which everyone can quickly try automatic melody generation.
- **Online tutorials**: We held online tutorials, in which the tutors taught the basic knowledge of MIDI and music as well as how to use the starter kits.
- **Community management of potential entrants**: We made a community of potential entrants, including those who had yet to determine entry. We encouraged communication among them by making a Slack channel and holding online meetings.

To achieve *fun for laypeople*, we introduce the following ideas to the event:

- **Popular vote**: The audience can get involved in determining the winner.
- **One-on-one battle scheme**: We adopted a tournament style based on a one-on-one battle scheme. This scheme makes the voting for each match simple, because all the participants have to do is to judge which is better of the presented two melodies.
- **Live melody generation**: For each match, the backing track is provided right when the match starts. Therefore, the entrants must generate melodies live during the event (they cannot generate melodies in advance).

### 3 Actions before the event

To encourage a wide range of people to join the event as entrants, we should promote this event widely and encourage as many people as possible to join the community of potential entrants. Therefore, we did the following (Figure 2).



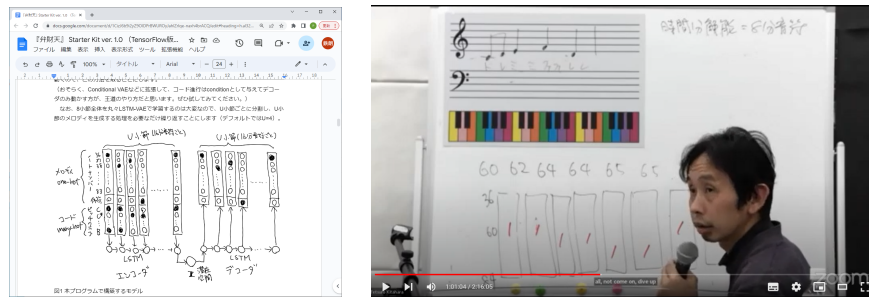


Fig. 3. Starter Kit (TensorFlow Edition) (left) and its tutorial (right)

### 3.1 Release of Starter Kit (TensorFlow Edition)

To encourage people who are not familiar with automatic melody generation, we developed the Starter Kit TensorFlow Edition (Figure 3.1 Left)<sup>1</sup>, in which users can easily try melody generation based on LSTM-VAE. This starter kit provides the codes for:

- Reading MusicXML files taken from Charlie Parker’s Omnibook MusicXML Data [6] and extracting melodies from them,
- Converting melodies to sequences of one-hot vectors,
- Training an LSTM-VAE model with prepared melodies using TensorFlow, and
- Generating a melody with the trained LSTM-VAE model in the MIDI format.

All codes are provided under the MIT License. As well, the starter kit provides several hints for extending those codes.

### 3.2 Tutorial on Starter Kit (TensorFlow Edition)

We conducted a tutorial aiming at allowing potential entrants to quickly understand the Starter Kit (TensorFlow Edition) on Zoom (Figure 3.1 Right). We encouraged a wide range of people to participate in this tutorial, including those yet to determined to join Benzaiten as entrants. 124 people participated, and some of them expressed their interest in the entry. This tutorial was well-received because we explained general knowledge of MIDI, the harmony theory as well as the codes of the starter kit.

### 3.3 Encouraging discussions on Slack

Because several participants of the above-mentioned tutorial expressed interest in joining Benzaiten as entrants, we invited them to the discussion channel for the community of potential entrants on Slack. We intended to let them exchange various information with each other and to foster a mood of friendly rivalry. They actually discussed some topics including: how to find publicly available datasets, and how to execute the starter kit’s codes on a local computing resource with MacOS on an M1 chip.

<sup>1</sup> Available at the following URL (written in Japanese): <https://docs.google.com/document/d/1CizJ6b9i2yZ9OIDPrBWUROyJahlZr1qe-naxh4brACQ/>

### **3.4 Starter Kit (Magenta Edition) and its tutorial**

We released the Starter Kit (Magenta Edition)<sup>2</sup>, which illustrates how to generate melodies using ImprovRNN, included in Magenta [7]. Because this edition uses the pre-trained ImprovRNN model, unlike the TensorFlow edition, users can try melody generation more simply, even though improving the model is complex. Furthermore, we held a tutorial for explaining this starter kit. 13 potential entrants participated in this tutorial.

### **3.5 Online meetings for progress sharing**

We held online meetings twice to enable potential entrants to share each other's progress. 16 people participated in the first meeting, and 13 participated in the second one.

## **4 Preliminary round**

15 people (or teams) entered Benzaiten even though the number of acceptable entrants was limited to eight. We, therefore, held an online preliminary round. On the designated web page, participants listened to all melodies submitted by 15 entrants and rated them on a scale of 0 to 10. To avoid bias caused by the order etc., the web page lists the melodies anonymously in a random order. We promoted this preliminary round on Twitter, and ratings by 70 participants were collected. Finally, the eight entrants with high ratings moved into the main round.

## **5 Implementation and results of Benzaiten (main round)**

### **5.1 Outline of the event**

Benzaiten adopts a single-elimination tournament style. Because we accepted eight entrants, the event consists of seven matches: four quarterfinals, two semifinals, and one final. The overall schedule is as follows:

**14:00–14:10** Opening  
**14:10–15:30** Four quarterfinals (20 mins for each match)  
**15:30–16:00** Lightning talks by all entrants on the techniques they use  
**16:00–16:40** Two semifinals  
**16:40–16:55** Sponsors' lightning talks  
**16:55–17:15** The final  
**17:15–17:35** Long talk by T. Kitahara  
**17:35–18:00** Awards ceremony & Ending

Each match was conducted as follows (Figure 7):

1. The two entrants go on the stage.
2. The organizer plays back the backing track for that match.

<sup>2</sup> Available at the following URL (written in Japanese): [https://colab.research.google.com/drive/1isnq\\_E2Mc-Fzeb8DKzYGL391-B30AwZK](https://colab.research.google.com/drive/1isnq_E2Mc-Fzeb8DKzYGL391-B30AwZK)



Fig. 4. Scene during a match (left) and winner announcement (right)

3. The backing track data (a MIDI file, a chord transcription file) is put into the online storage. The entrants are not permitted to download them before the match starts.
4. The match starts. The entrants must submit the generated melody's MIDI file before five minutes pass.
5. The melodies submitted by the two entrants are played back.
6. The popular vote starts. Google Forms is used as a voting platform.
7. The voting result (the winner) is announced.

## 5.2 Contest rule

To achieve a fair contest, we carefully designed the contest rule and presented the rule book on the Web. A distinctive rule is to allow entrants to compare more than one generated melody by listening and choose one within a five-minute time limit. We also decided the length and the timbre (program change) of melodies; The entrants cannot change them to focus on the quality of the melodies. Chord transcriptions are given as a text file.

## 5.3 Backing tracks

To make this event successful, we consider it important to present high-quality backing tracks that include a variety of chord progressions. To achieve this, a professional musician joined our team and composed seven high-quality backing tracks with different chord progressions.

According to the contest rules, every backing track has nine measures with the key of C major or A minor. The used chord progressions and keys are listed in Table 1. Backing tracks for Matches 1, 4, and 6 contain chords with non-diatonic roots, which make appropriate melody generation slightly difficult.

The chord transcription file (in the CSV format) as well as the MIDI file of the backing track are given to the entrants at each match.

## 5.4 Entrants

The eight entrants are listed in Table 2. Two entrants (*log5* and *Dekoboko Friends*) improved the post-processing to avoid musically unnatural notes without improving the starter kits' ML model. On the other hands, two entrants (*yatszhash* and *nayopu*) adopted completely different approaches, that is, melody generation by ABC-notated text generation or notation image generation. The other entrants adopted well-known ML models such as a Conditional VAE, a CNN-VAE, and a Transformer.

**Table 1.** Chord progressions and keys of backing tracks

Match	Round	Chord progression	Key
1		C   G/B   Bb   F/A   Fm/Ab   C/G   D/F#   G   C	C maj
2	Quarter-finals	C   C/E   F   G E   Am   C/G   D7/F#   Dm7G7   CM7	C maj
3		Am   F   G   C G   Am   F   G   E7   Am	A min
4		C   G   G7   C   C   Bb   F   C   C	C maj
5	Semi-finals	Am   DmEm   Am7   DmEm   Am7   DmEm   FM7 G   Am   Am	A min
6		Am   Am/F#   Dm   Bb   Am   Am/F#   Dm   Dm/B E7   Am	A min
7	Final	Dm7 F/G   CM7 FM7   Dm/B E7   A7   Dm7 F/G   CM7 FM7   Dm/B E7   Am   Am	A min

**Table 2.** Eight entrants and their melody generation techniques

No.	Name	Used techniques	Result
1	yatszhash	Generated a ABC-notated text with a language model	Champion
2	log5	Original post-processing based on music theory (with the unmodified ML model of the Starter Kit (TensorFlow))	
3	T. N.	Modified the Starter Kit (TensorFlow) to CNN-VAE	Semifinalist
4	nayopu	Generated notation images with DALL-E and converted it to the MIDI format	
5	AJI	Implemented MusicTransformer with their original dataset	
6	M. Y.	Generated melodies with ImprovRNN and then reconstructed them with MusicVAE	Semifinalist
7	konumaru	Modified the Starter Kit (TensorFlow) to Conditional VAEs.	Runner-up
8	Dekoboko Friends	Implemented post-processing to add grace notes and glissando (with the unmodified Starter Kit (Magenta))	

## 5.5 Number of participants

133 people (including the organizers) participated in Benzaiten on Zoom and about 40 to 50 people (including the organizers and entrants) participated on site. At each match, 54.7 votes were collected on average (max: 61, min: 51).

## 6 Discussions

### 6.1 Did we succeed in encouraging non-experts to participate?

Out of the eight entrants, five were non-music-related data scientists; they dealt with automatic MIDI generation for the first time. This fact shows that Benzaiten was able to reach out to a wide range of non-experts.

### 6.2 Did community management work well?

Our community management allowed potential entrants to exchange various information including: codes for porting the starter kit to a local environment, publicly available MIDI datasets, and chord notation.

Sharing MIDI data generated by entrants at online meetings enabled us to check if the MIDI data met our rules. It was effective to avoid errors occurring when we played them back during the event. In fact, no entrants generated erroneous MIDI data.

### 6.3 Technical trends in entrants

As we mentioned, entrants had various approaches ranging from improvements of the starter kits' post-processing (*log5*, *Dekoboko Friends*) to completely novel ones (*nay-opu*, *yatszhash*). The voting results showed that relatively *safe* melodies tended to win. In fact, melodies with some dissonant notes tended to be evaluated low.

Melodies generated by *log5* (the winner) and *Dekoboko Friends* (the runner-up) had different tendencies. The former was a sequence of close-packed short (e.g., 16th) notes, sometimes like arpeggios. The latter consisted of multiple phrases including rests between the phrases. Their system generated multiple melodies, and they selected one that included both notes and rests in a balanced way by checking them via piano rolls.

### 6.4 Future challenge

One issue in our community management was that discussions among the entrants had not gathered momentum until the date of Benzaiten was approaching. On the other hand, at Kaggle [8], an online platform for data science competitions, entrants can identify their current ranks through the leaderboard. It gives them motivation for continuous improvement. Also in Benzaiten, we need a mechanism like the leaderboard, which makes entrants' current ranks public. If Benzaiten has such a mechanism, entrants should have more motivation for continuously improving their melody generation models, and hence they would participate in the discussion more actively.

## 7 Conclusion

In this paper, we presented an automatic melody generation contest called *Benzaiten*. This event's features are to encourage non-experts to join it as entrants with starter kits and community management and to aim at a high-quality entertainment event to allow laypeople to enjoy it. Through these attempts, we let many non-expert people have interest in automatic melody generation. In the future, we would like to extend this attempt to let various people, ranging from hobbyists and musicians to ML researchers, begin music generation development. We believe that it would bring remarkable findings, contributing to further progress in our music generation field.

## References

1. J.-P. Briot, G. Hadjeres, and F.-D. Pachet: Deep Learning Techniques for Music Generation — A Survey, arXiv:1709.01620, 2017.
2. B. L. T. Sturm and H. Maruri-Aguilar: The Ai Music Generation Challenge 2020: Double Jigs in the Style of *O'Neill's 1011*, *J. Creat. Music Syst.* 5(1), 2021.
3. B. L. T. Sturm: The Ai Music Generation Challenge 2021: Summary and Results, *Proc. AIMC 2022*, 2022.
4. H. Katayose et al.: On Evaluating Systems for Generating Expressive Music Performance: the Rencon Experience, *J. New Music Res.*, 41(4), pp.299–310, 2012.
5. Y.-C. Yeh et al.: Automatic Melody Harmonization with Triad Chords: A Comparative Study, *J. New Music Res.*, 50(1), pp.37–51, 2021.
6. <https://homepages.loria.fr/evincent/omnibook/>
7. <https://magenta.tensorflow.org/>
8. <https://www.kaggle.com/>

# Pitch Class and Octave-Based Pitch Embedding Training Strategies for Symbolic Music Generation

Yuqiang Li<sup>1</sup> Shengchen Li<sup>1</sup> George Fazekas<sup>2</sup>

<sup>1</sup> Xi'an-Jiaotong Liverpool University

<sup>2</sup> Queen Mary University of London

yuqiang.li19@student.xjtlu.edu.cn

**Abstract.** This paper presents two strategies to prevent the pitch embeddings from being too close to the dataset characteristics so as to improve the pitch and pitch class distributions of generation. The first strategy is to switch the pitch representation from the MIDI number representation to an alternative representation that encodes a pitch into pitch class and octave, which forces musically similar pitches to share part of the embedding vectors. The second strategy freezes the pitch embeddings during training according to the proposed metrics that evaluate the quality of pitch embedding space, maintaining the robustness of the embedding obtained in the first strategy. The experiments show that, when both strategies are applied on the training in an auto-regressive melody generation task, the generated samples exhibit slightly improved pitch distribution but noticeably improved pitch class distribution, indicating the effectiveness of both strategies.

**Keywords:** Symbolic Music Generation, Word Embedding, Domain Knowledge

## 1 Introduction

The selection of an appropriate input music representation has been one of the key challenges in designing neural sequence models for symbolic generation, as multiple types of musical features must be serialized into sequences. Early MIDI event-like input representation (e.g. [26, 22]), suffered from the issues of being long and redundant to be handled by neural sequence models, and being implicit for models to reconstruct basic musical features (e.g. duration and metrical structures) [12]. Since then, solutions have been proposed to overcome these two problems, including applying constraints to the input representation using musical domain knowledge.

The REMI representation [13] uses the domain knowledge to recommend explicitly encoded durational and metrical features instead of MIDI-like note-on/note-off events, for a transformer to better capture durational and structural features on a sequential representation. The Compound Word representation (CPW) [11] improved the length limit and generation quality by shortening the input sequence length, based on the domain knowledge that tokens of the same type of musical features should be placed and treated



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

similarly in the input. The recent Music Fundamental Embedding (MFE) [10] avoids a type of generation failure by treating pitch, duration and metric position features as numeric features to ensure consistency of the implied relative musical features in the embedding space. We notice that the general approach here is to apply domain knowledge constraints on the model input such that the explicitness can benefit the model in capturing specific features related to the domain knowledge.

Comparatively, pitch feature domain knowledge constraints are less researched in symbolic music generation, with most models using the simple MIDI number encoding for input. Also, the current generation systems still struggle with capturing slightly complicated pitch and harmony features. For instance, the generated music usually lack a clear key center, without clear harmonic tension and releases. However, in discriminative tasks, such as chord estimation, music style clustering and automatic harmonic analysis [17, 6, 31], the pitch class feature is used more often than MIDI pitch number, indicating its effectiveness in capturing pitch-based features. Therefore, we consider the concept of pitch class important in the generation task as well.

It is therefore hypothesized that using pitch class and octave for the pitch feature would improve the learned pitch representation and the generation pitch and pitch class distribution by preserve more pitch proximity in the embedding space. First, an auxiliary metric SLD is proposed for the evaluation of pitch embedding space. The hypothesis is then evaluated through two experiments. Experiment 1 tests whether pitch class and octave can improve the pitch distribution compared to MIDI number encoding. Experiment 2 is based on the results of experiment 1, testing if freezing the pitch embeddings according to the SLD metric maintains high pitch performance during training.

In experiment 1, a Transformer-XL model is trained for melody generation under the two different pitch encoding methods multiple times with different pitch-unrelated hyper-parameters. Results show that melodies sampled from the group of models using class-octave encoding have better pitch and pitch class distributions compared to the MIDI-number encoding group. Also, the evaluation of SLD metric on the corresponding metric space is consistent with the pitch and pitch class performance in generation.

Although the class-octave pitch encoding outperforms the other, it exhibit a behavior of deterioration over epochs which is more obvious than the MIDI number encoding. Correspondingly, the SLD metrics of most class-octave models are observed to have reached an local minima when the the model at the best pitch performance. Therefore, in experiment 2, the best model of Experiment 1 is trained multiple times but the pitch embeddings are frozen at different epochs, respectively. The results reveal that the models whose pitch embeddings are frozen near the local minima of the SLC metric has better performance over longer training.

The outcomes of the two experiments show the effectiveness of the pitch class and octave constraints on the pitch representation, which informed the development of two practical pitch training strategies presented in this paper.

The rest of the paper would begin by a brief review of the previous methodologies in Section 2, followed by proposed methods in Section 3. The experiment and results are discussed in 4 and 5.

## 2 Related Work

### 2.1 Pitch representations

The one-hot representation is a widely used pitch representation in the literature [1, 14]. It does not assume any pitch structure or proximity, as all the one-hot pitch vectors are equidistant. However, Mozer [24] argued that equidistant one-hot pitch vectors are problematic for music generation. Mozer proposed a novel pitch representation called PHCCCF based on the spiral model by [25] and psychoacoustic experiments in 1979 [15, 16]. In PHCCCF, pitch vectors are closer in euclidean distance if they are closer as perceived by ears. While Mozer’s results has been able to learn some structure of diatonic scales [1], the psychoacoustic experiments were limited to isolated pitches without musical context, making the pitch representation less generalizable to music generation, where musical context is vital. In this work, we use the concept of pitch class and octave (both having been used in PHCCCF) but stick to the embedding representation learned through back propagation rather than static representation. To the best of our knowledge, PiRhDy [19] is the only recent music generation work that employed pitch class and octave, but the authors did not provide a comparison with the MIDI number encoding. Therefore, our work should be the first to compare these two different pitch encodings.

Alternative pitch encodings with domain knowledge have also been used in the symbolic music domain, but less frequently used in symbolic music generation. The tonnetz representation, proposed by Euler [7] in 1739, arranges pitch classes along major third, minor third, and perfect fifth dimensions. It has been successfully used for both feature extraction [3] and generative modelling in [20], but lacks smooth presentation of voice leading (namely the semitone or major second movements). Pitch classes are also effectively adopted in some discriminative tasks, e.g. chord classification[17] and style clustering [6, 31], but pitch-class-only representations ignore octave information needed for precise pitch description in generation tasks. This work, as a result, combines the pitch class and the octave feature for comparison with the MIDI-number encoding.

### 2.2 Word Embedding Training Strategies

Word embedding suffers from the representation degeneration problem [8], i.e. the embedding vector distribution is gradually distorted into a narrow cone shape, increasing the similarity of the word vectors with decreasing performance. [30] explained that rare token embeddings are pushed by their gradients away from the non-rare tokens, causing degeneration. Our observations, likewise, show that the pitch embedding space is biased towards the imbalanced pitch and pitch class distributions in the dataset. To prevent degeneration, [30] proposed a gradient gating strategy that freezes the rare tokens at early training, inspiring our strategy two.

Regarding poor numeracy performance of word embedding in language models [27], Gorishniy et al [9] demonstrated the advantages of using piecewise linear encoding (PLE) and sinusoidal activation functions (PAF) for numerical feature embedding. The FME [10], adopted an similar embedding scheme to embed pitch, duration and position features, ensuring the consistency of relative musical features such as intervals



and durations in the embedding space. Instead of enhancing the pitch feature numeracy, this work studies the robustness brought by periodicity of pitch class and octave.

### 3 Methods

#### 3.1 Pitch Encodings

In the commonly used music representations, (e.g. the MIDI event representation and the REMI representation), a pitch is encoded as a single token, indexed by the MIDI number, which we refer to as the MIDI number encoding. Being represented by one-hot vectors before embedding, the pitch vectors contain no domain knowledge information about frequency or pitch height as the dimensions are isotropic. This encoding is the baseline encoding.

The **class-octave** encoding is an alternative pitch encoding, which is less used in generation models [19] but more common in discriminative tasks as part of the input features [17, 6, 31, 18, 2]. It encodes a pitch to its pitch class (0 to 11) and the pitch octave number (0 to 9, if considering the highest valid MIDI pitch). If this encoding is used in a sequential music representation, a pitch is represented by two separate tokens in the sequence: the pitch class token ( $p \bmod 12$ ) followed by an octave token  $\lfloor \frac{p}{12} \rfloor$ . For instance, the pitch 60 (C4) is encoded into token `p0` and `o5`, corresponding to two different embedding vectors, respectively.

What is unique to about the class-octave encoding is its robustness to the slight pitch shifts, which manifests the proximity in listening experience before and after the shift. The pitch class-octave encoding is experimented to be compared with the baseline encoding because it has a much smaller vocabulary size ( $12 + \text{the number of octaves to be encoded}$ ), which reduces the chances of over-parameterization. The pitch class-octave encoding also explicitly provides the constraints on the translational invariance for octaves ( $\delta = 12$ ), i.e. all pitches that are octaves apart from each other share the same pitch class vector. Hence, it is expected to result in pitch embeddings that outperforms that of the MIDI number encoding.

#### 3.2 Freezing Pitch Embedding in Early Training

The decreasing trend of the pitch performance over epochs suggests the possibility of deterioration of the pitch embeddings. As proposed in [30], freezing the rare token embeddings at early stage can alleviate the performance decline by preventing the embedding degeneration problem. In the music generation task of interest, most datasets have imbalanced pitch and pitch class distributions. Likewise, if the pitch embeddings are frozen at the optimal state, the resulting pitch performance is expected to be better. Hence, freezing the pitch embeddings at different epochs of training is investigated.

#### 3.3 Metrics

This study employs two kinds of evaluative metrics to examine whether the proposed strategies effectively alleviate the pitch performance issue caused by imbalanced pitch (and pitch class) distribution in the dataset. The first kind evaluates the pitch embedding space itself and the other kind focuses on the generation quality, particularly about pitch.

**Embedding Space Evaluation Metrics** In order to obtain consistent embedding representations for intervals, (i.e. relative pitch features), the pitch vectors in the embedding space must follow certain constraints about intervals. According to FME [10], all the interval vectors  $\{\mathbf{p}_{i+\delta} - \mathbf{p}_{j+\delta} | \delta \in \mathbb{Z}\}$  that represent the same pitch distance  $|i - j|$  must have the same magnitude. As is not satisfied in most existing generation systems, this constraint is too strict. Therefore, we propose SLD, a metric that loosely measures the violation of such constraints. The Standard deviation of L2 Distances of pitch vectors<sup>3</sup> in the embedding space is defined as follows:

$$\text{SLD}(\mathbf{P}) := \sum_{\delta=1}^{\delta_{\max}} \left[ \text{Std}_{i=1..n-\delta} (|\mathbf{p}_{i+\delta} - \mathbf{p}_i|) \right]. \quad (1)$$

This metric penalizes the differences in magnitudes for all pitch vectors whose difference vector represents the interval of  $\delta$  semitone.  $\delta_{\max}$  is empirically set to 24 here for two octaves, since intervals larger than that are likely to have more different auditory experiences depending on the actual pitch height [23]. A better pitch embedding space is expected to have a lower SLD.

**Generation Quality Evaluation Metrics (for Pitch)** Admittedly, it is not practical to conduct a subjective listening test when the many models are experimented, also because the differences in the generated pitch distributions can be subtle to human audiences. Hence, objective metrics are adopted to evaluate the pitch performance in the generated samples. That is, the entropy of pitch class distribution  $H(\text{PC})$ , and for pitch  $H(\text{P})$ , as used in [28, 5]. These two metrics can accurately capture the lack of pitch diversity, or the repetition of very limited pitches when the  $H(\text{P})$  is lower than that of the dataset, while  $H(\text{PC})$  is an octave-agnostic version of  $H(\text{P})$ . The  $H(\text{P})$  and  $H(\text{PC})$  distributions of the test dataset are first approximated by Gaussian Kernel Density Estimation (KDE), and then compared to the KDEs of generation distributions. The overlapping area (OA) [29] between the fake and the true is used to score the generation quality, with the higher OA being the better.

## 4 Experiment Setup

### 4.1 Dataset

A cleaned version of the Wikifonia dataset<sup>4</sup> is used. Specifically, we only keep the songs with constant 4/4 time signatures. The training set (90%) contains 3,861 songs, and 429 songs for the test set. Note that quite a number of songs have modulations (key changes), so we do not do any kind of transposition for dataset balance as it will not completely balance the distribution. The imbalanced pitch class distribution is plotted in Figure 1. As can be seen, The frequent pitches come from the C major scales, the rest being rare in both subsets. The pitch class entropy  $H(\text{PC})$  of the train set and dataset are

<sup>3</sup> The pitch vectors must be  $z$ -score transformed before SLD calculation, so as to eliminate the influence of the scaling along different dimensions

<sup>4</sup> <http://www.wikifonia.org>

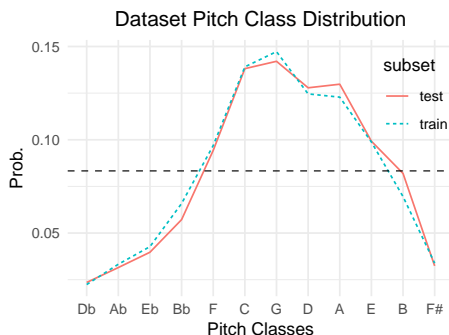


Fig. 1: Pitch Class Distributions of the Training Set and the Test Set

3.370 and 3.376 bits, respectively. Hence, the  $H(PC)$  of generated melodies should also be close to this dataset average value.

## 4.2 Data Representation

The input music representation resembles the REMI representation [13] because of the usage of duration, bar and position tokens. However, the features *chord*, *tempo* and *velocity* that are defined in REMI are ignored. In this work, the vocabulary set is formed by *pitch*, *octave* (if used), *duration*, *bar* and *position* tokens<sup>5</sup>. We also vary the beat resolution settings, allowing for the identification of consistent patterns in the model performance and a more robust analysis of the results.

## 4.3 Model and Training Specifications

A 4-layer transformer-XL network (proposed by [4] as used in [13, 28]) is employed to generate melodies in a next-token-prediction manner. The parameter size of the network is also cut down to 4M from the original, 12-layer model of 150M parameters in order to reduce the risks of over-fitting on such a small symbolic music dataset.

The experimented models in this work share most of the training hyper-parameters, including the cross-entropy loss, 0.9 to 0.1 train-test split, the optimizer AdamW [21], the learning rate  $8e-4$ , batch size 32 and the number of epochs. Since the Transformer-XL architecture does not have a limit on the sequence length, the training sequence length is set to 1,024 tokens chunked into 8 segments of 128 tokens. The model is saved at the end of each epoch. Top- $k$  sampling (at  $k = 5$ ) and softmax temperature  $\tau = 1.0$  is used for inference. For each model, 128 melodies are (unconditionally) sampled to evaluate the generation quality. However, only 512 tokens are sampled for each melody since longer sequences seem to be repetitive at the end.

<sup>5</sup> Miscellaneous tokens include a REST for silence that comes before duration, and PAD that pads the sequence

## 5 Results and Discussions

### 5.1 Experiment 1 - Comparison of Pitch Encodings

In this experiment, models are trained in pairs for token-by-token melody generation, teacher-forced. The two models in each pair share the common dataset, model architecture and only differ in the pitch encoding of the data representation: one uses the MIDI number encoding and the other uses the class-octave encoding. 24 pairs are set in order to compare the performance of two pitch encodings in different hyper-parameter configurations (e.g. the beat resolution).

**Generation Result Metrics** 128 melodies are sampled from each model for evaluation. The distributions of pitch entropy ( $H(P)$ ) and pitch class entropy ( $H(PC)$ ) are calculated for all the samples for each models. The overlapping area between the generation distribution KDE and the test dataset KDE are obtained to represent the performance of a model on a specific metric. Higher values are better.

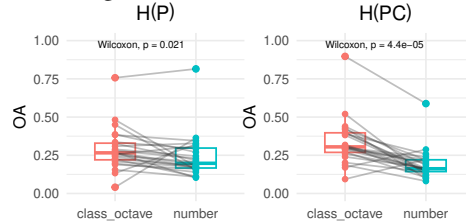


Fig. 2: Paired Box Plots of Model Performance Scores on Two Objective Metrics, Grouped by Pitch Encoding Used.

In Figure 2, each dot represents the generation metric distribution performance measured by OA of a model, grouped by the pitch encoding that the model uses. Line-connected dots are pairs of models that only differ in non-pitch hyper-parameters. The class-octave group on average outperforms the number group because of higher average performance. Paired Wilcoxon tests on show that, such mean differences are significant ( $p = 0.021$  for pitch, and  $p = 4.4 \times 10^{-5}$  for pitch class).

Note that there is a considerable gap between the two models with the highest OA  $H(PC)$ , where the class-octave has model learned about 81% of the true  $H(PC)$  distribution while the number pitch model only learned around 60%. This suggests that the best performance on pitch class is dominated by class-octave encoding. However, the best performance of the class-octave group on pitch is slightly inferior to the number encoding, which is not surprising since the number-encoding pitch vectors have more parameters directly fitted to pitch distributions more accurately.

**Embedding Space Metrics** The best model and the worst model judged by OA  $H(PC)$  of each group are picked out, with their embedding space visualized in Figure 3. PCA is used to reduce the dimensionality from 32 two the 3 primary components with the largest variances for visualization purpose. However, the two number pitch visualizations are obtained from Uniform Manifold Approximation and Projection (UMAP)

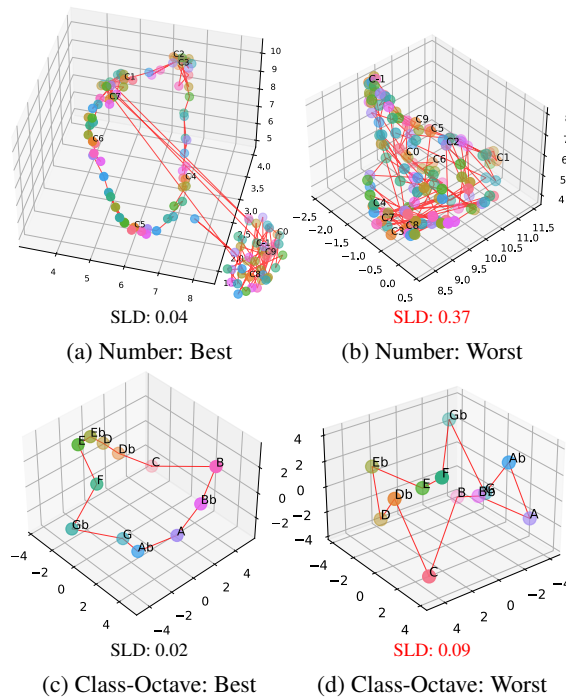


Fig. 3: The pitch / pitch class vectors are plotted as points, colored by pitch classes. Red segments represent semitone relationship. Clear proximity between semitone pitch vectors is shown in embedding spaces with low SLD values, but less clear when SLD is high. The low SLDs are consistent with the actual best generation performance on both pitch and pitch class.

since the large number of pitch vectors are crowded in the PCA results and can be better clustered in the UMAP results. Note that when calculating SLD for class-octave embedding spaces, we first take the sum of octaves to all the pitch classes to restore the 128 pitches<sup>6</sup>.

Overall, the visual differences between the best cases and the worst cases in Figure 3 suggest that pitch embedding space quality greatly contributes to the model performance on pitch performances. The two best cases demonstrate the success of modelling pitch distributions in early finished instances (because of other hyper-parameters e.g. beat resolution, that affects the model’s learning ability before over-fitting happens), while the two worse cases show how the embedding space deteriorates over epochs.

<sup>6</sup> Summation is just one way to approximate the vector representation of the pitch feature, under the assumption that the embedding vectors are semantic and they follow the analogy property of word embedding. However, this can lead to different expected ranges of SLDs from that of the number pitch vectors, because the vector differences cancel out octave vectors if the two pitches are from the same octave. After all, this approximation error does not change the overall trend of SLD, which is of interest, since the error is only on the formula.

By comparing Figure 3a with 3c, and 3b with 3d, the class-octave encoding shows strong robustness and the embedding spaces suffer much less from the rare-token degeneration problem. That is, in MIDI number encoding, the lowest and the highest pitches are always rare tokens, regardless of the data augmentation methods such as random transposition. As a result, the rare pitch tokens are pushed into a cluster during the optimization (as demonstrated in [30]), resulting in worse pitch performance.

In contrast, for the class-octave encoding, the rare pitches are represented by only a few octave tokens (e.g.  $\circ 0$  to  $\circ 3$ ,  $\circ 8$  to  $\circ 9$ ), and their pitch classes are no different from that of the non-rare pitches because they share the pitch classes. The degeneration problem can still be seen on the visualization (3d), i.e.  $\{D\flat, E\flat, F\sharp, A\flat, B\flat\}$  these rare pitch classes in this dataset (see Figure 1), are extruding out away from the non-rare pitch classes, causing worse pitch performance.

To conclude, the class-octave encoding is an underrated pitch encoding in the symbolic domain, outperforming the zero-domain-knowledge number encoding. It displays stronger robustness and interpretability. In addition, the results show that a low SLD is a necessary condition of models being able to precisely capturing pitch and pitch class distributions.

## 5.2 Experiment 2 - Freezing Pitch Embedding Space at Different Stages of Training

This experiment validates the existence of the optimal state of the pitch embeddings by freezing the pitch embedding vectors at different epochs of training and tracking the their states (SLD and pitch performance).

The best set of non-pitch hyper-parameters<sup>7</sup> used in experiment 1 was adopted. Specifically, both the number encoding models and the class-octave model achieved lowest test set loss around epoch 5, which ended way earlier than other models who were trained for around 30 – 40 epochs, suggesting that further training the models is prone to decreasing performance.

However, as previously discussed, the SLD of the number encoding model did not decrease (or slightly decreased but rose very quickly at the beginning), which is a general problem regardless of most hyper-parameter settings. Conversely, the SLD of the best class-octave model decreased in the first 5 epochs and started to increase, reaching the best OA H(PC) at epoch 5, too. Hence, this experiment is dedicated to **class-octave** encoding where the SLD can decrease more noticeably at the beginning of training.

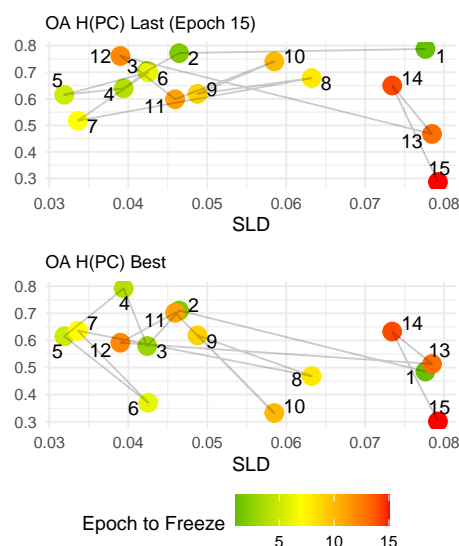
15 model instances were separately trained for 15 epochs from scratch (for better reproducibility), except that every time the pitch vectors are frozen 1 epoch later by zeroing out the gradients of pitch vectors. For each model, pitch embedding SLD was evaluated at each epoch until frozen. Note that the embedding vectors would still slightly change after frozen because of the existence of the projection layer between embedding output and the transformer input, which was not frozen as it is shared by all word vectors (including non-pitch vectors). In actual results, there was a very slight increase in the SLD for models but they did not change the ranking of different SLDs.

<sup>7</sup> Hyper-parameters: A beat resolution of 8 subdivisions per quarter note, a position grid similar to REMI [13] but each bar now has  $8 \times 4 = 32$  positions instead of 16 used by the authors.

**Metric Results** Each of the trained models was evaluated at two states: **Best**, referring to the epoch of lowest test NLL loss; **Last**, at the end of epoch 15. The embedding metric SLD and pitch performance metric overlapping area OA H(PC) are listed in Table 4a. Arrows near the metrics indicated whether the maximum or the minimum is desired.

Freezing Epoch	SLD Best ↓	H(PC) Last ↑	H(PC) Best ↑	Best Epoch
1	0.078	<b>0.788</b>	0.485	11
2	0.046	0.772	0.710	7
3	0.042	0.705	0.579	13
4	<b>0.039</b>	0.639	<b>0.792</b>	4
5	0.032	0.615	0.615	14
6	0.043	0.700	0.370	11
7	0.034	0.518	0.636	11
8	0.063	0.678	0.469	11
9	0.049	0.619	0.619	14
10	0.058	0.741	0.332	4
11	0.046	0.598	0.702	9
12	<b>0.039</b>	0.761	0.592	12
13	0.078	0.468	0.513	9
14	0.073	0.650	0.633	12
15	0.079	0.286	0.301	4

(a) The Embedding SLDs and Generation OA H(PC)s of 15 Models. SLDs are measured at model reaching lowest test error, not necessarily before or after the freezing moment.



(b) The plot traces the pair of both OA H(PC) and SLD over epochs of freezing. Higher positions stand for better pitch class performance while left positions for better embedding quality.

Fig. 4: Models with Pitch Vectors Frozen at Different Epochs

The 15 models display an interesting 3-phase training dynamics every 5 epochs.

- In phase 1, when frozen before epoch 5, the pitch embedding SLD decreased. The OA H(PC) of the resulting best models climbed up, reaching the maximal performance 0.79 at epoch 4. Models 1 to 4 at epoch 15 have OA H(PC) higher than 0.6, suggesting that the pitch performance is maintained in longer training.
- In phase 2, freezing happened between epoch 6 and 9, when the SLD was higher. Both *last* and *best* OA H(PC) slightly decreased, especially for the best models the OA H(PC) dropped below 0.4.
- In phrase 3, from epoch 10 onward, the pitch performance became much more unstable. The SLD for around epoch 13 to 15 quickly increases, with decreasing OA H(PC). Also notice that the “best epoch” numbers below the dashed line in Table 4a are all smaller than the freezing epoch, indicating over-fitting if pitch embeddings were frozen later than epoch 10. Conversely, if freezing happened before epoch 10, all except model 4 could last for longer training.

The results first suggest that it is effective to freeze pitch embeddings at low SLD level to retain pitch performance at higher levels for both the best and the last mod-

els. In addition, this strategy offers the benefit of being able to train a properly frozen embedding longer before the model is over-fitted.

## 6 Conclusion

This paper presents two strategies aiming at improving the pitch performance of a symbolic music generation model. Both involve incorporating domain knowledge that restricts the pitch representation in terms of feature encoding and feature representation, which effectively alleviate the problem of pitch performance deterioration. Strategy 1 introduces the concept of octave and pitch class, which preserves more pitch proximity than the MIDI number encoding while strategy 2 maintains the advantage of strategy 1 according to the proposed SLD, a loose version of translational invariance property. This study and also calls attention to the generation performance issues related to lack of prior knowledge when designing music generation models. In futural works, the authors plan to generalize such strategies for more advanced pitch features, such as intervals and harmony, or other non-pitch musical features with similar constraints.

## References

1. Briot, J.P., Hadjeres, G., Pachet, F.D.: Deep Learning Techniques for Music Generation – A Survey. arXiv:1709.01620 [cs] (Aug 2019)
2. Chawin, D., Rom, U.B.: Sliding-Window Pitch-Class Histograms as a Means of Modeling Musical Form. *Transactions of the International Society for Music Information Retrieval* **4**(1), 223–235 (Dec 2021). <https://doi.org/10.5334/tismir.83>
3. Chuan, C.H., Herremans, D.: Modeling Temporal Tonal Relations in Polyphonic Music Through Deep Networks With a Novel Image-Based Representation. *Proceedings of the AAAI Conference on Artificial Intelligence* **32**(1) (Apr 2018). <https://doi.org/10.1609/aaai.v32i1.11880>
4. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context (Jun 2019). <https://doi.org/10.48550/arXiv.1901.02860>
5. Dong, H.W., Chen, K., McAuley, J., Berg-Kirkpatrick, T.: MusPy: A Toolkit for Symbolic Music Generation (Aug 2020)
6. Ens, J., Pasquier, P.: Quantifying Musical Style: Ranking Symbolic Music based on Similarity to a Style (Mar 2020)
7. Euler, L.: *Tentamen novae theoriae musicae: ex certissimis harmoniae principiis dilucide expositae*. Saint Petersburg Academy (1739)
8. Gao, J., He, D., Tan, X., Qin, T., Wang, L., Liu, T.: Representation Degeneration Problem in Training Natural Language Generation Models. In: *International Conference on Learning Representations* (Feb 2022)
9. Gorishniy, Y., Rubachev, I., Babenko, A.: On Embeddings for Numerical Features in Tabular Deep Learning (Mar 2022)
10. Guo, Z., Kang, J., Herremans, D.: A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling (Dec 2022)
11. Hsiao, W.Y., Liu, J.Y., Yeh, Y.C., Yang, Y.H.: Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs (Jan 2021)
12. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music Transformer. In: *International Conference on Learning Representations* (2019)



13. Huang, Y.S., Yang, Y.H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1180–1188. Association for Computing Machinery, NY, USA (Oct 2020)
14. Ji, S., Luo, J., Yang, X.: A Comprehensive Survey on Deep Music Generation: Multi-level Representations, Algorithms, Evaluations, and Future Directions. arXiv:2011.06801 [cs, eess] (Nov 2020)
15. Krumhansl, C.L.: The Psychological Representation of Musical Pitch in a Tonal Context. *Cognitive Psychology* **11**(3), 346–374 (Jul 1979). [https://doi.org/10.1016/0010-0285\(79\)90016-1](https://doi.org/10.1016/0010-0285(79)90016-1)
16. Krumhansl, C.L., Kessler, E.J.: Tracing the Dynamic Changes in Perceived Tonal Organization in a Spatial Representation of Musical Keys. *Psychological Review* **89**(4), 334–368 (1982)
17. Laden, B., Keefe, D.H.: The Representation of Pitch in a Neural Net Model of Chord Classification. *Computer Music Journal* **13**(4), 12–26 (1989). <https://doi.org/10.2307/3679550>
18. Lazzari, N., Poltronieri, A., Presutti, V.: Pitchclass2vec: Symbolic Music Structure Segmentation with Chord Embeddings. Workshop on Artificial Intelligence and Creativity p. 17 (Nov 2022)
19. Liang, H., Lei, W., Chan, P.Y., Yang, Z., Sun, M., Chua, T.S.: PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-aware Embeddings for Symbolic Music. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 574–582 (Oct 2020). <https://doi.org/10.1145/3394171.3414032>
20. Lieck, R., Moss, F.C., Rohrmeier, M.: The Tonal Diffusion Model. *Transactions of the International Society for Music Information Retrieval* **3**(1), 153–164 (Oct 2020). <https://doi.org/10.5334/tismir.46>
21. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization (Jan 2019). <https://doi.org/10.48550/arXiv.1711.05101>
22. Meade, N., Barreyre, N., Lowe, S.C., Oore, S.: Exploring Conditioning for Generative Music Systems with Human-Interpretable Controls. arXiv:1907.04352 [cs, eess] (Aug 2019)
23. Moore, B.C.J.: *An Introduction to the Psychology of Hearing*. BRILL (2012)
24. Mozer, M.C.: Connectionist Music Composition Based On Melodic, Stylistic and psychophysical Constraints. *Computer Science Technical Reports* (476) (May 1990)
25. Shepard, R.N.: Geometrical approximations to the structure of musical pitch. *Psychological Review* **89**, 305–333 (1982). <https://doi.org/10.1037/0033-295X.89.4.305>
26. Simon, I., Oore, S.: Performance rNN: Generating music with expressive timing and dynamics. <https://magenta.tensorflow.org/performance-rnn> (2017)
27. Wallace, E., Wang, Y., Li, S., Singh, S., Gardner, M.: Do NLP Models Know Numbers? Probing Numeracy in Embeddings (Sep 2019)
28. Wu, S.L., Yang, Y.H.: The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures (Aug 2020)
29. Yang, L.C., Chou, S.Y., Yang, Y.H.: MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. arXiv:1703.10847 [cs] (Jul 2017)
30. Yu, S., Song, J., Kim, H., Lee, S.m., Ryu, W.J., Yoon, S.: Rare Tokens Degenerate All Tokens: Improving Neural Text Generation via Adaptive Gradient Gating for Rare Token Embeddings (Jun 2022). <https://doi.org/10.48550/arXiv.2109.03127>
31. Yust, J., Lee, J., Pinsky, E.: 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* **5**(1), 113–128 (Oct 2022). <https://doi.org/10.5334/tismir.114>

# VaryNote: A Method to Automatically Vary the Number of Notes in Symbolic Music

Juan M. Huerta<sup>1</sup> and Bo Liu<sup>1</sup> and Peter Stone<sup>1,2</sup>

<sup>1</sup> Department of Computer Science, The University of Texas at Austin

<sup>2</sup> Sony AI

jmhuer@utexas.edu, {bliu, pstone}@cs.utexas.edu

**Abstract.** Automatically varying the number of notes in symbolic music has various applications in assisting music creators to embellish simple tunes or to reduce complex music to its core idea. In this paper, we formulate the problem of varying the number of notes while preserving the essence of the original music. Our method, *VaryNote*, adopts an autoencoder architecture in combination with a masking mechanism to control the number of notes. To train the weights of the pitch autoencoder we present a novel surrogate divergence, combining the loss of pitch reconstructions with chord predictions end-to-end. We evaluate our results by plotting chord recognition accuracy with increasing and decreasing number of notes, analysing absolute and relative musical features with a probabilistic framework, and by conducting human surveys. The human survey results indicate humans prefer VaryNote output (with  $1.5, 1.9 \times$  notes) over the original music, suggesting that it can be a useful tool in music generation applications.<sup>3 4</sup>

**Keywords:** Pitch Autoencoder, Harmonic Analysis, Arrangement Generation, Automatic Ornamentation, Symbolic Music Generation, Chord Predictions

## 1 Introduction

Automating the process of varying the number of notes in a musical arrangement can have many applications. In the case of increasing notes, we can apply this technology to enhance compositions. This application has previously been explored, to some degree, when discussing automatic melody harmonization, arrangement generation, or automatic ornamentation [3, 4, 14–16]. However most of those methods require supervision in the form of labeled data such as Wang et al. POP909 MIDI dataset with segmented melody, arrangement, and bridge notes [17]. The other direction of reducing the number of notes is considered a useful research area relevant to voicing information, automatic melody extraction, and feature extraction in general. However, similar limitations exist

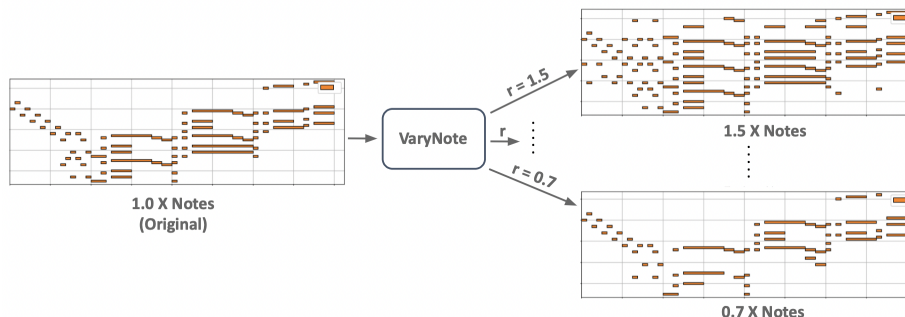
<sup>3</sup> Project page and listening examples: <https://varynote.github.io>

<sup>4</sup> Code: <https://github.com/varynote/varynote-code>



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

for this case: all the methods require segmented data and do not allow for continuous control over the number of notes. In the field of music theory, Schenkerian analysis, or similar variants, can be used to uncover the underlying hierarchical structure of music and use this information to both reduce and add notes to music. However to implement this process automatically, a corpus of analyzed examples is needed, in addition to heuristics to determine how to add or remove notes based on the analysis [7, 13].



**Fig. 1.** VaryNote example usage: given a piece of MIDI music we varying the number of notes according to a desired input-output ratio:  $r$ .

To approach the problem of varying the number of notes in symbolic music automatically, we introduce VaryNote, a novel method that uses an autoencoder trained on pitch reconstructions that preserve chord structure. This design is considering several studies that have surveyed human listeners and discovered maintaining harmonic chord structure, while removing other aspects can still allow human listeners to recognize the original tune [6]. An example is the common practice of describing the chord progression I-V-vi-iii-IV-I-IV-V in terms of Pachelbel’s Canon in D. In addition, VaryNotes’ design effectively preserves rhythmic features, which we believe is beneficial as listeners can recall a song based on its melody, even with different instrumentation or tempo [2, 5, 18]. In summary, this paper makes the following contributions:

1. Formulate the task of varying the number of notes in music as an optimization problem.
2. Introduce VaryNote, a novel deep learning method consisting of an autoencoder trained with a combined loss of pitch reconstructions and chord predictions. We demonstrate that VaryNote can significantly outperform a baseline based on heuristics from music theory at the task of varying the number of notes on the BTL MIDI dataset.

## 2 Problem Formulation and Background

In this section, we introduce a general problem formulation of varying the number of notes. Then, in Section 3, we provide a description of our proposed strategy to solve this problem.

## 2.1 Problem Formulation

Symbolic music information is a type of sequential information. We represent music using a piano roll representation, a frame-wise representation, where every time step is a multi-hot encoding of the pitches that are played at time  $t$ . Assuming a time-length  $H$ , with  $P$  possible note pitches, we denote  $\mathcal{X} = \{0, 1\}^{P \times H}$  as the input space. We define a piano roll matrix  $X \in \mathcal{X}$ , and quantify the number of notes as the sum of non-zero elements<sup>5</sup> in  $X$ :

$$\text{Number of Notes} : m := \|X\|_0. \quad (1)$$

The goal is to learn a mapping  $f_\theta(X | r) \rightarrow \hat{X} \in \mathcal{X}$  parameterized by  $\theta$  such that  $\hat{X}$  increases or decreases the number of notes in a piano roll  $X$ , given an *output-input ratio*,  $r \in \mathbb{R}^+$  of notes that controls the relative sparsity of the output. Formally, we view the problem of automatically varying the complexity of harmonies as the following optimization problem:

$$\min_{\theta} \mathcal{D}(f_\theta(X | r), X) \text{ s.t. } \frac{\|f_\theta(X | r)\|_0}{\|X\|_0} = r. \quad (2)$$

Conceptually,  $\mathcal{D}$  is a divergence that measures how similar the reconstructed  $f_\theta(X | r)$  is to the original piece of music  $X$ . Informally, it can be characterized as the degree to which an average human listener would consider the two passages to be "the same tune" and is related to cover song identification [9]. While both melodic contour and harmonic contour can be used to quantify music similarity in music theory, they may not provide a complete picture due to subjective differences in interpretation and other factors. Ultimately, this divergence is based on human judgement and is not easily measurable so we resort to using a surrogate loss defined in Section 3.2 that estimates the ability for a reconstructed piano roll to identify the original chords. We assume that when this loss is small, people will consider the passages to be the same tune. This assumption is considering the importance of harmonic structure in perceptual similarity [5, 6, 18]. However, we are not making any claim that this surrogate loss is the best possible quantitative estimation of the true divergence.

## 3 Method

The general problem presented in Eq. (2), is to conditionally generate music based on  $r$ . A straightforward approach is to first apply representation learning on the music and then reconstruct it conditioned on  $r$ , similar to autoencoder style models in machine learning. In this section, we introduce a novel autoencoder, named VaryNote. Specifically, VaryNote consists of two parts. The first is a pitch autoencoder (Section 3.1) where the encoder compresses a piece of music into a latent representation and the decoder reconstructs music from the latent representation. The second is a threshold mask (Section 3.1) that controls the sparsity in the output music. To train the weights of the pitch autoencoder we define a novel divergence in Section 3.2. This divergence is a combination of error on reconstruction and error on symbolic chord predictions.

<sup>5</sup> This definition is not exactly aligned with the music theory concept of a note, since we are ignoring note length, but it captures the amount of pitch information and is simple to calculate.

### 3.1 Architecture

**Pitch Autoencoder** An autoencoder is a model that seeks to learn a compressed representation of an input. It does so by passing the input through an information bottleneck of lower dimensionality than the original input. We apply an autoencoder to a piano roll  $X_t$ . We first breakdown the piano roll matrix by time-step, defining a sequence of pitch vectors such as:

$$X_t \triangleq x_{t-H:t} = [x_{t-H}, \dots, x_{t-1}]. \quad (3)$$

The goal is to learn to reconstruct a pitch vector  $x_t$  at time  $t$  using the encoder with  $d = 32$ :  $E_\phi : \mathbb{R}^{\mathcal{X}} \rightarrow \mathbb{R}^d$  and decoder  $D_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{\mathcal{X}}$ , parameterized by  $\phi$  and  $\theta$  respectively. In addition we test several non-linear activations  $\lambda$ :

- **ReLU**: rectified linear activation function.
- **k-WTA**: the  $k$ -largest neurons in the autoencoder’s hidden layer (or code) is kept and the rest, as well as their derivatives, are set to zero [11]

$$\lambda_{\text{WTA}}(\mathbf{y} \mid k)_j = \begin{cases} y_j, & y_j \in \{k\text{-th largest elements of } \mathbf{y}\} \\ 0, & \text{Otherwise.} \end{cases} \quad (4)$$

- **Lifetime sparsity**: this is the same as  $k$ -WTA constraints Eq. (4) except we apply percent sparsity  $\%k$  of the hidden layer across the entire mini-batch. This encourages a wider range of neurons to be active [12].

We encapsulate the autoencoder in a function  $\mathcal{A}$ , and define the autoencoder reconstruction as  $\hat{X}_t$ :

$$\hat{X}_t \triangleq \mathcal{A}_{\phi, \theta}(X_t) = D_\theta \circ \lambda[E_\phi(X_t)]. \quad (5)$$

**Thresholding Piano Rolls** After training the autoencoder, VaryNote reduces or increases the number of notes in a piano roll using a threshold mask. This mask essentially zeros out everything except the top- $k$  values in the autoencoder output. Specifically, consider the autoencoder output of size  $S = P \times H$  with  $\hat{X}_t \in \mathbb{R}^S$  as defined in Eq. (5). Denote the  $k$ -th smallest element of  $\hat{X}_t$  as  $\hat{x}^{(k)}$ . We define a mask  $M$ :

$$M(\hat{X}_t) = \mathbf{1}(\hat{x}_{i,j} \geq \hat{x}^{(k)}). \quad (6)$$

For any desired *output-input ratio*  $r$ , and  $m$  number of notes in the original piano roll  $X_t$ , we find a  $k$ -th order that achieves  $r$ :

$$k(r) = \lfloor S - rm \rfloor. \quad (7)$$

Now we can write Eq. (6) using a target  $r$ :

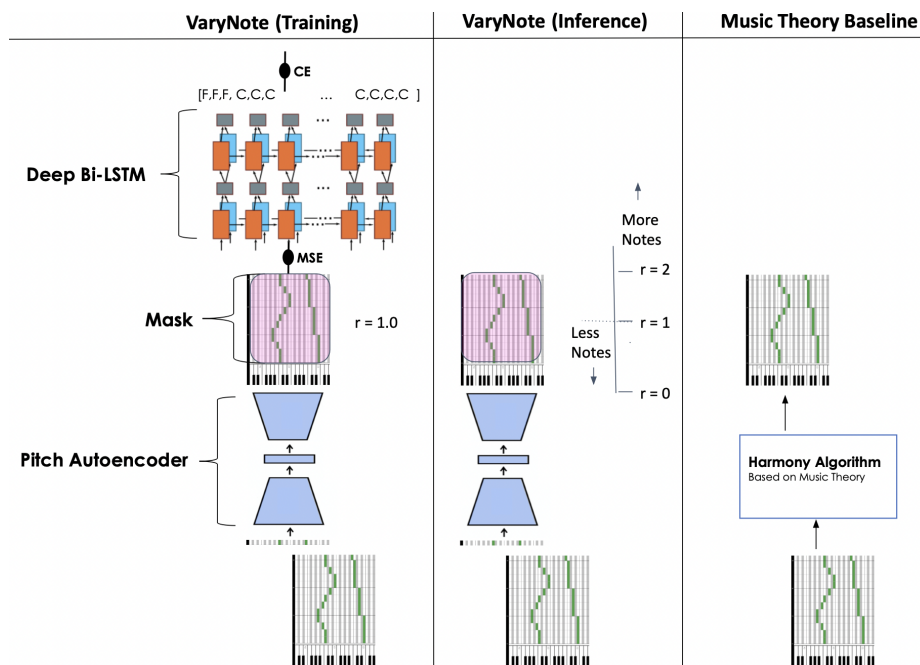
$$M(\hat{X}_t \mid r) = \mathbf{1}(\hat{x}_{i,j} \geq \hat{x}^{(k(r))}). \quad (8)$$

Applying the mask defined in Eq. (8) on  $\hat{X}_t$  assures we end up with  $rm$  number of notes:

$$\|M(\hat{X}_t \mid r)\|_0 = S - k \approx rm. \quad (9)$$

**VaryNote Architecture** At this point we have described all the required components of VaryNote. We have slightly different treatment for increasing or decreasing notes. *Increasing Number of Notes:* to apply a relative increase in number of notes (output-input ratio  $r \geq 1$ ), we add the pitch autoencoder output with the original music and apply the mask in Eq. (8) that assures we meet the desired output-input ratio constraints. *Decreasing Number of Notes:* to apply a relative decrease in number of notes (output-input ratio  $r < 1$ ), we multiply, element-wise, the pitch autoencoder output with the original music and apply the mask in Eq. (8) that assures we meet the desired output-input ratio constraints. In summary:

$$F_{vc}(X | r) = \begin{cases} \mathbf{M}(\mathcal{A}_{\phi,\theta}(X) + X | r), & \text{if } r \geq 1 \\ \mathbf{M}([X + \mathcal{A}_{\phi,\theta}(X)] * X | r), & \text{if } r < 1. \end{cases} \quad (10)$$



**Fig. 2.** During training VaryNote combines MSE loss and softmax cross entropy loss. Note the mask requires an output-input ratio  $r$ . During training we can fix  $r$ ; or train without masking, and apply the mask during inference. During inference,  $r$  controls the number of notes.

**Bi-LSTM Architecture for Chord Recognition** To train the weights of the autoencoder  $\mathcal{A}_{\phi,\theta}$ , VaryNote temporarily attaches a Bi-LSTM [1] that uses the output of the autoencoder to make chord predictions as a downstream task. This addition helps our

pitch reconstructions maintain the original chord structure. The task is to find a mapping from  $X_t = [x_{t-H}, \dots, x_{t-1}] \in \{0, 1\}^{P \times H}$  to a corresponding chord sequence per time step  $Y_t = [y_{t-H}, \dots, y_{t-1}] \in \mathbb{Z}^C$  where  $C$  is the number of symbolic chord classes. The output sequence is passed through a softmax layer that generates the probability for each pitch vector.

### 3.2 VaryNote Surrogate Loss: MSE and Chord Recognition

Ideally we want a differential metric,  $\mathcal{D}$ , that measures music similarity across different arrangement representations as described in Eq. (2). This divergence is not easily quantifiable, so we resort to designing a combined loss that preserves chord structure during pitch reconstructions. Specifically, we propose a combined loss of mean-square error on the pitch autoencoder reconstruction and cross entropy on symbolic chord targets between the Bi-LSTM output  $o_t \in \mathbb{R}^{C \times H}$  and target sequence. In our study we reduce the number of pitches to  $P = 64$  and predict  $N = 24$  possible chords. The total loss can be described as:

$$\begin{aligned} \mathcal{D} &= L_{\text{total}} = L_{\text{MSE}} + cL_{\text{CE}} \\ &= \frac{1}{P} \sum_{t=1}^P (x_t - \hat{x}_t)^2 - \frac{c}{N} \sum_{i=1}^N \log \frac{\exp(o_t[y_i])}{\sum_{y=1}^K \exp(o_t[y])}. \end{aligned} \quad (11)$$

Finally, VaryNote trains with the presented  $L_{\text{total}}$

$$\min_{\theta, \phi} L_{\text{total}}(F_{vc}(X | r), X) \quad \text{s.t.} \quad \frac{\|F_{vc}(X | r)\|_0}{\|X\|_0} = r. \quad (12)$$

The constraints are automatically met by the mask  $M$ . During training we can fix  $r$ . Alternatively, VaryNote can train without a mask by using the autoencoder output with no threshold, and then apply a mask during inference.

### 3.3 Music Theory Baseline

VaryNote enables varying the number of notes along a continuous spectrum from very sparse to very dense orchestration. There are no existing rule-based methods that can similarly control the number of notes in the same way. There are relevant examples of music algorithms based on theory rules such as voice leading applied to automatic harmonization. However, none of these methods provide a comparison as we increase or decrease notes. So we designed a method that can automatically generate harmonic intervals and automatically remove notes. To add notes, the algorithm requires two steps. First we sample harmonic intervals from a probability distribution computed from aggregating music theory rules used in prior work [10]. Table 3 in the Appendix summarizes the weighted probabilities of harmonic intervals. To remove notes, we randomly find a note with probability proportional to the density of notes at each time step.

## 4 Experiment

We conduct experiments to compare the original music with the output of VaryNote variants and the baseline method, using three different criteria. Section 4.3 examines the impact on chord structure when notes are added or removed, Section 4.4 compares various musical features using a probabilistic framework, and Section 4.5 evaluates the perception of VaryNote’s output through a human survey. To verify that the surrogate loss in Eq. (11) is superior to a standard MSE loss, we compare VaryNote variants with a standard autoencoder. We also test  $k$ -WTA and Lifetime sparsity constraints on the autoencoders with the expectation they will achieve better generalization on pitch reconstructions. In more detail:

- **Lifetime:** VaryNote with Lifetime ( $k = 3$ ) sparsity constraints, described in Section 3.1.
- **k-WTA:** VaryNote with  $k$ -WTA ( $k = 3$ ) sparsity constraints, described in Section 3.1
- **Ordinary:** VaryNote with no sparsity constraints, using a standard ReLU activation, described in Section 3.1
- **AE:** VaryNote with no sparsity constraints, trained only with  $L_{MSE}$ . That is  $c = 0$  in Eq. (11).
- **Rules:** This is the Music Theory baseline: a simple algorithm that can generate harmonic intervals sampled from weighted probabilities in Table 3 in the Appendix, see Section 3.3 for more details.

### 4.1 Dataset

We use the BPS-FH dataset with 32 movements of Beethoven Piano Sonatas [1]. The musical pieces in the repertoire are represented as binary piano rolls with the time resolution of one 16th note. A sliding window of length 128 time-steps (equal to 32 quarter notes) with a hop size of 16 is applied to the piano rolls to generate the instances for recognition. For chord recognition, we use the maj-min chord vocabulary (including 24 major and minor chords plus an additional ‘others’ class which is excluded from evaluation). We only consider 64 pitches, excluding the lowest and highest octave of the standard 88 key piano notes.

### 4.2 Model Training

We train VaryNote, Eq. (12), without a threshold mask  $\mathbf{M}$  and apply the mask during inference. All VaryNote models are trained with the same train-validation BPS-FH dataset. We train each method for 20 epochs using Adam optimizer, and use  $c = \frac{1}{3}$  for our loss Eq. (11) (i.e:  $MSE + \frac{1}{3} CE$ ). In the interest of reproducibility, all experimental parameters are stored in the code repository.<sup>6</sup>

<sup>6</sup> See the README.md file in the code repository



### 4.3 Recovering Chord Information

To verify that the added or reduced notes do not significantly affect the harmonic structure of music we test if we can recover ground truth chords from the original piano roll (Fig. 3). To accomplish this, first we train each method. Then we transform the validation data using note multiples:  $r \in [0.3, 0.5, 0.7, 1, 1.3, 1.5, 1.9]$ . Finally, using a separate and isolated Bi-LSTM model trained on the original data, we predict symbolic chords for each note multiple.

### 4.4 Music Similarity with Kullback-Leibler Divergence

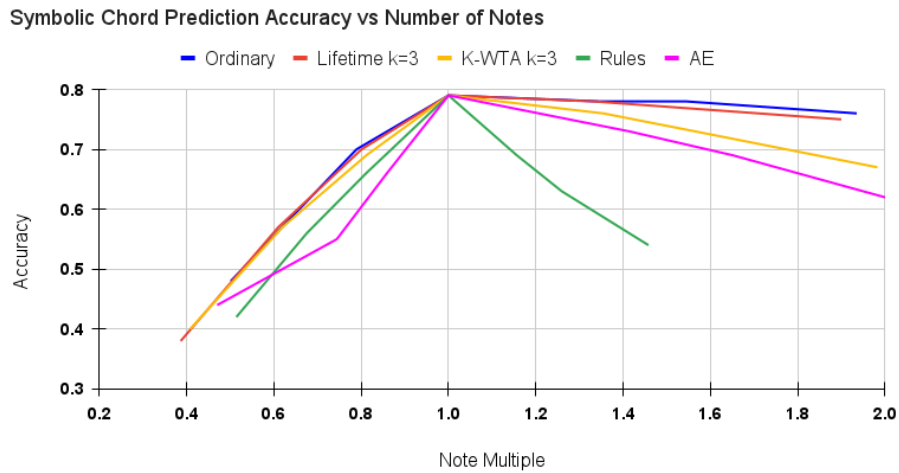
To get a sense of the music similarity without using a human analyst, we apply Lerch et. al. multi-criteria evaluation metrics based on probabilistic measures of musical features [8]. We compare the original MIDI music datasets against every method with  $1.5 \times$  notes by applying kernel density estimation (Gaussian kernel) to find a Probability Density Function (PDF) for each musical feature, and plot them in Fig. 4. Related to harmony, we measure *Pitch Count (PC)*: the number of different pitches within a sample, *Pitch Range (PR)*: the difference of the highest and lowest used pitch in semitones, and *Average Pitch Interval (PI)*: the average value of the interval between two consecutive pitches in semitones. Related to rhythm, we measure *Average Inter-Onset-Interval (IOI)*: the time between two consecutive notes.

### 4.5 Human Evaluations

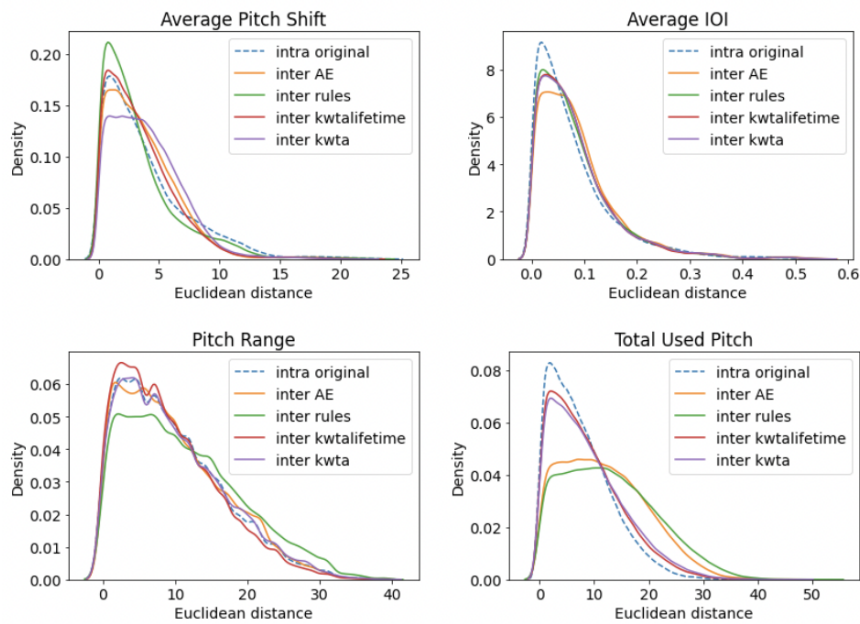
In order to evaluate the practical use of this method, we conduct a small survey designed to understand how human listeners, musically trained and untrained, judge reduced/added note transformations<sup>7</sup>. The goal is to understand if the transformations sound realistic, pleasant, and match our expectation of complexity. 11/30 participants self-report knowing how to play an instrument. We test results for VaryNote Lifetime since it is the best performing method. The survey has three sections. The first is *Musical Preference*: the participants are asked to score VaryNote output from 1-5, 1 being the lowest appeal, and 5 being the highest appeal. The second is *Perceived Musical Complexity*: the participants are asked to score VaryNote output from 1-5, 1 being the lowest complexity, and 5 being the highest complexity<sup>8</sup>. The final is *Music Turing Tests (MTT)*: the participants are given two examples, VaryNote output, and the original music and are asked to identify the piece of music that was fully composed by a human—we do this for piano, and multi-instrument output. The piece the participant selects as being composed by a human receives a score of 1. We sum the total scores and divide by the total number of participants to get a proportion of times humans select the VaryNote output over the original music. To generate a multi-instrument output we simply isolate the notes from the VaryNote output and synthesize the MIDI with a new instrument. Results are summarized in Table 1 and Table 2, the best mean for each question is shown in bold.

<sup>7</sup> The survey form is available in the code repository.

<sup>8</sup> This question is intended to provide insight into how listeners perceive and differentiate between music with different note multiples. It is worth noting that the use of the term "complexity" was chosen to align with a previous version of the paper.



**Fig. 3.** Symbolic chord prediction accuracy using a Bi-LSTM model trained on the original data as we transform our validation data using VaryNote



**Fig. 4.** We extract certain features and use kernel density estimation (Gaussian kernel) to find a probability density function for specific dataset generated by a model. "Intra" refers to comparisons made within a single group of the original music. "Inter," on the other hand, refers to comparisons made between two different groups or categories, in this case comparisons made between the altered music and the original music.

**Table 1.** Human survey results for preference and complexity. Participants are asked to rate the VaryNote output based on preference on a scale of 1-5, 1 being the lowest appeal, and 5 being the highest appeal. Participants also rate complexity from 1-5, 1 being the lowest complexity, and 5 being the highest complexity. There were 30 total participants; 11/30 participants self-reported knowing how to play an instrument. The highest mean for each question is shown in bold.

Experiment	Score Report				
	Original	$\times 0.5$ Notes	$\times 0.7$ Notes	$\times 1.5$ Notes	$\times 1.9$ Notes
<b>Preference Mean</b>	3.09	2.15	2.73	<b>3.62</b>	3.41
Std. Deviation	1.33	1.23	1.23	1.11	1.35
<b>Complexity Mean</b>	3.25	1.62	2.52	<b>3.92</b>	3.85
Std. Deviation	1.61	1.21	1.46	1.24	1.32

**Table 2.** This table includes results for Music Turing Tests (MTT). The participants are given two examples, VaryNote output, and the original music, and are asked to identify the piece of music that is fully composed by a human. The piece the participant selects as being composed by a human receives a score of 1. We sum the total scores and divide by the total number of participants to get a proportion of times humans select the VaryNote output over the original music. The multi-instrument question uses string and woodwind MIDI instruments.

Experiment	$\times 0.5$ Notes	$\times 1.5$ Notes	$\times 1.9$ Notes
<b>Music Turing Test (MTT) - Piano</b>	0.22	<b>0.36</b>	0.17
<b>MTT - Multi-Instrument</b>	N/A	<b>0.57</b>	N/A

## 5 Discussion

As we vary the note multiple  $r$ , the Ordinary and Lifetime methods achieve the highest accuracy in chord recognition according to Fig. 3. We also measure similarity between the original music and  $1.5 \times$  note outputs using KL-divergence. All methods have very similar IOI values. Other harmonic features such as PC, PR, and PI, closely match the original music for all VaryNote methods, and the Rules method is clearly inferior at matching the distribution of the original music.

The human survey results in Table 1 indicate humans prefer VaryNote output, with  $1.5, 1.9 \times$  notes, over the original music. Table 2 indicates humans perceive increased complexity with higher note multiples, except that  $1.5 \times$  notes seems to be perceived with higher complexity than  $1.9 \times$  notes. On the MTT-Piano, participants identify the original music 64% of the time. In comparison, participants only identify MTT-Multi-instrument pieces 43% of the time.

## 6 Conclusion

In summary, we have introduced the task of automatic note variation in music and proposed a novel method, VaryNote, that outperforms a music theory baseline. The proposed method offers significant advantages over traditional approaches by generating a coherent range of outputs for any given note multiple. Notably, our method requires only a corpus of chord labels for training, and it can be easily extended to other divergence metrics beyond chord predictions. Moreover, our results indicate that VaryNote’s output is preferred over the original music, suggesting that VaryNote can be a useful tool in music generation applications.

## Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (CPS-1739964, IIS-1724157, FAIN-2019844), ONR (N00014-18-2243), ARO (W911NF-19-2-0333), DARPA, GM, Bosch, and UT Austin’s Good Systems grand challenge. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

## References

1. Tsung-Ping Chen and Li Su. Attend to Chords: Improving Harmonic Analysis of Symbolic Music Using Transformer-Based Models. *Transactions of the International Society for Music Information Retrieval*, 4(1):1–13, 2021.
2. W. Jay Dowling, James C. Bartlett, Andrea R. Halpern, and Melinda W. Andrews. Melody recognition at fast and slow tempos: Effects of age, experience, and familiarity. *Perception & Psychophysics*, 70(3):496–502, 2008.
3. Kemal Ebcioglu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3):43–51, 1988.
4. Benjamin Evans, Satoru Fukayama, Masataka Goto, Nagisa Munekata, Tetsuo Ono, et al. Autochoruscreator: Four-part chorus generator with musical feature control, using search spaces constructed from rules of music theory. In *ICMC*, 2014.
5. Andrea R. Halpern, James C. Bartlett, and W. Jay Dowling. Perception of Mode, Rhythm, and Contour in Unfamiliar Melodies: Effects of Age and Experience. *Music Perception*, 15(4):335–355, 07 1998.
6. Ivan Jimenez and Tuire Kuusi. Connecting chord progressions with specific pieces of music. *Psychology of Music*, 46:716–733, 9 2018.
7. Phillip B. Kirlin and Jason Yust. Analysis of analysis: Using machine learning to evaluate the importance of music parameters for schenkerian analysis. *Journal of Mathematics and Music*, 10(2):127–148, 2016.
8. Li-chia Lerch and Alexander Yang. On the evaluation of generative models in music. *Neural Computing and Applications*, 32:4773–4784, 2020.
9. Elad Liebman, Peter Stone, Supervisor Kristen, Grauman Scott, Niekum Maytal, Saar-Tsechansky Roger, and B Dannenberg. Sequential decision making in artificial musical intelligence.

10. Chien-Hung Liu and Chuan-Kang Ting. Polyphonic accompaniment using genetic algorithm with music theory. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2012.
11. Alireza Makhzani and Brendan Frey. Winner-take-all autoencoders. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS’15*, page 2791–2799, 2015.
12. Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. *CoRR*, abs/1312.5663, 2013.
13. Alan Marsden. Software for schenkerian analysis. In *ICMC*, 2011.
14. Montserrat Puiggròs, Emilia Gómez Gutiérrez, Rafael Ramírez, Xavier Serra, and Roberto Bresin. Automatic characterization of ornamentation from bassoon recordings for expressive synthesis. In *Baroni M, Addessi AR, Caterina R, Costa M, editors. 9th International Conference on Music Perception and Cognition; 2006 Aug 22-26; Bologna, Italy. Bologna: Bononia University Press; 2006*. Bononia University Press, 2006.
15. Rafael Ramirez and Amaury Hazan. A tool for generating and explaining expressive music performances of monophonic jazz melodies. *International Journal on Artificial Intelligence Tools*, 15(04):673–691, 2006.
16. Miguel Sarabia, Kyuhwa Lee, and Yiannis Demiris. Towards a synchronised grammars framework for adaptive musical human-robot collaboration. In *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 715–721. IEEE, 2015.
17. Ziyu Wang, Ke Chen, Junyan Jiang, Yiyi Zhang, Maoran Xu, Shuqi Dai, Xianbin Gu, and Gus Xia. Pop909: A pop-song dataset for music arrangement generation. *arXiv preprint arXiv:2008.07142*, 2020.
18. Richard M. Warren, Daniel A. Gardner, Bradley S. Brubaker, and James A. Jr. Bashford. Melodic and Nonmelodic Sequences of Tones: Effects of Duration on Perception. *Music Perception*, 8(3):277–289, 04 1991.

## Appendix

### A Weighted Probabilities of Harmonic Intervals

To add notes using the Rules approach, we sample harmonic intervals from a probability distribution computed from aggregating music theory rules used in prior work. The harmonic intervals are summarised in Table 3 below.

**Table 3.** Assigned probabilities  $p$  for intervals according to music theory rules from prior work [10]. To add a new note, a random note from the original music is selected uniformly and harmonized with a random interval drawn with probability  $p$ .

Assigned Prob. ( $p$ )		
$p = 0.19$	$p = 0.10$	$p = 0.003$
Perfect fourth	Minor third	Minor second
Perfect fifth	Major third	Major second
	Minor sixth	Minor seventh
	Major sixth	Major seventh
	Perfect octave	Augmented interval
	Perfect unison	Diminished interval

# ShredGP: Guitarist Style-Conditioned Tablature Generation with Transformers

Pedro Sarmiento<sup>1\*</sup>, Adarsh Kumar<sup>2</sup>, Dekun Xie<sup>1</sup>, CJ Carr<sup>3</sup>, and Zack Zukowski<sup>3</sup>, and Mathieu Barthe<sup>1</sup>

<sup>1</sup> Queen Mary University of London

<sup>2</sup> Indian Institute of Technology Kharagpur

<sup>3</sup> Dadabots

p.p.sarmiento@qmul.ac.uk

**Abstract.** GuitarPro format tablatures are a type of digital music notation that encapsulates information about guitar playing techniques and fingerings. We introduce ShredGP, a GuitarPro tablature generative Transformer-based model conditioned to imitate the style of four distinct iconic electric guitarists. In order to assess the idiosyncrasies of each guitar player, we adopt a computational musicology methodology by analysing features computed from the tokens yielded by the DadaGP encoding scheme. Statistical analyses of the features evidence significant differences between the four guitarists. We trained two variants of the ShredGP model, one using a multi-instrument corpus, the other using solo guitar data. We present a BERT-based model for guitar player classification and use it to evaluate the generated examples. Overall, results from the classifier show that ShredGP is able to generate content congruent with the style of the targeted guitar player. Finally, we reflect on prospective applications for ShredGP for human-AI music interaction.

**Keywords:** Tablature Generation, Computational Musicology, Transformers

## 1 Introduction

Historically, symbolic music generation research has initially relied on datasets using formats such as MIDI, MusicXML, and ABC [8]. The publication of the DadaGP dataset [23] has fostered research on *guitar-focused symbolic music generation*, adopting a symbolic format supporting multiple instruments including tablature information for string instruments. The dataset is built-upon the GuitarPro (GP) format, supporting fingering and expressive information specific to fretted string instruments, features not supported by MIDI. Related works include GTR-CTRL [24], a Transformer-based

---

\* This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (Grant no. EP/S022694/1).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

generative model for guitar tablatures that can be conditioned on musical genre and instrumentation. In this work, we follow some of the future work suggestions from the GTR-CTRL paper, namely the development of a model conditioned on artist style. As a use case, we selected four guitar players of distinct styles to assess the ability of the model to capture and reproduce their stylistic idiosyncrasies: David Gilmour (DG), Jimi Hendrix (JH), Steve Vai (SV) and Yngwie Malmsteen (YM). Token-based heuristics to analyse the style of the guitarists are presented in Section 4, following a computational musicology approach. We present ShredGP, a model that leverages the approach used in GTR-CTRL to condition tablature generation based on guitarist style. The main contributions of this paper are: (1) a method for the generation of multi-instrument guitar tablatures conditioned on artist style; (2) ShredGP, a model for the generation of guitar tablatures in the style of specific guitarists; (3) heuristics to analyze guitar playing styles in the symbolic domain using the token format from DadaGP, that can find applications in computational musicology; (4) SoloGPBERT, a classification model for the task of identifying performances from different guitarists, fine-tuned on the specific four guitar players as a use case.

## **2 Background**

### **2.1 Deep Learning for Symbolic Music Generation**

The task of music generation with deep learning has been steadily demonstrating promising results and achieving state-of-the-art [19][11][15][22]. Architectural choices for generative music models range from Variational Autoencoders (VAEs) [21][25], to Generative Adversarial Networks (GANs) [9][10], and natural language processing (NLP) inspired models, such as Recurrent Neural Networks (RNNs) [18] and, most notably, Transformers [13]. This work explores the use of the Transformer, a sequence-to-sequence model capable of learning the dependencies and patterns among elements of a given sequence by incorporating the notion of self-attention, which has achieved state-of-the-art results in many NLP tasks. Huang et al.’s Music Transformer [13] was the pioneering work to employ self-attention mechanisms for generating longer sequences of symbolic piano music. Other noteworthy contributions in this area include Musenet [20], that used the GPT-2 Transformer model to produce symbolic multi-instrument music across various musical genres; the Pop Music Transformer [14], which used the Transformer-XL architecture and demonstrated better rhythmic structure in generating pop piano symbolic music, and the Compound Word Transformer [12], which explores innovative and more efficient approaches to tokenizing symbolic music during training.

### **2.2 Automatic Guitar Tablature Music Generation**

Despite the widespread availability of guitar tablatures [16][3], there has been limited research on generating guitar tablature music prior to the release of the DadaGP dataset [23]. McVicar [17] proposed an automatic guitar solo generator in tablature format, which utilized probabilistic models and relied on input chord and key sequences. In terms of guitar tab music generation using deep learning techniques, Chen et al. [4] developed a fingerstyle guitar generator, trained on a dataset of 333 examples that did not

use the GuitarPro format. With the release of the DadaGP dataset [23], works regarding automatic guitar tablature generation include GTR-CTRL, a Transformer-XL based model that can control instrumentation and musical genre [24], and LooperGP, a model that can create loops and which was designed having in mind live coding performance applications [2].

### 3 Datasets

#### 3.1 DadaGP Dataset

The DadaGP dataset [23] contains a collection of 26,181 songs, available in two different representations: the *token format*, a form of a textual representation of the songs, and the *GuitarPro format*, named after the GuitarPro software used for tablature editing and playback. The conversion between these two file formats is facilitated by a tool that uses PyGuitarPro [1], a Python library that can parse GuitarPro files. The songs in the DadaGP's *token format* begin with specific tokens such as `artist`, `downtune`, `tempo`, and `start`. Notes played on pitched instruments are represented by tokens in the format `instrument:note:string:fret`. Although this syntax is primarily suitable for string instruments, the combination of string and fret is eventually mapped to a MIDI note, thus supporting other pitched instruments. Percussive instruments, such as the drumkit, are represented using tokens in the form `drums:note:type`. To quantify note durations, the dataset employs the `wait:ticks` token, which uses a resolution of 960 ticks per quarter note. In terms of notating guitar playing techniques, DadaGP uses the note effect (`nfx`) and beat effect (`bfx`) tokens. These include expressive guitar techniques such as *palm mute* (a technique in which the player dampens the strings with their right hand palm), bends and vibratos, tappings, slides, hammer-ons and pull-offs (both represented under the `nfx:hammer` token).

#### 3.2 SoloGP Dataset

In order to create a subset of DadaGP that consisted only of solo guitar parts, we developed a method to extract solo sections from the dataset. By leveraging PyGuitarPro, we searched for *Solo* markers on the files, textual indications of where a guitar solo section is located, then extracted the corresponding guitar part at that section. With this procedure we assembled SoloGP, containing 3,308 guitar solos from more than 1,000 guitarists (12,7% of the tracks in DadaGP), with a total duration of over than 43h.

### 4 Computational Musicology for Guitarist Style Analysis

In order to experiment with guitarist style-conditioned guitar tablature generation, we gathered a corpus of 50 songs from four distinct iconic electric guitar players: David Gilmour (DG), Jimi Hendrix (JH), Steve Vai (SV) and Yngwie Malmsteen (YM). These guitar players are known to have different styles. To validate this, we use a computational musicology approach [6] by comparing features computed on a corpus of examples from each guitarist. We use a Type I error  $\alpha$  of .05 in statistical analyses. General



descriptive statistics about the corpus of each guitar player can be seen in Table 1. We can observe, for example, that YM not only plays a higher number of notes than the other three guitar players, but it does it on average at faster tempos. By opposition, DG usually resorts on slower tempos and fewer notes. Additionally, JH and SV seem to make use of specific guitar techniques (e.g. *bends* and *tapping*, respectively) more often, evidenced by a larger number of `nfx` and `bfx` tokens.

Table 1: Overall statistics of the conditioning subset, per guitar player.

Guitarist	Avg. Tempo	Num. Notes	Num. FXs
David Gilmour	94 bpm	13,534	5,921
Jimi Hendrix	111 bpm	28,843	14,625
Steve Vai	123 bpm	31,715	14,457
Yngwie Malmsteen	142 bpm	33,206	7,093

Figure 1 presents a distribution of the note durations used by each guitarist in the corpus. The results suggest that, whereas DG and JH seem to predominantly use 16th and 8th notes (240 and 480 ticks, respectively), YM plays 16th note triples more frequently (160 ticks). Furthermore, a Kruskal-Wallis rank sum test yielded significant statistical differences between the four note duration distributions ( $H(3) = 12.848$ ,  $p < .005$ ).

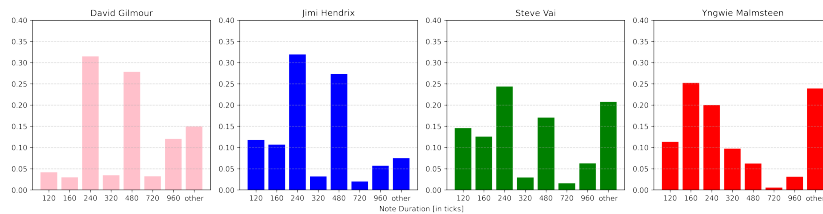


Fig. 1: Note duration distribution (in ticks, 960 ticks per quarter note), per guitar player.

Figure 2 shows the distributions of guitar playing expressive techniques for each guitar player. An overall analysis indicates that DG relies mostly on *bends* by comparison with the other remaining five techniques. YM seems to prefer *hammer-ons* and *pull-offs* (i.e. left hand *legatos*). It is interesting to note that, for the analyzed corpus, SV is the only guitarist using *tapping* (i.e. a guitar technique in which the player hits a fretted note with a finger from the right hand). A Kruskal-Wallis rank sum test showed significant statistical differences between the four guitar playing techniques' distributions ( $H(3) = 24.312$ ,  $p < .001$ ).

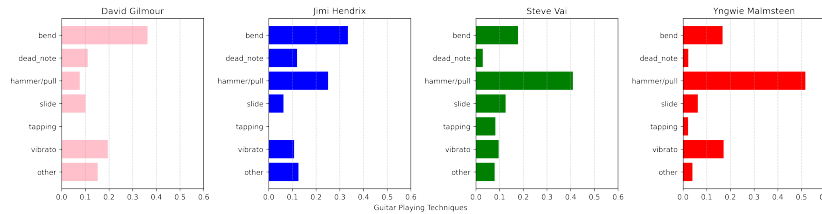


Fig. 2: Guitar playing techniques distribution, per guitar player.

In order to have a better melodic/harmonic understanding of each players' performances, we computed the **pitch class entropy** (PCE) and **scale consistency** (SC) metrics, as defined by [8]. Applied to tonal music, PCE can indicate indirectly how tonal a piece is. Applied to a corpus of tonal music, it reflects the consistency in the keys used. The SC is defined as the largest pitch-in-scale rate over all major and minor scales.

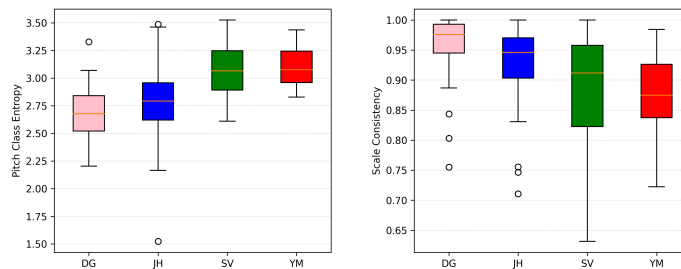


Fig. 3: Box plots for pitch class entropy (left) and scale consistency (right) per guitar player.

Both plots in Figure 3 suggest that SV and YM are more diverse in terms of the keys and pitches used. For PCE, a Kruskal-Wallis rank sum test showed significant statistical differences between the four guitar players ( $H(3) = 73.602, p < .001$ ). Likewise, a Kruskal-Wallis rank sum test yielded significant statistical differences between the four guitarists' SC distributions ( $H(3) = 915.960, p < .001$ ).

Some additional information about the results from PCE and SC can be observed in Figure 4. For example, by analyzing the plot for JH, we notice that the five pitch class peaks could correspond to a *E<sub>b</sub>* minor pentatonic scale (i.e. *E<sub>b</sub>, G<sub>b</sub>, A<sub>b</sub>, B<sub>b</sub>, D<sub>b</sub>*). This is particularly relevant because JH mostly plays in a half-step down guitar tuning (i.e. from the lowest to the highest string: *E<sub>b</sub>-A<sub>b</sub>-D<sub>b</sub>-G<sub>b</sub>-B<sub>b</sub>-E<sub>b</sub>*) and is famous for his use of the minor pentatonic scale. Regarding YM, the other guitarist from the corpus that plays with a half-step down tuning, we can observe that the highest peak also falls on *E<sub>b</sub>*. Finally, DG's distribution has visibly lower entropy than the others, in accordance with the plots in Figure 3.

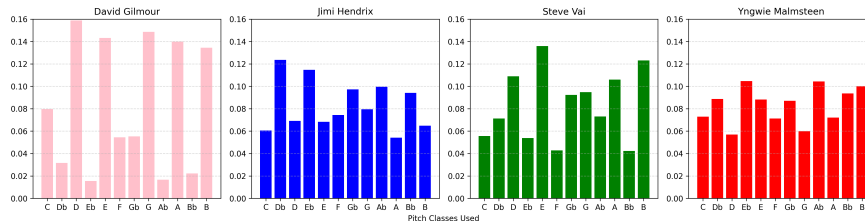


Fig. 4: Distributions of pitch classes used, per guitar player.

## 5 Experiments

Previous work in [23] demonstrated that the use of *control tokens* succeeded in conditioning a guitar tablature generation model on either instrumentation or musical genre. Following a similar approach, we used the *artist* tokens at the beginning of every song to condition the generation on the style of the four guitarists. Although we investigate here a use case for guitarist style imitation, our approach can be used to condition generation for a whole band (e.g. Pink Floyd). Thus, at the start of every respective song we used a *control token* in the form of `artist:pink_floyd` for DG, a *control token* in the form of `artist:pink_floyd` for DG, a *control token* in the form of `artist:jimi_hendrix` for JH, a *control token* in the form of `artist:steve_vai` for SV and a *control token* in the form of `artist:yngwie_malmsteen` for YM. By training the model with these control tokens, we aim to stir the generation output at the time of inference. In order to obtain varied generated songs, we followed two distinct strategies, one using **multi-instrument** compositions (ShredGP-M), trained on the DadaGP dataset and fine-tuned on the multi-instrument version of the conditioning subset, and another using only **solo-instrument** parts (ShredGP-S), trained on the SoloGP dataset and fine-tuned on a solo-instrument version of the conditioning subset (i.e. we manually filtered the guitar parts of every song in this corpus).

### 5.1 Model Description

Regarding architectural choices, we followed a similar procedure as in our previous work [23], namely a Transformer-XL model [7] as backbone architecture. Concerning ShredGP-M, the model’s configuration comprised 12 self-attention layers with 8 multi-attention heads, trained for 200 epochs on the whole DadaGP dataset, and fine-tuned for 20 epochs on the multi-instrument conditioning set, with a learning rate of  $1e-4$  and a batch size of 8 samples. Regarding the ShredGP-S model, due to the lesser complexity of the task (i.e. generative procedure for a single instrument vs. many instruments), we reduced the models’ complexity to 2 self-attention layers and 4 multi-attention heads. ShredGP-S was trained on the SoloGP dataset for 300 epochs, and finally fine-tuned on the conditioning subset for 200 epochs. Both models were training using NVIDIA QUADRO RTX 600 GPUs. Model parameters were heuristically tuned based on prior experiments.

## 5.2 Inference Procedures

The results from GTR-CTRL [24] showed a significant effect of the prompting strategy in conditioning guitar tablature generative models on instrumentation and musical genre. As another source of variability for generated outputs, we also use two distinct prompts for both the ShredGP-M and ShredGP-S models: a *full-prompt*, consisting of the first two measures from compositions of the target guitarist, and an *empty-prompt*, comprising only one initial note. In the case of the *full-prompt* for ShredGP (ShredGP-M-FP), we used a multi-instrument version of said two measures, and used a single-instrument version for the ShredGP-S case (ShredGP-S-FP). A similar reasoning was followed for the *empty-prompt* on both ShredGP-M (ShredGP-M-EP) and ShredGP-S (ShredGP-S-EP). We generated 400 examples per model/prompt configuration, comprising a total of 100 songs per guitar player. For ShredGP-S we defined a limit of 256 generated tokens per song, and for ShredGP-M a limit of 2,048 tokens, as ShredGP-M was set to generate multi-instrument compositions, thus needing more tokens to accommodate for that factor.

## 5.3 Listening Examples

For the experiment settings described in section 5.2, we cherry-picked examples of generated songs for each guitar player. These examples, together with all the generated compositions, without any post-processing, are made available for listening<sup>4</sup>.

## 6 Objective Analysis

Assessing the quality of generative music models is a difficult task, as it usually involves conducting subjective listening tests that are challenging to design and require significant expertise and resources. For the particular case of this study, a listening test would need participants that are familiar with the differences in playing style of the four guitarists. Thus, we resorted on an objective computational analysis based on the metrics described in the next subsection. Finally, we compared these results against the ones obtained for the groundtruth data and presented on Section 4.

### 6.1 Metrics

**Note Duration Distributions:** we calculated note duration distributions on the generated corpus. We computed the Kullback-Leibler divergence (KLD) between the note duration distributions of the generated examples and of the groundtruth data to assess the similarities between these sets. Here, a smaller value indicates less divergences, hence more similarity. **Guitar Playing Techniques Distributions:** we computed these distributions for every guitarist/prompt configuration and calculated the KLD between the groundtruth and generated examples. **SoloGPBERT Classifier:** inspired by the

<sup>4</sup> Currently available at: [https://drive.google.com/drive/folders/1vmaKGYFgp-02fGuEvz9BXtWDZuvHk0Hc?usp=share\\_link](https://drive.google.com/drive/folders/1vmaKGYFgp-02fGuEvz9BXtWDZuvHk0Hc?usp=share_link)

work in GTR-CTRL [24], we here propose SoloGPBERT, a variant of the model introduced in [5], MIDIBERT, as a Bidirectional Encoder Representations from Transformers (BERT)-based masked language able to be configured for downstream classification tasks concerning piano MIDI songs. For SoloGPBERT, we first pre-trained it on the SoloGP dataset for 50 epochs, finally fine-tuning it for two epochs on the conditioning subset for the task of classifying songs of each of the four guitar players, with a split of 55/20/25 between training, validation and test sets. After the fine-tuning, we obtained an accuracy of 89.09% on the test data, thus deeming this model suitable to distinguish the style of each guitarist with a high confidence.

## 6.2 Results

Results for the Kullback-Leibler divergence (KLD) figures for both the note duration and guitar playing techniques’ distributions can be seen in Table 2.

Table 2: KLD scores between the groundtruth distributions for the note duration (left) and guitar playing techniques (right) and the distributions for the generations for each of the four guitar players. Best results in **bold**.

		Note Durations						Guitar Playing Techniques			
		DG	JH	SV	YM			DG	JH	SV	YM
DG	M-FP	0.2808	<b>0.2051</b>	0.0797	0.3045	DG	M-FP	0.3077	<b>0.0552</b>	0.2305	0.2076
	M-EP	<b>0.0975</b>	0.2610	0.1435	0.3963		M-EP	0.2108	<b>0.0875</b>	0.2511	0.2008
	S-FP	<b>0.0497</b>	0.2575	0.1503	0.4455		S-FP	<b>0.1497</b>	0.2952	0.6834	0.6269
	S-EP	<b>0.0546</b>	0.0990	0.2775	0.7579		S-EP	<b>0.1239</b>	0.2832	0.5308	0.4330
JH	M-FP	<b>0.0877</b>	0.2160	0.3442	0.9227	JH	M-FP	<b>0.3052</b>	<b>0.3159</b>	0.7550	0.8674
	M-EP	0.2721	0.5388	<b>0.2050</b>	0.3859		M-EP	0.05351	<b>0.2849</b>	0.6000	0.6267
	S-FP	<b>0.2133</b>	0.2805	0.4967	1.1245		S-FP	0.1683	<b>0.0990</b>	0.3063	0.3687
	S-EP	0.2300	<b>0.2120</b>	0.4862	1.094		S-EP	0.1618	<b>0.0933</b>	0.3707	0.3268
SV	M-FP	0.1542	0.1998	<b>0.1114</b>	0.3814	SV	M-FP	0.3053	<b>0.1121</b>	<b>0.2891</b>	0.4883
	M-EP	0.0920	0.1465	<b>0.0884</b>	0.3844		M-EP	0.7508	<b>0.2293</b>	0.2924	0.3234
	S-FP	0.2263	<b>0.1790</b>	0.3727	0.6814		S-FP	<b>0.2601</b>	0.3232	0.4413	0.7312
	S-EP	0.2343	<b>0.0582</b>	0.1428	0.5225		S-EP	<b>0.2415</b>	0.2656	<b>0.3217</b>	0.2628
YM	M-FP	1.4008	1.3423	<b>0.7131</b>	0.7169	YM	M-FP	0.9105	0.3299	0.1876	<b>0.0449</b>
	M-EP	0.9753	0.7833	0.3224	<b>0.2919</b>		M-EP	1.3512	0.5029	0.2511	<b>0.1439</b>
	S-FP	1.3478	0.6474	0.4610	<b>0.4452</b>		S-FP	0.3894	0.3820	0.5410	<b>0.3192</b>
	S-EP	1.6598	0.8734	0.7884	<b>0.4136</b>		S-EP	1.9688	0.8898	0.4014	<b>0.2360</b>

Concerning **note duration**’s distributions (left side table), the generative outputs conditioned on DG and YM seem to have obtained the best classifications (i.e. 3 best classifications out of 4 possible model/prompt configurations). Generating compositions with a note duration distribution similar to the groundtruth from JH obtains the worst scores (i.e. only 1 best classification out of 4 possible). Regarding the **guitar playing techniques**’ distributions (right side table), YM obtained the best results, while SV-conditioned generations failed to match the groundtruth distribution. Considering the figures of both tables together, an overall analysis suggests that the style from SV is the hardest to model (2/8 best results), whilst YM obtains the highest number of best scores (7/8 best results), followed by DG (5/8) and JH (4/8).

Table 3: Guitar player classification softmax scores from SoloGPBERT, for the generations from every guitarist/prompt configuration. Best results in **bold**.

		Guitar Player Classification Score			
		DG	JH	SV	YM
DG	M-FP	<b>0.5691</b>	0.2103	0.1175	0.1031
	M-EP	<b>0.5086</b>	0.2033	0.1474	0.1407
	S-FP	<b>0.6037</b>	0.2238	0.0945	0.0780
	S-EP	<b>0.5952</b>	0.2034	0.1080	0.1006
JH	M-FP	0.1577	<b>0.5785</b>	0.1358	0.1280
	M-EP	0.2850	<b>0.4054</b>	0.1338	0.1757
	S-FP	0.1229	<b>0.6285</b>	0.1105	0.1380
	S-EP	0.1090	<b>0.6207</b>	0.0835	0.1868
SV	M-FP	0.1839	0.3146	<b>0.3318</b>	0.1697
	M-EP	0.1859	0.2146	<b>0.3498</b>	0.2497
	S-FP	0.1692	<b>0.3499</b>	0.3042	0.1768
	S-EP	0.0619	0.2831	<b>0.3273</b>	0.2827
YM	M-FP	0.0848	0.2364	0.0979	<b>0.5810</b>
	M-EP	0.1085	0.2512	0.1161	<b>0.5242</b>
	S-FP	0.0619	0.2156	0.0755	<b>0.6470</b>
	S-EP	0.0461	0.1501	0.0557	<b>0.7480</b>

The results obtained from the SoloGPBERT classifier can be observed in Table 3. Overall, the generations from all guitarist/prompt configurations were classified correctly, with the exception of ShredGP-S-FP when conditioned on SV, thus showcasing ShredGP’s ability to recreate compositions on the style of each guitarist. It is interesting to note that this matches the conclusions from the results in the previous metrics, where SV also proved to be harder to model. Similarly, the results for YM obtain the best classifications on all the prompt configurations.

## 7 Subjective Analysis

In order to complement the quantitative evaluation, we conducted a subjective analysis of some of the cherry-picked examples. In this section, underlinked song ids in figures’ captions are hyperlinked to facilitate listening. We would like to highlight that what we define as *style* in this paper is viewed from the perspective of a symbolic representation of these guitarists’ playing techniques, thus not taking into account timbral features that express identifiable, unique characteristics of each guitar player. In Figure 5 we display a few measures from a song from ShredGP-M-FP conditioned on JH. We can observe a stylistic phrasing that emphasizes the minor pentatonic scale, composed of patterns with bends that are characteristic of JH.



that the token format in DadaGP opens up new possibilities for the assessment of guitarist playing styles. We anticipate that applying these heuristics to a wider corpus of guitar players could potentially lead to the creation of continuous space of guitar playing style, classifying different guitarists and positioning them in said space accordingly. However, it's worth noticing that these methods do not account for a disentanglement of the guitarist style from the style of the group/band they are playing in, as many times the composition will put creative constraints on the guitar players' part. For our particular case, while JH, SV and YM are theoretically the lead composers in their own groups, the same cannot be said about DG and Pink Floyd.

## 9 Conclusion and Future Work

In this paper we presented ShredGP, a Transformer-based model for guitar tablature generation, conditioned on the style of four distinct iconic electric guitarists. Furthermore, in order to justify the choice of these guitar players as a conditioning subset, we proposed and implemented a computational musicology-driven approach that leverages DadaGP's token format to analyze guitar players' style on different aspects. Generative outcomes from ShredGP were overall able to match the style of each guitarist, obtaining better results when modelling the guitar playing of YM and worst results for SV. These conclusions are supported by both the SoloGPBERT classifier analysis and the comparison of note duration and guitar playing techniques distributions against the groundtruth data. In future work we plan to expand the musicological analysis on a wider selection of artists. Regarding the evaluation of our generative results, we expect to better support these findings with listening tests targetting expert guitar players. Finally, we aim to use the methods in this paper for human-AI co-creative collaborations with guitar players.

## References

1. Abalumov, S.: PyGuitarPro (2014), <https://github.com/Perlence/PyGuitarPro>, Last accessed: 3 Nov 2022
2. Adkins, S., Sarmiento, P., Barthet, M.: LooperGP: A loopable sequence model for Live Coding Performance using Guitarpro Tablature. In: Proceedings of the EvoMUSART Conference (2023)
3. Barthet, M., Anglade, A., Fazekas, G., Kolozali, S., Macrae, R.: Music Recommendation for Music Learning: Hotttabs, a Multimedia Guitar Tutor. In: Workshop on Music Recommendation and Discovery. pp. 7–13. Chicago, IL, USA (2011)
4. Chen, Y.H., Huang, Y.H., Hsiao, W.Y., Yang, Y.H.: Automatic Composition of Guitar Tabs by Transformers and Groove Modelling. In: Proc. of the 21st Int. Soc. for Music Information Retrieval Conference. pp. 756–763 (2020)
5. Chou, Y.H., Chen, I.C., Chang, C.J., Ching, J., Yang, Y.H.: MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding. Tech. rep. (2021)
6. Cook, N.: Computational and Comparative Musicology. In: Clarke, E., Cook, N. (eds.) Empirical Musicology: Aims, Methods, Prospects, chap. 6, pp. 103–127. Oxford University Press (2004)
7. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In: Proc. of the 57th Annual Meeting of the Ass. for Computational Linguistics. pp. 2978–2989. Florence, Italy (2019)



8. Dong, H.W., Chen, K., McAuley, J., Berg-Kirkpatrick, T.: MusPY: A Toolkit for Symbolic Music Generation. In: Proc. of the 21st Int. Soc. for Music Information Retrieval. pp. 101–108. Montréal, Canada (2020)
9. Dong, H.W., Hsiao, W.Y., Yang, L.C., Yang, Y.H.: MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence (AAAI) (2018)
10. Dong, H.W., Yang, Y.H.: Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation. In: Proc. of the 19th Int. Soc. for Music Information Retrieval Conf. pp. 190–198. Paris, France (2018)
11. Eck, D., Schmidhuber, J.: Learning the Long-Term Structure of the Blues. In: Proc. of the Int. Conf. on Artificial Neural Networks. vol. 12, pp. 284–289 (2002)
12. Hsiao, W.Y., Liu, J.Y., Yeh, Y.C., Yang, Y.H.: Compound Word Transformer: Learning to Compose Full-Song Music Over Dynamic Directed Hypergraphs. In: Proc. of the AAAI Conf. on Artificial Intelligence. pp. 178–187 (2021)
13. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music Transformer: Generating Music with Long-term Structure. In: Proc. of the 7th Int. Conf. on Learning Representations. New Orleans, LA, USA (2019)
14. Huang, Y.S., Yang, Y.H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proc. of the 28th ACM Int. Conf. on Multimedia. pp. 1180–1188. Seattle, WA, USA (2020)
15. Kaliakatsos-Papakostas, M., Floros, A., Vrahatis, M.N.: Artificial Intelligence Methods for Music Generation: A Review and Future Perspectives. In: Nature-Inspired Computation and Swarm Intelligence, pp. 217–245 (2020)
16. Macrae, R., Dixon, S.: Guitar Tab Mining, Analysis and Ranking. In: Proc. of the 12th Int. Soc. for Music Information Retrieval Conf. pp. 453–459. Miami, FL, USA (2011)
17. McVicar, M., Fukayama, S., Goto, M.: AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space. Int. Conf. on Signal Processing Proc. pp. 599–604 (2014)
18. Meade, N., Barreyre, N., Lowe, S.C., Oore, S.: Exploring Conditioning for Generative Music Systems with Human-Interpretable Controls. Tech. rep. (2019)
19. Mozer, M.C.: Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing. *Connection Science* **6**(2-3), 247–280 (1994). <https://doi.org/10.1080/09540099408915726>
20. Payne, C.: Musenet (2019), <https://openai.com/blog/musenet>, Last accessed: 12 Jun 2022
21. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. In: Proc. of the 35 th Int. Conf. on Machine Learning. Stockholm, Sweden (2018)
22. Sarmiento, P.: Perspectives on the Future for Sonic Writers. *Journal of Science and Technology of the Arts* **13**(1), 110–114 (2021)
23. Sarmiento, P., Kumar, A., Carr, C., Zukowski, Z., Barthet, M., Yang, Y.H.: DadaGP: a Dataset of Tokenized GuitarPro Songs for Sequence Models. In: Proc. of the 22nd Int. Soc. for Music Information Retrieval Conf. pp. 610–618 (2021)
24. Sarmiento, P., Kumar, A., Chen, Y.H., Carr, C., Zukowski, Z., Barthet, M.: GTR-CTRL: Instrument and genre conditioning for guitar-focused music generation with transformers. In: Proceedings of the EvoMUSART Conference (2023)
25. Tan, H.H., Herremans, D.: Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling. In: Proc. of the 21st Int. Soc. for Music Information Retrieval Conf. pp. 109–116. Montréal, Canada (2020)

# ProgGP: From GuitarPro Tablature Neural Generation To Progressive Metal Production

Jackson Loth<sup>1</sup>, Pedro Sarmiento<sup>1</sup>, CJ Carr<sup>2</sup>, Zack Zukowski<sup>2</sup> and Mathieu Barthe<sup>1</sup> \*

<sup>1</sup> Queen Mary University of London, United Kingdom

<sup>2</sup> Dadabots, <https://dadabots.com/>  
j.j.loth@qmul.ac.uk

**Abstract.** Recent work in the field of symbolic music generation has shown value in using a tokenization based on the GuitarPro format, a symbolic representation supporting guitar expressive attributes, as an input and output representation. We extend this work by fine-tuning a pre-trained Transformer model on ProgGP, a custom dataset of 173 progressive metal songs, for the purposes of creating compositions from that genre through a human-AI partnership. Our model is able to generate multiple guitar, bass guitar, drums, piano and orchestral parts. We examine the validity of the generated music using a mixed methods approach by combining quantitative analyses following a computational musicology paradigm and qualitative analyses following a practice-based research paradigm. Finally, we demonstrate the value of the model by using it as a tool to create a progressive metal song, fully produced and mixed by a human metal producer based on AI-generated music.

**Keywords:** Controllable Music Generation, Transformers, Interactive Music AI, Guitar Tablatures, Human-AI Interaction, Practice-Based Research

## 1 Introduction

With advancements in computing power, new approaches to music generation have emerged. In recent years, deep learning has become a popular approach for automatic music generation, with research focusing on both the audio domain and the symbolic domain. This work extends previous work by Sarmiento et al. [18] using a symbolic music generation model trained on DadaGP, a symbolic music dataset consisting 26k songs of various genres [17]. We follow here a practice-based research approach where a human expert music producer and music AI researchers collaborate to produce music based on machine-generated outputs. We fine tuned the DadaGP-based model with a custom dataset of 173 progressive metal songs, which we refer to in this paper as

\* This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (Grant no. EP/S022694/1). First and second author have equal contributions.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

ProgGP, with the intent of using the model to generate songs, which can be recorded and turned into a fully produced progressive metal song. The model used in this work generates music in the GuitarPro format, rather than formats such as MIDI, MusicXML and ABC seen in other symbolic music generation works [7]. For guitar parts, GuitarPro not only encodes the pitch of each note, but also the location on a guitar fretboard where the note is meant to be played, as well as various expressive techniques (e.g. *vibrato* and *string bending*). We suggest that for certain musical genres, this format is very advantageous for a practice-based approach, as it provides much more information to an artist on how to perform the music that is generated, while still leaving room for creative interpretation. This paper presents the work that went into creating a brand new progressive metal song using neurally generated riffs and ideas that are relevant to the progressive metal genre. As per its main contributions, we highlight: (1) ProgGP, a manually curated progressive metal GuitarPro dataset made available to the community for research purposes; (2) a fine-tuned guitar tablature generative model for the creation of progressive metal tablatures; (3) heuristics for assessing whether generated music holds traits of the desired genre; (4) a practice-based research approach relying on a human-AI partnership where neurally-generated music is selected, edited, and integrated into a composition by a human producer. We also critically examine how to use neurally-generated music to foster creativity, inspire new ideas and improve the writing workflow of artists. We hope that this work will stir more research into human-AI interaction in the musical domain.

## 2 Background

### 2.1 Symbolic Music Generation Using Deep Learning

Recent advances in deep learning have led to promising results in the field of music generation [16], with techniques such as Variational Autoencoders (VAEs) [21], Generative Adversarial Networks (GANs) [8], Recurrent Neural Networks (RNNs) [13] [20], and Transformers [10] being increasingly used. The Transformer model [22] has enabled steep improvements in natural language processing (NLP) tasks and has been adapted for generating symbolic piano music in Huang et al.'s Music Transformer [10]. Other notable works, such as Musenet [14] and Pop Music Transformer [11], have further built on this approach to generate multi-instrument music and improve the generated music's rhythmic structure. However, the task of guitar tablature music generation has received limited research attention until the recent release of the DadaGP [17] dataset, comprising songs in both GuitarPro format, a tablature edition software, and a dedicated textual token format. An initial example of guitar tablature generation work is Chen et al.'s fingerstyle guitar generator [5], despite not being based on the GuitarPro format. More recent works that explore the DadaGP dataset include GTR-CTRL [18], proposing a method for guitar tablature generation with control over instrumentation and musical genre, as well as LooperGP [1], enabling to generate loopable music excerpts with applications for live coding performance.

## 2.2 Practice-Based Research and Computer Music

Many works deal with the notion of ‘practice’ in research. Practice-based research is generally concerned with the knowledge gained through practice and the outcomes of that practice, while practice-led research leads to new understandings about practice itself [4]. Benford et al. describe this kind of research as consisting of three interconnected activities which inform each other in different ways: *practice*, *theory* and *studies* [3]. However, they note challenges in conducting this research with balancing potentially different researcher and artist goals, as well as ethical concerns that can arise through artistic use of new technologies. Artistic uses of new technologies involving AI can be difficult due to the difficulty of prototyping new AI systems and the number of ways that AI can respond to users in different contexts [23]. Amershi et al. [2] provide guidelines on dealing with such unpredictable AI systems, mostly focusing on keeping the user informed on the system’s capabilities and understanding its outputs. AI systems have seen use in musical practice-based research [12] [19] with the *Folk-RNN* model by Sturm et al. being noted to have a number of impacts on musical creation such as a way to inspire ideas, break habits, and a sense of creating something that could not have been created otherwise.

## 3 Practice-Based Research Methodology

### 3.1 Human-AI Partnership

In this work, the first author, a music AI researcher and progressive metal producer, adopted the practice-based research approach described below:

1. Use a deep learning model to generate music in the style of the producer’s preferred genre, progressive metal;
2. Evaluate the outputs of the model using a mixed method evaluation approach, combining objective metrics with subjective evaluation;
3. Craft a song using generated outputs based on outcomes from the evaluation;
4. Learn and record the song;
5. Analyse and reflect on the overall music production process.

The work aims to better understand the successes and issues of the deep learning model in order to help the research community use and improve the model. We also publicly release the dataset used to fine-tune the deep learning model to support similar kinds of research. Finally, we develop a music production process which can be used to efficiently integrate neurally-generated content within a human composition. The artistic content that was recorded can be listened to online and could lead to public performances.

For the neural music generation, we use a model pre-trained on the DadaGP [17] dataset, a dataset consisting of over 26k songs of various genres. The model is trained to produce songs in a tokenized symbolic format, which can be converted to the more commonly used GuitarPro format. This model is further fine-tuned on ProgGP, a curated dataset of progressive metal songs. This fine-tuned model can then be used to generate new songs in the style of progressive metal. For clarification, we do not assess timbre

quality aspects of progressive metal since we are working in the symbolic domain, despite timbre playing an important role in the genre (e.g. heavily distorted guitars, loud and punchy snare and kick drums, etc). However, we do take into account timbre identity through a distinction between distorted and clean guitars in our model.

### 3.2 Fine-Tuning Dataset

ProgGP, the fine-tuning dataset used in our experiments, consists of 173 songs largely from the progressive metal genre<sup>3</sup>. The songs were obtained using Songsterr<sup>4</sup>, a website that hosts GuitarPro files and allows playback using an web-based GuitarPro player. The tablatures (tabs) obtained from this website were not official tabs created by the artists of the songs, but rather created and maintained by the online community. Due to this, there is no guarantee that the tabs used in the dataset are perfectly accurate to the songs they are based on. However, each was verified to at least mostly capture the spirit of the original performance during the construction of the dataset. We limited the dataset to only songs in which the bass guitar and drums have also been transcribed, since the pre-trained model was trained on fully transcribed songs. This however limited the scope of the dataset, as many songs were only available with guitar transcriptions, rather than the full band. Additionally, the model only supports a few common guitar tunings, and only 6 and 7 string guitars. Many bands in this genre use more unique guitar tunings and/or 8 string guitars, so some artists that might be important in the genre of progressive metal may have limited songs or be absent entirely from the dataset. All this led to some artists dominating the dataset more than others. A word cloud representation of the artists used in the ProgGP dataset can be seen in Figure 1. We made ProgGP<sup>5</sup> available upon request, together with a list of songs per artist.



Fig. 1: Word cloud representation of ProgGP's songs per artist distribution.

<sup>3</sup> Some songs included in the dataset are from adjacent genres (e.g. technical death metal).

<sup>4</sup> <https://www.songsterr.com/>

<sup>5</sup> <https://github.com/otnemrasordep/ProgGP>

### 3.3 Model Fine-Tuning

The pre-trained model is based on the Transformer-XL [6] architecture, a modified version of the original Transformer [22] that is more capable of learning longer-term dependency. The pre-trained model used in our experiments was trained for 200 epochs on the DadaGP [17] dataset. We trained the model on the fine-tuning dataset for an additional 65 epochs, at which the loss dropped low enough to trigger early stopping. Checkpoints were saved at every five epochs or training, resulting in 13 models at various stages of fine tuning.

### 3.4 Neural Generation

A new song can be generated by feeding the model a prompt (set of instructions) in the form of a tokenized GuitarPro file. This will be the starting point of the generation, and the model will attempt to continue the song after the prompt. The tempo (in BPM) used for the generated song is taken from the prompt and the number of tokens to be generated is used as a parameter during inference. In DadaGP token format, a token can be a single note, rest, or expressive technique. Prompts used in the generation experiments ranged from a single note, a few measures from songs in the training set, and a few measures of songs not in the training set. The number of generated songs and the model from which to generate the songs can also be specified. Empirical analysis of the generated songs have allowed us to identify common structural patterns in generated songs, which we refer to as ‘sections’, typically consisting of a *riff* that is repeated one or more times with slight variations. The songs will typically start by repeating the notes from the prompt, with minor changes. It will then generate two or three sections afterward, each somewhat changing the feel of the song. While progressive metal songs can contain a large number of different riffs, they tend to build on one another and use references to musical motifs found throughout the song and throughout other songs by the same artist. Between The Buried And Me, a band with a large presence in ProgGP, is particularly well known for this [9]. This is a difficult thing to capture within a model however, as while the different sections seem to fit together naturally, they do not necessarily reference one another. Together with this submission, we release all the generated compositions on the undertaken experiments, cherry-picking some examples<sup>6</sup>.

## 4 Analysing AI-Generated Music

We used a mixed method approach to better understand the outputs of the fine-tuned models, their strengths and weaknesses, and to help the producer select a model for further music production use. This was done by analysing the generated music from each model objectively through the use of common symbolic music metrics, as well as listening through many generated examples and analysing them subjectively in the context of the author’s own knowledge of progressive metal.

---

<sup>6</sup> Available at: [https://drive.google.com/drive/folders/1xaejTcUrPncE4hoyONhSzgS0a5TRo6G\\_?usp=share\\_link](https://drive.google.com/drive/folders/1xaejTcUrPncE4hoyONhSzgS0a5TRo6G_?usp=share_link)

#### 4.1 Objective Metrics

Given the difficulties in assessing the quality of neurally-generated music without using a listening test, specially in the symbolic domain, we resorted on commonly used metrics from the literature, implemented in the MusPy package [7]. For this evaluation, 173 songs were generated from each of the thirteen fine-tuned models, the same number of songs present within ProgGP, in order to maintain consistency when comparing the songs generated to the songs present in ProgGP. The prompt used in this analysis was a single low E note on guitar and bass guitar, and a kick and cymbal hit on drums. This was chosen in order to minimize the influence of the prompt as much as possible, as per the findings in [18].

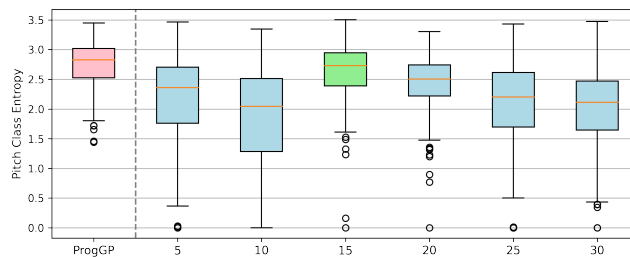


Fig. 2: Pitch class entropy calculated for the songs in ProgGP (pink) and the generated songs from the fine-tuned models for different epochs (blue and green). Model with lowest KL-divergence highlighted (in green).

In previous work, Sarmiento et al. [18] used **pitch class entropy** (PCE), a measure of the entropy of pitch classes used within a song, to evaluate their model. The PCE of the fine tuned models can be seen in Figure 2 (to ease visualization, we omit plots from models after epoch 30). The models fine-tuned for 15 and 20 epochs seem to have a distribution closer to ProgGP. The models fine-tuned for 5 and 10 epochs and beyond 20 epochs generally have a lower mean than the 15 and 20 epoch models. We hypothesize that this could be due to overfitting, causing the model to get stuck on certain sections or notes and repeating them, something seen in the generated songs by the more fine-tuned models. This would lower the pitch class entropy of a model's outputs rather than push it closer to that of the training data which is higher. The rest of the metrics can be seen in Figure 3. They include **drum pattern consistency** (DPC), **number of pitch classes** (NPC), **number of pitches** (NP), **pitch entropy** (PE), **pitch range** (PR), **scale consistency** (SC), **polyphony** (Pol) and **polyphony rate** (PolR). These metrics, while not necessarily giving a definitive idea of the performance of a model, help us understand how the output of certain models matches the training data. They also give an idea of certain characteristics of the music that each model tends to generate. An in-depth definition of each can be found in MusPy's package documentation<sup>7</sup>.

<sup>7</sup> <https://salu133445.github.io/muspy/metrics.html>

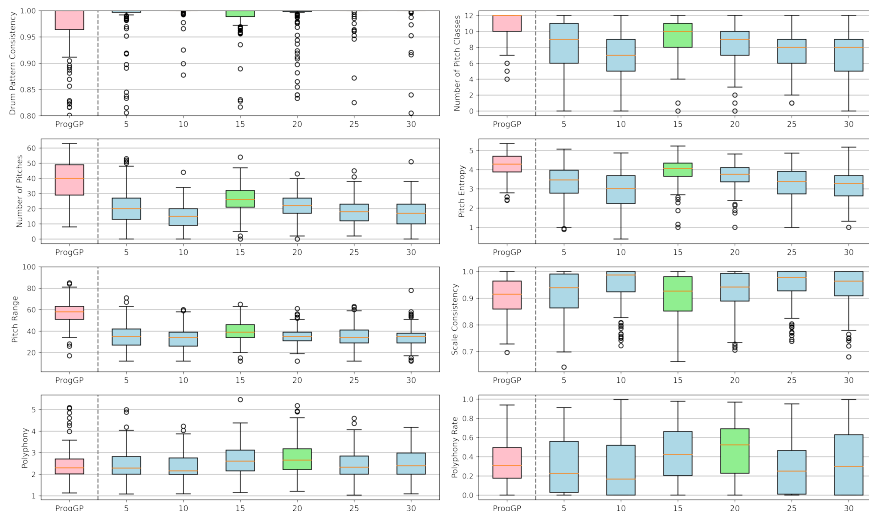


Fig. 3: Metrics calculated for the songs in ProgGP (pink) and the generated songs for each fine-tuned model (blue and green). Model with lowest KLD highlighted (green).

The Kullback-Leibler divergence (KLD), a measure of relative entropy between the true probability distribution and a sample probability distribution, was calculated for each of the fine-tuned models (ProgGP is used as groundtruth to compared against generated songs). The KLD results can be seen in Table 1.

Table 1: KLD scores for each fine-tuned model against ProgGP. Bold and green coloring indicates the lowest KLD per column.

Epoch	PCE	DPC	NPC	NP	PE	PR	Pol	PolR	SC
5	0.473	0.513	0.608	0.799	0.638	0.762	0.497	0.495	0.263
10	0.665	0.696	1.599	1.052	0.800	0.845	0.570	0.573	0.433
15	<b>0.262</b>	<b>0.442</b>	<b>0.491</b>	<b>0.746</b>	<b>0.442</b>	<b>0.591</b>	0.365	0.353	<b>0.216</b>
20	0.425	0.478	0.999	0.914	0.616	1.062	<b>0.301</b>	<b>0.247</b>	0.286
25	0.673	0.596	1.641	0.998	0.670	0.912	0.484	0.559	0.491
30	0.707	0.640	1.200	1.043	0.851	1.054	0.400	0.509	0.312
35	0.625	0.625	1.144	0.939	0.743	0.974	0.376	0.493	0.376
40	0.480	0.611	1.050	0.970	0.717	1.121	0.513	0.544	0.274
45	0.702	0.746	1.554	1.059	0.910	1.089	0.420	0.486	0.336
55	0.648	0.679	1.510	1.040	0.813	1.092	0.517	0.504	0.317
55	0.595	0.690	1.358	1.039	0.818	1.092	0.471	0.485	0.346
60	0.681	0.677	1.513	1.018	0.816	1.157	0.579	0.575	0.375
65	0.757	0.730	2.069	1.126	0.842	1.041	0.394	0.484	0.379

The model fine-tuned for 15 epochs scores the lowest for most metrics. The only exceptions are polyphony and polyphony rate, in which the model fine-tuned for 20



epochs scores the lowest. This is expected given that the model trained for 15 epochs seems to be more similar to ProgGP for most of the metrics than the other models.

## 4.2 Subjective Analysis

Subjectively evaluating generated progressive metal songs first requires a definition of progressive metal. This definition is hard to specify, as music genres are not always straightforward. Nevertheless, there are a number of tropes that progressive metal songs tend to have. Robinson [15] describes several of these such as polyrhythms, syncopated chugging on low notes and uncommon time signatures. These can be seen in many generated songs, particularly uncommon time signatures and syncopated rhythms. Similarly to the conclusions from GTR-CTRL [18], we empirically found that the prompt has a reasonably large amount of influence over the generated song, but this varies between songs. The model tends to only generate notes for instruments contained in the prompt (e.g. if there exists two guitars, one bass guitar and drums within the prompt, the model will only generate new notes for those instruments). It does however occasionally generate an extra guitar or keyboard track ([id-00](#))<sup>8</sup>, but these scenarios were found to be rare. Generated guitar parts for multiple guitar tracks tend to be mostly identical, mirroring the recording technique of two guitars playing identical parts in order to create width in a song mix. Interestingly however, the model will sometimes generate a harmony for a particular guitar line where one guitar plays some kind of melodic line and the other playing the same line with the pitch shifted ([id-01](#)). It also occasionally generates guitar solos and rhythmic accompaniment ([id-002](#)), with one guitar playing low-pitched chords while the other plays fast single high-pitched notes. The model generates very impressive drum parts in addition to the guitar and bass guitar ([id-03](#)). The timing of the kick drum consistently lines up with the notes of the bass guitar ([id-04](#)). Additionally, several common drum beats heard in many metal songs can be generated (e.g. blast beats ([id-05](#))). Many songs also feature drum fills at the end of a section before transitioning into a new section. It is possible that the model excels at generating drum parts due to the limited number of possible notes compared to pitch-based instruments such as guitar and bass guitar. This being said, the generated drum parts would likely need further editing if used in an actual song in order to convey more of the nuance heard in progressive metal drumming.

## 5 Song Production

A short progressive metal song was recorded, produced and mixed using one of the fine-tuned models to generate the initial musical ideas and song structure. This was done by the first author, himself a progressive metal producer and music AI researcher. The intention with this production was to utilize the generated songs as a way to bolster creativity and inspire ideas for music in a way in which the artist's creativity can still be applied to integrate the generated content into a song of their own. Section 5.1 describes a high level overview of the song creation process using the AI system in collaboration

<sup>8</sup> Song ids are hyperlinked to facilitate listening.

with a music producer, while Section 5.2 presents a detailed analysis of the generated song and what was changed in order to suit the production.

### 5.1 Process

The process of creating the song can be broken into the following steps:

1. A prompt is selected and songs are generated using one a fine-tuned model. One is chosen to be the starting point of the song based on how it inspires the producer.
2. The generated song is loaded into a guitar tab reader software (e.g. GuitarPro).
3. Drums and bass are exported to MIDI format and loaded into a digital audio workstation (DAW), along with appropriate virtual instruments.
4. The guitar parts are learned by the guitarist producer from the generated guitar tab and subsequently recording in the DAW. During the recording of the guitar, changes can be made to suit the producer's idea of the direction of the song.
5. The drum and bass guitar MIDI are edited to suit any changes made to the guitar, or to better serve the song. This may be done in conjunction with the previous step and may require some back and forth in order to fully develop the song.

These steps can be repeated as many times as desired to build out a complete song. They may even be skipped if the producer is inspired by the ideas to create their own parts based on what was already generated. In the next section we focus on a particular example generated using the first two measures of "Stabwound" by Necrophagist as the prompt. The song was generated using the model fine-tuned on ProgGP for 15 epochs. The structure of the generated song was not changed, as we felt that it had many interesting qualities. The guitar, drums and bass were changed slightly to better fit the vision that the generated song inspired. Additional sounds such as synths, organs and impact samples were also added to flesh out the song and increase interest in the production. The final mix and the original generated song in both PDF and GuitarPro format are available online<sup>9</sup>.

### 5.2 Song and Production Analysis

The first section of the song is made up of an idea which takes up 4 measures. This idea is repeated with the second repetition skipping the first measure of the motif and adding on a new lick in the final measure which helps transition the section into the next one. Each repetition has a similar structure: three measures of 4/4 and a final measure with an odd time signature. The first repetition adds a 5/4 time signature to the end, while the second section uses a 6/4 time signature. Time signature changes are common in progressive metal [15], and it is interesting to see the model generate this time signature change in both repetitions of the initial idea without simply repeating the idea. The changes in the second repetition of the idea feel like something a real songwriter might intentionally write, as if the model is building on the initial idea to create more excitement before the next section. The second section shows off a major flaw of the model:

<sup>9</sup> Available at: <https://drive.google.com/drive/folders/1y2xX3WlQeOz6Z8F0N2VP3kzWvOqYk8QI?usp=sharing>

it does not always generate tabs or ideas that can be reasonably played by a human. Since a specific pitch can be played at multiple different areas of the guitar fretboard, tabs specify exactly which fret and string a note should be played on. However, the model will sometimes generate fretboard locations that are very unnatural to play by a guitarist. The tabs had to be slightly modified in order to record this section, however keeping the same notes. The main idea in this section is a repeated line of seven 8th notes followed by a chromatic note run and a lick that changes the modality from major to minor halfway through. It is difficult to know if this is something the model learned through training or if this note selection was more random. The section ends with four simple chords to transition into the next one. These were changed to be more dissonant chords in the recorded version. The final section is another repeated riff of seven notes used in a slightly more musical way than the previous section. Each repetition uses the same relative intervals between notes to outline two different chords, F# minor and G# minor. It then ends the section with two measures of 4/4, helping the song end in a slightly more familiar and natural way. A lick from the previous section is used in this ending in the tab, which helps tying the two sections together and increases cohesion.

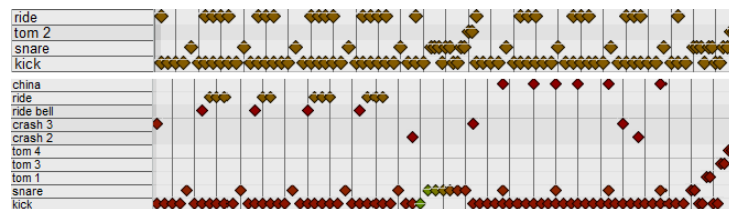


Fig. 4: Original generated drum MIDI (top) vs. the final edited drum MIDI (bottom).

While the structures and guitar riffs remained largely unchanged, the drums did not support the rest of the song as well as they could have. While many generated songs have impressive sounding drums, the drum parts generated in this particular song did not quite hold up to professional standards. The first section mostly had a snare fill which did not enhance the interesting aspects of the guitar and bass parts. This was changed to use a more steady snare hit and cymbals on the downbeats of the measure. A stack cymbal was used in the first repetition, but was changed to a china cymbal in the second repetition to add excitement to the changes between the two repetitions. A drum fill was also added in during the last few beats of the section to help highlight the transition between the two sections. The drums for the second section were mostly the same as the generated drums. The generated snare drum placement in this section accents the 7/4 time signature. However, the ride cymbals in the second repetition were changed to china cymbals which hit on the downbeats of the measure, and the kick drum was changed to be constant eighth notes. This was done to push the energy up as the section finishes. The drums in the final section were kept mostly unchanged, with a small change to the drum fill at the end. A comparison from a section of the song of the originally generated MIDI and the edited MIDI can be seen in Figure 4. The process showed that while the model can excel at generating inspiring progressive

metal ideas, a decent amount of work is still needed to make the ideas playable and professional sounding. Drums in particular, while containing good initial ideas, need a lot of editing to make them sound natural and support the ideas in the guitar and bass guitar parts. It is not as simple as directly importing the drum and bass MIDI from the generated song, a human producer is still required to make the ideas into something that is satisfying to listen to and convey emotion properly. That being said, the entire writing and production process only took three to four hours over two sessions, with most of the time being spent practicing the guitar parts in order to play them to a sufficient level for recording. The producer felt that the AI system helps inspiring new ideas and producing a good sounding demo extremely quickly, with an amazing level of detail in both the kinds of notes generated and song structure. It is easy to imagine combining multiple generated ideas together in this way to produce a full length song.

## 6 Conclusion and Future Work

We have presented a deep learning model capable of generating songs in the style of progressive metal. We released ProgGP, a symbolic music dataset consisting of 173 progressive metal songs, which was constructed and used to fine-tune a pretrained transformer model. The models fine-tuned for only a relatively small number of epochs, such as 15 and 20 epochs, produce interesting results and are shown to exemplify traits of the fine-tuning data in nine different symbolic music metrics. This analysis was used to inform the selection of a generated song, which was then turned into a full progressive metal production. Finally, we presented an analysis of the generated song and how it was used to augment the producer's own creativity. This work could be further improved through extending the dataset with additional high quality tabs in the genre, as well as a DAW integration to streamline the process of generating tabs and editing them into a song in a DAW. Additionally, the effects of prompting the model could be further explored, particularly with prompts of different genres both within and outside of metal. We hope to continue this collaboration between human musicians and the AI system in a possible professionally recorded album and live performance of AI-assisted progressive metal songs.

## References

1. Adkins, S., Sarmiento, P., Barthet, M.: LooperGP: A Loopable Sequence Model for Live Coding Performance using GuitarPro Tablature. In: Proc. of the EvoMUSART Conf. (2023)
2. Amershi, S., Weld, D., Vorvoreanu, M., Fournay, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., et al.: Guidelines for human-ai interaction. In: Proc. of the 2019 chi conference on human factors in computing systems. pp. 1–13 (2019)
3. Benford, S., Greenhalgh, C., Crabtree, A., Flinham, M., Walker, B., Marshall, J., Koleva, B., Rennick Egglestone, S., Giannachi, G., Adams, M., et al.: Performance-led research in the wild. *ACM Transactions on Computer-Human Interaction (TOCHI)* **20**(3), 1–22 (2013)
4. Candy, L.: Practice based research: A guide. *CCS report* **1**(2), 1–19 (2006)
5. Chen, Y.H., Huang, Y.H., Hsiao, W.Y., Yang, Y.H.: Automatic Composition of Guitar Tabs by Transformers and Groove Modelling. In: Proc. of the 21st Int. Soc. for Music Information Retrieval Conf. pp. 756–763 (2020)

6. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. In: Proc. of the 57th Annual Meeting of the Ass. for Computational Linguistics. pp. 2978–2989. Florence, Italy (2019)
7. Dong, H.W., Chen, K., McAuley, J., Berg-Kirkpatrick, T.: MusPY: A Toolkit for Symbolic Music Generation. In: Proc. of the 21st Int. Soc. for Music Information Retrieval (2020)
8. Dong, H.W., Yang, Y.H.: Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation. In: Proc. of the 19th Int. Soc. for Music Information Retrieval Conf. pp. 190–198. Paris, France (2018)
9. Hannan, C.: Hearing form in progressive metal: Motivic return, genre borrowing, and Sonata form in *Between the Buried and Me's Parallax II*. Ph.D. thesis, Columbia University New York (2019)
10. Huang, C.Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D.: Music Transformer: Generating Music with Long-term Structure. In: Proc. of the 7th Int. Conf. on Learning Representations (2019)
11. Huang, Y.S., Yang, Y.H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proc. of the 28th ACM Int. Conf. on Multimedia. pp. 1180–1188. Seattle, WA, USA (2020)
12. Martelloni, A., McPherson, A., Barthelet, M.: Guitar augmentation for percussive fingerstyle: Combining self-reflexive practice and user-centred design. In: Proc. of the Int. Conf. on New Interfaces for Musical Expression (2021)
13. Meade, N., Barreyre, N., Lowe, S.C., Oore, S.: Exploring Conditioning for Generative Music Systems with Human-Interpretable Controls. Tech. rep. (2019)
14. Payne, C.: Musenet (2019), <https://openai.com/blog/musenet>, Last accessed: 12 Jun 2022
15. Robinson, D.: An Exploration of the Various Compositional Approaches to Modern Progressive Metal. Ph.D. thesis, University of Huddersfield (2019)
16. Sarmiento, P.: Perspectives on the Future for Sonic Writers. *Journal of Science and Technology of the Arts* **13**(1), 110–114 (2021)
17. Sarmiento, P., Kumar, A., Carr, C., Zukowski, Z., Barthelet, M., Yang, Y.H.: DadaGP: a Dataset of Tokenized GuitarPro Songs for Sequence Models. In: Proc. of the 22nd Int. Soc. for Music Information Retrieval Conf. pp. 610–618 (2021)
18. Sarmiento, P., Kumar, A., Chen, Y.H., Carr, C., Zukowski, Z., Barthelet, M.: GTR-CTRL: Instrument and genre conditioning for guitar-focused music generation with transformers. In: Proc. of the EvoMUSART Conf. pp. 260–275. Springer (2023)
19. Sturm, B.L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., Hadjeres, G., Deruty, E., Pachet, F.: Machine learning research that matters for music creation: A case study. *Journal of New Music Research* **48**(1), 36–55 (2019)
20. Sturm, B.L., Santos, J.F., Ben-Tal, O., Korshunova, I.: Music transcription modelling and composition using deep learning. In: Proc. on the 1st Conf. on Computer Simulation of Musical Creativity (2016)
21. Tan, H.H., Herremans, D.: Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling. In: Proc. of the 21st Int. Soc. for Music Information Retrieval Conf. pp. 109–116. Montréal, Canada (2020)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention Is All You Need. In: Proc. of the 31st Conf. on Neural Information Processing Systems. Long Beach, CA, USA (2017)
23. Yang, Q., Steinfeld, A., Rosé, C., Zimmerman, J.: Re-examining whether, why, and how human-ai interaction is uniquely difficult to design. In: Proc. of the 2020 CHI Conf. on Human Factors in Computing Systems. pp. 1–13 (2020)

# Reconstructing Human Expressiveness in Piano Performances with a Transformer Network

Jingjing Tang<sup>1</sup>, Geraint Wiggins<sup>1,2</sup>, and György Fazekas<sup>1</sup> \*

<sup>1</sup> Center for Digital Music, Queen Mary University of London

<sup>2</sup> Vrije Universiteit Brussel

jingjing.tang@qmul.ac.uk

**Abstract.** Capturing intricate and subtle variations in human expressiveness in music performance using computational approaches is challenging. In this paper, we propose a novel approach for reconstructing human expressiveness in piano performance with a multi-layer bi-directional Transformer encoder. To address the needs for large amounts of accurately captured and score-aligned performance data in training neural networks, we use transcribed scores obtained from an existing transcription model to train our model. We integrate pianist identities to control the sampling process and explore the ability of our system to model variations in expressiveness for different pianists. The system is evaluated through statistical analysis of generated expressive performances and a listening test. Overall, the results suggest that our method achieves state-of-the-art in generating human-like piano performances from transcribed scores, while fully and consistently reconstructing human expressiveness poses further challenges. Our codes are released at <https://github.com/BetsyTang/RHEPP-Transformer>.

**Keywords:** music generation, expressive music performance, transformer model

## 1 Introduction

An expressive music performance goes beyond playing the notes in the score correctly. Following annotations in music sheets, performers interpret the music with different degrees of expressive control including articulation and dynamics to express emotions and provide an individual rendition of the music, resulting in different performance styles [6]. A common way of rendering expressive performances with computational models is to meaningfully tune the velocity and timing of notes in the score to reconstruct

---

\* This work is supported by the UKRI Centre for Doctoral Training in Artificial Intelligence and Music. J.Tang is a research student supported jointly by the China Scholarship Council and Queen Mary University of London. G. Wiggins received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen”. We would like to thank Lele Liu, Jiawen Huang and the reviewers for their valuable feedback to improve our work.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

human expressiveness [2]. Generally modelling human expressiveness requires capturing the differences between scores and human performances in expressive features including tempo, timing, dynamics, and so on. Learning the subtle nuances in expression among individual pianists demands the model to learn much smaller perceivable differences within those expressive features.

In recent years, deep learning (DL) models have shown promising results in music generation and representation learning. In particular, the Transformer architecture has gained popularity due to its ability to capture long-range dependencies and contextual information in sequential data. This capability positions the Transformer as a potential solution for modeling performance actions such as adjusting tempo and loudness, and capturing a performer's structural interpretation of music. However, while many studies have successfully applied Transformer architecture to algorithmic music composition [4, 9, 10, 11] and representation learning for symbolic music [5, 24], few works pay attention to modeling human performance expressiveness independently. In the field of expressive performance rendering (EPR), recent studies have achieved convincing results for the purpose of reconstructing general human expressiveness and controlling style using DL architectures including Recurrent Neural Network [12], Graph Neural Network [13] and conditional Variational Autoencoder [21]. These models require large-scale accurate alignments of well-annotated music scores and performances. However, due to the limited quality and size of the currently available datasets, including the Vienna 4x22 Piano Corpus [8] and ASAP [7], these systems still have difficulty dealing with playing techniques such as pedalling and trills, recovering expressiveness overarching longer passages of music, as well as modeling the performance style of individual players.

In this paper, we propose a novel approach for reconstructing human expressiveness with a multi-layer bi-directional Transformer encoder. Training a Transformer model for this task demands large amounts of accurately recorded and score-aligned performance data, which is not currently readily available. A recently released performance-to-score transcription system [15] and the transcribed expressive piano performance dataset ATEPP [25] allow us to use transcribed scores and performances to train our model. Using transcribed scores in the EPR task can be beneficial when the canonical score is not representative enough. For example, jazz performances rely heavily on improvisation, making it difficult to align canonical scores with performances. Even in classical music, ornaments such as trills may not be explicitly notated in canonical scores, which poses problems for the alignment process. Moreover, the reconstruction of human expressiveness from transcribed scores can support research in musical style transfer, particularly when people aim to change a performance by one pianist into the style of another. Considering this, we investigate the ability of our system to model the expressiveness for individual pianists and evaluate it through statistical analysis of the generated performances and a listening test comparing our model to state-of-the-art expressive performance rendering systems.

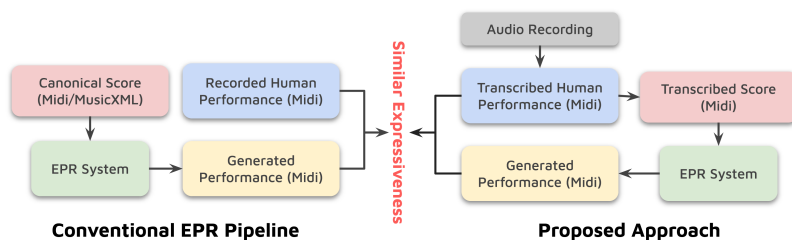
The rest of this paper is organized as follows: Section 2 describes the methodology detailing the dataset, the process of feature extraction and the model architecture. Section 3 introduces the experiment setting-ups for training our model. Section 4 presents

the results of quantitative analysis and the listening test as well as discussions upon the results, and finally, Section 5 concludes the paper.

## 2 Methodology

### 2.1 Problem Definition

Expressive performance rendering (EPR) is commonly defined as *the task of generating human-like performances with music sheets as input*. Most existing work [12, 13, 21] proposes systems using recorded performances and canonical scores to solve the problem. All of these systems require alignment between the canonical scores and performances, which is limited in accuracy given the available datasets and alignment algorithms. With the purpose of reconstructing human expressiveness given a composition, we reformulate the task by relaxing the requirement for using conventional music sheets as input, in order to take advantage of the recent performance-to-score transcription algorithms [15] and large transcribed performance datasets [25]. We will provide more details about the transcription algorithm and the dataset used in this work in Sections 2.2 and 2.3. As shown in Fig. 1, the EPR task, in our definition, is to take the transcribed scores as input and reconstruct human expressiveness by generating expressive performances that are similar to the transcribed human performances.



**Fig. 1.** Comparison of the conventional expressive performance rendering (EPR) pipeline with our proposed method

### 2.2 Dataset

The recently released ATEPP dataset [25] provides high-quality transcribed piano performances by world-renowned pianists. According to a listening test conducted by Zhang et al., the transcribed performance MIDIs reliably retain the expressiveness of performers. The dataset includes multiple performances of the same composition by different pianists, allowing comparison in expressiveness among different performers. However, since the ATEPP dataset has a highly skewed distribution of performers, rather than using the whole dataset, we use a subset [19] that balances the number of performances by six pianists: Alfred Brendel, Claudio Arrau, Daniel Barenboim, Friedrich Gulda, Sviatoslav Richter, and Wilhelm Kempff. Compositions in this subset are mainly composed by Beethoven with only two pieces by Mozart. Each of the compositions corresponds to at least one performance by each pianist. Table 1 presents statistics of the subset in comparison with datasets used by other EPR systems.



**Table 1.** Comparison of datasets used in different EPR systems. \*NN stands for the number of notes. † denotes that the information not provided.

Systems	Performances	Pianists	Compositions	Composer(s)	Total NN*
VirtuosoNet [12]	1052	/†	226	16	3301K
Sketching-Internal [21]	356	/	34	1	/
Sketching-External [21]	116	/	23	10	/
<b>Ours</b>	457	6	36	2	1341K

### 2.3 Data Processing

**Score Transcription** Similarly to other EPR systems [12, 13, 21], our method requires note-to-note alignment between the input score MIDI and the output performance MIDI. Despite the convincing alignment results of the state-of-the-art algorithm proposed by Nakamura et al. [18], the algorithm shows difficulty in dealing with repeated sections as well as trills in classical piano music, which causes unexpected loss of information during the alignment process. Instead of using the original or manually edited scores of the compositions, we obtained the transcribed scores of the performances through a performance-to-score transcription algorithm proposed by Liu et al. [15]. The transcribed score midi data can be aligned with the performances at the note level without losing any structural generality in the music [23].

The transcription algorithm performs rhythm quantisation through a convolutional-recurrent neural network and a beat tracking algorithm to remove expressive variations in timing, velocity, and pedalling. While expressiveness regarding velocity and pedalling is certainly erased through the process, how much expressiveness is remained in timing is implicit and will be discussed further in Section 4. A further constraint of this algorithm is its inability to retrieve performance directives like dynamics, phrase markings, and beam directions set by the composer. As a result, we were limited to leveraging only the note-related features the algorithm offered.

**Data Augmentation** The transcribed scores are first scaled to the same length as the corresponding performances. We then augment the data by changing the tempo for both performances and the scores. For each pair of performance and score midis, the onset time, offset time and duration of each note are multiplied by a ratio  $r_i \in [0.75, 1.25]$ . In total, we have each pair augmented by multiplying 10 different ratios that are evenly spaced along the interval grid.

**Table 2.** Vocabulary size of the tokenized note-level features

Features	Pitch	Velocity	Duration	Position	Bar
<b>Size</b>	89	66	4609	1537	518

**Feature Encoding** Features related to performance expressiveness are extracted and tokenized to reduce the the dimensionality of the input space. Following the tokenisation method, OctupleMIDI, proposed by Zeng et al. [24], we encode the note-level

features including pitch, velocity, duration, bar, and position. Table 2 shows the vocabulary size of our tokens for each feature. When using OctupleMIDI, the onset time of a note  $N_i$  is represented jointly by its bar number  $B_i$  and position number  $P_i$ , where  $i = 1, 2, \dots, n$  and  $n$  denotes the length of the note sequences. Given that we use a piano music dataset, we consider only pitches with numbers ranging from 21 to 109. The duration of notes is set to be linearly proportional to the token value  $D_i$ . All of the midi files have a resolution of 384 ticks per beat, and we default each bar to have 4 beats, resulting in  $384 \times 4 = 1536$  different positions per bar. We calculate values of other two note-level performance features which are commonly used for capturing the expressiveness of piano performances [12, 20, 21] based on the tokens:

- *Inter-Onset Interval (IOI)*: the time interval between the onset time (OT) of the note  $N_i$  and that of the next note  $N_{i+1}$ :

$$IOI_i = \begin{cases} OT_{i+1} - OT_i, & i = 1, 2, \dots, n-1 \\ 0, & i = n \end{cases} \quad (1)$$

where  $OT_i = B_i \times 1536 + P_i$ ,  $i = 1, 2, \dots, n$

- *Duration Deviation (DD)*: the difference between duration token values of a note in performance midi and score midi

$$DD_i = Dp_i - Ds_i, i = 1, 2, \dots, n \quad (2)$$

where  $Dp$  is the duration obtained from the performance midi and  $Ds$  is that from the score midi.

## 2.4 Generation with Transformer Encoder

**Input and Output Features** Input and output features are carefully designed to preserve the score content while allowing changes in the performance control of each note. The input features include pitch, velocity, duration, bar, position, and inter-onset interval from the score midis. As for the output, we infer values of three features including velocity, DD, and IOI in the performance midis. Following Eq. 1 and Eq. 2, we can calculate the predicted token values of duration, position, and bar for each note based on DD and IOI. Combined with the predicted token values for velocity, we can construct a performance MIDI file through detokenization.

**Model Architecture** Inspired by the MidiBert model proposed by Chou et al. [5], we design a multi-layer bi-directional Transformer encoder with 4 layers of multi-head self-attention where each has 4 heads and a hidden space dimension of 128. The pianist’s identity is represented using a one-hot encoding embedding, which is then concatenated to the last hidden state before the final prediction, as shown in Fig. 2. As velocity and timing in music are continuous variables, the interval between two token values is informative in representing the distinction of playing a note. Most transformers trained for music generation [9, 4, 11, 10] take different token values as independent

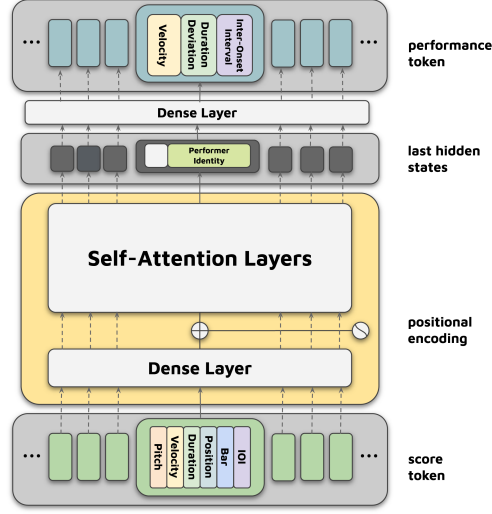


Fig. 2. Model architecture of the Transformer encoder

classes which makes this information implicit to the model. Our system instead uses the tokens without creating embeddings, and predicts the token values for different features through regression. In addition, we add activation functions after the inference layer to clamp the predicted values, ensuring that they fall into the ranges of different features.

**Loss Design** The losses  $\mathcal{L}_v$ ,  $\mathcal{L}_{dd}$ , and  $\mathcal{L}_{ioi}$  for velocity, DD, and IOI features are calculated respectively, following the loss function defined in Eq. 3 which represents the percentage of how much the predicted values  $y$  deviated from the target values  $\hat{y}$ . Masks are created to exclude loss calculation for padded tokens.

$$\mathcal{L}_{feature} = \sum_{i=0}^n l(y_i)m_i, \quad (3)$$

where  $m_i$  represents the loss mask for the  $i$ -th note and

$$l(y_i) = \begin{cases} \frac{|y_i - \hat{y}_i|}{|\hat{y}_i|}, & \text{if } \hat{y}_i \neq 0 \\ \alpha|y - \hat{y}_i|, & \text{if } \hat{y}_i = 0 \end{cases}$$

The parameter  $\alpha$  regularizes the loss calculation when the target value is zero and is experimentally set to 0.001. The total loss is calculated by

$$\mathcal{L}_{total} = w_v\mathcal{L}_v + w_{dd}\mathcal{L}_{dd} + w_{ioi}\mathcal{L}_{ioi} \quad (4)$$

where weights are empirically initialized and assigned to each feature loss respectively.

## 2.5 Evaluation

The system is objectively evaluated through validation losses and statistical distributions of expressive parameters in generations, presented in Section 4.1. Additionally, we evaluate the perceived expressiveness of generated performances through a subjective listening test. As the aim of EPR task is to generate performances with human-like expressiveness [2], we assume that the more similar a model's output is to a human performance, the more effectively expressive it is. We recruit participants who have experience in playing musical instruments and who are engaged with classical music, and ask them to rate the presented samples by evaluating how expressive, natural, and human-like they are. The detailed experiment design and conditions and the results of the listening test are presented in Section 4.2.

## 3 Experimental Setup

We implement our model based on the PyTorch. We have a 8:1:1 data split in the number of piece and performance, and we cut or pad the token sequences into sequences of 1000 notes before inputting into our transformer. The model is trained with a batch size of 16 sequences for at most 400 epochs, using the Adam optimizer with an initial learning rate of  $1e-4$  and a weight decay rate of  $1e-7$ . We update the learning rate using the cosine annealing warm restart scheduler [17] since it has been shown to result in faster convergence during training, compared with other learning rate scheduling strategies. If the validation loss does not improve for 30 consecutive epochs, we stop the training process early. The training converges in 2 days on two RTX A5000 GPUs.

Different vocabulary sizes of expressive features shown in Table 2 result in different degrees of complexity when modeling. Consequently, we observed unbalanced decrease in losses and overfitting across learning for different features with constant weights assigned to each feature loss. To balance training and reduce overfitting, we optimize the training process using the GradNorm algorithm proposed by Zhao et al. [3] to dynamically update weights based on gradients calculated at the end of each training epoch.

## 4 Results

### 4.1 Quantitative Evaluation

Quantitative methods for evaluating expressive performance rendering systems are limited. One approach [2] is to calculate the loss for each performance feature. Unlike existing approaches [13, 12, 21] where the features are not tokenised, our system computes the losses using the token values. Based on the feature encoding process and the loss design discussed in Section 2, we estimate the average prediction errors in MIDI quantised velocity value and seconds, shown in Table 3.

Although the results are not directly comparable to existing works because of the differences in feature extraction and loss design, they indicate that the transformer model could learn the patterns of expressive variations and reproduce them in the transcribed scores. However, the average errors at the note level in generations are still

**Table 3.** Loss and average prediction error in MIDI velocity value and seconds for note-level expressive features on the test dataset

Features	Loss	Average Error
<b>Velocity</b>	0.1267	$\pm 16.2048$
<b>Duration Deviation</b>	0.6280	$\pm 0.0473s$
<b>Inter-Onset Interval</b>	0.2389	$\pm 0.0183s$

noticeable to human ears [16], and can affect the perceived expressiveness of the generated music in comparison to human performances.

Since the level of expressiveness regarding timing left in the transcribed scores is implicit as discussed in Section 2, we evaluate the ability of our system to reconstruct the expressiveness for individual pianists through the velocity distributions obtained from kernel density estimation [20, 26].

**Fig. 3.** Velocity distributions for the human performances (P) and the our generations (G-TS) on all pieces in the test set, grouped by different pianists.

As shown in Fig. 3, velocity distributions for each pianists are distinguishable, indicating different performing styles. However, performance recording environments may have impact on the transcribed velocity values [14] and contribute to differences of the distributions. The distributions of the generations based on transcribed scores (G-TS) and those of the human performances (P) have a high degree of overlap, providing evidence of learning individual expressiveness through the training.

## 4.2 Subjective Evaluation

A listening test was performed to evaluate the perceived expressiveness of our model’s output. We recruited 19 people who had some level of music training through email. All participants have learned a musical instrument, while over half of our participants had been engaged with classical music for over 5 years. The participants completed the study anonymously.

The stimuli consisted of four 20s classical piano excerpts detailed in Table 4. For each excerpt, the human performance (P) was provided as a reference to be compared with four MIDI renderings: the generation based on the transcribed score (G-TS), the

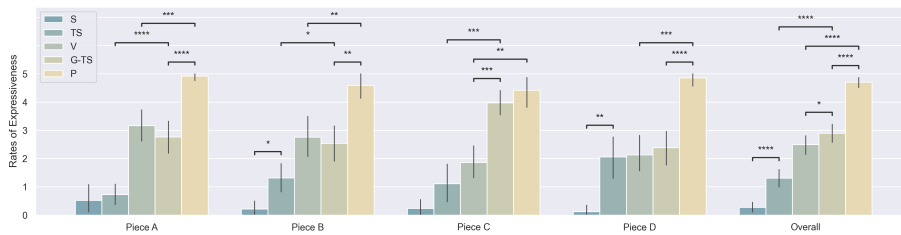
generation by the state-of-the-art VirtuosoNet [12] using the canonical score (**V**), a direct rendering of the transcribed score (**TS**), and finally the canonical score (**S**) without expression. The human performances were transcribed piano performance MIDI files from the ATEPP dataset [25] and were included as one of the stimuli as well. All the MIDI files were synthesised into audio recordings through GarageBand to ensure consistency in the listening experience. For each piano excerpt, six recordings, the reference plus 5 stimuli, were presented in the test<sup>3</sup>.

Participants were asked to listen to five stimuli, and rate the degree of expressiveness for them on a 100-point scale by comparing each of them with the reference human performance. During the test, we explicitly ask participants to rate based on the expressive differences among the stimulus with more focus on the performance features such as the dynamics and tempo changes rather than the compositional content. We encouraged them to use the full scale, rating the best sample higher than 80 and the worst lower than 20. We adopt the MUSHRA framework [22] to conduct the test using the Go Listen platform [1].

**Table 4.** Compositions used for the listening test

Annotation	Composer	Composition
Piece A	Beethoven	<i>Piano Sonata No. 19 in G Minor, Op. 49 No. 1: II. Rondo (Allegro)</i>
Piece B	Beethoven	<i>Piano Sonata No. 7 in D Major, Op. 10 No. 3: III. Menuetto (Allegro)</i>
Piece C	Haydn	<i>Piano Sonata in C Major, Hob. XVI:48: II. Rondo (Presto)</i>
Piece D	Bach	<i>French Suite No. 5 in G, BWV 816: 7. Gigue</i>

In total, 380 ratings from the 19 listeners were collected. We filtered out raters who could not identify the difference in expressiveness between the anchor (**S**) and the reference (**P**). Fig 4 shows the mean opinion scores (MOS) and the results of Wilcoxon signed rank test for the differences between: (a) **TS** versus **S**, (b) **G-TS** versus **V**, (c) **P** versus **G-TS**, (d) **P** versus **V**, (e) **G-TS** versus **TS**.



**Fig. 4.** Results of listening test. The mean opinion scores (converted to a 5-point scale) and 95% confidence intervals are presented for each test piece and the overall results. Wilcoxon signed-rank test are performed to test the significance of the differences. \* ( $0.01 < p < 0.05$ ), \*\* ( $0.001 < p < 0.01$ ), \*\*\* ( $0.0001 < p < 0.001$ ), \*\*\*\* ( $p < 0.0001$ )

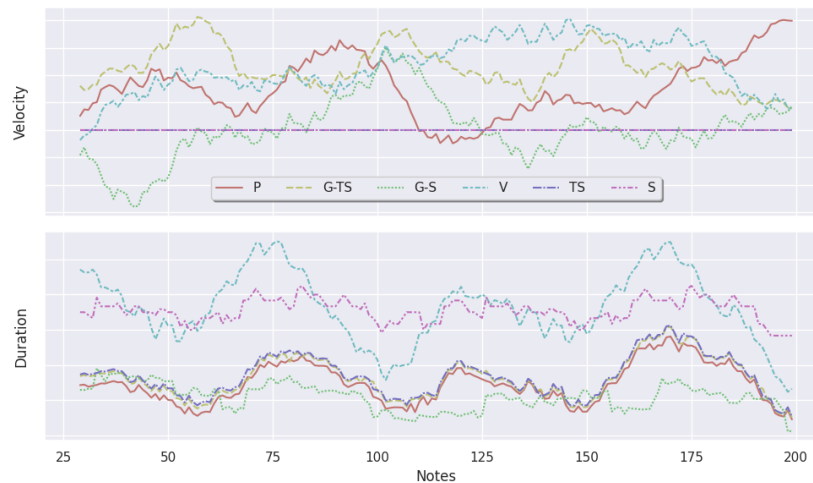
<sup>3</sup> Listening samples are provided at <https://drive.google.com/drive/folders/1nfaZ23vr8xZHlyhTAApK2hl-aHQpIqP?usp=sharing>

According to the results, human performances (**P**) are significantly different from generations of our model (**G-TS**) and VirtuosoNet (**V**) in most situations. The outputs of our model (**G-TS**) are overall preferred over the performances produced by VirtuosoNet (**V**) significantly ( $0.01 < p < 0.05$ ), receiving trivially lower (not significant) ratings for piece **A** and **B** but higher (significant for **C** and not significant for **D**) ratings for the compositions that never appear in the training dataset. Comparing with canonical scores (**S**), transcribed scores (**TS**) get significantly higher ratings from listeners. Ratings of the generations by our system (**G-TS**) are significantly higher than those of the direct audio rendering of transcribed scores (**TS**) for most pieces except **D**.

These results suggest that our system achieves the state-of-the-art and even outperforms the VirtuosoNet [12] in some cases, although neither of the systems can consistently generate the same level of expressiveness as human performances. On the other hand, while the transcribed scores (**TS**) could have more expressiveness than the canonical scores (**S**), the generations from the transcribed scores (**G-TS**) are perceptually more expressive than the transcribed scores (**TS**) in most cases, indicating the success of reconstructing human expressiveness. The success has also been proven by the overall difference ( $0.01 < p < 0.05$ ) in MOS between our generations (**G-TS**) and generations from the VirtuosoNet (**V**).

### 4.3 Case Study: Comparison in Dynamics and Duration

Building on the promising results of our system in the listening test of Piece **C**, we conducted a more detailed analysis to compare the expressive variations in dynamics and duration among human performances, system-generated performances, and scores. Specifically, in Fig. 5, we present the fluctuations in velocity and duration across the note sequences. Compared with the VirtuosoNet generation (**V**), the generation of our



**Fig. 5.** Standardized and smoothed velocity and duration changes across note sequences from *Piano Sonata in C Major; Hob. XVI:48: II. Rondo (Presto)* for enhanced trend comparison. G-S represents the generation of our system based on the canonical scores.

system (**G-TS**) could capture both short-term and long-term velocity variations better.

Even when inputting the unseen canonical score, the generation of our system (**G-S**) outperforms the other model in terms of reconstructing velocity variations. Meanwhile, the strong similarity between duration changes in the human performance (**P**) and transcribed score (**TS**) suggest that the transcription algorithm [15] alters the timing information of the notes cautiously with only limited modification of the duration. Therefore, the reconstruction of the expressive variations in timing through our system could be restricted. The limitation is also demonstrated by the duration changes of our system's generation based on the canonical score (**G-S**).

## 5 Conclusion

This paper presents a novel method for reconstructing human expressiveness in classical piano performances. Our expressive performance rendering system consists of a Transformer encoder trained on transcribed scores and performances. The quantitative evaluation and listening test show that the proposed method succeeded in generating human-like expressive variations, especially for dynamics. Moreover, our method could be used for modeling the differences in expressiveness among individual pianists.

In future work, we will train our system with a mixture of the canonical scores and transcribed scores to create a more robust system. We will further improve the capacity of our system on modeling individual performance styles possibly through contrastive learning. In addition, we will consider a separate system to model pedalling techniques in performances or try to integrate the pedalling information into the current feature encoding.

## References

1. Barry, D., Zhang, Q., Sun, P.W., and Hines, A.: Go Listen: An End-to-End Online Listening Test Platform. *Journal of Open Research Software* (2021)
2. Cancino-Chacón, C.E., Grachten, M., Goebel, W., and Widmer, G.: Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities* 5, 25 (2018)
3. Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A.: GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In: Dy, J., and Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, pp. 794–803. PMLR (2018)
4. Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., and Engel, J.: Encoding Musical Style with Transformer Autoencoders. In: III, H.D., and Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning*. *Proceedings of Machine Learning Research*, pp. 1899–1908. PMLR (2020)
5. Chou, Y.-H., Chen, I., Chang, C.-J., Ching, J., Yang, Y.-H., *et al.*: MidiBERT-piano: Large-scale pre-training for symbolic music understanding. *arXiv preprint arXiv:2107.05223* (2021)
6. Dai, S., Zhang, Z., and Xia, G.G.: Music Style Transfer: A Position Paper. *arXiv:1803.06841 [cs, eess]* (2018)
7. Foscarin, F., Mcleod, A., Rigaux, P., Jacquemard, F., and Sakai, M.: ASAP: a dataset of aligned scores and performances for piano transcription. In: *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pp. 534–541 (2020)
8. Goebel, W.: Melody lead in piano performance: Expressive device or artifact? *The Journal of the Acoustical Society of America* 110(1), 563–572 (2001)



9. Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C., and Yang, Y.-H.: Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 178–186 (2021)
10. Huang, C.-Z.A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A.M., Hoffman, M.D., Dinculescu, M., and Eck, D.: Music Transformer: Generating Music with Long-Term Structure. In: International Conference on Learning Representations (2018)
11. Huang, Y.-S., and Yang, Y.-H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1180–1188 (2020)
12. Jeong, D., Kwon, T., Kim, Y., Lee, K., and Nam, J.: VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance. In: Proceedings of the 20th International Society for Music Information Retrieval Conference (2019)
13. Jeong, D., Kwon, T., Kim, Y., and Nam, J.: Graph neural network for music score data and modeling expressive piano performance. In: International Conference on Machine Learning, pp. 3060–3070 (2019)
14. Kong, Q., Li, B., Song, X., Wan, Y., and Wang, Y.: High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, 3707–3717 (2021)
15. Liu, L., Kong, Q., Morfi, V., Benetos, E., *et al.*: Performance MIDI-to-score conversion by neural beat tracking. In: Proceedings of the 23rd International Society for Music Information Retrieval Conference (2022)
16. London, J.: *Hearing in time: Psychological aspects of musical meter*. Oxford University Press (2012)
17. Loshchilov, I., and Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. In: International Conference on Learning Representations (2017)
18. Nakamura, E., Yoshii, K., and Katayose, H.: 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 (2017)
19. Rafee, S., Fazekas, G., and Wiggins, G.: HIPI: A Hierarchical Performer Identification Model Based on Symbolic Representation of Music. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing* (2023)
20. Rafee, S., Fazekas, G., and Wiggins, G.: Performer identification from symbolic representation of music using statistical models. In: *Proceedings of the International Computer Music Conference 2021*, pp. 178–184 (2021)
21. Rhyu, S., Kim, S., and Lee, K.: Sketching the Expression: Flexible Rendering of Expressive Piano Performance with Self-Supervised Learning. In: *International Society for Music Information Retrieval Conference*, pp. 178–185 (2022)
22. Series, B.: Method for the subjective assessment of intermediate quality level of audio systems. *International Telecommunication Union Radiocommunication Assembly* (2014)
23. Wiggins, G.A., Miranda, E., Smaill, A., and Harris, M.: A framework for the evaluation of music representation systems. *Computer Music Journal* 17(3), 31–42 (1993)
24. Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., and Liu, T.-Y.: MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 791–800, Online (2021)
25. Zhang, H., Tang, J., Rafee, S.R., Dixon, S., Fazekas, G., and Wiggins, G.A.: ATEPP: A Dataset of Automatically Transcribed Expressive Piano Performance. In: *International Society for Music Information Retrieval Conference*, pp. 446–453 (2022)
26. Zhao, Y., Wang, C., Fazekas, G., Benetos, E., and Sandler, M.: Violinist identification based on vibrato features. In: *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 381–385 (2021)

# Effective Textual Feedback in Musical Performance Education: A Quantitative Analysis Across Oboe, Piano, and Guitar

Rina Kagawa<sup>1</sup>, Nami Iino<sup>2,3</sup>, Hideaki Takeda<sup>2</sup>, and Masaki Matsubara<sup>1</sup> \*

<sup>1</sup> University of Tsukuba

<sup>2</sup> National Institute of Informatics

<sup>3</sup> RIKEN Center for Advanced Intelligence Project

kagawa-r@md.tsukuba.ac.jp

**Abstract.** Despite the importance of feedback in musical performance education, there is a lack of quantitative and cross-instrumental examination on what feedback is effective for students. This study collected recordings of performances by students on three instruments (oboe, piano, and guitar) and gathered written feedback from multiple teachers for each performance. Quantitative analysis revealed that the usefulness of feedback varied significantly among teachers, independent of musical instruments, compared to pieces or students. We then conducted multilevel modeling based on hierarchy among teachers for each instrument, and found that the number of sentences *giving objective information* significantly contributed to the usefulness of feedback. Our findings have high generalizability and can be applicable to face-to-face lessons. The collected recordings and written feedback have been published, and can provide valuable resources for music educators seeking to improve their teaching practices.

**Keywords:** Database, Music Education, Verbal Information

## 1 Introduction

Traditionally, people are taught to play musical instruments face-to-face. However, remote lessons can enable the provision of instruction in remote areas, flexible scheduling, reduced travel, security and cost, and can enhance teachers' and students' creative learning and critical thinking by reducing time and distance between teacher-student or teachers [2, 45]. To increase the value of remote lessons, a set of remote lesson modules

---

\* This study was partially supported by JST-Mirai Program Grant Number JPMJMI19G8, JSPS KAKENHI Grant Number JP19K19347, and Kayamori Foundation of Informational Science Advancement. We would like to thank all the performers and teachers who participated in the data collection of this study. We would also thank to those who helped us with data annotation and evaluation.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

(e.g., the Swing project<sup>4</sup>) and an online platform for music distance learning education and practice (e.g., Intermusic project<sup>5</sup>) has developed.

During the COVID-19 pandemic, the demand for remote lessons increased significantly [1, 19]. In fact, online lessons and asynchronous lessons in which critiques are given to recordings were provided at some music colleges, and students and teachers deeply recognized the benefits and shortcomings of remote lessons [27]. In the post-pandemic era, remote lessons will lead to a new group of students joining the music education environment, which will bring diversity to music education and facilitate its development. We believe that music performance instruction will take place in a variety of learning formats, including online, real-world, and hybrid formats.

In traditional face-to-face lessons, the teachers use non-verbal information, such as singing melodies and making gestures, and verbal information, such as their knowledge of the piece [9, 13, 24]. However, in remote lessons, it is difficult to convey detailed body movements and high-quality sound due to the low resolution of video and audio. Therefore, the quality of verbal feedback must be improved to continue to increase the value of remote lessons. However, what kind of content should be included in verbal feedback for students across musical instruments has not been clarified.

Therefore, this study asks, what kind of verbal information related to musical performance is useful for music students? In order to present generalized findings, this study collects and integratively analyzes musical performance data and verbal feedback data with respect to performances on multiple musical instruments.

We have been conducting this study since 2020 and have already published data for the oboe [26]. In this study, we collected additional data for two more musical instruments and analyzed them in an integrated manner.

The contributions of this paper are as follows.

- Musical performance data and corresponding verbal instruction data have been collected and published for three musical instruments.
- The usefulness of verbal feedback on performance was found to depend on the teacher, not the piece or the student, and this finding was independent of the musical instruments.
- The contents of verbal feedback that significantly affected the usefulness of the feedback independent of the musical instruments were clarified.

## **2 Related Work**

### **2.1 Music Database for Research**

Several datasets have been published as music knowledge resources [32]. They adopt various perspectives, including performance recordings [14, 16], metadata (genre, composer, lyrics, fingering, music analysis, etc. [17, 18, 31, 35, 38]), musical scores [12, 22, 23, 42], other multimodal information [25, 43], emotions [6, 7, 46], and students' interpretations [20, 21, 30, 33]. To the best of our knowledge, none have focused on teaching behavior for musical performance. There are also several datasets for music listen-

<sup>4</sup> <https://aec-music.eu/project/swing-2018-2021-erasmus-strategic-partnership/>

<sup>5</sup> <https://aec-music.eu/project/intermusic-2017-2020-erasmus-strategic-partnership/>

ing events [34, 40], but these are datasets gathered from online music services such as Last.fm<sup>6</sup> and they are not intended for pieces for student to learn performance.

Compared to these databases, this study is novel as it collected a dataset that allows for the evaluation of the relationship between the content of the verbal information in the instructions and the musical performances.

## 2.2 Effects of Teaching Behavior on Musical Performance Education

The effects of teaching behavior on musical performance education have been widely studied within the field of music education. Prior studies compared teacher levels [15], analyzed time allocation [5], compared [44] and categorized [36, 37] verbal and non-verbal information, and examined teacher-student interaction [11]. These studies all depended upon the transcription of speech in interactive instruction. There are also studies on supporting the learning of musical performance by presenting nonverbal information [39, 41]. Unlike these studies, our study focused on verbal feedback, which is more applicable to asynchronous education.

One study compared verbal and non-verbal instruction [8], and another study evaluated and summarized the usefulness of instruction [10]. Both were based on five or fewer performances. In contrast, we have conducted a large-scale experiment to clarify the relationship between verbal information and its usefulness.

## 3 Materials

In our previous studies [28, 29], we collected the performance recordings of the oboe and corresponding textual feedback and published them as CROCUS (CRitique dOCumentS of musical performance) dataset.<sup>7</sup> In this study, we collected similar data for the piano as a keyboard instrument and the guitar as a string instrument and published them.<sup>8</sup> Then, each sentence of textual feedback was annotated to indicate what was described, and the perceived utility of each piece of textual feedback was evaluated.<sup>9</sup>

All procedures have been approved by the ethical review board of University of Tsukuba, Sensoku Gakuen College of Music, and Kunitachi College of Music.

### 3.1 Recording

An overview of materials is presented in Table 1 and 2. We selected the pieces shown in Table 3 considering the balance of difficulty, style, form, and era.

**piano:** We used the recording data collected and published in the previous study [20]

**oboe:** We used the recording data collected and published in the previous study [28, 29]. Each student played all 10 pieces in a less reverberant and less noisy environment at home, about one meter away from the recording device (Roland R-07).

<sup>6</sup> <https://www.last.fm/>

<sup>7</sup> <https://masaki-cb.github.io/crocus/>

<sup>8</sup> **piano:** <https://zenodo.org/record/7753365>, **guitar:** <https://zenodo.org/record/7778923>

<sup>9</sup> For oboe, the procedure of questionnaire survey regarding the usefulness and annotation of the textual feedback was the reprint of [29].

**Table 1.** Overview of Materials

Instrument	Recording			
	$N_{student}$	Level of student	$N_{piece}$	recording
piano	4	professional players	10	home
oboe	9	music college student	10	home
guitar	12	Students who have participated in national or international competitions	7	home

**Table 2.** Overview of Materials

Instrument	Textual feedback		
	$N_{teacher}$	Level of teachers	$N_{textualfeedback}$
piano	24	professional teachers	144
oboe	12	music college teacher	239
guitar	13	professional players or teachers	252

**guitar:** Each student played all seven pieces in a less reverberant and less noisy environment at home, about one meter away from the recording device (Roland R-07).

### 3.2 Textual Feedback for Each Performance Recording

As online lessons have become the norm in music colleges due to COVID-19, a similar lesson plan was adopted in our method.

Each teacher wrote one textual feedback for each performance recording, as if you were giving a daily lesson, and each teacher in total wrote 6 (for piano), 20 (for oboe), and 20 or 19 (for guitar) pieces of textual feedback. A total number of textual feedback for each instrument is shown in Table 2. The performances were selected in a counterbalanced manner with the following constraints: each teacher reviewed two performances for each piece, and each student was reviewed by all the teachers throughout the performances. Audio files of performance recordings were sent to each teacher, along with the following introduction: “Please provide textual feedback for each recording assuming the daily lessons.” They listened to each recording and either wrote or typed their feedback. For oboe, one piece of textual feedback was lost during the collection process.

### 3.3 Questionnaire Survey of the Usefulness of the Textual Feedback

We conducted an online questionnaire survey via a crowdsourcing platform<sup>10</sup> to evaluate the usefulness of the textual feedback. We recruited 400 (100 for piano/ 200 for oboe/ 100 for guitar) people who had musical experience outside of school and asked them to provide their demographic informations and answer the question “Do you think that this feedback is useful for future performances?” using a 11-point Likert scale (10: useful – 0: useless). Each participant responded to 46 (for piano), 50 (for oboe), and

<sup>10</sup> <https://www.lancers.jp>

**Table 3.** List of Pieces

Instrument	ID	Composer	Piece
piano	n01	F. Chopin	“Tristesse”, Op. 10-3
	n02	F. Chopin	“24 Préludes”, Op. 28-7
	n03	J. S. Bach	Invention No. 1 in C major, BWV 772
	n04	J. S. Bach	Invention No. 15 in M minor, BWV 786
	n05	L. v. Beethoven	Sonata No. 8 in A flat major, Op. 13
	n06	L. v. Beethoven	Sonata No. 8 in C minor, Op. 13
	n07	R. Schumann	“Traumerai”, Kinderszenen No.7, Op. 15
	n08	W.A. Mozart	Sonata No.32 in A major, KV. 331
	n09	C. Debussy	La Fille aux Cheveux de Lin
	n10	C. Debussy	Rêverie
oboe	n01	L. v. Beethoven	Symphony No. 3 in E flat major ‘Eroica’, Op. 55
	n02	G. A. Rossini	‘La Scala di seta’ Overture
	n03	F. Schubert	Symphony No. 8 in B minor D.759 ‘Unfinished’
	n04	J. Brahms	Violin Concerto in D major, Op. 77
	n05	P. I. Tchaikovsky	Symphony No. 4 in F minor, Op. 36
	n06	P. I. Tchaikovsky	“Swan Lake”, Ballet Suite, Op.20a
	n07	N. Rimsky-Korsakov	“Scheherazade”, Symphonic Suite, Op. 35
	n08	R. Strauss	“Don Juan”, Symphonic Poem, Op. 20
	n09	M. Ravel	Le Tombeau de Couperin I.Prelude
	n10	S. Prokofiev	“Peter and the Wolf”, Symphonic Tale, Op. 67
guitar	n01	F. Sor	Etude No. 1, Op. 31-1
	n02	F. Sor	Etude No. 5, Op. 35-22
	n03	M. Carcassi	Etude, Op. 60-3
	n04	Anonymous	Romance: Jeux interdits
	n05	F. Tárrega	Lágrima
	n06	L. Walker	Kleine Romanze
	n07	J. S. Sagreras	Maria Luisa

100 (for guitar) randomly selected pieces of textual feedback. Different participants were recruited for each musical instrument.

Hereinafter, in this paper, the average value for each textual feedback will be referred to as its usefulness.

### 3.4 Annotation of Types of Sentences in Textual Feedback

The purpose of this study was to identify which instructional contents that are significantly more useful for performance students as the more they are mentioned in the textual feedback. An annotation was assigned to each sentence of textual feedback to categorize them by content. Then, we obtained the number of sentences of each content type in each piece of textual feedback. Periods, exclamation marks, or question marks were considered as sentence breaks.

**Table 4.** Types of Instruction Contents

Types	Definition	Example of sentence
Giving Subjective Information (GSI)	Teacher providing general and/or specific conceptual information based on teacher's subjectivity.	<i>The tone is soft and comfortable to listen to.</i>
Giving Objective Information (GOI)	Teacher providing general and/or specific conceptual information based on objectively referable events or concepts.	<i>Too much arpeggio on the chords in bar 32 would sound unnatural.</i>
Asking Question (AQ)	Enquiring.	<i>Is there a problem with the tuning of the instrument?</i>
Giving Feedback (GF)	Teacher evaluation of a student's applied and/or conceptual knowledge.	<i>The detailed phrasing of the melody is well expressed.</i>
Giving Practice (GP)	Providing suggestions of ways to practice a particular passage or discussing a practicing schedule.	<i>The first step in practicing is to play only the melody.</i>
Giving Advice (GA)	Giving a specific opinion or recommendation without demonstration or modelling to guide the student's action towards the achievement of certain specific musical aims.	<i>I think it would be better to be more aware of the larger phrases and not stop the music so much within these phrases.</i>

The content types were adapted from the works of Simone [37], Carlin [4], and Zhukov [47] as shown in Table 4.<sup>11</sup> Each sentence was annotated as one of these six types of content. If a sentence was judged to consist of descriptions that could be classified as more than one type of content, the sentence was separated using commas. Two annotators annotated all the textual feedback. If their annotations for a sentence differed, they discussed the sentence and settled on a final annotation. The Cohen's Kappa coefficient, which was a statistic to measure inter-rater reliability, was 0.96 for the oboe dataset.

#### 4 Contents that Contribute to the Usefulness of Textual Feedbacks

In this section, we used the usefulness of each textual feedback (Section 3.3) and the number of sentences that meant each content in each textual feedback (Section 3.4) to identify content that significantly improves the usefulness of textual feedback.

<sup>11</sup> Types of "Demonstrating", "Modelling", and "Listening/Observing" were omitted because these actions might be not observed in textual feedback.

#### 4.1 Usefulness

First, this subsection showed demographic data on usefulness of each textual feedback. The average usefulness is following<sup>12</sup>; piano: 6.75 ( $\pm 2.01$ ), oboe: 7.27 ( $\pm 2.02$ )<sup>13</sup>, and guitar: 7.15 ( $\pm 1.90$ ).

The textual feedback with the highest usefulness and the lowest usefulness for the piano are presented below.

**The highest rated textual feedback for piano (p06-s02-c21, usefulness: 8.12  $\pm 1.68$ )**

*You have read the score carefully. The tempo was a little slow compared to the Allegro, so do your best to play faster as you continue to practice.*

*Check the Es in the third beat of the left hand in the 10th bar because you played it incorrectly.*

*You played the two-hand staccato in the 17th bar too long, so cut it a little shorter (same in the 78th bar).*

*You played the left-hand note in the 19th bar by extending it to the first beat of the 20th bar, rather than the whole beat, so you should cut it off properly on beat 1 (same for the left-hand in the 23rd bar).*

*I don't feel the sforzando in the 33rd and 34th bars at all. Play it with more force (don't hit the keyboard).*

*Don't extend the left-hand note in bar 40 a whole beat.*

*The beginning of the fourth beat of the right hand in the 43rd bar was slow.*

*I understand that you want to rit. from the descending flow of the right-hand note in the 42nd bar, but you should play it without slowing down the tempo as the score shows.*

*The half note in the 46th bar was long. Since it is staccato, let's play it as long as a quarter note.*

*I am concerned about the right hand note that comes in without a pause after the fermata in the 61st bar. As you can see when you actually sing it, you always need to breathe to start a new song after such a long note. Be sure to breathe at the eighth rest.*

**The lowest rated textual feedback for piano (p08-s02-c20, usefulness: 3.75  $\pm 2.27$ )**

*The melody sounded good and the harmony was well-balanced.*

*It was a very nice performance.*

#### 4.2 Hierarchy of Usefulness

This subsection explores whether the usefulness of text feedback depended on the teacher, the student, or the piece, independent of the musical instruments.

<sup>12</sup> Since the crowdworkers who participated in the questionnaire survey of usefulness were different for each musical instrument, an absolute value comparison of usefulness among musical instruments is not very meaningful.

<sup>13</sup> For oboe, this result was reprint of [29].



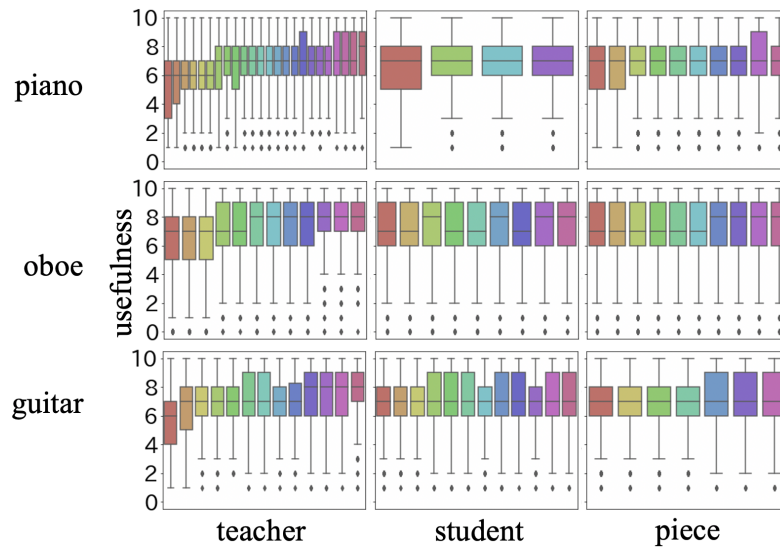


Fig. 1. The average usefulness for each teacher, student, and piece (sorted by usefulness score)

Figure 1<sup>14</sup> shows the average usefulness for each teacher, student, and piece. This result implies that the usefulness of the text feedback differed more by the teacher than by the piece or student.

For teachers, players, and pieces, the intraclass correlation coefficient (ICC) and the design effect (DE)<sup>15</sup> were calculated for each musical instrument.

**piano:** For teachers, ICC was 0.43 and DE was 3.14. For students, ICC was 0.006 and DE was 1.21. For pieces, ICC was 0.0, and DE was 1.0.

**oboe:** For teachers, ICC was 0.45 and DE was 9.43. For students or pieces, ICCs were 0.0, and DEs were 1.0<sup>16</sup>.

**guitar:** For teachers, ICC was 0.55 and DE was 11.2. For students or pieces, ICCs were 0.0, and DEs were 1.0.

In summary, independent of musical instruments, the usefulness showed hierarchy among teachers.

### 4.3 Factors Contributing to the Usefulness of Textual Feedback

For each of the three musical instruments, the content that significantly improves the usefulness of textual feedback was analyzed.

<sup>14</sup> For oboe, the figure was reprint of [29].

<sup>15</sup> DE is a criterion that takes into account both the average number of data in the group and ICC.  $DE = 1 + (k^* - 1)ICC$ , where  $k^*$  is the average number of data in the group. An ICC greater than 0.05 or a DE greater than two suggested that the data were hierarchical.

<sup>16</sup> For oboe, this result was reprint of [29].

**Method** Multilevel modeling was conducted to quantitatively analyze the effect of number of sentences annotated as each type of content for each of the three musical instruments. Multilevel modeling enables analysis assuming that the behavior of individual data changes depending on the hierarchy of data. In other words, in this study, not only the change in usefulness among textual feedback but also the influence of the teachers could be analyzed. R 4.1.0 and brms 2.15.0 were used for this multilevel modeling.

In the  $i$ -th feedback of the  $j$ -th participants, the usefulness of the  $k$ -th content  $U_{ij}$  is designated as follows:

$$U_{ij} = \alpha + \sum_{k=1}^6 \beta_k n_{ik} + \eta_k^{(z_{ijk}g)} + \sum_{k=1}^6 \gamma_k^{(z_{ijk}g)} n_{ik} + e_{ij}$$

Let  $\alpha$  be intercept,  $k$  be each type of contents,  $\beta_k (k = 1, \dots, 6)$  be the coefficient of  $n_{ik}$ ,  $n_{ik}$  be the number of descriptions for the  $k$ -th category. Here,  $z_{ijk}$  indicates each teacher who wrote the  $i$ -th feedback for each musical instrument.  $g$  indicates each teacher;  $g \in \{1, \dots, 24\}$  for piano,  $g \in \{1, \dots, 12\}$  for oboe, and  $g \in \{1, \dots, 13\}$  for guitar.  $\eta_k^{(z_{ijk}g)}$  is the random effect of the teacher on the intercept for the  $k$ -th content category of the  $i$ -th feedback.  $\gamma_k^{(z_{ijk}g)}$  is the random effect of the teacher on the coefficient for  $n_{ik}$ .

The model parameters were fitted with four Markov chain Monte Carlo chains with 2,000 iterations and 1,000 burn-in samples with a thinning parameter of one. Specifically, we used  $\beta_k \sim N(0, 100)$ ,  $\alpha \sim StudentT(3, 0, 2.5)$ , and  $\sigma_e \sim StudentT(3, 0, 2.5)$  as the prior distributions of the fixed effects,  $StudentT(3, 0, 2.5)$  as the prior distribution of SD of random effects, and  $LKJCholesky(1)$  as the prior distribution of the correlation matrix between  $\eta_k^{(g)}$  and  $\gamma_k^{(g)}$  for  $k \in \{1, \dots, 6\}$ .

**Results** Correlation coefficients between the six variables were checked for each musical instrument and all were less than 0.8, so all variables were used in the analysis. For each instrument, the estimates and 95% credible intervals of each content are shown in Table 5. R-hats for all features were under 1.05.

**Table 5.** Overview of the Results

	piano	oboe	guitar
$\beta_{GSI}$	-0.11[-0.24,-0.01]	0.07[-0.09, 0.23]	-0.02[-0.09, 0.05]
$\beta_{GOI}$	<b>0.07 [0.00, 0.14]</b>	<b>0.13 [0.06, 0.21]</b>	<b>0.12 [0.07, 0.19]</b>
$\beta_{AQ}$	0.05[-0.23, 0.32]	0.41[-1.11, 1.86]	-0.05[-0.34, 0.28]
$\beta_{GF}$	0.03[-0.09, 0.15]	<b>0.14 [0.04, 0.25]</b>	0.07[-0.01, 0.15]
$\beta_{GP}$	-0.08[-0.82, 0.14]	<b>0.27 [0.10, 0.45]</b>	<b>0.13 [0.04, 0.22]</b>
$\beta_{GA}$	<b>0.16 [0.09, 0.24]</b>	<b>0.15 [0.08, 0.22]</b>	0.08[-0.00, 0.17]

These results showed that the number of sentences that conveyed *GOI* were significantly positive for all of the three instruments. Therefore, the number of sentences

conveying *GOI* significantly contributes to the usefulness of the textual feedback independent of musical instruments. Moreover, the number of sentences conveying *GP* or *GA* significantly contributes to the usefulness of the textual feedback for the two musical instruments.

## 5 Discussion

In this study, textual feedback on musical recordings of oboe, piano, and guitar pieces were collected and analyzed. We quantitatively found that the usefulness of the textual feedback differed most significantly by teachers independent of musical instruments. Moreover, the number of sentences conveying *GOI* was found to significantly contribute usefulness of the textual feedback independent of musical instruments. In this study, different levels of students and teachers were involved in the collected recordings and textual feedback for each musical instrument (Table 1). Therefore, the instrument-independent results suggested that the results may be independent of the level of the player and the instructor.

Our result that the number of sentences conveying *GOI* significantly contributes to the usefulness of the textual feedback has high generalizability because the results can apply to face-to-face lessons.

In this study, the experiments were conducted only in Japanese. In the future, it will be necessary to conduct comparisons across multiple languages and discuss the differences between languages and cultures [3]. Another limitation was that this study used textual data as verbal information. We cannot deny the possibility that the verbal information in face-to-face speech shows different characteristics from the verbal information in textual data. An exploration of whether feedback should be psychologically supportive or what words should be used should be undertaken in the future.

## 6 Conclusion

We published the dataset for investigating the use of verbal feedback for three musical instruments. This dataset clarified that the content of text feedback were different between the teachers, and the feedback conveying *giving objective information* was critical for students independent of the musical instrument. In the future, we would like to utilize these findings in the development of educational programs.

## References

1. Bayley, J.G., Waldron, J.: “it’s never too late”: Adult students and music learning in one online and offline convergent community music school. *Int. J. Music. Educ.* **38**(1), 36–51 (2020)
2. Biasutti, M., Antonini Philippe, R., Schiavio, A.: Assessing teachers’ perspectives on giving music lessons remotely during the covid-19 lockdown period. *Musicae Scientiae* **26**(3), 585–603 (2022)
3. Campbell, P.S.: *Lessons from the world: A cross-cultural guide to music teaching and learning*. MacMillan Publishing Company (1991)

4. Carlin, K.D.: Piano pedagogue perception of teaching effectiveness in the preadolescent elementary level applied piano lesson as a function of teacher behavior. Ph.D. thesis, Indiana University (1997)
5. Cavitt, M.E.: A descriptive analysis of error correction in instrumental music rehearsals. *J. Res. Music. Educ.* **51**(3), 218–230 (2003)
6. Chen, Y.A., Yang, Y.H., Wang, J.C., Chen, H.: The amg1608 dataset for music emotion recognition. In: ICASSP. pp. 693–697 (2015)
7. Choi, E., Chung, Y., Lee, S., Jeon, J., Kwon, T., Nam, J.: Ym2413-mdb: A multi-instrumental fm video game music dataset with emotion annotations. In: *Ismir 2022 Hybrid Conference* (2022)
8. Dickey, M.R.: A comparison of verbal instruction and nonverbal teacher-student modeling in instrumental ensembles. *J. Res. Music. Educ.* **39**(2), 132–142 (1991)
9. Dorman, P.E.: A review of research on observational systems in the analysis of music teaching. *Bull. Counc. Res. Music. Educ.* pp. 35–44 (1978)
10. Duke, R.A.: Measures of instructional effectiveness in music research. *Bull. Counc. Res. Music. Educ.* pp. 1–48 (1999)
11. Duke, R.A., Simmons, A.L.: The nature of expertise: Narrative descriptions of 19 common elements observed in the lessons of three renowned artist-teachers. *Bull. Counc. Res. Music. Educ.* pp. 7–19 (2006)
12. Foscarin, F., McLeod, A., Rigaux, P., Jacquemard, F., Sakai, M.: ASAP: a dataset of aligned scores and performances for piano transcription. In: *ISMIR*. pp. 534–541 (2020)
13. Froehlich, H.: Measurement dependability in the systematic observation of music instruction: A review, some questions, and possibilities for a (new?) approach. *Psychomusicology* **14**(1-2), 182 (1995)
14. Gillman, D., Goyat, U., Kutlay, A.: Teach yourself georgian folk songs dataset: An annotated corpus of traditional vocal polyphony. In: *Ismir 2022 Hybrid Conference* (2022)
15. Goolsby, T.W.: Verbal instruction in instrumental rehearsals: A comparison of three career levels and preservice teachers. *J. Res. Music. Educ.* **45**(1), 21–40 (1997)
16. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: Rwc music database: Popular, classical and jazz music databases. In: *ISMIR*. pp. 287–288 (2002)
17. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: Rwc music database: Music genre database and musical instrument sound database. In: *ISMIR*. pp. 229–230 (2003)
18. Hamanaka, M., Hirata, K., Tojo, S.: Gtm database and manual time-span tree generation tool. In: *SMC*. pp. 462–467 (2018)
19. Hash, P.M.: Remote learning in school bands during the covid-19 shutdown. *J. Res. Music. Educ.* **68**(4), 381–397 (2021)
20. Hashida, M., Matsui, T., Katayose, H.: A new music database describing deviation information of performance expressions. In: *ISMIR*. pp. 489–494 (2008)
21. Hashida, M., Nakamura, E., Katayose, H.: Constructing pedb 2nd edition: a music performance database with phrase information. In: *SMC*. pp. 359–364 (2017)
22. Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.Z.A., Dieleman, S., Elsen, E., Engel, J., Eck, D.: Enabling factorized piano music modeling and generation with the MAESTRO dataset. In: *ICLR* (2019)
23. Hentschel, J., Neuwirth, M., Rohrmeier, M.: The annotated mozart sonatas: Score, harmony, and cadence. *Transactions of the International Society for Music Information Retrieval* **4**(1) (2021)
24. Lehmann, A.C., Sloboda, J.A., Woody, R.H., Woody, R.H., et al.: *Psychology for musicians: Understanding and acquiring the skills*. Oxford University Press (2007)
25. Li, B., Liu, X., Dinesh, K., Duan, Z., Sharma, G.: Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Tran. Multimedia* **21**(2), 522–535 (2018)

26. Matsubara, M.: Crocus: Dataset of musical performance critique (Jun 2021). <https://doi.org/10.5281/zenodo.4748243>
27. Matsubara, M., Kagawa, R., Hirano, T., Tsuji, I.: Analysis of the usefulness of critique documents on musical performance: Toward a better instructional document format. In: Ke, H.R., Lee, C.S., Sugiyama, K. (eds.) *Towards Open and Trustworthy Digital Societies*. pp. 344–353. Springer International Publishing (2021)
28. Matsubara, M., Kagawa, R., Hirano, T., Tsuji, I.: Crocus: Dataset of musical performance critiques: Relationship between critique content and its utility. In: *CMMR (2021)*
29. Matsubara, M., Kagawa, R., Hirano, T., Tsuji, I.: Useful feedback in asynchronous lessons of music performance: A pilot study of oboes. In: *The Journal of the Society for Art and Science*. pp. 241–255 (2022)
30. Miragaia, R., Reis, G., de Vega, F.F., Chávez, F.: Multi pitch estimation of piano music using cartesian genetic programming with spectral harmonic mask. In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. pp. 1800–1807. IEEE (2020)
31. Nakamura, E., Saito, Y., Yoshii, K.: Statistical learning and estimation of piano fingering. *Information Sciences* **517**, 68–85 (2020)
32. Salamon, J.: What’s broken in music informatics research? three uncomfortable statements. In: *36th International Conference on Machine Learning (ICML), Workshop on Machine Learning for Music Discovery*. Long Beach, CA, USA (2019)
33. Sapp, C.S.: Comparative analysis of multiple musical performances. In: *ISMIR*. pp. 497–500 (2007)
34. Schedl, M.: The lfm-1b dataset for music retrieval and recommendation. In: *ICMR*. pp. 103–110 (2016)
35. Silla Jr, C.N., Koerich, A.L., Kaestner, C.A.: The latin music database. In: *ISMIR*. pp. 451–456 (2008)
36. Simones, L., Schroeder, F., Rodger, M.: Categorizations of physical gesture in piano teaching: A preliminary enquiry. *Psychology of Music* **43**(1), 103–121 (2015)
37. Simones, L.L., Rodger, M., Schroeder, F.: Communicating musical knowledge through gesture: Piano teachers’ gestural behaviours across different levels of student proficiency. *Psychology of Music* **43**(5), 723–735 (2015)
38. Sturm, B.L.: An analysis of the gtzan music genre dataset. In: *ACM Workshop MIRUM*. pp. 7–12. MIRUM ’12 (2012)
39. Ueda, K., Takegawa, Y., Hirata, K.: Evaluation of a piano learning support system focusing on visualization of keying information and annotation. In: *E-Learn*. pp. 1198–1204 (2015)
40. Vigliensoni, G., Fujinaga, I.: The music listening histories dataset. In: *ISMIR*. pp. 96–102 (2017)
41. Visentin, P., Shan, G., Wasiak, E.B.: Informing music teaching and learning using movement analysis technology. *Int. J. Music. Educ.* **26**(1), 73–87 (2008)
42. Wang, Z., Chen, K., Jiang, J., Zhang, Y., Xu, M., Dai, S., Bin, G., Xia, G.: Pop909: A pop-song dataset for music arrangement generation. In: *ISMIR (2020)*
43. Weiß, C., Zalkow, F., Arifi-Müller, V., Müller, M., Koops, H.V., Volk, A., Grohgan, H.G.: Schubert winterreise dataset: A multimodal scenario for music analysis. *J. Comp. Cult. Herit.* **14**(2), 1–18 (2021)
44. Whitaker, J.A.: High school band students’ and directors’ perceptions of verbal and nonverbal teaching behaviors. *J. Res. Music. Educ.* **59**(3), 290–309 (2011)
45. Yalcinalp, S., Avci, Ü.: Creativity and emerging digital educational technologies: A systematic review. *Turkish Online Journal of Educational Technology-TOJET* **18**(3), 25–45 (2019)
46. Zhang, K., Zhang, H., Li, S., Yang, C., Sun, L.: The pmemo dataset for music emotion recognition. In: *ICMR*. pp. 135–142 (2018)
47. Zhukov, K.: Teaching styles and student behaviour in instrumental music lessons in Australian conservatoriums. Ph.D. thesis, University of New South Wales (2005)

# A Melody Input Support Interface by Presenting Subsequent Candidates based on a Connection Cost

Tatsunori Hirai\*

Komazawa University  
thirai@komazawa-u.ac.jp

**Abstract.** In this paper, we present a melody input support interface that offers multiple pre-existing melody fragments as potential continuations for the melody being composed. The proposed interface utilizes the connection cost between melody fragments, based on the BiLSTM approach proposed by the author [1]. It provides subsequent candidate melodies or notes when the user encounters difficulties or needs fresh ideas during the melody composition process. Specifically, we consider a melody composition scenario in which the user inputs melodies onto a piano roll. We propose an interface that searches and presents subsequent candidate melodies or notes from a database comprised of existing melodies, based on the user's inputted melody. We conducted a user study on melody composition utilizing the proposed interface and assessed the effectiveness of the interface, as well as the quality of the generated melodies. The results confirmed the effectiveness of the proposed interface.

**Keywords:** Composition support; Connection Cost; LSTM

## 1 Introduction

Melody is a crucial element that characterizes a musical piece, and its creation is prioritized in music production. Melodies can exhibit a wide variety of characteristics, ranging from simple motifs repeated multiple times to intricate compositions that do not feature identical melodies from beginning to end. A common aspect among numerous musical pieces is the requirement for a melody to possess adequate length to constitute an entire song, compelling a composer to craft such a melody from the ground up. However, there is a constraint on the length of a melody that can be conceived at once, often resulting in the creation of only a small portion of the entire song at a time. A prevalent approach in melody composition, albeit with numerous exceptions, involves generating short, phrase-sized melodies and connecting them sequentially.

Generating short melodies through humming is relatively easy and is considered achievable even for individuals without expertise in music composition. Conversely,



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS KAKENHI Grant Number JP19K20301 and JP23K17023.

crafting a melody for an entire song from start to finish is not something that everyone can do easily. Based on this observation, we consider that the difficulty in creating a melody primarily lies in effectively connecting short melody phrases. Consequently, we propose an interface designed to facilitate melody creation by presenting multiple candidate melodies that can follow the melody being created, utilizing the melody connection cost, a metric quantifying the naturalness of the connection between melody fragments.

By connecting short melody fragments conceived by an individual, it is possible to create longer melodies, potentially transforming a simple act of humming a tune into a more professional music production. Furthermore, if one can measure which melodies naturally connect together when creating mashup music comprising multiple tracks, irrespective of whether they are original or pre-existing, it could pave the way for supporting music production. As an exploration of the potential for music production support, this paper examines an interface designed to support melody input using connection costs.

In recent years, deep generative models such as Music Transformer [2] and MusicVAE [3] have been proposed for melody generation, yielding high-quality results. Many of these approaches are categorized as “automatic composition” models, implying a significant machine contribution when users employ them for creative purposes. In this study, we investigate the potential of supporting melody composition while maintaining a balance between human creativity and machine involvement.

The melody connection cost employed in this interface is based on a previously proposed model by the author, which utilizes BiLSTM [1]. This model can also be adapted for automatic melody generation through minor modifications to the network configuration. However, in this study, we refrain from generating melodies and solely use the model to calculate the naturalness of connections between melodies. The objective is to develop a system capable of suggesting melodies that can be connected to the melody currently being produced, drawing upon a vast collection of existing melodies.

Our interface does not generate melodies; rather, it provides existing melodies when necessary. Consequently, our objective is to develop a support interface that functions similarly to predictive text input. Its use is not obligatory, but it can be employed when beneficial candidate options are presented. In the proposed interface, the subsequent candidate melodies are not machine-generated but are manually created melodies. Furthermore, the machine’s role is minimized, as the final selection of subsequent melody candidates is left to the user’s discretion.

## **2 Related Work**

Bretan et al. proposed a melody generation technique employing existing melodies based on the connection cost of melodies, referred to as the unit selection method [4]. In this approach, new melodies are automatically generated by reusing pre-existing melodies. However, Bretan et al. did not focus on developing user-oriented support for music creation or associated interfaces. Furthermore, their method takes into account not only the connection cost between melodies but also their semantic relationships in order to narrow down the search space.

Cope also proposed an approach for generating new music by connecting existing melodies [5]. This method involves dividing a musical piece into small fragments, labeling each fragment according to its characteristics, and subsequently creating new music through the reuse and recombination of these fragments. The approach by Cope differs from the method presented in this paper as it relies on rule-based melody reconstruction rather than machine learning-based modeling.

The concept of employing existing melodies in music generation has been previously proposed. Pachet introduced a system called “The Continuator” that generates new melodies by dividing existing melodies into small fragments, modeling transitions between fragments using a tree-structured Markov chain, and searching for appropriate subsequent melodies from the training data [6]. Kitahara et al. proposed JamSketch, which generates improvised melodies in real-time using a genetic algorithm and existing melodies, based on the user’s rough outline of the melody input [7]. Although JamSketch does not utilize existing melodies in their original form, it is one example of utilizing existing melodies for melody generation.

The approach of generating new content by reusing existing content has been explored in various domains beyond music. For instance, it has been applied to image synthesis [8] and music video generation [9]. In this study, we focus on melody creation and propose an interface that utilizes the connection cost between melody fragments [1] to present existing melodies as candidates for subsequent melodies.

### **3 An Interface for Melody Input Support based on Connection Cost**

Our interface is designed as a melody input support tool that utilizes the connection cost between melody fragments based on the BiLSTM proposed by the author [1]. A piano roll is commonly employed when composing a melody using a computer. Consequently, the input support interface in this study aims to facilitate melody creation utilizing a piano roll.

#### **3.1 Basic Configuration of the Proposed Interface**

The proposed interface is implemented as an additional feature on top of the conventional piano roll. Users can input notes by dragging the piano roll using a pen-style input tool. As fundamental functionalities, the interface incorporates quantization features for aligning the onset timing and length of notes, a function to move, modify, and delete input notes, and capabilities to play, pause, and stop the entered melody. The interface is designed for inputting melodies by note, employing a grid in the time direction using a 4/4 time signature, with four beats per bar and a 16th note as the smallest unit.

This interface offers all the fundamental features typically present in a standard piano roll, facilitating users to accomplish all the essential tasks for melody input. By incorporating a function that presents information based on connection cost, melody input can be supported.



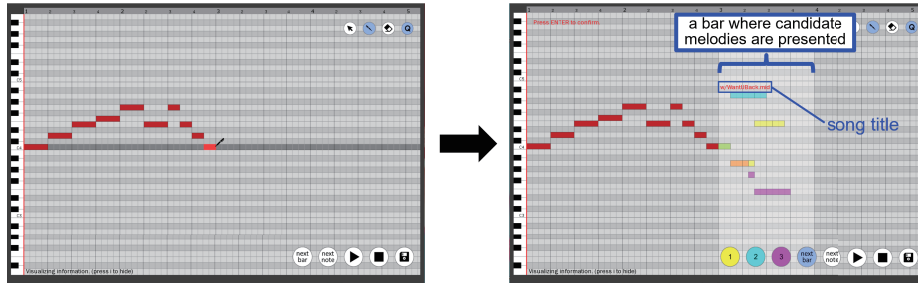


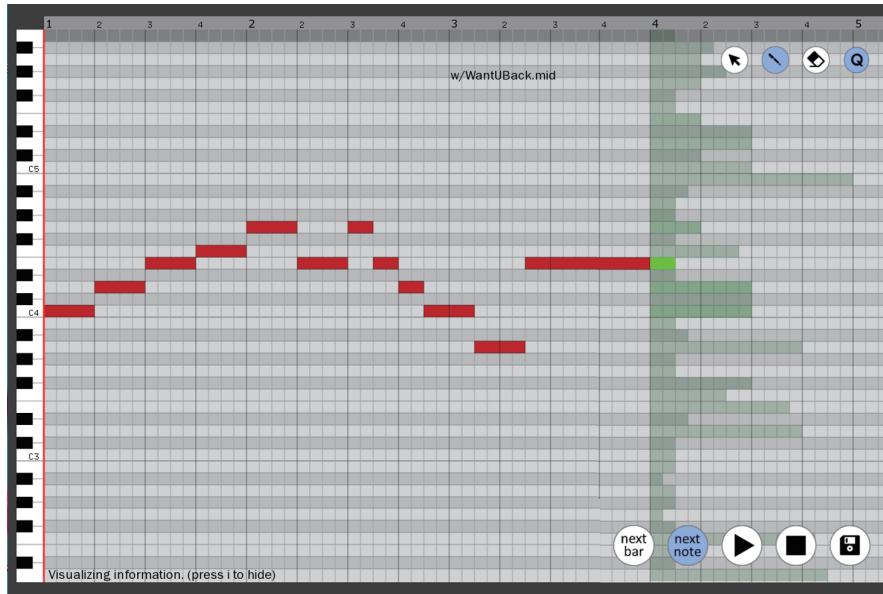
Fig. 1. Melody candidate suggestion based on connection costs

### 3.2 Melody Candidate Suggestion based on Connection Costs

We propose a function that suggests subsequent melody candidates based on the connection cost between melody fragments. The proposed function searches a pre-existing melody database for the three fragments with the lowest connection cost that are most likely to continue the inputted melody, and presents them to the user as recommendations for melody continuation. Upon inputting one or more bars of melody using the piano roll and pressing the "next bar" button, the interface calculates the connection costs between the last bar of the user's inputted melody and pre-existing melodies in the database that are one bar in length. The interface then suggests three possible melody candidates for the user to continue their melody, based on this calculation. Fig.1 shows an example of the subsequent melody candidate presentation. The left side of the Fig.1 shows the melody that the user manually inputted, while the right side of the Fig.1 shows the result screen after pressing the "next bar" button. The bright background in the piano roll indicates the bar where the subsequent melody candidates are presented. Users can listen to the three suggested subsequent melody candidates and select one to connect with their input melody. If users find a suggested melody that they like, they can incorporate it into their composition.

The name of the MIDI file from which the suggested melody candidates were extracted is displayed on the piano roll. If a candidate is selected, the corresponding song title will continue to be displayed on the melody of the corresponding bar. After selecting a melody from the presented candidates, the user can edit it further as with a typical piano roll interface. If the user doesn't like a certain part of the candidate melody, they can modify it to fit their own image while keeping the original style. At this time, the user is not required to adopt the presented melody into their own composition, so it can be used only as a reference when the user gets stuck in their composition process. This feature is positioned to assist users only when necessary, as it is not a mandatory function.

Although automatic composition methods that generate subsequent melodies have been previously proposed, a notable aspect of our interface is that the suggested melodies are based on existing melodies, which are manually created rather than generated automatically. With this function, the user can add preferred melody data to the database and search for melodies that are more likely to be connected to the current melody from a large number of existing melody dataset. They can then adopt these



**Fig. 2.** Visualization of subsequent note candidates

melodies as part of their own composition. The title of the original melody is displayed at the top of the corresponding melody, making it possible to create the user’s own melody while inheriting and citing existing melodies.

When the “next bar” button is pressed, inference is performed in the background to calculate the connection cost between the input melody and the melodies in the database. Therefore, the more melodies there are to search, the longer the wait time until candidate melodies are presented. Currently, when searching for candidates for 10,000 bars, it takes approximately 20 seconds on a machine with 32.0GB memory, Intel Core i9-1088H 2.40GHz, and NVIDIA GeForce RTX 2060. When using this feature, shorter wait times are desirable as they allow for more attempts to be made, and faster feedback can be obtained. The waiting time can be shortened by improving the implementation, and reducing it further is our future challenge.

### 3.3 Visualization of Subsequent Note Candidates

When calculating the melody connection cost, the validity of note-level connections is also considered, and by visualizing it during melody input, a user can examine what would be appropriate as the next input note. Pressing the “next note” button reveals the candidates for subsequent notes, including the type and likelihood of notes that are likely to follow the last note the user inputted. Fig.2 shows how the subsequent note candidates are visualized.

As shown in Fig.2, the interface visualizes which pitch and duration the user would be preferable to input as the next note after the last note they inputted. This visualization

is based on the frequency of note transitions in the melodies of the dataset used to train the original melody connection cost calculation model. Therefore, it simply shows more common note transitions in a darker green color. Since many existing melodies have frequent transitions to the same pitch, this function often suggests notes of the same pitch as the most probable candidates. It should be noted that this is simply an information visualization, and users are not obligated to input the next note based on this information. This information can serve as a reference when transitioning to less common notes, and is intended as a suggestion to the user while they actively input the melody.

The proposed interface provides two functions to assist with melody input: suggesting subsequent melody candidates by bar and by note. The suggestion of candidates is entirely optional, and both functions are designed to be utilized only when the user needs them.

## **4 User Study**

A user study is conducted to evaluate the effectiveness of the proposed melody input support interface.

### **4.1 Conditions of the User Study**

We conducted a user study with four participants who used the proposed melody input support interface. Each participant completed six melody input trials, three with and three without using the function for suggesting candidate melodies. After each trial, participants responded to a questionnaire to evaluate the system's effectiveness. Before starting the user study, the author demonstrated how to operate the interface to the participants, and they were given the opportunity to try it out after learning the basic operation method. We also provided a document that explained the details of each button, which participants could refer to if they were unsure of how to operate the interface during the trials. The participants' musical experience for this user study was as follows:

- User A: Less than 1 year of music experience, no experience in DTM (desktop music: music production software), and some experience in composing songs at a level of humming.
- User B: No music experience, no DTM experience, no composition experience.
- User C: Over 10 years of musical experience, experience with DTM, and some experience in composing songs at a level of humming.
- User D: No music experience, no DTM experience, no composition experience.

Participants were asked to input short melodies consisting of 2 to 4 bars with the proposed interface, and the interface was evaluated through multiple trials. During the trials where the melody candidate suggestion function was utilized, participants were instructed to use the function within a 4 bars, while the subsequent note candidate suggestion function was optional and used only when necessary. For each participant's

six trials, the subsequent melody candidate suggestion function was used on even-numbered trials, alternating between trials with and without its use. Trials excluding the melody candidate suggestion were utilized as our baseline. In the baseline trial, participants inputted a melody of 2 to 4 bars into the piano roll interface without any guidance.

The database used for the melody candidate suggestion function consisted of 10,000 bars of melody randomly extracted from test data that were not used for training the connection cost calculation model. When the suggestion function is used under these conditions, it takes about 20 seconds to process.

In addition to assessing the interface, upon completion of all trials by the participants, we further evaluated the melodies themselves. Each participant's set of 6 melodies was reviewed by three other participants, who were not the original creators, for evaluation.

#### **4.2 Evaluation Items**

Participants were asked to evaluate each melody creation trial based on the following four evaluation criteria.

1. Able to create a desired melody
2. Able to create a unexpected melody
3. Able to create a satisfactory melody
4. Able to create melodies easily

The melody was created six times in total, with three times using the melody candidate suggestion function and three times without using it. After completing each melody, the participants were asked to rate the four evaluation criteria mentioned above on a 4-point scale, with options "1: Does not apply", "2: Somewhat does not apply", "3: Somewhat apply", and "4: Apply".

After the 6 trials and responses to the evaluation items were completed, an overall evaluation was conducted. For each subsequent melody and subsequent note candidate suggestion function, participants were asked to rate their effectiveness on a 4-point scale: "1: Not effective", "2: Somewhat not effective", "3: Somewhat effective", "4: Effective". Furthermore, regarding the subsequent note candidate function, each participant was asked to evaluate the degree of use of the optional subsequent note candidate function, which was evaluated in four levels: "1: Almost never used", "2: Rarely used", "3: Used several times", "4: Used frequently". Finally, participants were asked to give their general opinions and feedback in an open-ended format.

All user trials were recorded with screen captures, and the duration of each trial was measured. Furthermore, the influence of the feature on the time needed to create a melody was assessed.

The evaluation of all melodies created by the participants in the user study was conducted by asking them to rate each melody on a 4-point scale, ranging from "1: not a good melody", "2: not a very good melody", "3: somewhat a good melody", to "4: a good melody". Additionally, the evaluation was conducted by the remaining three participants of the user study who listened to each melody without knowledge of how it was created.

**Table 1.** Evaluation results of melody creation trials

	Evaluation items			
	(1)	(2)	(3)	(4)
<u>without</u> candidate suggestion	2.17	2.25	2.08	2.42
<u>with</u> candidate suggestion	<b>3.25</b>	<b>3.58</b>	<b>3.17</b>	<b>3.75</b>

**Table 2.** Evaluation results of each function's effectiveness

	average evaluation score
Effectiveness of melody candidate suggestion	3.75
Effectiveness of note candidate suggestion	2.67
Frequency of using note candidate suggestion	1.5

All evaluation items were rated on a 4-point scale, where higher ratings denote better performance. The intermediate value is 2.5, with ratings above this value indicating a positive outcome.

### 4.3 Evaluation Results

The results of the user study are presented in Table 1, 2, and 3. Table 1 shows the evaluation results for each melody creation trial. It presents the average evaluation scores separately calculated for the presence and absence of the candidate suggestion function. Table 2 presents the evaluation results regarding the effectiveness of the candidate suggestion function after all trials were completed. It shows the average evaluation values for each item. Table 3 shows the evaluation results of the six melodies created by each participant, as evaluated by the remaining three participants. It shows the average evaluation values for all six melodies produced by each participant, including the average score with/without the candidate suggestion function. The evaluation scores range from 1 to 4, with higher values indicating better performance.

Based on the results presented in Table 1, all evaluation items received higher scores when using the subsequent melody candidate suggestion function compared to when it was not used. Notably, the use of the candidate melody suggestion function resulted in higher scores even for the evaluation item “able to create a desired melody.” These results imply that the presented candidate melodies are more aligned with the melody that users imagine. Specifically, for participants who were creating a melody with piano roll for the first time, it appeared challenging to compose musically pleasing melodies. In such a situation, the melodies suggested by the candidate suggestion function are actual melodies that possess musical sense. Therefore, it is inferred that the support provided by the function fulfilled the users' requirements and facilitated them in achieving their melody creation goals.

As shown in Table 2, the average evaluation score for the subsequent melody candidate suggestion function's effectiveness was 3.75, with all four participants indicating that it was effective. In contrast, the note candidate suggestion function's average evaluation score for effectiveness was 2.67 and was not evaluated as particularly effective. In terms of usage frequency, three out of the four participants reported that they “almost

**Table 3.** Evaluation results of composed melodies

User	Evaluation score					
	without candidate suggestion			with candidate suggestion		
A	3.33	2.33	2.00	3.67	3.33	2.67
B	1.67	2.33	2.33	2.33	2.33	3.33
C	3.00	3.33	3.00	3.00	3.67	3.33
D	1.67	1.67	3.33	3.00	2.67	3.00
average	2.50			<b>3.03</b>		

never used” the feature, indicating that it did not contribute significantly to melody creation support.

Based on the evaluation results of the melodies produced by the four participants, as presented in Table 3, the melodies created using the melody candidate suggestion function received higher overall ratings than those created without using the function. The quality of the created melodies varied among users. For instance, user C, who had the most musical experience, received evaluation scores of 3 or higher for all of their created melodies. Examining the evaluation values for each melodies based on whether they used the function or not, it can be seen that every user was able to create higher-quality melodies by using the function. These results indicate that the interface support has improved the quality of the melodies produced.

The following are some of the comments obtained through the open-ended section at the end of the trial<sup>1</sup>.

- I would like the system to propose other melodies when I don’t like the suggested melody.
- The note suggestion function kept suggesting the same notes.
- I was glad that the created song didn’t become monotonous because the system suggested melodies that I wouldn’t have thought of myself.
- After repeating the process, I gained a sense of what makes a melody work and felt that as I became better at creating melodies, the suggested melodies also improved.

The feedback obtained suggests that the interface provided a certain level of useful assistance; however, there is still room for improvement in the subsequent note suggestion function. We intend to incorporate the feedback received to enhance the interface in the future.

Finally, we evaluated the impact of using the melody suggestion functions on the time required for creating melodies. Table 4 shows the time required for all six melody creation trials for each user. In the condition with the melody suggestion function, the waiting time for suggestions (approximately 20 seconds per use) was also included in the total time. Users with less experience tended to use the candidate suggestion function multiple times, resulting in longer overall required times due to the waiting times that occurred each time. Moreover, the time required for comparing and listening to the three proposed melody candidates also added up to the required time. Therefore, it can be concluded that the current interface does not contribute to the efficiency of

<sup>1</sup> The comments originally provided in Japanese have been translated into English by the author.

**Table 4.** Evaluation of the duration required for each trial

User	Time required for trial					
	without candidate suggestion [s]			with candidate suggestion[s]		
A	681	184	233	450	663	551
B	138	134	244	208	261	231
C	166	124	193	181	222	216
D	218	191	181	297	238	368
average	<b>223.9</b>			322.2		

melody creation in terms of time. We aim to address this issue by improving the system speed and providing more suitable candidate melodies based on user needs in the future.

Through this user study, it became apparent that users improved their melody creation skills as they repeated the trials. Additionally, some users gained an understanding of what kind of melodies to input to receive better candidate suggestions. These findings suggest that, like traditional music production tools, repeated use of this tool can lead to greater proficiency, making it more convenient to use. The observation that humans adapt their behavior to the tool suggests the potential for collaboration between artificial intelligence technology and human music creation, making it an intriguing research topic for future studies.

## 5 Discussion

In this chapter, we discuss the potential and concerns of the interface introduced in Section 3.2, which enables the reuse of existing melodies.

As mentioned in Chapter 1, this interface was developed with the idea that if the act of inputting short melody phrases such as humming can be connected to the creation of longer melodies for an entire song, anyone can easily engage in music production. The interface is designed to support such endeavors, and the results of the user study in Chapter 4 demonstrate the effectiveness in melody creation.

When reusing existing melodies, it can encourage the reuse of other people's creative works, which can be both positive and negative. Creative activities are often inspired by the works of others, and in music, for example, it is a legitimate practice to compose based on chord progressions of songs created by others. While it is difficult to deal with melodies and not permitted to use them as is, paying homage to past music by incorporating someone else's melodies into one's own work is a common practice. Short units such as a single bar have countless examples of songs that share melodies with other works. Sampling has emerged as a well-established musical genre and technique that involves incorporating segments of pre-existing music or sounds into one's own compositions. Our interface can be viewed as an interface that enables the direct sampling of melodies.

When using the function in our interface to suggest subsequent melody candidates based on existing melodies, the original song file name is displayed on the piano roll, providing an opportunity to credit the reused music information in the final composition. This allows for the creation of works that include citations, akin to the culture of

fan fiction. However, the interface not only enables reusing melodies as they are but also re-editing them to fit one's own melody, posing a challenging issue from a copyright perspective on how to treat a reused melody that no longer retains its original form.

Using the proposed interface, one can extract melodies from short phrases previously created by oneself, in a manner akin to predictive text input, even without utilizing others' works as the database. By accumulating many short phrases on their own, users can conveniently extract their own melodies. As this process involves reusing materials created by oneself, there are no rights-related issues. We anticipate that the proposed interface will continue to serve as a useful tool when employed in this manner.

The proposed interface opens up new possibilities for collaborative music creation among multiple creators. Drawing inspiration from the way short sentences are retweeted and attached to other tweets on Twitter, we envision the possibility of expanding the system further by incorporating a mechanism that facilitates the reuse of short melody phrases shared by multiple users on social networking services (SNS). Such an approach would enable the construction of a single composition through the amalgamation of diverse phrases contributed by numerous users. This could lead to a future where someone's casual humming could be incorporated into a professional musician's new song.

Sound libraries such as Splice<sup>2</sup> offer numerous publicly available short audio materials that are utilized by creators worldwide as components of their works. Just as there are cases where lyrics are completed by collecting words submitted by fans and having professional artists write the final version, a collaborative production approach can also be applied to musical elements such as melodies. The proposed interface is one example of how such a production style can be implemented.

## 6 Conclusion

In this paper, we proposed an interface that supports melody input by presenting candidate melodies based on the connection cost between melody fragments. We conducted a user study to evaluate the effectiveness of the proposed interface for assisting melody input and confirmed its effectiveness by evaluating melodies created by users using the interface.

The proposed interface enables users to combine short melody fragments to construct longer melodies, seamlessly incorporating melodies created by themselves or other users as necessary. It includes a function similar to culture of fan-created content, allowing users to credit the sources of melodies used. This is particularly important since there is no clear legal definition of the maximum length of a melody that can be reused without infringing on copyright law. However, additional deliberation is required to judge whether edited melodies are also permissible for use. This interface can be used without infringing on any rights issues if users utilize melodies that they have previously created. In such cases, there are no copyright infringement issues as it involves reusing one's own material.

---

<sup>2</sup> <https://sounds.splice.com/>



A potential future direction for this research is to improve the response speed of the interface. To present candidate melodies for the subsequent phrase, the interface needs to perform inference to calculate connection costs between the input melody and all melodies in the dataset. Consequently, the current waiting time to compute the connection cost between the input melody and the 10,000-bar search candidates is approximately 20 seconds. Bretan's unit-based melody generation [4] narrows down the search space by utilizing the semantic relationship between melodies. Preprocessing, such as this, is crucial for enhancing processing speed. In the future, we aim to enhance the functionality of this interface to make it more practical and develop it into a tool that can be used with actual DAW software in formats such as VST plugins.

## References

1. Hirai, T., Sawada, S.: A Method for Calculating Melody Concatenation Cost based on BiLSTM. *Journal of Global Media Studies*, Volume 31, pp.55–64 (2022)
2. Huang, A., Vaswani, A., Uszkoreit, J., Shazeer, N., Hawthorne, C., Dai, A.M., Hoffman, M.D., Eck, D.: Music Transformer: Generating Music with Long-Term Structure. In: *Proceedings of the International Conference on Learning Representations* (2018)
3. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., Eck, D.: A hierarchical latent vector model for learning long-term structure in music. In: *Proceedings of the 35th International Conference on Machine Learning*, pp.4364–4373 (2018)
4. Bretan, M., Weinberg, G., Heck, L.: A unit selection methodology for music generation using deep neural networks. In: *Proceedings of the International Conference on Computational Creativity* (2017)
5. Cope, D.: One approach to musical intelligence. *IEEE Intelligent Systems and their Applications*, Volume 14, No.3, pp.21–25 (1999)
6. Pachet, F.: The continuator: Musical interaction with style. *Journal of New Music Research*, Volume 32, No.3, pp.333–341 (2003)
7. Kitahara, T., Giraldo, S., Ramírez, R.: JamSketch: Improvisation Support System with GA-Based Melody Creation from User's Drawing. In: *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, pp.509–pp.521 (2017)
8. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, Volume 28, No.3 (2009)
9. Nakano, T., Murofushi, S., Goto, M., Morishima, S.: DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Dance Video Clips on the Web. In: *Proceedings of the 8th Sound and Music Computing Conference*, pp.183–pp.189 (2011)

# Phoneme-inspired playing technique representation and its alignment method for electric bass database

Junya Koguchi and Masanori Morise \*

Meiji University  
{korguchi, mmorise}@meiji.ac.jp

**Abstract.** In plucked string instruments such as electric bass, the attack phase is dominated by non-periodic components resulting from picking noise, while the sustain phase is dominated by periodic components resulting from string vibrations. This phenomenon is analogous to unvoiced consonants and voiced vowels in speech, suggesting the possibility of applying speech phoneme representations to plucked string instrument playing techniques. In this study, we design playing technique labels for an electric bass database by treating the attack phase as consonants and the sustain-to-decay phase as vowels. Furthermore, we employ a phoneme alignment algorithm to obtain the alignment between the playing technique labels and the acoustic signals of the electric bass. To conduct experiments, we construct an electric bass database and apply methods based on hidden Markov models and dynamic time warping. As a result, methods based on dynamic time warping, particularly those incorporating timbre transformations, provided the most accurate alignment.

**Keywords:** Electric bass, playing technique, phoneme alignment, hidden Markov model, dynamic time warping

## 1 Introduction

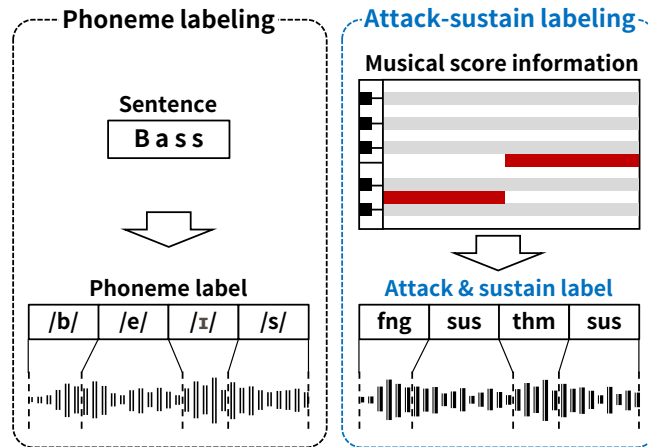
The advancement of musical information retrieval research is supported not only by machine learning and signal processing techniques, but also by open sound databases. Many of these databases include not only sound data but also annotation data. Sound databases with useful annotations accelerate research and enhance reproducibility. For instance, the presence of musical score information like MIDI can assist in automatic transcription and sound synthesis [1, 2], while attributes such as genre can aid in music information retrieval [3]. Furthermore, playing technique information plays a crucial role in accurately representing their timbre and articulations.

When considering applications for controllable instrument sound synthesis [4] and playing technique recognition [5], it is essential to include detailed information on

\* This work was supported by JSPS KAKENHI Grant Number JP22J22158 and JP21H04900.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



**Fig. 1.** Proposed attack-sustain label contrasted with phoneme label. For example, two notes played by finger picking and thumping are converted to the labels “fng-sus” and “thm-sus”, respectively.

changes in playing techniques in addition to musical score information. However, there is no standard format for expressing playing techniques in MIDI. Some software synthesizers implement out-of-range notes as key switches for changing playing techniques [6, 7], but the types and assignments of these techniques vary among developers. Many databases provide only note information but playing technique information. This is due to the time-consuming annotation process. In addition, since performance techniques may change independently for each note of a multipitch instrument, it is difficult to track them on a single time axis.

This problem might be solvable, at least for electric bass signals, by applying insights from speech processing. Firstly, it is reasonable to assume a monophonic melody in normal performances. Although electric basses with multiple strings can play chords, their role within an ensemble is to provide a monophonic bass and rhythm part. Moreover, in the attack phase of electric bass, non-periodic components dominate due to picking noise, while periodic components dominate during the sustain phase due to string vibrations. Electric bass playing techniques can be broadly divided into those that change the attack phase, such as fingerpicking and slapping, and those that change the sustain phase, like harmonics and muting [8]. This is similar to the relationship between consonants and vowels in speech. Furthermore, string vibrations result in integer harmonic components, which are then shaped through pickups. This suggests that the source-filter model [9], which approximates vocal fold vibrations as a periodic impulse train and filters the vocal tract characteristics, is also a valid approximation for electric bass. Promising acoustic features and analysis algorithms based on the source-filter model are expected to be applicable.

In this study, we propose the Attack-sustain label for annotating electric bass playing techniques (**Fig. 1**). The Attack-sustain label treats playing techniques that depend on changes in the attack phase as consonants and those that depend on the sustain phase as vowels. This label is provided as a temporally aligned sequence of playing technique

symbols, separate from MIDI, similar to phoneme labels in singing voice. This provides detailed annotation data on the temporal transitions of playing techniques, which can be useful for instrument sound synthesis and playing technique recognition. Additionally, by focusing on the acoustic similarity between electric bass and speech, it is possible to automate segmentation using high-precision phoneme alignment methods.

In our experiments, we aligned our Attack-sustain labels with acoustic signals. We constructed a new electric bass database and applied conventional alignment methods which are based on viterbi algorithm of hidden Markov model [10] and dynamic-time-warping (DTW) [4], DTW with timbre conversion based on a voice conversion (VC) [11]. Our results demonstrate that our Attack-sustain labels provide temporally accurate annotations of playing techniques.

## 2 Attack-sustain label

### 2.1 Label design

A naive annotation method of a technique to a note is an assignment of a single technique to a single note (hereinafter referred to as "note-wise"). For example, for a note played by plectrum picking, "plectrum" is assigned to that note. However, annotating a performance that combines multiple techniques, such as a muted string played with plectrum picking, requires multiple symbol sequences, complicating the annotation process.

We focus on the acoustic properties of the electric bass signal. Electric bass signals are generated by plucking the strings with a finger or pick. The strings collide with the pick/finger/fret depending on the playing technique, generating aperiodic noise. Then, depending on the playing technique (mute/harmonics/etc.), periodic string vibrations are generated and slowly decay. Focusing on this generative process suggests that the acoustic differences in playing techniques can be broadly classified into those that appear in the attack phase and those that appear in the sustain phase [8].

**Table 1** lists techniques corresponding to attack and sustain (hereinafter, they are called "attack technique" and "sustain technique", respectively). Techniques that affect string vibration, such as mute and harmonic techniques, are distinguished. We assign "pause" to a silent segment such as a rest.

**Table 1.** The list of playing techniques corresponding attack and sustain labels.

Attack	Sustain
Finger, pick, thump, thumb up, pluck, hammer on, pull off	Sustain, mute, harmonics, slide up, slide down

### 2.2 Automatic alignment method

**Viterbi alignment of HMM** Because controllable systems typically uses explicit temporal segmented data [4,12]. However, the manual annotation requires well-experienced

annotators in detecting segment boundaries. A common automatic method in speech processing is a Viterbi alignment based on hidden Markov models (HMMs) [10]. HMMs are trained using pairs of label sequences and acoustic features, and the Viterbi path is a temporal alignment of the technique label sequences [13]. The HMM perform robustly for performances that contain some disturbance such as noise and small fluctuation. However, because the HMM is based on switching stationary signal sources, it is difficult to model slowly decaying string vibration. The effects of its improvements such as hidden semi-Markov models [14] and trajectory HMMs [15] are also limited, because not only the playing technique, but also the pitch and duration of the notes vary depending on the musical context. In addition, the accuracy of data-driven approaches is highly dependent on the amount of data.

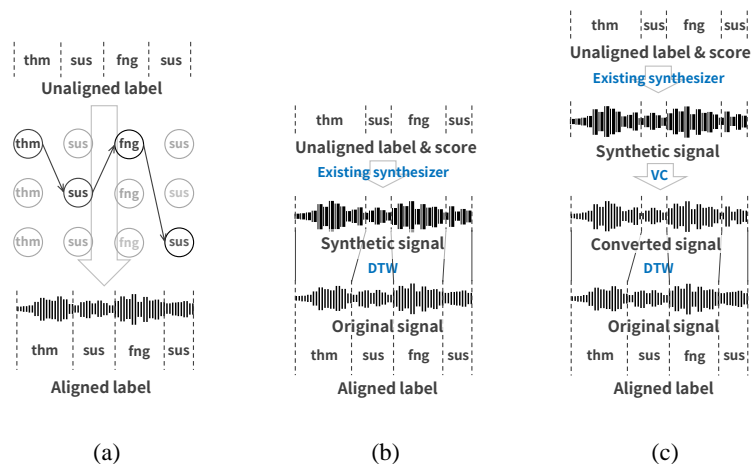
**DTW** Another method is synthesizing electric bass signals from the musical scores using existing synthesizers (e.g., sample concatenative synthesizer), and obtaining the alignment with the recorded signal by DTW [16]. Since the synthesizer generates a faithful performance to the musical score, the label's temporal offset can be obtained from the alignment of the synthetic and recorded sound.

**DTW with timbre conversion** Since the timbres differ between synthetic and recorded sound, this affects the alignment accuracy of the DTW. To reduce this problem, we utilize timbre conversion during the DTW using a VC technique. It has shown efficacy in singing voice alignment [11] and is also promising for electric bass with acoustic similarity to speech. First, the alignment of synthesized and recorded speech is obtained as described above. Next, using the aligned sound, a VC model (e.g., affine transformation [17] or Gaussian mixture model (GMM) [18]) is trained to transform the synthetic sound's timbre into the recorded one's timbre. Finally, the DTW takes between the converted and the recorded sound. This method is expected to be more accurate in alignment because the distribution of acoustic features is closer to the recorded sound. In addition, it is known that the DTW and timbre conversion can be sufficiently accurate in a single iteration [17].

### 3 Experimental evaluation

#### 3.1 Dataset

A new electric bass sound database was constructed to evaluate the accuracy with respect to actual acoustic signals. The sounds used were 180 phrases of four bars of monophonic bass line (approximately 112 minutes), containing all techniques in the list (**Table 1**), and each with a various tempo between from 60 to 120 beat per minute (BPM). The label series before alignment was given manually. The note-wise label gave the attack label and sustain label pair as a single symbol. Finger picking, for example, is annotated as “fng-sus” for a single note. The electric bass used was a Fender custom shop 1962 Jazz Bass [19], the audio interface was an RME ADI-2 Pro FS R [20], and the performance was recorded by an experienced player.



**Fig. 2.** Overview of automatic alignment algorithms. Each figure shows (a) Viterbi alignment of HMM, (b) DTW and (c) DTW with timbre conversion.

### 3.2 Conditions

We apply the alignment algorithms to the proposed attack-sustain label and evaluate its accuracy. The most straightforward evaluation in comparing alignment methods is to calculate the error to ground truth. However, it is difficult to manually obtain ground truth for all the data. Therefore, we performed manual labeling on randomly selected pieces and calculated the mean absolute error (MAE) [21] on the rest of pieces for each attack and sustain technique.

In addition, for all data, we segmented acoustic features following the resulting alignment, and we calculated a separation metric (SM)  $R$  [11] defined as

$$R = \sum_D \frac{\sum_a \omega_a (\mu_a - \mu)^2}{\sum_a \omega_a \sigma_a^2}. \quad (1)$$

The subscript  $a$  indicates a technique label.  $\mu_a$  and  $\mu$  is the mean in the segment of technique label  $a$  and the global mean, respectively.  $\sigma_a$  is the standard deviation in the segment of technique label  $a$ .  $\omega_a$  is the amount ratio of  $a$ : the number of frames in  $a$  segment divided by the total number of frames. These values are calculated from each dimension of  $D$ -dimensional acoustic features segmented following the resulting alignment.  $\mu$  is the global mean calculated from the whole of database. When the resulting alignment can segment acoustic features for each label accurately, intra-technique standard deviation (i.e.,  $\sigma_a$ ) becomes smaller, and  $R$  becomes larger.

We first evaluated whether the Note-wise label or our attack-sustain label gives a more accurate alignment. The SM and MAE for the HMM-based alignment result were calculated for the two labels. To ensure fair conditions, Attack-sustain labels were compared to the start and end times of the Note-wise label, while Attack-sustain labels were compared to the start time of the Attack label and the end time of the Sustain label. The performance of the DTW-based method was omitted because it depends only on the acoustic signal.

We secondly compared alignment methods described in **Section 2.2** as follows.

- **HMM**: Viterbi alignment of the HMMs [10]
- **DTW**: DTW between synthetic and recorded sound [16].
- **DTW+AF**: DTW with Affin-transform-based timbre conversion [17].
- **DTW+GMM**: DTW with GMM-based timbre conversion [18].

10% of the dataset was manually annotated, and 20% was evaluated by SM and the remaining 70% was used to train the HMM. These subsets were randomly selected. The sound was recorded at 48 kHz sampling/16-bit PCM and were downsampled to 16 kHz for acoustic feature extraction. Mel cepstrum was downsampled to 16 kHz with a window length of 1024 and a hop size of 5 ms. 24-dimensional mel-cepstral coefficients were used as acoustic features. The number of Gaussian mixtures was set to 4 for “HMM” and 16 for “DTW+GMM”. For the DTW-based method, we used Standard Bass V2 [22] as a sample concatenative synthesizer. Each sample was manually labeled and aligned in advance. The cost of DTW was calculated as the mean squared error between the acoustic features.

### 3.3 Result and discussion

**Table 2** lists the result of the label comparison. The alignment accuracy with our attack-sustain label became higher. This is because the note-wise HMM assumes a stationary signal for the steep acoustic change from attack to sustain. On the other hand, our attack-sustain label improved the accuracy by distinguishing between harmonic and non-harmonic states. However, there are still estimation errors in DTW-based methods. Focusing on the MAE, there were about 10 ms of errors in the time boundary of the technique. It is possible that noise from the release of the pressed strings interfered with the DTW path and was incorrectly estimated as the attack phase.

**Table 3** lists the results. First, there are no large differences of SM and MAE in attack technique. This is considered that aperiodic components were dominant in the attack segment, and the acoustic features varied steeply. On the other hand, “HMM” scored the worst in sustain technique, and “DTW”, “DTW+AF”, and “DTW+GMM” scored better in that order. This indicated that the DTW-based method worked robustly because the synthesizer replaced the modeling of non-stationary decay of the string vibration. In addition, the affine-transformation-based conversion is equivalent to the single-component GMM. “DTW+GMM” therefore enhanced the performance because of the higher accuracy of the timbre conversion.

In this experiment, both DTW and HMM were performed on a single player’s performance. Different players perform different types of electric bass and in different styles, resulting in different acoustic characteristics. Thus, the accuracy may vary depending on a performer. This difference correspond to speaker differences in speech. Parallel voice conversion also uses the DTW between different speakers and performs high quality conversion. Since the electric bass signal exhibits similar acoustic characteristics to speech, it is expected to produce similar results in the signals of different performers.

**Table 2.** Comparison of alignment accuracy between note-wise label (Note) and our attack-sustain labels (AS). Separation metric (SM) and mean absolute error (MAE) from the ground truth of alignment methods. Higher SM value and lower MAE indicate more accurate.

Method	SM		MAE [ms]	
	AS (ours)	Note	AS (ours)	Note
HMM	<b>35.73</b>	20.01	<b>24.15</b>	32.00

**Table 3.** Accuracy comparison of automatic alignment methods. Separation metric (SM) and mean absolute error (MAE) from the ground truth of alignment methods. Higher SM value and lower MAE indicate more accurate.

Method	SM		MAE [ms]	
	Attack	Sustain	Attack	Sustain
HMM	11.98	40.03	35.14	20.06
DTW	13.07	60.03	21.23	11.69
DTW+AF	12.31	67.79	19.45	<b>11.28</b>
DTW+GMM	<b>13.98</b>	<b>68.15</b>	<b>19.24</b>	12.42

## 4 Conclusion

This paper proposed the attack-sustain label inspired by phoneme representation. By labeling the playing technique changes separately into attack and sustain techniques, as in the case of vowels and consonants, the method in speech processing can also be applied to electric bass signals.

We investigated automatic labeling method to align the label sequence to the acoustic signal. The experimental evaluation demonstrated that 1) our attack-sustain label is effective for accurate alignment 2) the method based on DTW with timbre conversion achieved better accuracy. In our future work, we will increase the data and train DNN-based synthesis models using our the label and acoustic signal pairs. Moreover, constructed sound database will be available in the public domain.



## References

1. 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 ICLR*, 2019.
2. Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.
3. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical, and jazz music databases," in *Proc. ISMIR*, Paris, France, Oct. 2002, vol. 2, pp. 287–288.
4. Hao-Wen Dong, Cong Zhou, Taylor Berg-Kirkpatrick, and Julian McAuley, "Deep performer: Score-to-audio music performance synthesis," in *Proc. ICASSP*, 2022, pp. 951–955.
5. Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, and Elaine Chew, "Adaptive scattering transforms for playing technique recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1407–1421, 2022.
6. Toontrack, "Ezbass," <https://www.toontrack.com/product/ezbass/>.
7. IK Multimedia, "Modo bass 2," <https://www.ikmultimedia.com/products/modobass2/>.
8. Jakob Abeßer, Hanna Lukashovich, and Gerald Schuller, "Feature-based extraction of plucking and expression styles of the electric bass guitar," in *Proc. ICASSP*, 2010, pp. 2290–2293.
9. Gunnar Fant, *Acoustic Theory of Speech Production*, De Gruyter Mouton, Berlin, Boston, 1971.
10. F. Brugnara, D. Falavigna, and M. Omologo, "Automatic segmentation and labeling of speech based on Hidden Markov Models," *Speech Communication*, vol. 1, no. 4, pp. 357–370, 1993.
11. J. Koguchi, S. Takamichi, and M. Morise, "PJS: phoneme-balanced japanese singing-voice corpus," in *Proc. APSIPA ASC*, 2020, pp. 487–491.
12. E. Cooper, X. Wang, and J. Yamagishi, "Text-to-Speech Synthesis Techniques for MIDI-to-Audio Synthesis," in *Proc. SSW 11*, 2021, pp. 130–135.
13. K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "HMM-based singing voice synthesis and its application to Japanese and English," in *Proc. ICASSP*, 2014, pp. 265–269.
14. Shun-Zheng Yu, "Hidden semi-markov models," *Artificial Intelligence*, vol. 174, no. 2, pp. 215–243, 2010.
15. H. Zen, K. Tokuda, and T. Kitamura, "A viterbi algorithm for a trajectory model derived from hmm with explicit relationship between static and dynamic features," in *Proc. ICASSP*, 2004, vol. 1, pp. I-837.
16. N. Hu, R.B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Proc. WASPAA*, 2003, pp. 185–188.
17. G. Kotani, H. Suda, D. Saito, and N. Minematsu, "Experimental investigation on the efficacy of affine-dtw in the quality of voice conversion," in *Proc. APSIPA ASC*, 2019, pp. 119–124.
18. T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.
19. Fender Custom Shop, "1962 jazz bass," <https://www.fendercustomshop.com/basses/jazz-bass/>.
20. RME, "ADI-2 Pro FS R," <https://www.rme-audio.de/adi-2-pro-fs-be.html>.
21. A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *Proc. ISMIR*, 2007, pp. 315–316.
22. Purple\_Shikibu\_, "Standard Bass V2," <https://unreal-instruments.wixsite.com/unreal-instruments/standard-bass>.

# An Audio-to-Audio Approach to Generate Bass Lines from Guitar's Chord Backing

Tomoo Kouzai<sup>1</sup> and Tetsuro Kitahara<sup>1</sup> \*

<sup>1</sup>Graduate School of Integrated Basic Sciences, Nihon University  
Setagaya-ku, Tokyo, Japan  
{kouzai, kitahara}@kthrlab.jp

**Abstract.** In this paper, we address a system that generates a bass line from a chord backing played on the electric guitar in an audio-to-audio manner. Yielding bass lines for guitar chord backings would be helpful for amateur musicians composing band music. Conventional music arrangement systems targeted MIDI-like symbolic music representations, but accurately obtaining symbolic representations from guitars takes work. To solve this problem, we attempt an audio-to-audio approach; Once the user gives an audio recording of the guitar's chord backing, the system extracts some audio features (spectrogram, mel-spectrogram, or chromagram) and then generates an audio signal of bass lines using a convolutional neural network. The experimental results showed that the model with chromagrams generates bass lines the most robustly.

**Keywords:** music, CNN, guitar, band arrangements, audio-to-audio

## 1 Introduction

The electric guitar is one of the central instruments in light music, especially in band music. Therefore many amateur guitarists enjoy playing in a band. When they try to play their original songs in a band, a particular member (such as the guitarist) often composes a melody and a chord progression. They often collaboratively decide the phrases of instrumental parts (e.g., bass, drums). However, it is a challenging task because it requires musical knowledge, like typical phrases of each instrument. If the phrases of each instrument part can be automatically decided on a computer and the band members can listen to them, creating original songs may be more efficient.

Most of the existing studies on automatic music arrangements have been for the piano, such as piano arrangement from band or orchestra pieces[1, 2] and score reduction of piano pieces for beginners[3]. Although some studies targeting guitar, most of them are systems for arranging solo guitar scores, such as generating solo guitar scores

---

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

from orchestral scores[4] and from audio signals[5]. No research has been done on band arrangements that add drums, bass, or other sounds from the guitar sound.

Our goal is to develop a system that automatically makes band arrangements for a given song. This system is intended to be used by people who can play a simple backing for their original song on the guitar but cannot create phrases of other instruments, such as the bass and drums. As the first step, this paper addresses a system that generates a bass line for a given chord backing played on the guitar.

Most existing systems of music arrangement use MIDI-like symbol music representations, but it is not easy to accurately obtain a symbolic representation from recordings of guitar performances (Although there are commercial products of MIDI guitars, their audio-to-MIDI conversion is not necessarily accurate). We, therefore, adopt an audio-to-audio approach in which both inputs (guitar backings) and outputs (bass lines) are audio signals.

## 2 Proposed Method

Given an audio signal of a chord performance played on the electric guitar, our method generates an audio signal of a bass performance that fits the given guitar performance. For simplicity, the tempo and length are fixed (120 BPM and four measures in the current implementation). First, the given guitar signal is converted to a feature representation (i.e., spectrogram, mel-spectrogram, or chromagram). Then, it is segmented by 0.5 seconds, and each segment is input to a convolutional neural network (CNN), which generates a bass spectrogram. Finally, the bass spectrogram is converted to an audio signal. To train the CNN model, we use a pairwise dataset consisting of guitar feature representations and bass spectrograms.

### 2.1 Calculation of the spectrogram of the input sound source

The spectrogram is computed from a given guitar audio signal (and the bass source when learning) using the short-time Fourier transform (STFT) after downsampled to 22050 Hz. The Hann window is used. The window size is set to 2048, and the hop size is set to 1/1000 of the sampling frequency.

### 2.2 Feature extraction

We attempt three different feature representations:

- Spectrogram: The amplitude spectrogram obtained in Section 2.1 is used without conversions.
- Mel-spectrogram: This is calculated from the spectrogram using Librosa.
- Chromagram: This is also calculated from the spectrogram using Librosa. The hop size for the chromagram is set to 512.

Below, the models with a spectrogram, a mel-spectrogram, and a chromagram are called the *STFT model*, *Mel model*, and *Chroma model*, respectively.

### 2.3 Generation of bass spectrogram

The feature representation (spectrogram, mel-spectrogram, or chromagram) of the given guitar signal is converted into a spectrogram of a bass performance using a CNN model, because CNNs are widely used for analyzing spectrograms[6–11]. Our CNN model (Figure 1) consists of convolution layers and deconvolution layers as follows:

**Convolution layers** For the *STFT* model, the convolution layers consist of:

- 1st layer: The frequency axis of the guitar’s feature representation is compressed to one dimension. The input spectrogram of dimension  $1025 \times 500$  (frequency axis: 1025, time axis: 500) is compressed to a  $1 \times 500$  matrix with a  $1025 \times 1$  filter.
- 2nd layer: The  $1 \times 500$  matrix is compressed to a  $1 \times 250$  with a  $1 \times 2$  filter.
- 3rd layer: The  $1 \times 250$  matrix is compressed to a  $1 \times 50$  with a  $1 \times 5$  filter.
- 4th and later layers: A  $1 \times 2$  filter and a  $1 \times 5$  filter are alternately applied until a  $1 \times 5$  matrix is obtained.

The number of filter channels in each layer is 1024. The stride is 1. No padding is used. A ReLU function is used for the activation.

For the *Mel* model, the filter size for the 1st layer is  $128 \times 1$  because the input matrix size is  $128 \times 500$ . Apart from this, the same configurations are used.

For the *Chroma* model, the following convolution layers are used:

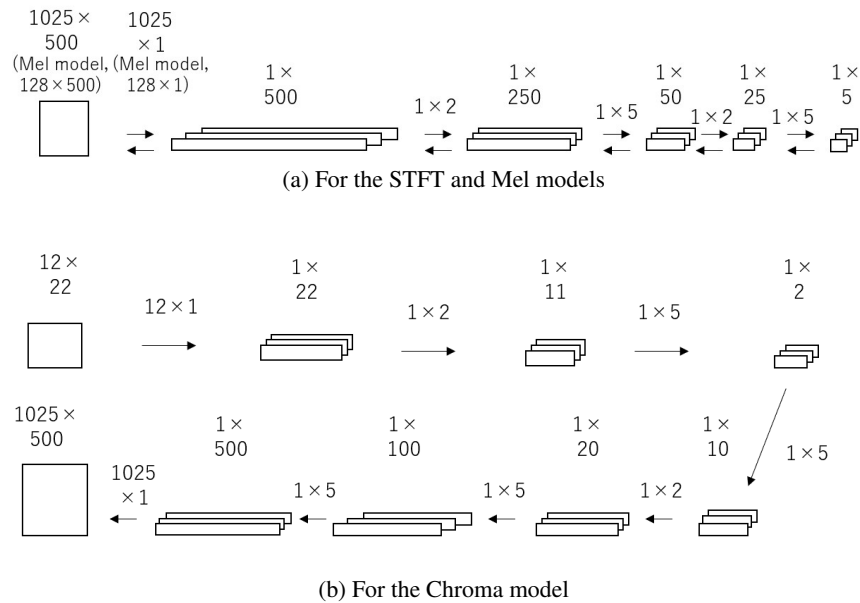
- 1st layer: The  $12 \times 22$  chromagram is compressed to a  $1 \times 22$  (filter size:  $12 \times 1$ ).
- 2nd layer: The  $1 \times 22$  matrix is compressed to a  $1 \times 11$  matrix (filter size:  $1 \times 2$ ).
- 3rd layer: The  $1 \times 11$  matrix is compressed to a  $1 \times 2$  matrix (filter size:  $1 \times 5$ ).

**Deconvolution layers** The set of deconvolution layers generates a bass spectrogram independently of the feature representation used for guitar signals. It consists of:

- Multiple decomposition layers with filter sizes of  $1 \times 5$  and  $1 \times 2$  are alternatively applied. These layers convert a  $1 \times 5$  matrix (a  $1 \times 2$  matrix for the *Chroma* model) to a  $1 \times 500$  matrix.
- After that, a deconvolution layer expanding the frequency axis is applied. This layer has a filter size of  $1025 \times 1$ , which converts a  $1 \times 500$  matrix to a  $1025 \times 500$  matrix. This matrix represents a bass spectrogram.

### 2.4 Generation of the bass’s audio signals

The audio signal of the bass part is obtained by using inverse Fourier transform and phase restoration on the spectrogram generated from the CNN. The Griffin-Lim algorithm is used for phase restoration. The number of iterations is 32, the window size is 2048, and the hop size is 1/1000 of the sampling frequency. To reduce impulsive noises, we use harmonic percussive source separation (HPSS) because impulsive noises are similar to percussive sounds.



**Fig. 1.** Architecture of the CNN model. The numbers above the rectangles represent the shape of the data, and the numbers above the arrows represent the shape of the filter. Right-pointing arrows indicate the convolution layer and left-pointing arrows indicate the inverse convolution layer.

### 3 Experiment

We conducted an experiment to confirm whether an appropriate bass sound can be generated in several conditions.

#### 3.1 Dataset

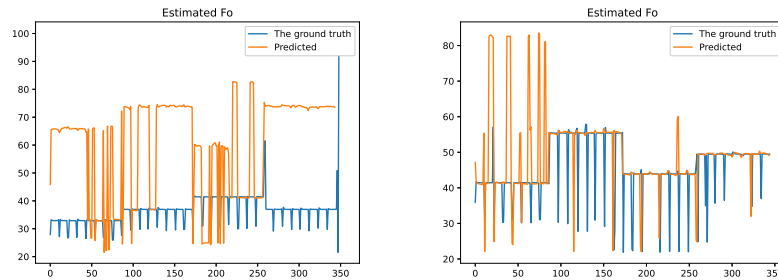
We made MIDI sequences that consisted of the guitar chord performances and bass lines using Cakewalk by BandLab. For simplicity, we only used four-bar chord progressions that consisted of one chord per measure. The guitar and bass performances are a sequence of eighth notes (for the bass, root eighth notes). Those MIDI sequences were converted to waveforms using software synthesizers (sforzando for the guitar and SI-Bass Guitar for the bass, included in Cakewalk by BandLab). The tempo for all sequences was set to 120 BPM. Based on these criteria, 20 pairs of guitar and bass signals were created. These pairs include those of the same chord progression but with different voicings. An example is shown in Fig.2. Out of them, 10 were allocated for training and 10 for testing.

#### 3.2 Experimental conditions

The following three conditions were set.



**Fig. 2.** Examples of guitar and bass scores created



**Fig. 3.** Condition 1: F0 of generated bass lines with the Chroma model and the ground truth (Left: CDmEmD, the lowest accuracy; Right: EmAmFG, the highest accuracy)

**Condition 1** The chord progressions or voicings are different between the training and test data, but all conditions in generating audio signals are the same.

**Condition 2** In addition to Condition 1, the acoustic features are different between the training and test data. Specifically, a low-pass filter (setting:  $-3\text{dB}$  per octave increase) was applied to the test data.

**Condition 3** The training data were those described above, while the test data was a recording of a performance by the first author on a real guitar. It was recorded using M-Audio's M-Track.

The generated bass signals were evaluated by calculating the ratio of *correct* frames. When the difference of the fundamental frequency (F0) at each frame from the signal given as the ground truth is lower than 50 cents, that frame is regarded as a *correct* frame. This ratio is called *accuracy* here. We also calculated *octave-ignored accuracy*, in which the difference of 1200 cents was considered correct.

### 3.3 Experimental results

The experimental results, listed in Table 1, can be summarized below<sup>1</sup>.

**Experimental condition 1** The model with the highest average accuracy, both with and without octave ignorance, was the Chroma model, and the model with the lowest average accuracy was the Mel model. When the octave is not ignored, the Mel model

<sup>1</sup> Audio samples are available at: <https://sites.google.com/kthrlab.jp/cmmr2023-kouzai>

**Table 1.** Accuracy and Octave-ignored accuracy for each test data

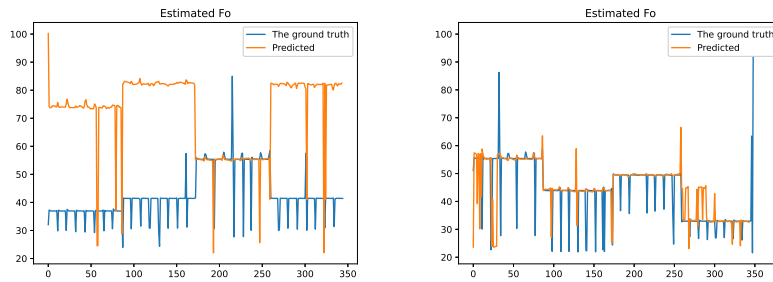
Condition	Filename	Accuracy			Octave-ignored accuracy		
		STFT	Mel	Chroma	STFT	Mel	Chroma
Condition1	A#CDmEm_voicing	0.37	0.32	0.66	0.55	0.65	0.83
	EABmC#m_voicing	0.39	0.20	0.64	0.48	0.29	0.67
	CDEmAm_voicing	0.39	0.49	0.53	0.53	0.57	0.81
	GABmD_voicing	0.55	0.54	0.62	0.62	0.56	0.77
	GCDEm	0.58	0.56	0.62	0.76	0.57	0.83
	CDmEmDm	0.42	0.21	0.17	0.66	0.26	0.77
	DmEmAmEm	0.57	0.40	0.32	0.65	0.40	0.84
	EmAmFG	0.59	0.39	0.81	0.69	0.42	0.87
	AmFGC	0.70	0.54	0.79	0.81	0.58	0.88
	FAmGDm	0.58	0.35	0.63	0.69	0.38	0.90
	Average	0.51	0.40	0.58	0.65	0.47	0.82
Condition2	A#CDmEm_voicing	0.29	0.26	0.58	0.53	0.58	0.78
	EABmC#m_voicing	0.22	0.26	0.62	0.27	0.41	0.75
	CDEmAm_voicing	0.17	0.30	0.51	0.29	0.43	0.82
	GABmD_voicing	0.28	0.49	0.67	0.34	0.51	0.74
	GCDEm	0.24	0.31	0.52	0.32	0.34	0.78
	CDmEmDm	0.15	0.11	0.23	0.20	0.10	0.85
	DmEmAmEm	0.23	0.17	0.22	0.27	0.18	0.85
	EmAmFG	0.35	0.26	0.68	0.43	0.28	0.88
	AmFGC	0.17	0.35	0.76	0.21	0.39	0.79
	FAmGDm	0.41	0.42	0.58	0.43	0.46	0.83
	Average	0.25	0.29	0.54	0.33	0.37	0.81
Condition3	CDEmAm_Audio	0.20	0.09	0.35	0.21	0.11	0.69

The name of the test data represents the chord progression. The same chord progression used for training, but with different voicing, was given "\_voicing".

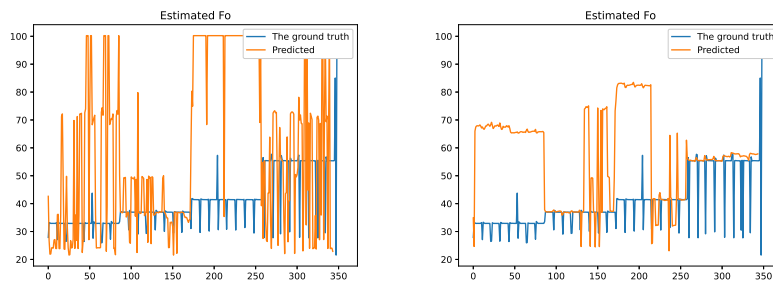
had the lowest accuracy among the three models in 6 out of the 10 data. When the octave is ignored, the Mel model had the lowest accuracy among the three models in 8 out of the 10 data.

Looking at Figure 3, which shows generated bass lines' F0 with the highest and lowest accuracy in the Chroma model, we can see that F0 is moving up and down. This is because the estimated F0s often contain double-pitch errors. In fact, the octave-ignored accuracy for these data is 0.77 and 0.6, respectively. For data containing three or more minor chords, the STFT model showed higher accuracy than the Chroma model, but again, the octave-ignored accuracy was high with the Chroma model.

**Experimental condition 2** As in Condition 1, the model with the highest average accuracy with and without octave ignorance was the Chroma model. Especially for the Chroma model, the average accuracy was almost the same as for Condition 1. On the other hand, the model with the lowest average accuracy with and without octaves ignorance was the STFT model.



**Fig. 4.** Condition 2: F0 of generated bass lines with the Chroma model and the ground truth (Left: DmEmAmDm, the lowest accuracy; Right: AmFGC, the highest accuracy)



**Fig. 5.** Estimation of the fundamental frequency of CDEmAm\_audio in the lowest accuracy Mel model and the highest accuracy Chroma model

Compared to Condition 1, the average accuracy for the STFT model dropped by more than 0.2, while the average accuracy for the Chroma model did not drop as much. This would be because the chromagram is a robust feature to timbral changes caused by the low-pass filter. Compared to Condition 1, the accuracy for data with many minor chords was lower for all models, especially for the STFT model; the accuracy dropped to less than half of the accuracy in Condition 1.

Figure 4 showed that the F0 fluctuates less up and down than in Condition 1. Instead, for DmEmAmEm, the double pitch was stably estimated. This is why the octave-ignored accuracy is high (0.85) while the accuracy is low (0.22).

**Experimental condition 3** Although the accuracy was lower than in conditions 1 and 2, the model with the highest accuracy was the Chroma model, while the model with the lowest accuracy was the Mel model. The Mel model generated no harmonic tone in the first two measures. Because there were no harmonic tones, the F0 estimator showed erroneous values, as shown in Figure 5. This is why this model showed the lowest accuracy. With the Chroma model, bass-like harmonic tones were generated but were slightly distorted. This distortion caused double-pitch errors in F0 estimation; in fact, the accuracy and octave-ignored accuracy had a large difference.



## 4 Conclusion

In this paper, we proposed a method for generating bass signals from given guitar signals using a convolutional neural network. The experimental results show that the accuracy of the model using the chromagram is the best in all conditions, while the accuracy of the model using the mel-spectrogram and STFT is considerably low for guitar signals with a low-pass filter.

However, these models have been tested only with simple bass lines that consist of only root notes. To enable to generate more complex bass lines, the models need to learn various bass lines, ranging from rhythmic to melodious ones, played in real songs. To achieve this, we must consider longer contexts in the models. Therefore, we would like to extend our models, for example by increasing context layers.

## References

1. S. Onuma, M. Hamanaka.: Piano Arrangement System Based on Composers' Arrangement Processes, ICMC, pp.191–194 (2010)
2. E. Nakamura, S. Sagayama.: Automatic Piano Reduction from Ensemble Scores Based on Merged-Output Hidden Markov Model, ICMC, pp.298–305 (2015)
3. M. Terao, Y. Hiramatsu, R. Ishizuka, Y. Wu, K. Yoshii.: Difficulty-Aware Neural Band-To-Piano Score Arrangement Based on Note-and Statistic-Level Criteria, ICASSP, pp.196–200 (2022)
4. S. Ariga, S. Fukayama, M. Goto.: Song2Guitar: A Difficulty-aware Arrangement System for Generating Guitar Solo Covers from Polyphonic Audio of Popular Music, ISMIR, pp568–574 (2017)
5. Daniel R. Tuohy, Walter D. Potter.: GA-based Music Arranging for Guitar, Congress on Evolutionary Computation, pp.1065–1070 (2006)
6. A. Ferraro, D. Bogdanov, X. Serra, Jay H. Jeon, J. Yoon.: How Low Can You Go? Reducing Frequency and Time Resolution in Current CNN Architectures for Music Auto-tagging, EUSIPCO2020 (2020)
7. M. Dong.: Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification, arXiv:1802.09697 (2018)
8. Y. Gong, S. Khurana, A. Rouditchenko, J. Glass.: CMKD: CNN/Transformer-Based Cross-Model Knowledge Distillation for Audio Classification, arXiv:2203.06760 (2022)
9. G. Jiang, A. Biswas, C. Bergler, A. Maier.: InSE-NET: A Perceptually Coded Audio Quality Model based on CNN, AES (2021)
10. C. Ji, Y. Pan.: Infant Vocal Tract Development Analysis and Diagnosis by Cry Signals with CNN Age Classification, SpeD, pp.37–41 (2021)
11. Ephrem A. Retta, R. Sutcliffe, E. Almekhlaf, Yosef K. Enku, E. Alemu, Tigist D. Gemechu, Michael A. Berwo, M. Mhamed, J. Feng.: Kinit Classification in Ethiopian Chants, Azmaris and Modern Music: A New Dataset and CNN Benchmark, arXiv:2201.08448 (2022)

# Teaching Chorale Generation Model to Avoid Parallel Motions

Eunjin Choi<sup>1</sup>, Hyerin Kim<sup>2</sup>, Juhan Nam<sup>1</sup>, and Dasaem Jeong<sup>2</sup>

<sup>1</sup> Graduate School of Culture Technology, KAIST

<sup>2</sup> Department of Art & Technology, Sogang University

**Abstract.** This paper presents a music generation model trained with Bach’s chorales and classical music theory rules. Although previous work has shown promising results in generating the four-part harmony, one of the limitations is the frequent appearance of parallel 5th or 8th, which are prohibited in music theory and rarely used in Bach’s chorale. To address this issue, we propose an additional loss that minimizes the probability of prohibited patterns, comparing the results with those from inference using a post-hoc probability manipulation to prevent parallel 5th and 8th. The experimental result shows that applying the proposed loss term can help to reduce parallel motion without losing other quality.

**Keywords:** Music generation, Bach chorale, Domain knowledge injection

## 1 Introduction

Music generation is a fascinating research topic that has received much attention for centuries. From W.A. Mozart’s *Musikalisches Würfelspiel* (musical dice game), there have been several works conducted in a rule-based approach, such as David Cope’s *Experiments in Musical Intelligence* [1]. Since the success of deep learning, however, data-driven approaches using neural networks have been dominating the music generation. Especially, several Bach chorale generation models have shown promising results [2, 3]. However, previous works [4, 5] pointed out that these models tend to generate note patterns that were avoided by Bach, such as parallel 5th and 8th, which are shown in Figure 1. Fang et al. showed that these parallel 5th and 8th patterns are the most distinctive characteristics to distinguish the model’s generation from Bach’s original chorales [5].

It is not surprising that the data-driven model could generate prohibited patterns, because most music generation models, including language-modeling-based, use likelihood maximization for the training objective; models only learn the pattern that exists in the training dataset. Therefore, this training strategy is not effective in teaching the model what patterns to avoid.

In this context, we raise three research questions. First, how can we inject music domain knowledge or rules into the data-driven generation model? Second, how can we



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

The image shows a musical score for the chorale 'Wer nur den lieben Gott läßt walten' (mm. 1-2). The score is in G major (one sharp) and 4/4 time. It features a grand staff with treble and bass clefs. A red dashed box highlights a parallel 5th interval between two notes in the treble clef staff, with the number '5' written in red next to it. The text 'Parallel 5' is written in red below the staff. The score ends with a double bar line and a fermata.

**Fig. 1.** Example of parallel 5th in four voice chorale. The highlighted notes are intentionally manipulated to demonstrate parallel 5th, which are rare in Bach’s original chorale.

teach the generation model to recognize the absence of something rather than its presence? Third, can we improve the generation model performance using music theory?

To address this issue, we propose a novel musically-informed loss term for training a music generation model. We compared the experimental results with the JS Fake Chorale dataset [6]. Generated music samples are available on the online webpage.<sup>3</sup>

## 2 Related Works

### 2.1 Deep Music Generation with Domain Knowledge

While early deep learning-based music generation studies were conducted by bringing models from computer vision [7] or natural language processing (NLP) [8, 9] domains, there are an increasing number of music generation models inspired by intuition from music domain. We have summarized these cases into two main groups. One group focuses on the structure and repetition of music defined as a unit of pattern such as theme [10], loop [11], bar relations [12], skeleton [13], or hierarchical structure [14], and extracted the unit with a rule-based approach and utilized it for modeling, or focused on modeling the structure of music using hierarchical encoding methods [15–17]. The other group utilizes intuition from music domain knowledge to suggest novel music generation system paradigms: non-unidirectional music generation system suggested by Coconet [3], and BERT-based music generation system [18, 19], combinatorial music generation system [20], harmonic expectation-based music harmonization system [21].

<sup>3</sup> <https://bit.ly/3LMajEK>

To the best of our knowledge, so far there have been no (or few) studies that have used music theory directly in the training procedure of deep learning-based music generation. We conjecture that it is not necessary to follow the rules strictly unless it is classical music, and the stricter the rules we impose on the model, the less diverse the music produced. Rather, studies indirectly utilize some music knowledge by steering models to learn specific features of music.

[22] proposed a contrastive loss that steers a music transformer to have arbitrary logical music features. Studies on the disentanglement of latent representation [23, 24] proposed models to learn specific music features in certain latent dimensions, which can be useful to steer the model in a controllable way. Studies that focus on the chord-conditioned music generation [25, 26] could be an example that uses music theory in a music generation as well. However, as we noted in the introduction section, the likelihood-based models ended up rarely generating notes prohibited in music theory. In this context, our approach has novelty in that it uses music knowledge directly in the model training scheme so that the model could learn to avoid prohibited patterns according to music theory.

## **2.2 Bach Chorale Generation**

As a representative corpus of Western classical polyphonic music, Bach's chorale has been widely adopted for music generation research. Among many, we introduce previous deep learning-based approaches to Bach chorale. BachBot [27] is one of the earliest examples of success in modeling Bach chorale with the deep neural network, or long short-term memory (LSTM) more specifically. BachBot uses 16th-grid sampling, along with additional tokens for time grid delimiter. Fermata, which plays a critical role in notating the phrase boundary in Bach's chorale, was also considered as an additional token in BachBot.

DeepBach [2] proposed to apply pseudo-Gibbs sampling instead of generating the music in sequential order. While DeepBach used LSTM as a main neural network block, CocoNet [3] applied a convolutional neural network using similar ideas of applying Gibbs sampling. The model was employed as a backbone to serve Google's first AI-powered doodle, Bach doodle, which generates Bach-like harmonization for a user's input melody.

Another recently proposed model, TonicNet[28], is closer to BachBot in the sense that it uses the 16th-grid sampling with ancestral sampling. Here, the author proposed a feature-rich encoding scheme, such as a number of sustain counts for each voice and adding a chord token at the head of each time frame. The author later proposed the JS fake Chorale [6], a dataset of machine-generated chorales, even though an explanation of the model used for the generation was not provided along.

The frequent appearance of parallel 5th and 8th is considered as a problem with deep learning-based Bach chorales generation. However, no previous research has made a direct attempt to reduce the parallel motions of the generated chorale. This paper suggests a novel loss function that directly prohibits parallel motion.

### 3 Methods

#### 3.1 Problem Formulation

In ordinary language modeling, the problem can be defined as modeling the probability distribution of the next token for given previous tokens, such as  $P(x_{t+1}|x_0, \dots, x_t)$ . However, in the symbolic music generation, one can provide more information about the current time step before predicting the token, such as a beat position or which voice the current token has to belong to. We can group this information as a condition  $c$  and formulate the music language modeling as Equation 1

$$P(x_t|x_0, \dots, x_{t-1}, c_0, \dots, c_t) \quad (1)$$

where  $x_t$  and  $c_t$  represents a predicting token and a condition token of timestep  $t$ , respectively. During the inference,  $c$  can be calculated by a rule-based approach for every timestep. Since the condition of the current time step is given explicitly, the model does not have to implicitly predict the information, such as to which beat the current time step belongs. While providing the condition also can be done synchronously with the predicting token  $x_t$ , separating the  $c_t$  from  $x_{t-1}$  has several advantages. Even though  $c_t$  is easily predictable for a given  $c_{t-1}$  in many cases, there are some exceptional cases, such as measure boundary with different time signatures. If we notate the offset of the current time step from measure starting in the sixteenth notes, the next offset for 11 is 0 for time signature 3/4 and 12 for time signature 4/4. By providing  $c_t$  instead of  $c_{t-1}$ , we can eliminate this type of ambiguity.

While any causal model, such as a transformer decoder, can be used for this task, we used a stack of uni-directional GRU as our model. We also tried a stack of transformer decoder module, but the result was not better than GRU.

#### 3.2 Data Representation

Following the previous works on Bach chorales generation [2, 27, 28], we use 16th-grid sampling so that a single bar of 4/4 time signature is represented with sixteen-time frames. A single voice is represented as a sequence of  $F$ -dimensional tokens  $v \in \mathbb{Z}^{T \times F}$ , where  $T$  represents the number of total time frames and  $F$  represents the number of features. Thus, an entire four-voice chorale can be represented as  $c \in \mathbb{Z}^{(T \times 4) \times F}$  and this flattened voices as (S, A, T, B)-repeated order was fed to the GRU model.

To extract metadata such as the number of sharp in the key signature and time signature, we used the music21 [29] library. Analyzing the major minor tonality was done using the Krumhansl-Schmuckler key-finding algorithm [30] in music21. For pitch representation, we adapted a sustain token for representing the same repetitive pitch without onset, following [2]. The selected features we used are described in Table 1. Besides the features in Table 1, we also considered the tonality(major or minor), num beat in 3/4 or 4/4 time, voice index of current time step, the recent previous MIDI pitch value of current voice, the number of time step current voice sustained, beat distance from last fermata, and remaining number of fermata. However, our preliminary ablation study showed that using Pitch, Fermata, Beat position, Beat strength, and Num sharp in key

**Table 1.** List of considered features for note encoding.

Feature	Description	Type
Pitch	MIDI pitch value of current time step	I & O
Fermata	1 when the fermata starts at current time step, 0 otherwise	I & O
Beat position	Beat position in sixteenth note grid (0 - 15 for 4/4)	I
Beat strength	Beat strength in sixteenth note grid	I
Num sharp in key	The number of [-flats/+sharps] in key signature.	I

(PFBBN) was most effective in modeling Bach-like music for our generation model. Therefore, we used PFBBN as a default encoding scheme of experiments in this paper.

It’s important to highlight that the dimensions of our input and output features differ in our study. For our input, we utilized all the features previously described. However, for our output feature,  $x_t$ , which the model is tasked with predicting, we only incorporated pitch and fermata. The features that aren’t predicted,  $c_t$  are initially fed into the model shifted to the left by one step so that  $x_{t-1}$  and  $c_t$  are concatenated together. This allows the model to anticipate the subsequent token  $x_t$  based on  $(x_0, x_1, \dots, x_{t-1}, c_0, c_1, \dots, c_t)$ . During the inference process,  $c_{t+1}$  was obtained using a rule-based approach. For the initial condition token  $c_0$ , we derived it from the pre-established distribution for each feature across the entire dataset.

### 3.3 Pitch Onset Loss

As we used note sustain as an independent token, we found that this sustain token appears 2.5 times more often than note onset, or change of pitch in the dataset. This can lead the model to predict sustain too frequently, as this single token occupies 70 % of entire pitch values. Therefore, we additionally imposed pitch onset loss, a pitch loss of a time step where onset exists, to enforce the model to focus more on the note onset and not hold the same pitch too much time during inference. The onset boolean can be represented as  $o \in \{0, 1\}^T$  for a single voice  $v \in \mathbb{Z}^{T \times F}$ , where  $o_t = 1$  if the voice has a note onset at time frame  $t$  and otherwise  $o_t = 0$ . The pitch onset loss  $L_{po}$  can be represented as an equation below.

$$L_{po} = \frac{1}{T} \sum_t o_t \cdot (-\log \hat{y}_t) \quad (2)$$

### 3.4 Loss Function Design According to Music Theory

**Parallel Prohibition Loss** We designed a loss function that penalizes parallel 5th and parallel 8th, which imitates one of the most marked rules for composing counterpoint. Even though we can also penalize concealed 5th and 8th along with the parallel, we only focus on the parallel error in this work. To force the model to avoid these prohibited patterns, our system calculates prohibition matrix  $\mathbf{Pr}$  for a given preceding voice  $v \in \mathbb{Z}^{T \times F}$  and the following voice  $w \in \mathbb{Z}^{T \times F}$  using a rule-based algorithm. The result can be denoted as  $\mathbf{Pr}_{v,w} \in \{0, 1\}^{T \times P}$ , where  $T$  and  $P$  represent the number of time frames and total note pitch in the vocabulary, respectively.

The prohibited pitches  $f(p, q, t)$  at time  $t$  for a sequence of MIDI pitch for voice,  $q \in \mathbb{N}^T$ , for a sequence of MIDI pitch for the preceding voice,  $p \in \mathbb{N}^T$ , can be represented as below:

$$f(p, q, t) = \begin{cases} q_{t-1} + (p_t - p_{t-1}) & \text{if } |p_{t-1} - q_{t-1}| \equiv 7 \text{ or } 0 \pmod{12} \\ & \text{and } p_{t-1} \neq p_t \text{ and } q_t \neq p_t - p_{t-1} + q_{t-1} \\ 0 & \text{else} \end{cases} \quad (3)$$

We intentionally did not prohibit the parallel progression that actually occurred in training set for two primary reasons. Firstly, there are instances where Bach himself did not adhere to the prohibition rule. Secondly, the log-likelihood loss and the prohibit loss directly conflict with each other. While the log-likelihood loss seeks to maximize the probability of a particular note, the prohibition loss aims to minimize the probability of that very same note. Therefore, we did not apply the prohibition rule in these cases.

Using  $f(p, q, t)$ , the piano-roll-like prohibition matrix  $\mathbf{Pr}_{v,w} \in \{0, 1\}^{T \times P}$  for the voice  $w$  and its preceding voice  $v$ , can be represented as Equation 4.

$$\mathbf{Pr}_{v,w}[n, t] = \begin{cases} 1 & \text{if } f(v, w, t) = n \\ 0 & \text{else} \end{cases} \quad (4)$$

The integrated  $\mathbf{Pr}_i$ , the prohibition matrix for  $i$ -th voice for every preceding voice, can be represented as Equation 5, where  $u_i$  represents a sequence of features for  $i$ -th voice. For example, if the voice order is soprano, alto, tenor, and bass,  $u_0$  is soprano, and  $u_3$  is bass.

$$\mathbf{Pr}_i = \sum_j^{i-1} \mathbf{Pr}_{u_j, u_i} \quad (5)$$

After the language model predicts the shifted events, we calculated the prohibition loss  $L_{phb}$ , the cross entropy loss between the predicted pitch token probabilities  $\hat{y} \in (0, 1)^{T \times P}$  and the prohibit matrix  $\mathbf{Pr}$ , which can be represented as an equation below:

$$L_{phb} = -\frac{1}{T} \sum_t \mathbf{Pr}[t] \cdot (\log(1 - \hat{y}_t^\alpha)) \quad (6)$$

where  $\alpha$  is a hyperparameter, which helps to preserve loss and gradient for small  $\hat{y}$ . In our experiment, we used  $\alpha = 0.5$ . Minimizing  $L_{phb}$  forces  $\hat{y}$  to be close to zero in the case of prohibited pitches. We have also tested to maximize  $-\log(\hat{y})$ , but this often results in unstable training since the gradient explodes around  $\log(0)$ .

Our final loss function is formulated as follows:

$$L_{total} = L_{LM} + \lambda_{phb} \cdot L_{phb} + \lambda_{po} \cdot L_{po} \quad (7)$$

where  $L_{LM}$  is Cross Entropy loss between predicted pitch and fermata tokens and target,  $L_{phb}$  and  $L_{po}$  are prohibited and pitch onset loss, and  $\lambda_{phb}$  and  $\lambda_{po}$  are weights for prohibit and pitch onset loss. We applied weight annealing for  $\lambda_{phb}$ , so that  $\lambda_{phb} = 0$  for the first 10% of iteration, and apply sigmoid annealing, so that the prohibition loss is gradually applied after the training becomes stable.

**Rule-based Parallel Masking** To compare the effectiveness of applying parallel prohibition loss, we also tested a rule-based parallel masking that avoids parallel progress during inference. Using a similar approach in Equation 3, we calculated the possible prohibited pitch for every step of the autoregressive inference. While we only prohibited parallel progression with exactly the same interval in the prohibited pitch during the training, we prohibited every possible pitch across the entire octave that makes the same 5th or 8th interval as a pitch class during the inference so that we could achieve zero parallel errors in the evaluation metric.

## 4 Experiments

In this section, we describe the experiment to investigate the effect of our suggested prohibit loss and pitch onset loss terms.

### 4.1 Dataset

For the train and validation dataset, we used 366 Chorales of Johann Sebastian Bach, which are provided in the format of Humdrum kern [31]. The data provides note information of each of the four voices, including the fermata symbol.

### 4.2 Experiment Setting

We split the dataset as 9:1 for the train and validation dataset. For the model, we used a 4-layer GRU model with a hidden size of 512 and a dropout rate of 0.2. For the hyperparameter, we used batch size 8, Adam optimizer with learning rate  $1e-3$ , and Step LR scheduler with step size 2k and gamma 0.8. Since the model normally converges within 30k steps, we used 30k steps to train the model. For the embedding size of used features, the default feature embedding size is 512 and all features have a feature dimension ratio of 0.1 of the default feature embedding size, which corresponds to 51. As the pitch feature is the most important feature, we used dimension ratio 0.75 for the pitch, which makes 384 dimensions. The embeddings from each feature are all concatenated, forming a total of 588 dimensions.

### 4.3 Evaluation Metric

Since we change loss weight with different values, total validation loss values are not directly comparable to evaluate the model performance. Therefore, for evaluation, we used the metric suggested by [5], which calculates Wasserstein distance of distribution of generated note, rhythm, parallel errors, harmonic quality, intervals of each voice (S, A, T, B), repeated sequence, and overall grade values compared to the Bach's original chorale dataset. To compare with the previous works, we evaluate our suggested model with JS Fake Chorale Dataset [6].

The original implementation of the metric [5] distinguishes enharmonic like  $C\sharp$  and  $D\flat$ , which are encoded in the same MIDI pitch. MIDI files generated from our proposed method or from JS Fake Chorale get severe distortion when converted by music21 in



the evaluation code. For example, MIDI pitch from 64 to 63 can be encoded either E4 to D#3 (minor second) or E4 to Eb4 (augmented first). If this interval is interpreted as E4 to Eb4 by music21, this makes a large error in Wasserstein distance because Bach’s original chorale corpus uses a lot of minor seconds but not augmented first. Therefore, we modified the code to use interval and note pitch classes in MIDI pitch, not distinguishing enharmonic notes. Note that we calculated the distribution of each feature in the Bach chorale corpus based on the dataset from [31], which is slightly different from the one used in [5].

**Table 2.** Experiment results for suggested losses. PE: parallel error, HQ: harmonic quality, B Intervals: bass intervals, RS: repeated sequence. Lower values mean better chorales. Here, the first row means using only PFBBN feature. **Bold** values are minimum values among our model conditions.

Conditions			Metrics						
$L_{phb}$	Masking	$L_{po}$	Note	Rhythm	PE	HQ	B Intervals	RS	Grade
×	×	×	0.30 (0.16)	0.22 (0.16)	0.94 (2.39)	0.62 (0.38)	0.42 (0.21)	1.38 (0.93)	4.77 (2.76)
✓	×	×	0.30 (0.17)	0.22 (0.15)	0.74 (1.77)	0.63 (0.41)	0.41 (0.23)	1.38 (0.91)	4.58 (2.28)
×	✓	×	0.31 (0.20)	0.23 (0.12)	<b>0.0 (0.0)</b>	0.65 (0.38)	0.54 (0.33)	1.52 (2.11)	<b>4.19 (2.47)</b>
✓	✓	×	0.36 (0.23)	0.21 (0.10)	<b>0.0 (0.0)</b>	0.69 (0.39)	0.68 (0.42)	1.40 (0.77)	4.25 (1.31)
×	×	0.2	<b>0.29 (0.16)</b>	0.21 (0.09)	1.00 (2.60)	0.61 (0.36)	0.41 (0.19)	1.30 (0.81)	4.72 (2.94)
×	×	0.5	0.31 (0.18)	0.21 (0.14)	0.67 (1.56)	0.61 (0.36)	<b>0.40 (0.19)</b>	1.33 (0.86)	4.42 (1.94)
×	×	1.0	0.30 (0.17)	0.20 (0.11)	0.76 (2.24)	<b>0.59 (0.37)</b>	<b>0.40 (0.19)</b>	<b>1.26 (0.61)</b>	4.40 (2.43)
×	×	2.0	0.31 (0.17)	0.21 (0.10)	0.69 (1.86)	0.61 (0.37)	0.40 (0.20)	1.33 (0.72)	4.46 (2.27)
✓	×	1.0	0.31 (0.18)	<b>0.20 (0.09)</b>	0.52 (1.3)	0.63 (0.42)	0.40 (0.20)	1.28 (0.63)	4.27 (1.61)
	Bach [31]		0.27 (0.15)	0.25 (0.16)	0.30 (0.88)	0.57 (0.32)	0.40 (0.21)	1.43 (0.92)	4.14 (1.60)
	JS Fake [6]		0.29 (0.14)	0.17 (0.07)	3.91 (4.01)	1.03 (0.77)	0.37 (0.16)	1.12 (0.40)	7.73 (4.43)

#### 4.4 Effect of Losses

To investigate whether the suggested loss terms are effective, we conducted an ablation study of prohibition loss and pitch onset loss. As we mentioned earlier, we selected PFBBN as the baseline to apply the loss. Since the voice intervals except bass (S, A, T) are not significantly different among the conditions, we omit the column in the result table 2.

For prohibition loss, we experimented with  $\lambda_{phb} = 1k$ , which yields the lowest parallel error in our preliminary experiment. Rule-based parallel masking was also compared with the condition using  $\lambda_{phb}$ . Similarly, we tested the effect of the pitch onset loss with  $\lambda_{po} = 0, 0.2, 0.5, 1.0, 2.0$ . The results are shown in Table 2. The result shows that parallel prohibition loss helped to reduce parallel errors but could not completely avoid them. This is also partially due to the fact that the training data itself does not perfectly exclude parallel motion. For the pitch onset loss experiment, we found that using  $\lambda_{po} = 1.0$  results in the best performance for most of the metrics. The combination of prohibition loss and the pitch onset loss showed the best performance, which is nearly similar to the metric of Bach’s original corpus.

Table 2 reveals that our rule-based masking method can effectively eliminate parallel errors during inference. However, this approach also resulted in a degradation of the metric for harmonic quality or bass intervals, as the model must sample lower-probability pitches to avoid producing parallel fifths. The most significant impact is observed in the Wasserstein distance of the bass interval, as the bass voice is influenced by three preceding voices, resulting in a more densely constrained inference process. Therefore, rule-based hard masking has to be carefully considered.

## 5 Conclusion

In this paper, we suggested a music theory-based novel loss term and applied it to the Bach chorale generation. Using the previously suggested quantitative evaluation metric, we showed that the model can generate chorales in quality that follow a similar distribution of musical characteristics as Bach’s corpus. We found that the suggested loss terms could improve the sample quality of generated chorale in terms of parallel errors, which was one of the main critical limitations of previous chorale generation models.

For further study, we will continue investigating the effect of voice order and pitch augmentation to improve the generated sample quality. Also, we can apply other well-known prohibitions such as concealed fifth and voice crossing into our prohibit loss term. Although our current work only studied hard prohibition (strict prohibition), soft prohibition is another topic of imposing the rule to the model.

## 6 Acknowledgment

This work was supported by the Year 2022 Culture Technology R&D Program by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Name: Research Talent Training Program for Emerging Technologies in Games, Project Number: R2020040211) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) (NRF-2022R1F1A1074566).

## References

1. Alcedo Coenen. David Cope, Experiments in Musical Intelligence. A-R Editions, Madison, Wisconsin, USA. Vol. 12 1996. *Organised Sound*, 2(1):57–60, 1997.
2. Gaëtan Hadjeres, François Pachet, and Frank Nielsen. DeepBach: A Steerable Model for Bach Chorales Generation. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1362–1371, 2017.
3. Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. Counterpoint by Convolution. In *Proceedings of 18th International Conference on Music Information Retrieval (ISMIR)*, pages 211–218, 2017.
4. Cheng-Zhi Anna Huang, Curtis Hawthorne, Adam Roberts, Monica Dinulescu, James Wexler, Leon Hong, and Jacob Howcroft. Approachable Music Composition with Machine Learning at Scale. In *Proceedings of the 20th International Conference on Music Information Retrieval (ISMIR)*, pages 793–800, 2019.

5. Alexander Fang, Alisa Liu, Prem Seetharaman, and Bryan Pardo. Bach or Mock? A Grading Function for Chorales in the Style of J.S Bach. In *Machine Learning for Media Discovery (MLAMD) Workshop at the International Conference on Machine Learning (ICML)*, 2020.
6. Omar Peracha. JS Fake Chorales: A Synthetic Dataset of Polyphonic Music with Human Annotation. *arXiv preprint arXiv:2107.10388*, 2021.
7. Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, volume 32, 2018.
8. Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music Transformer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.
9. Yu-Siang Huang and Yi-Hsuan Yang. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.
10. Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Muller, and Yi-Hsuan Yang. Theme Transformer: Symbolic Music Generation with Theme-conditioned Transformer. *IEEE Transactions on Multimedia*, 2022.
11. Sangjun Han, Hyeongrae Ihm, Moontae Lee, and Woohyung Lim. Symbolic Music Loop Generation with Neural Discrete Representations. In *Proceedings of 23rd International Conference on Music Information Retrieval (ISMIR)*, pages 403–410, 2022.
12. Yi Zou, Pei Zou, Yi Zhao, Kaixiang Zhang, Ran Zhang, and Xiaorui Wang. MELONS: Generating Melody with Long-term Structure using Transformers and Structure Graph. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 191–195, 2022.
13. Kejun Zhang, Xinda Wu, Tiejiao Zhang, Zhijie Huang, Xu Tan, Qihao Liang, Songruoyao Wu, and Lingyun Sun. Wuyun: Exploring Hierarchical Skeleton-guided Melody Generation using Knowledge-enhanced Deep Learning. *arXiv preprint arXiv:2301.04488*, 2023.
14. Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B Dannenberg. Controllable Deep Melody Generation via Hierarchical Music Structure Representation. In *Proceedings of 22nd International Conference on Music Information Retrieval (ISMIR)*, 2021.
15. Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music. *arXiv preprint arXiv:2005.00341*, 2020.
16. Ziyu Wang, Yiyi Zhang, Yixiao Zhang, Junyan Jiang, Ruihan Yang, Junbo Zhao, and Gus Xia. Pianotree VAE: Structured Representation Learning for Polyphonic Music. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.
17. Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. Transformer VAE: A Hierarchical Model for Structure-aware and Interpretable Music Representation Learning. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 516–520, 2020.
18. Ke Chen, Cheng-i Wang, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Music SketchNet: Controllable Music Generation via Factorized Representations of Pitch and Rhythm. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.
19. Ziyu Wang and Gus Xia. MuseBERT: Pre-training Music Representation for Music Understanding and Controllable Generation. In *Proceedings of 22nd International Conference on Music Information Retrieval (ISMIR)*, pages 722–729, 2021.
20. Hyun Lee, Taehyun Kim, Hyolim Kang, Minjoo Ki, Hyeonchan Hwang, Sharang Han, Seon Joo Kim, et al. ComMU: Dataset for Combinatorial Music Generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 39103–39114, 2022.

21. Yi-Wei Chen, Hung-Shin Lee, Yen-Hsing Chen, and Hsin-Min Wang. SurpriseNet: Melody Harmonization Conditioning on User-controlled Surprise Contours. In *Proceedings of 22nd International Conference on Music Information Retrieval (ISMIR)*, pages 105–112, 2021.
22. Halley Young, Vincent Dumoulin, Pablo S Castro, Jesse Engel, and Cheng-Zhi Anna Huang. Compositional Steering of Music Transformers. 2022.
23. Hao Hao Tan and Dorien Herremans. Music Fadernets: Controllable Music Generation based on High-level Features via Low-level Feature Modelling. In *Proceedings of 19th International Conference on Music Information Retrieval (ISMIR)*, 2020.
24. Ashis Pati and Alexander Lerch. Is Disentanglement Enough? On Latent Representations for Controllable Music Generation. In *Proceedings of 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
25. Kyoyun Choi, Jonggwon Park, Wan Heo, Sungwook Jeon, and Jonghun Park. Chord Conditioned Melody Generation with Transformer based Decoders. *IEEE Access*, 9:42071–42080, 2021.
26. Seungyeon Rhyu, Hyeonseok Choi, Sarah Kim, and Kyogu Lee. Translating Melody to Chord: Structured and Flexible Harmonization of Melody with Transformer. *IEEE Access*, 10:28261–28273, 2022.
27. Feynman T Liang, Mark Gotham, Matthew Johnson, and Jamie Shotton. Automatic Stylistic Composition of Bach Chorales with Deep LSTM. In *Proceedings of 18th International Conference on Music Information Retrieval (ISMIR)*, pages 449–456, 2017.
28. Omar Peracha. Improving Polyphonic Music Models with Feature-rich Encoding. In *Proceedings of 21st International Conference on Music Information Retrieval (ISMIR)*, 2020.
29. Michael Scott Cuthbert and Christopher Ariza. music21: A Toolkit for Computer-aided Musicology and Symbolic Music Data. In *Proceedings of 11th International Conference on Music Information Retrieval (ISMIR)*, pages 637–642, 2010.
30. David Temperley. What’s Key for Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. *Music Perception*, 17(1):65–100, 10 1999.
31. Nathaniel Condit-Schultz, Yaolong Ju, and Ichiro Fujinaga. A Flexible Approach to Automated Harmonic Analysis: Multiple Annotations of Chorales by Bach and Pr torius. In *Proceedings of 19th International Conference on Music Information Retrieval (ISMIR)*, pages 66–73, 2018.

# DiffVel: Note-Level MIDI Velocity Estimation for Piano Performance by A Double Conditioned Diffusion Model

Hyon Kim<sup>1</sup> and Xavier Serra<sup>1</sup> \*

Music Technology Group, Universitat Pompeu Fabra  
hyon.kim@upf.edu, xavier.serra@upf.edu

**Abstract.** In any piano performance, expressiveness is paramount for effectively conveying the intent of the performer, and one of the most significant aspects of expressiveness is the loudness at the individual key or note level. However, accurately detecting note-level loudness poses a considerable technical challenge due to the polyphonic nature of piano performances, wherein multiple notes are played simultaneously, as well as the similarity of harmonic elements. MIDI velocity is crucial for indicating loudness in piano notes. This study conducted experiments for estimating a note-level MIDI velocity expanding the DiffRoll model: the Diffusion Model for piano performance transcription. By adopting double conditioning—audio and score information—and implementing noise removal as a post-processing, our findings highlight the model’s potential in estimating note level MIDI velocity.

**Keywords:** MIDI Velocity Estimation, Diffusion Model, Conditioned Deep Neural Network, FiLM Conditioning

## 1 Introduction

The assessment of piano performance can be attributed to three key factors, namely loudness, rhythm, and accurate key strokes [1]. Owing to the polyphonic nature of piano performances, multiple auditory streams coexist, such as melody line and accompaniment. This intricate aspect allows for enhanced distinguishability in the interpretations of expert pianists [2]. The expressiveness of a musical piece is significantly influenced by the series of loudness values associated with each note in the score, which contribute to the dynamic alterations throughout the composition [3].

Within the realm of music education, research has demonstrated the effectiveness of utilizing visual feedback in enhancing students’ abilities [4, 5]. In this regard, the comprehension and management of loudness become especially significant [1] when

\* This research was carried out under the project Musical AI - PID2019- 111403GB-I00/AEI/10.13039/501100011033, funded by the Spanish Ministerio de Ciencia e Innovación and the Agencia Estatal de Investigación.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

it is visualised. The employment of techniques for estimating and visualizing loudness fulfills the crucial system requirement, enabling the provision of valuable feedback to learners.

[8,9,12] researched mapping from perceptual loudness value in dB scale to dynamic symbols for piano performance such as *forte*, *mezzo forte*, *piano*, *pianissimo*, *crescendo*, etc. Note that using MIDI velocity, we predict loudness at a lower granularity, i.e. finer scale than the dynamic markings which are explicitly written in most of music scores and indicate how loud the piece should be played. Furthermore, each note in a piano performance may have a different loudness depending on the texture of the music [11, 14]. Therefore, the note-level loudness itself has special meaning in piano performance, considering its polyphonic characteristics.

To prevent ambiguity, we use the term "loudness" to denote the combined MIDI velocities within a specific time frame as measured by an electronic piano device. On the other hand, "intensity" refers to the maximum value of the frequency sum for a note frame, as defined in [15]. It is essential to recognize that MIDI velocity does not have a direct correlation with loudness as experienced by the human auditory system. Studies have been conducted to explore the relationship between MIDI velocity and loudness measured in decibels (dB) [30]. While the research demonstrates a consistent increase in perceived loudness (in dB) with increasing MIDI velocity, it also reveals that this relationship is non-linear [13].

Since this research aims to detect MIDI velocity on each note performed, automatic piano performance transcription is a closely related area for this purpose. Piano performance transcription is also an actively researched topic [10, 23, 24]. However, these studies primarily focus on detecting individual notes, rather than note loudness or dynamic symbols in a score. Additionally, the transcription process is not yet fully accurate and reproducible of performance.

Several studies have explored the note-level loudness estimation task [6, 15–18]. These researchers employed NMF and DNN methods to isolate piano performance audio into 88 distinct keys and estimated MIDI velocity or intensity for each note. We consider this area of research as an application of Automatic Music Transcription (AMT) and to be applied to expressiveness performance modeling. The piano note-level MIDI velocity estimation task involves solving a two-fold problem. One aspect is a regression problem, requiring the estimation of numbers within the 0-127 range for MIDI velocity. The other issue is audio classification, which involves sorting audio into each piano key, typically consisting of 88 keys. To address these challenges, we propose the DiffVel as an expansion of DiffRoll [7], a diffusion model for AMT, and Feature-wise Linear Modulation (FiLM) conditioning layers [20] to incorporate score information into the DNN. We conducted experiments to estimate the MIDI velocity using this approach.

## **2 Related Work**

### **2.1 Automatic Music Transcription**

The piano performance transcription is one of the closest problems for classification from audio input. [29] proposed a CNN-GRU combined acoustic model which branches

into four outputs: velocity regression, onset, offset, and note frame estimation. The note frame estimation is the final goal of this model and the other three estimations are gathered as input to another acoustic model to estimate the notes at the frame level. Therefore, the estimated MIDI velocity regression is not evaluated in the paper since it is out of the scope.

Recently, diffusion models have been explored as an alternative approach. DiffWave, a state-of-the-art generative model for audio synthesis, leverages the diffusion probabilistic framework and exhibits remarkable capabilities in generating high-quality audio samples from various sources. The core idea behind DiffWave involves employing a series of de-noising score matching steps, iteratively refining the generated audio samples to achieve accurate and precise output. Building upon DiffWave, DiffRoll has been researched [7]. DiffRoll expands DiffWave into a two-dimensional representation of sound and output, taking Mel Spectrogram as a condition and forming the two-dimensional Gaussian noise input into MIDI roll. The generative model’s characteristics offer considerable potential to simultaneously address classification and regression problems by tuning conditions. Exploring conditions with not only one but also multiple conditions would contribute to estimating MIDI velocity more accurately. However, the model disregards velocity estimation in the model evaluation.

For conditioning, existing research utilizes score information to inform musical instrument separation in polyphonic music [19, 21, 28]. These works employ score or video information to enhance source separation results by creating an additional neural network to extract features from the supplementary data, which are then fed into the original DNN.

## 2.2 Feature-wise Linear Modulation (FiLM)

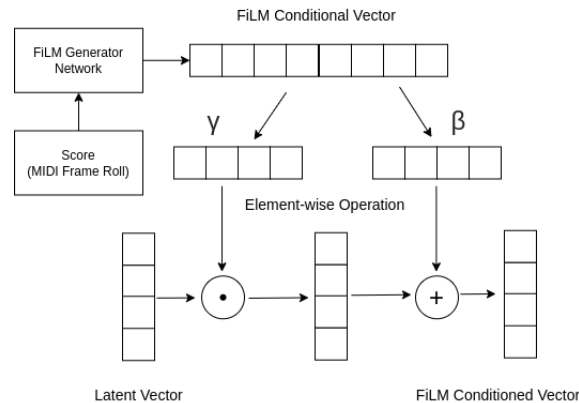
In this paper, we utilized the FiLM conditioning [20] to insert score information in order to estimate note-level MIDI velocity for piano performance. FiLM conditioning is used in the image processing area and has gained improved results on object detection [20]. In previous research, natural language is used as an external condition to indicate the existence of target objects to be detected. This idea has been applied to audio source separation tasks by conditioning audio with video and score information [28].

The FiLM comprises a set of neural network layers that generate an affine transformation for a given input layer in a neural network. It consists of a base DNN which is trained in a supervised fashion and a condition generator which takes conditions such as score as input and generates  $\beta$  and  $\gamma$  to make an element-wise affine transformation in the latent space of the base DNN. In the math formula, it is described as follows;

$$FiLM(x) = \gamma(z) \cdot x + \beta(z) \quad (1)$$

where vector  $z$  is a conditional vector.

The Figure 1 shows the architecture of FiLM conditioning. This condition embedding model generates parameters,  $\beta$  and  $\gamma$ , to make an affine transformation on the latent vector  $x$  from the base DNN.



**Fig. 1.** The diagram illustrates the operation flow for inserting a FiLM condition into a latent vector

### 2.3 Note Level MIDI Velocity Estimation with Score Information

Only three papers have considered note-level MIDI velocity in music performance, employing NMF [15, 18] and DNN methods [6]. NMF methods have been used for source separation problems and effectively applied to music source separation as well [22]. [15] examined an NMF method with score information to estimate note-level intensity before creating a linear regression model to obtain note-level MIDI velocity estimation. This research provided a detailed analysis of NMF method errors and their causes. The DNN method attempted to address the estimation problem by applying the AMT method and score conditioning. The DNN architecture involves stacking convolution blocks and GRU block and inserting a FiLM conditioning generated by a fully connected linear layer. Although it did not surpass the results of the NMF method, it was the first attempt to estimate MIDI velocity using a DNN method and to generalize the model for unseen classical music inputs, as opposed to the NMF method which optimizes parameters for each test data. [16] aimed to estimate the note level intensity, rather than MIDI velocity, from the spectrogram by filtering it according to the frequency of each note.

In our study, we compare our results with the NMF method proposed in [15] and the DNN method [6] as our benchmark.

## 3 Method

There are two models experimented in this study: the diffusion models with and without score information by FiLM conditioning. The entire architecture is based on the DiffRoll model and the conditions, Mel Spectrogram and score, are inserted as an expansion. We used the MIDI velocity data on note frame level as supervised data for training for both models. Score information is represented in a note frame roll in the MIDI roll.



For the training data, we used the Maestro dataset [26]. The data segmentation is 20 seconds, and the number of data frames is 31 in one second. Therefore, each output from the models is a (620, 88) matrix containing onset, offset, and velocity information.

### 3.1 Model Architecture

The simplified overall model architecture is illustrated in Figure 2. In the diffusion model, each residual layer takes the conditions. The Mel Spectrogram transformed from input audio is added as another condition to each residual layer before the FiLM conditional vector insertion.

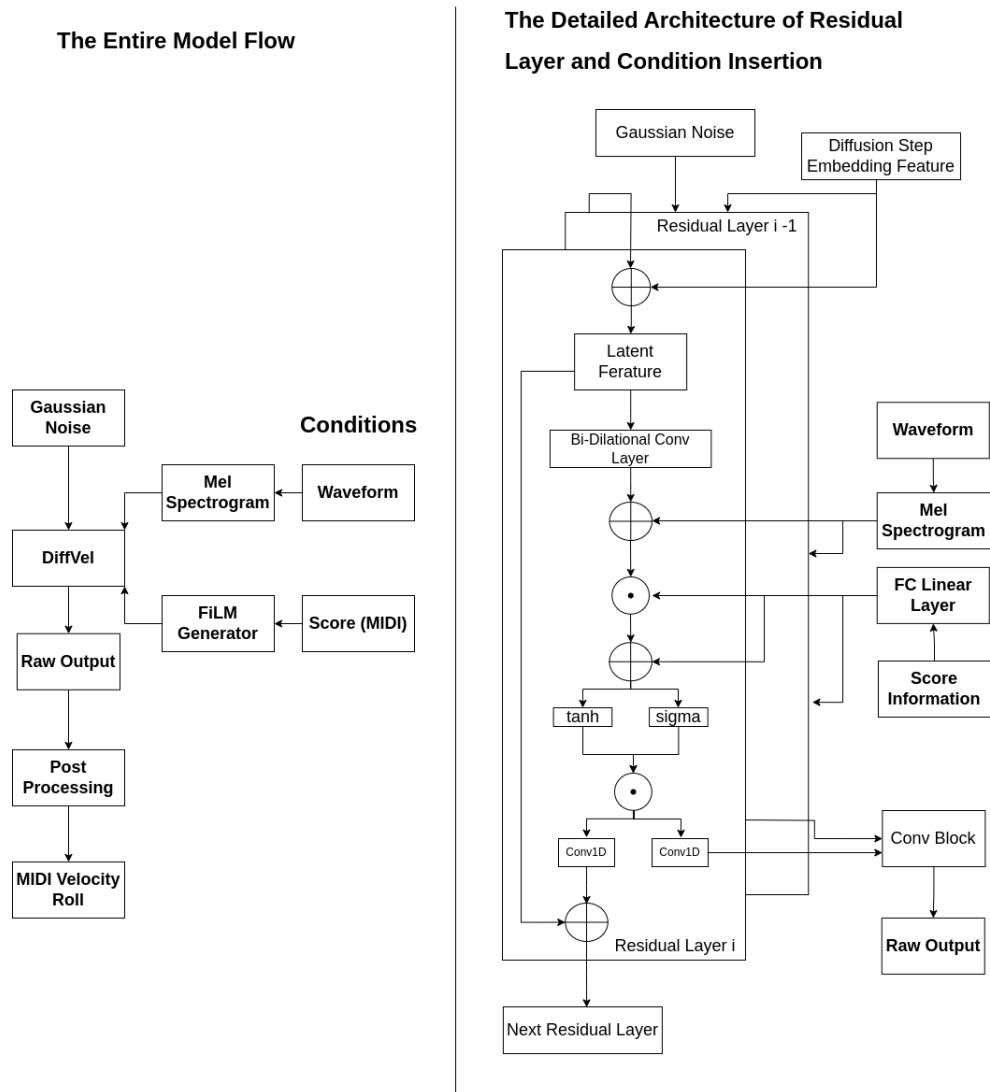
For the purpose of inserting the score information, we also added a FiLM conditioning layer as it is introduced in Section 2. We have tested the element-wise operations for multiplication and addition. However, the scalar multiplication and addition gave us better results. The FiLM generator is designed as a fully connected layer to generate conditioning parameters, and it is inserted after Mel Spectrum conditioning in each residual layer, i.e. the generated conditional vectors are sliced for each residual layer for the affine transformation.

For the parameter setup for DiffVel, the original setup is employed from DiffRoll: 15 residual layers, sampling rate 16000Hz, the hop size 512, the drop rate 0 to be fully supervised learning fashion, and the convolutional kernel size 9 for the residual layers. The loss function is L2 (mean square error) loss for the entire data segment, not note-level MIDI velocity error. We have tried Binary Cross Entropy (BCE) for better classification and L1 loss for MIDI velocity estimation. However, they did not work well in this diffusion model setup. Due to the limitation of computational resources, the epoch is stopped at 2000 for each training.

In the task of MIDI velocity estimation, which aims to get a number as value from the output, dealing with the input Gaussian noise is crucial. When the Gaussian noise is generated, it has a mean value of 0 and variance of 1. The diffusion step to denoise the Gaussian noise is set to 200 steps. However this noise is not perfectly removed after the diffusion steps and we need to perform denoising to each output by the post-processing.

During the post-processing, Gaussian noise removal is performed, which remained after the diffusion steps. This remainder causes a problem that it is considered as velocity during the evaluation process and causes 100% of the recall score and its note level estimation error is calculated high since the error is calculated where the note is not actually detected by the model. In this research, velocity estimation evaluation is made only on correctly detected notes. This evaluation constraint is applied to the other two models to be compared, the DNN [6] and NMF [15] models.

In order to remove the remaining noise, three methods are considered; one is to increase the diffusion steps since each step of diffusion step reduces the Gaussian noise. The second way is using a post-processing method employed by SegDiff, which averages the output from multiple inferences [25], at the expense of computational resources. However, these methods were not chosen due to the limitation of computational resources. In the process of removing the remaining Gaussian noise in the output, we calculated the distribution where the note does not exist in the ground truth score and defined a threshold to set output value 0. More precisely, a right  $Z$ -score is set based



**Fig. 2.** The simplified overall process (right) and the detailed condition insertion into each residual layer (left)

on the distribution in order to find a threshold to set the output value to 0.  $Z$ -score is a number derived by following equation;

$$Z = \frac{x - \mu}{\sigma} \quad (2)$$

where  $x$  is observed value,  $\mu$  and  $\sigma$  are mean and standard deviation of all output values for each score.

The reason why the right  $Z$ -scores is chosen, rather than taking the highest value in the area where the note does not exist in the ground truth, is that there are wrongly detected extra notes which have proper values to represent a MIDI velocity. These values are considered above the  $Z$ -score in the distribution of the remaining noise and do not affect the correctly detected values during the post-processing by not setting them to 0.

After removing the remaining Gaussian noise, we normalized the output to be in the range  $[0, 1]$  looking at entire output value of each excerpt, not just for each output, and then scaled back to  $[0, 127]$ .

### 3.2 Evaluation

For testing purposes, we used the Saarland Music Data (SMD) dataset [27], which is also used for testing in previous researches [6, 15]. The dataset consists of students' piano performances, both audio data and MIDI data, which are perfectly aligned. The amount of data includes 50 classical piano excerpts, performed on Yamaha Disklavier. The original sampling frequency is 44.1kHz and down-sampled to 16kHz. We chose 49 excerpts from this dataset which are used in the score-informed NMF method by [15].

The model evaluation is made by taking an  $L1$  distance of MIDI velocities for each note between ground truth and inference by the models, similarly to the previous research [15].

$$Error = \frac{\sum_i |V(i)_{\text{ground truth}} - V(i)_{\text{inference}}|}{N} \quad (3)$$

where  $i$  is each note and  $N$  is the number of correctly detected notes in the score.

The inferred MIDI velocity is the maximum value within the interval of each detected and classified velocity frame against the ground truth velocity frame for each note. This is because the detected velocity tends to fade after having the maximum value in the estimated MIDI velocity in a note frame as if depicting attack and fades of loudness of each note.

To evaluate the classification accuracy, recall score is chosen as the evaluation metric. This is because the estimation is masked by the given score, and recall is considered as the most appropriate evaluation metric for this classification problem when score is informed, as it takes into account both true positive and false negative. It measures the proportion of the total actual positive cases that are correctly identified by the classifier.

In this study, only correctly detected notes are evaluated, since we separate the MIDI velocity estimation accuracy and note detection accuracy as different research problem statements; the AMT and the MIDI velocity estimation as mentioned in the Section 2.

## 4 Results and Analysis

As we can see from the Table 1, FiLM conditioning to incorporate score information helped the estimation accuracy among the two models we have tested. The results show that FiLM conditioning improved MIDI velocity estimation but did not help with note detection for any setup. The result represents all note-wise errors inferred on the SMD dataset.

In terms of Gaussian noise removal, the right  $Z$ -score = 3 improved the overall accuracy significantly by sorting output values to correctly detected notes and the remaining noise after the diffusion steps. When post-processing is not performed and the noise remains, the evaluation method considers MIDI velocity detected and recall score is always 100%.

$Z$ -score	Single Conditioning			Double Conditioning with Score		
	Mean	SD	Recall	Mean	SD	Recall
Raw Output	32.8	20.5	100%	28.6	19.2	100%
1	24.7	16.7	60%	21.0	14.5	56%
2	24.0	16.1	58%	20.2	13.6	54%
3	23.7	15.8	56%	<b>19.7</b>	<b>13.1</b>	53%

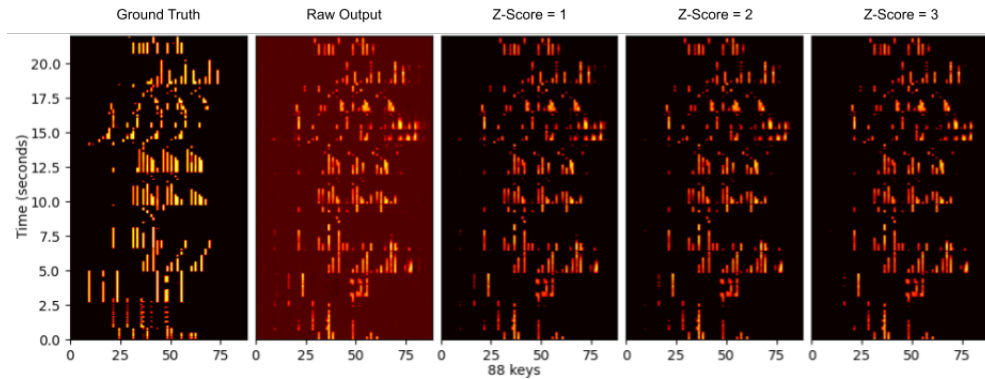
**Table 1.** The mean and standard deviation (SD) of the MIDI velocity estimation error for the models are based on  $Z$ -score for noise removal

The Figure 4 shows an example of the remaining Gaussian noise removal. The pale red color shown in the raw output is the remaining Gaussian noise from the input to the model, and setting  $Z$ -score determines the threshold to set the value to zero, attempting not to touch the detected notes. It can be intuitively seen that the remaining noise is removed without changing the value of the detected note velocities based on  $Z$ -score values.

Proposed Model		Conv-FiLM with Score [6]		NMF with Score [15]	
Mean	SD	Mean	SD	Mean	SD
19.7	13.1	15.1	12.3	4.1	5

**Table 2.** The comparison of results for the proposed model and previous research

We also compared the results to the previous models that have the same setup: a score-informed MIDI velocity estimation task. The Table 2 displays the mean and standard deviation (SD) values for proposed and previously researched models. The proposed model exhibited a mean value of 19.7 and an SD of 13.1, indicating the poorest performance among the three methods. In contrast, the Conv-FiLM DNN with Score



**Fig. 3.** The visualization and comparison of Gaussian noise removal for raw output and after removal are based on  $Z$ -score = 1, 2 and 3.

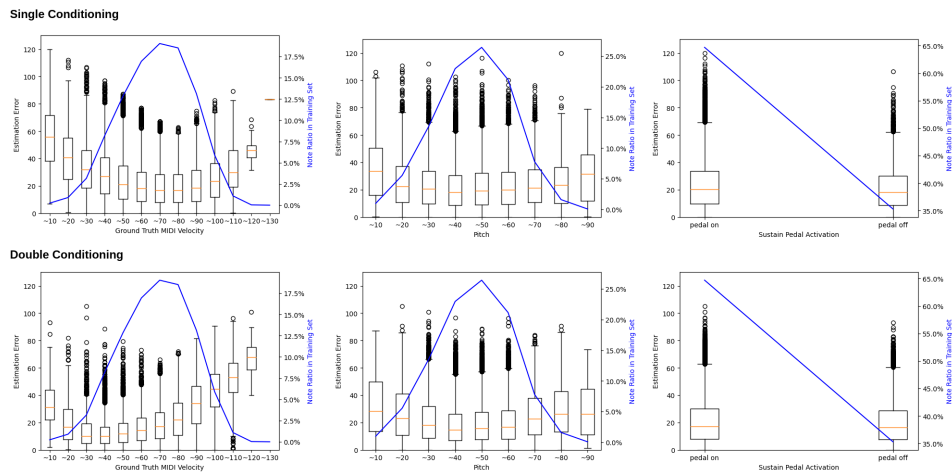
approach achieved a mean of 15.1 and an SD of 12.3, while the NMF with score method demonstrated the best performance with the lowest mean and SD values, at 4.1 and 5, respectively. Although the proposed model currently underperforms compared to the other models, it is important to note that the difference between their mean values is not substantial and we do not know significance in the sense of perceptual loudness yet.

Figure 4 displays the deviation of the error in each range of MIDI velocity, pitch, and sustain pedal activation respectively for both models. The box-charts for pitch and sustain pedal are similar figure for both models. These charts demonstrate that the more training data notes you have, the more accurate your MIDI velocity estimation will be, looking at the note ratio in the training dataset. This implies that data augmentation, such as pitch shifting, is necessary for low and high pitch notes in the training data. When looking at the error based on the MIDI velocity group, it is interesting to observe that FiLM conditioning improved the model’s estimation for lower velocity notes, but resulted in worse estimation for higher velocity notes compared to the model without score information. It was also observed that both models tend to estimate MIDI velocity lower than the ground truth. Further analysis is required to interpret this phenomenon.

## 5 Discussion and Future Work

In this study, experiments on a diffusion model with double conditions for note level MIDI velocity estimation for piano performance have been conducted. We discovered that FiLM conditioning for score information insertion improved the estimation error and standard deviation on the overall test data.

We need to investigate the how the MIDI velocity error gives us the human perceptual sense to give the true evaluation of the model. As is mentioned in the introduction, there is still no research has been conducted for creating mapping from MIDI velocity to perceptual loudness. This will be one of our future works to keep this research move forward.



**Fig. 4.** The error analysis is based on ground truth MIDI velocity, pitch and sustain pedal activation. The box charts in the upper row display results for the single-conditioned model. Similarly, the box charts in the lower row show the results of the model with score information.

One of the downside of the models is they take significant amount of time and computational cost for an inference and model convergence in training phase. In this study, for example, it took about 2.5 minutes for 20 seconds of MIDI velocity roll output. This problem would be a blocker for a use case which requires real time processing.

The model achieved a similar result to the DNN used in a previous study [6], which indicates that the direction of this research is promising, and further exploration is warranted. Due to computational limitations, the training was stopped at 2000 epochs. However, the losses on validation set are still showing the trend of decrease on each model. This indicates that further training could improve accuracy within a short period of time with high confidence. Moreover, the recent rapid development and evolution on generative models including the diffusion model will improve its transcription accuracy, and at the same time, it would lead more attention to the FiLM conditioning to realize multi-modality for certain use-case scenarios such as education purposes which needs score and audio information.

Since it has been observed that FiLM conditioning improves the estimation results, further investigation into the condition generator is necessary for better estimation and note detection, rather than a simple fully connected linear layer. Moreover, the proposed diffusion model is adaptable to multiple conditioning techniques, making feature engineering a particularly suitable strategy for optimization within the DiffVel setup. By refining the features used in the model, it may be possible to extract more meaningful patterns and relationships from the data, ultimately leading to improved results. Additionally, incorporating more data and extending the training process could potentially enhance the proposed model's performance. Therefore, future research can focus on these aspects to optimize the proposed model and potentially achieve better performance than the existing approaches.

In real-world use cases, such as music education, score alignment must be taken into account for conditioning. A Dynamic Time Warping will be used to address this issue in a future work.

The code and the dataset used for this research would be provided upon request.

## References

1. Kim, Hyon and Ramoneda, Pedro and Miron, Marius and Serra : An overview of automatic piano performance assessment within the music education context, Xavier : 2022 : SCITEPRESS–Science and Technology Publications
2. Federico Simonetta, Federico Avanzini, Stavros Ntalampiras : A Perceptual Measure for Evaluating the Resynthesis of Automatic Music Transcriptions : arXiv:2202.12257 [cs.SD]
3. Grachten, Maarten and Widmer, Gerhard : Linear Basis Models for Prediction and Analysis of Musical Expression : Journal of New Music Research, volume 41, number 4, pages 311–322, 2012
4. Hamond, Luciana Fernandes and Welch, Graham and Himonides, Evangelos : The pedagogical use of visual feedback for enhancing dynamics in higher education piano learning and performance : Opus, 25, 3, pages 581–601 year 2019
5. Hamond, Luciana Fernandes: The pedagogical use of technology-mediated feedback in a higher education piano studio: an exploratory action case study : 2017 UCL (University College London)
6. H. Kim, M. Miron, X. Serra : Score-Informed MIDI Velocity Estimation for Piano Performance by FiLM Conditioning : Proc. Int. Conf. Sound and Music Computing, 2023
7. Kin Wai Cheuk, Ryosuke Sawata, Toshimitsu Uesaka, Naoki Murata, Naoya Takahashi, Shusuke Takahashi, Dorien Herremans, Yuki Mitsufuji : DiffRoll:Diffusion-based Generative Music Transcription with Unsupervised Pretraining Capability : arXiv:2210.05148 [cs.SD]
8. Kosta, Katerina and Ramírez, Rafael and Bandtlow, Oscar F and Chew, Elaine : Mapping between dynamic markings and performed loudness: a machine learning approach : Journal of Mathematics and Music, volume 10, number 2 pages 149–172, 2016, Taylor & Francis
9. Kosta, K., O. F. Bandtlow, E. Chew : Outliers in Performed Loudness Transitions: An Analysis of Chopin Mazurka Recordings. : International Conference for Music Perception and Cognition (ICMPC), pages 601-604, 2016, California, USA
10. Benetos, Emmanouil and Dixon, Simon and Giannoulis, Dimitrios and Kirchhoff, Holger and Klapuri, Anssi : Automatic music transcription: challenges and future directions : Journal of Intelligent Information Systems, volume 41, page 407–434, 2013, Springer
11. Sarah Kim and Jeong Mi Park and Seungyeon Rhyu and Juhan Nam and Kyogu Lee : Quantitative analysis of piano performance proficiency focusing on difference between hands, PLoS ONE volume 16, 2021
12. Katerina Kosta and Oscar F. Bandtlow and Elaine Chew : Dynamics and relativity: Practical implications of dynamic markings in the score : Journal of New Music Research, volume 47, number 5, pages 438-461, 2018, Routledge : <https://doi.org/10.1080/09298215.2018.1486430>
13. Qu, Yang and Qin, Yutian and Chao, Lecheng and Qian, Hangkai and Wang, Ziyu and Xia, Gus : Modeling Perceptual Loudness of Piano Tone: Theory and Applications : arXiv preprint arXiv:2209.10674
14. Goebel, W. : Melody lead in piano performance: expressive device or artifact? : The Journal of the Acoustical Society of America, volume 110, number 1, pages 563-72, 2001, Acoustical Society of America

15. Jeong, Dasaem and Kwon, Taegyun and Nam, Juhan : Note-Intensity Estimation of Piano Recordings Using Coarsely Aligned MIDI Score, volume 68, pages 34–47, number 1, Journal of the Audio Engineering Society, JAES, Audio Engineering Society
16. Ewert, Sebastian and Müller, Meinard : Estimating note intensities in music recordings : 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 385–388, 2011, IEEE
17. Devaney, Johanna and Mandel, Michael : An evaluation of score-informed methods for estimating fundamental frequency and power from polyphonic audio : 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 181–185
18. Jeong, Dasaem and Nam, Juhan : Note intensity estimation of piano recordings by score-informed NMF : Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio, 2017, Audio Engineering Society
19. Manilow, Ethan and Pardo, Bryan : Bespoke neural networks for score-informed source separation : arXiv preprint arXiv:2009.13729, 2020
20. Perez, Ethan and Strub, Florian and de Vries, Harm and Dumoulin, Vincent and Courville, Aaron : FiLM: Visual Reasoning with a General Conditioning Layer : <http://arxiv.org/abs/1709.07871>,
21. Meseguer-Brocal, Gabriel and Peeters, Geoffroy : Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations : arXiv preprint arXiv:1907.01277, 2019
22. Miron, Marius and Carabias Orti, Julio J and Janer Mestres, Jordi : Improving score-informed source separation for classical music through note refinement : Proceedings of the 16th International Society for Music Information Retrieval (ISMIR) Conference; 2015 Oct 26-30; Málaga, Spain. Canada: International Society for Music Information Retrieval; 2015.
23. Kim, Jong Wook and Bello, Juan Pablo : Adversarial learning for improved onsets and frames music transcription : arXiv preprint arXiv:1906.08512, 2019
24. Kelz, Rainer and Dorfer, Matthias and Korzeniowski, Filip and Böck, Sebastian and Arzt, Andreas and Widmer, Gerhard : On the Potential of Simple Framewise Approaches to Piano Transcription : arXiv:1612.05153 [cs]
25. Tomer Amit, Tal Shaharbany, Eliya Nachmani, Lior Wolf : SegDiff: Image Segmentation with Diffusion Probabilistic Models : arXiv:2112.00390 [cs.CV]
26. Curtis Hawthorne and Andriy Stasyuk and Adam Roberts and Ian Simon and Cheng-Zhi Anna Huang and Sander Dieleman and Erich Elsen and Jesse Engel and Douglas Eck : Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset : International Conference on Learning Representations, 2019
27. Müller, Meinard and Konz, Verena and Bogler, Wolfgang and Arifi-Müller, Vlora : Saarland music data (SMD), 2011
28. Slizovskaia, Olga and Haro, Gloria and Gómez, Emilia : Conditioned source separation for musical instrument performances : IEEE/ACM Transactions on Audio, Speech, and Language Processing : volume 29, pages 2083–2095, 2021,
29. Kong, Qiuqiang and Li, Bochen and Song, Xuchen and Wan, Yuan and Wang, Yuxuan : High-resolution piano transcription with pedals by regressing onset and offset times : IEEE/ACM Transactions on Audio, Speech, and Language Processing : volume 29, pages 3707–3717, 2021
30. Roger B. Dannenberg : The Interpretation of MIDI Velocity : International Conference on Mathematics and Computing



## 8+8=4: Formalizing Time Units to Handle Symbolic Music Durations

Emmanouil Karystinaios<sup>1</sup>, Francesco Foscarin<sup>1</sup>, Florent Jacquemard<sup>2</sup>, Masahiko Sakai<sup>3</sup>, Satoshi Tojo<sup>4</sup>, and Gerhard Widmer<sup>1,5</sup>

<sup>1</sup> Institute of Computational Perception, Johannes Kepler University Linz, Austria

<sup>2</sup> CNAM Paris, France

<sup>3</sup> Nagoya University, Japan

<sup>4</sup> Asia University, Japan

<sup>5</sup> LIT AI Lab, Linz Institute of Technology, Austria

emmanouil.karystinaios@jku.at

francesco.foscarin@jku.at

**Abstract.** This paper focuses on the nominal durations of musical events (notes and rests) in a symbolic musical score, and on how to conveniently handle these in computer applications. We propose the usage of a temporal unit that is directly related to the graphical symbols in musical scores, and pair this with a set of operations that cover typical computations in music applications. We formalise this time unit and the more commonly used approach in a single mathematical framework, as semirings, algebraic structures that enable an abstract description of algorithms / processing pipelines. We then discuss some practical use cases and highlight when our system can improve such pipelines by making them more efficient in terms of data type used and the number of computations.

**Keywords:** symbolic music; musical score; duration encoding.

### 1 Introduction

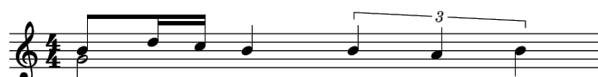
In a musical score, the duration of musical events (i.e., notes and rests) is defined by a finite set of symbols, and their temporal position by summing the duration of the previous musical events. Computer applications that deal with musical scores typically work with *Relative Symbolic Duration* (RSD) units, i.e., they choose a reference note duration and model all temporal information as ratios of that reference. For example, for the first four notes of the upper voice in Figure 1, one can choose a quarter note ♩ as a reference and represent the durations in the first two beats as the sequence  $[0.5, 0.25, 0.25, 1]$ . This kind of encoding shows its limits for certain durations, typically those produced by irregular groupings (also called tuplets). The 5th note in the top voice in the figure would have a duration of  $2/3$ , which is a periodic number not representable as a floating point value in computer applications, thus requiring a truncation. This introduces an



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

error that propagates to all subsequent musical events and creates a number of problems for applications that require exact matching of temporal positions.

Two main approaches have been proposed to solve this problem. The first is the *fraction approach*, implemented, for example, by the Python library Music21 [CA10]. It involves the representation of durations with specific Python objects made to mimic a fraction. This eliminates the rounding problems, but the fraction object is inefficient to handle with respect to native Python types and is not supported by libraries for heavy computations such as Numba, Pytorch, or TensorFlow. The second method, the *common divisor approach*, consists of setting the aforementioned reference duration to a value that is a common divisor of all durations appearing in a given piece or set of pieces. All temporal information can then be expressed with natural numbers, enabling very efficient computations. This solution is adopted by the Python library Partitura [Can+22], in some musical score storage formats such as MIDI, MEI, MusicXML, and in other computer music frameworks (e.g., [Fos+19]). However, this solution is still problematic for real-time scenarios when we do not know all duration in advance or when the piece can be modified. When a new duration is added that is not a multiple of the reference, the reference must be recomputed and all values updated.



**Fig. 1.** A musical score example with two problematic configurations: a tuplet and an incomplete bottom voice.

*Example 1.* Let us consider a toy application on the score of Figure 1. We are interested in importing it from an MEI file, splitting the third note (the C) in the top voice into two, and producing a pianoroll representation. The notes in the top voice have durations, in RSD units (with a quarter note as reference), of  $[\frac{1}{2}, \frac{1}{4}, \frac{1}{4}, 1, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}]$ . In the common division approach, we first need to compute as reference value  $\delta$  a common divisor of all absolute note durations, the largest (i.e., the greatest common divisor GCD) if we want to optimize memory usage. In this case, this is  $\frac{1}{12}$  of a quarter note. We then express each duration as a multiple of the reference, i.e.,  $[6, 3, 3, 12, 8, 8, 8]$ . If we want to split the third note, we need to recalculate the value of  $\delta$  as  $1/24$ , update the durations to  $[12, 6, 6, 24, 16, 16, 16]$ , and finally split the third note in two notes with duration 3. We can produce the pianorolls of the two voices independently and then perform an element-wise sum to obtain the score pianoroll. However, the second voice is logically incomplete in the score, missing an explicit half-note rest.<sup>6</sup> Thus, we first need to compute the maximum between the total duration of the two voices and insert the missing rests in the second voice. We compute the onset of each note by summing the durations of all previous notes in the same voice.  $\diamond$

<sup>6</sup> Ideally, both voices will have the same duration, but in real scores, this is often not the case; see [Fos+20; FRT21] for a discussion about score quality.

Music Symbol	$\circ$	$\text{♩}$	$\text{♪}$	$\overset{3}{\text{♩}}$	$\overset{3}{\text{♪}}$	$\overset{3}{\text{♩}}$
Relative Symbolic Duration (1 = $\text{♩}$ )	4	2	1	$0.\bar{6}$	0.5	$0.\bar{3}$
Absolute Symbolic Duration	1	2	4	6	8	12
						16

**Table 1.** Examples of music symbols and corresponding durations in RSD and ASD units. Notes with “3” on top are notes that are part of a triplet.

This paper discusses an alternative approach to handling durations: the use of *Absolute Symbolic Duration* (ASD) units. The core idea is to consider the integers implied by the names of the graphical symbols. For example, a quarter note  $\text{♩}$  as 4, an eighth note  $\text{♪}$  as 8, a 16th note  $\text{♫}$  as 16, and so on. Durations produced from irregular groupings are also expressed as integers (see Table 1). ASD units are already used by the Humdrum `**kern` file format, and (in a mixed representation with the `divs` approach) by MEI and MusicXML. However, they are only used to encode single note/rest durations. The typical pipeline procedure is to translate this duration format into relative symbolic durations, as a preprocessing step before any other operation.

On the contrary, we explore the usage of ASD, as “standalone” units to manipulate musical score durations. To make this practicable, we define two operations that cover typical use cases and we prove that, like RSD, ASD units form a *semiring*, an algebraic structure that enables a more abstract general description of processing pipelines. The actual computations can later be performed in ASD or RSD (or a mixture of the two), depending on the situation. This is enabled by an isomorphism that we provide to translate between the two units. Finally, we discuss some practical cases where one unit is to be preferred over the other to make the pipeline more efficient in terms of the number of operations and data types that are considered. We implement some algorithms that use ASD units in the Python library Partitura [Can+22].

## 2 Definitions

In this section, we first introduce the semiring; then we formally define *ASD* and *RSD* units and a morphism between them. Our goal with the introduction of this formalism is to give a general, abstract way of describing algorithms on music durations which is valid for both ASD and RSD units. Such algorithms can practically be performed in one unit or the other (or a mix of the two) depending on the specific application (see Section 3).

### 2.1 Semiring

Formally, a *semiring*  $\mathcal{S} = \langle \mathbb{S}, \oplus, \otimes \rangle$  is an algebraic structure that consists of a domain  $\mathbb{S} = \text{dom}(\mathcal{S})$ , and two associative binary operators  $\oplus$  and  $\otimes$ . Some properties must be verified:  $\oplus$  is commutative, and  $\otimes$  distributes over  $\oplus$ , i.e.,  $\forall x, y, z \in \mathbb{S}, x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$ .

Note that there is no complete agreement in the literature over the exact definition of a semiring. Other research (e.g., [Pin98]) defines the two operations of a semiring

with a neutral element ( $0$  and  $1$  respectively), such that  $0$  is absorbing for  $\otimes$ :  $\forall x \in \mathbb{S}, 0 \otimes x = x \otimes 0 = 0$ . Then the semiring without neutral elements is called *hemiring*. However, similarly to [DSA10; SMS21], we just use the term semiring, without including neutral elements (and in particular the absorbing propriety of  $0$ ). The motivation is that verifying the absorbing propriety requires changes that would take our framework further away from its use for practically useful operations on music duration (more on this in Section 2.3). Components of any semiring  $\mathcal{S}$  may be superscripted by  $\mathcal{S}$  when needed. By abuse of notation, we write  $x \in \mathcal{S}$  to denote  $x \in \mathbb{S}$ .

A semiring  $\mathcal{S}$  is *commutative* if  $\otimes$  is commutative. It is *idempotent* if for all  $x \in \mathcal{S}$ ,  $x \oplus x = x$ . It is *monotonic* w.r.t. a partial ordering  $\leq$  iff for all  $x, y, z$ ,  $x \leq y$  implies  $x \oplus z \leq y \oplus z$ ,  $x \otimes z \leq y \otimes z$  and  $z \otimes x \leq z \otimes y$ . Every idempotent semiring  $\mathcal{S}$  induces a partial ordering  $\leq_{\mathcal{S}}$  called the *natural ordering* of  $\mathcal{S}$  and defined by: for all  $x$  and  $y$ ,  $x \leq_{\mathcal{S}} y$  iff  $x \oplus y = x$ . It holds then that  $\mathcal{S}$  is monotonic w.r.t.  $\leq_{\mathcal{S}}$ .  $\mathcal{S}$  is called *total* if it is idempotent and  $\leq_{\mathcal{S}}$  is total, i.e., when for all  $x$  and  $y$ , either  $x \oplus y = x$  or  $x \oplus y = y$ .

Given the particular algebraic properties above, semirings can be used as a weight domain for optimization problems such as the search for shortest paths in weighted graphs or hypergraphs [Moh02; Hua08]. Indeed, the theory of semirings and in particular the min-plus and max-plus Tropical Algebras [GP97] is commonly applied in decision theory and operational research, performance evaluation and control of dynamic systems, and also formal language theory, for quantitative extensions of formal computation models [DK09] (weighted automata and grammars). They have also been recently used for the formalization of musical elements, e.g., harmonic/melodic intervals by Albini [AB19], and to describe algorithms for musical tasks, e.g., music transcription by  $n$ -best parsing [Fos+19], and melodic distance computation [GJ22].

This work focuses on formalisations of musical duration that form idempotent and commutative semirings. Intuitively, in the applications presented in this paper,  $\oplus$  selects the longest duration and  $\otimes$  aggregates two durations in a single one.

## 2.2 Absolute Symbolic Durations

Let us define the semiring of Absolute Music Duration units  $\mathcal{A} = \langle \mathbb{Q}^+ \cup \{\infty\}, \oplus, \otimes \rangle$  by detailing its domain and the two operations.

### The domain

The domain  $dom(\mathcal{A}) = \mathbb{Q}^+ \cup \{\infty\}$  of  $\mathcal{A}$  contains (but is not limited to) non-null integers implied by the graphical symbol of notes and rests, e.g., quarter notes, eighth notes, 16th notes, 32th notes, etc. Intuitively, larger values correspond to shorter notes. The limiting case is the null musical duration (used, for example, for grace notes), which is denoted by  $\infty$ .  $dom(\mathcal{A})$  also includes other values that can result from the use of *duration modifiers* in the musical score, such as dots and tuplets, and will be described later in this section. We define  $\prec^{\mathcal{A}}$  to be the strict order of absolute musical durations on the domain of  $\mathcal{A}$ . Elements of  $\mathcal{A}$  are defined such that  $\forall a, b \in dom(\mathcal{A}), a \prec^{\mathcal{A}} b \iff a > b$ .

### Operations

We are interested in two operations: a *selection* operation to find the longest duration,

and a *concatenation* to combine two or more musical durations. We define  $\oplus^{\mathcal{A}}$  such that  $a \oplus^{\mathcal{A}} b = \min(a, b)$ , as the selection operation. Practically, this operation can be used to select the longest voice within a measure, when their durations do not correspond, like in Example 1.

The concatenation operation  $\otimes^{\mathcal{A}}$  is defined as  $a \otimes^{\mathcal{A}} b = \frac{ab}{a+b}$ . This operation expresses mathematically the well-known musical rules about aggregating durations. For example, the concatenation of two eighth notes yields a quarter note, which in our framework can be written as  $8 \otimes^{\mathcal{A}} 8 = 4$ . A more advanced usage for ties and dots is also exemplified in Section 2.4. Readers who are not familiar with the semiring formalisms may find confusing that this concatenation operation, which looks very much like a sum, is denoted with the symbol  $\otimes$ , but this is what is commonly used and we keep it for consistency.

To prove that  $\mathcal{A}$  is a semiring we need to prove that we have closure for both operations and that the multiplication distributes over additions. We go slightly further than proving closure and prove that both operations are commutative monoids (i.e. that they are commutative, associative, and there is an identity element) since this could be useful for further extension of our framework. Remember that for simplicity we write  $x \in \mathcal{A}$  to denote  $x \in \text{dom}(\mathcal{A})$ .

**Lemma 1.**  $\langle \text{dom}(\mathcal{A}), \oplus^{\mathcal{A}} \rangle$  is a commutative monoid.

*Proof.* Let  $a, b, c \in \mathcal{A}$ . By definition  $a \oplus^{\mathcal{A}} b = \min(a, b)$ .

Then from the commutativity and associativity properties of the min operation,  $\oplus$  is also commutative and associative. The closure is trivial for min. The identity element is  $\infty$ , i.e.,  $\forall a \in \mathcal{A}, a \oplus \infty = a$ .

**Lemma 2.**  $\langle \text{dom}(\mathcal{A}), \otimes^{\mathcal{A}} \rangle$  is a commutative monoid.

*Proof.* Let  $a, b, c \in \mathcal{A}$ . By definition  $a \otimes^{\mathcal{A}} b = \frac{ab}{a+b}$ . By the commutativity of addition and multiplication, it follows that  $\otimes^{\mathcal{A}}$  is also commutative and associative. Closure is also verified for the same reason. Let us investigate if the relationship also holds for the case of the null durations, i.e.  $\infty$ . We define  $a \otimes^{\mathcal{A}} \infty$  as the limit  $\lim_{b \rightarrow \infty} (a \otimes^{\mathcal{A}} b)$ .

$$\begin{aligned} a \otimes^{\mathcal{A}} \infty &= \lim_{b \rightarrow \infty} (a \otimes^{\mathcal{A}} b) = \lim_{b \rightarrow \infty} \left( \frac{ab}{a+b} \right) = a \lim_{b \rightarrow \infty} \left( \frac{b}{a+b} \right) = a \lim_{b \rightarrow \infty} \left( \frac{1}{\frac{a}{b} + 1} \right) = \\ &= a \left( \frac{\lim_{b \rightarrow \infty} 1}{\lim_{b \rightarrow \infty} \frac{a}{b} + 1} \right) = a \left( \frac{1}{\lim_{b \rightarrow \infty} \frac{a}{b} + 1} \right) = a \left( \frac{1}{0 + 1} \right) = a \end{aligned}$$

Since  $\otimes^{\mathcal{A}}$  is commutative, this also holds for the case  $\infty \otimes^{\mathcal{A}} a$ . We also proved that  $\infty$  is the neutral element of  $\otimes^{\mathcal{A}}$ . ◇

We will now prove some Lemmas that will be useful for the proof of Theorem 1.

**Lemma 3.** Let  $a, b, c \in \mathcal{A}$ , then  $b < c \iff a \otimes^{\mathcal{A}} b < a \otimes^{\mathcal{A}} c$ .

*Proof.*

$$\begin{aligned}
 b < c &\stackrel{a>0}{\iff} ab < ac \stackrel{bc>0}{\iff} ab + bc < ac + bc \iff b(a + c) < c(a + b) \stackrel{a>0}{\iff} \\
 &ab(a + c) < ac(a + b) \stackrel{a+c>0, a+b>0}{\iff} \frac{ab}{a + b} < \frac{ac}{a + c} \equiv a \otimes^A b < a \otimes^A c
 \end{aligned}$$

◇

**Lemma 4.**  $\otimes^A$  is left and right distributive over  $\oplus^A$ .

*Proof.* Let  $a, b, c \in \mathcal{A}$ : By induction on the order relation between  $b, c$ :

- $b = c$ :  
Trivial.
- $b < c$ :

$$a \otimes (b \oplus c) = \frac{a \min(b, c)}{a + \min(b, c)} \stackrel{\text{by IH}}{=} \frac{ab}{a + b} = a \otimes b \quad (1)$$

$$(a \otimes b) \oplus (a \otimes c) = \min((a \otimes b), (a \otimes c)) \stackrel{\text{by Lemma 3 and IH}}{=} a \otimes b \quad (2)$$

Then our proof is completed by using reflexivity on 1 and 2.

- $b > c$ :  
Similar to  $b < c$ .

The right side distributivity follows by the commutativity properties of the operations.

◇

**Theorem 1.**  $(\text{dom}(\mathcal{A}), \oplus^A, \otimes^A)$  is a semiring.

*Proof.* We have all the elements to conclude the proof:

- $(\text{dom}(\mathcal{A}), \oplus^A)$  is associative and satisfies the closure property (by Lemma 1);
- $(\text{dom}(\mathcal{A}), \otimes^A)$  is associative and satisfies the closure property (by Lemma 2);
- Multiplication distributes over addition (by Lemma 4)

◇

When dealing with multiple equal durations in music, it is practical to extend the  $\oplus$  operation to define a scalar multiplication. For a duration  $a \in \mathcal{A}$  and a scalar  $n \in \mathbb{Q}$ , it is denoted by the function  $\text{repeat}^A(a, n) = a/n$ .

### 2.3 Relative Symbolic Durations

We define the semiring of Relative Symbolic Duration units  $\mathcal{R}^\delta = (\mathbb{Q}^+ \cup \{0\}, \oplus, \otimes)$  relative to the reference duration  $\delta$ .

**The domain** The domain  $dom(\mathcal{R}^\delta) = \mathbb{Q}^+ \cup \{0\}$  of  $\mathcal{R}$  contains durations measured relative to a reference duration value. Intuitively, smaller values correspond to shorter notes. The limiting case is the duration 0, which can be used, for example, for grace notes. We define  $\prec^{\mathcal{R}}$  to be the strict order of absolute musical durations on the domain of  $\mathcal{A}$ . Elements of  $\mathcal{R}$  are defined such that  $\forall a, b \in dom(\mathcal{R}), a \prec^{\mathcal{R}} b \iff a < b$ .

**Operations** Similarly to  $\mathcal{A}$ ,  $\oplus^{\mathcal{R}} \equiv \max$  is used to select the larger duration and  $\otimes^{\mathcal{R}} \equiv +$  is used to add two durations together. The repeat operation can be defined as  $repeat^{\mathcal{R}}(a, n) = a * n$ .

We skip the proof of  $\mathcal{R}$  being a semiring for brevity. It can also be noted that the operations and domain we defined are equivalent to those of a tropical semiring [Pin98], so the proof for tropical semirings is also valid for our case. Differently from a tropical semiring, however, we don't have the absorption property of the  $\otimes$  neutral elements  $0$ , i.e.,  $\forall x \in \mathcal{R}, 0 \otimes x = x \otimes 0 = 0$ . In order to verify this, we would need to swap the min with the max (and vice-versa) for the  $\otimes$  in our two semirings, but this would make for a non-musically useful operation, violating the ultimate objective of this research.

## 2.4 A General Duration Framework

Table 2 summarises our formalization of ASD and RSD units. In the following, we introduce a morphing function to convert between these two units. Finally, we include in our framework the duration modifiers that are used in musical scores, i.e., ties, dots, and tuplets.

### Morphing between time units

Given a reference duration value  $\delta$ , we define the reciprocal function  $f(x) = \delta/x$  that maps every element  $x \in dom(\mathcal{A})$  to its correspondent in  $\mathcal{R}^\delta$ , and vice-versa. It is trivial to see that this function is isomorphic and order-preserving (it preserves the ordering in the respective source/target domains, even though the order in  $\mathcal{A}$  is reversed with respect to  $\mathcal{R}$ ); it follows that  $f$  is a *Homomorphism*, i.e.  $\forall a, b \in dom(\mathcal{R}), f(a \otimes^{\mathcal{R}} b) = f(a) \otimes^{\mathcal{A}} f(b)$ . The choice of  $\delta$  has interesting practical implications. For example, by setting it to a beat duration (which depends on the time signature), we obtain units typically used in music research to reduce the dependency on the time signature. By setting it to a quarter note duration we obtain the so-called *quarter length* durations, commonly used for general applications since they do not depend on other score parameters.

	S	$\oplus$	$\otimes$
ASD $\mathcal{A}$	$\mathbb{Q}^+ \cup \{\infty\}$	$\min(a, b)$	$\frac{ab}{a+b}$
RSD $\mathcal{R}$	$\mathbb{Q}^+ \cup \{0\}$	$\max(a, b)$	$a + b$

**Table 2.** Table comparing the semirings of Absolute and Relative Symbolic Durations.

### Duration modifiers

In a musical score, there are some graphical symbols, i.e., ties, dots, and tuplet groupings, that modify the duration of the notes/rests they are assigned to. In this section, we will define ties, dots, and tuplets as functions applied to elements of either  $\mathcal{A}$  or  $\mathcal{R}$ . We use the symbol  $\mathcal{X}$  to refer to either of the structures  $\mathcal{A}$  or  $\mathcal{R}$ .

First, let us consider the ties between notes. The total duration of two tied notes  $a, b \in \mathcal{X}$  can be easily captured by the  $\otimes$  operation.

**Definition 1.** *The total duration of two tied notes  $a, b \in \mathcal{X}$  is given by function  $\text{tie} : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{X}$*

$$\text{tie}(a, b) = a \otimes^{\mathcal{X}} b \quad (3)$$

Another musical concept that can prolong the duration of a musical note is the dot. A dotted note  $a$  can be seen as a function  $\text{dot}$  applied to the note  $a$ . This can be generalized for an arbitrary number of dots:

**Definition 2.** *The function  $\text{dot} : \mathcal{X} \times \mathbb{N} \rightarrow \mathcal{X}$  applied to a note  $a \in \mathcal{X}$  is inductively defined as follows:*

$$\text{dot}(a, 0) = a \quad (4)$$

$$\text{dot}(a, n + 1) = \text{repeat}\left(a, \frac{1}{2^{n+1}}\right) \otimes^{\mathcal{X}} \text{dot}(a, n) \quad (5)$$

Another function that can be used to construct musical duration is the tuplet function. The duration of a note in a tuplet of total duration  $a \in \mathcal{X}$  can be seen as a function with two parameters, the base note duration  $a$  and the type of tuple  $\gamma$  (in this case 3 for triplet).

**Definition 3.** *Let  $a \in \mathcal{X}$ ,  $\gamma \in \mathbb{N}_{>2}$ . The tuplet function,  $t : \mathcal{X} \times \mathbb{N}^* \rightarrow \mathcal{X}$ , is defined as:*

$$t(a, \gamma) = \text{repeat}\left(a, \frac{2}{\gamma}\right) \quad (6)$$

*Example 2.* We use the formalisms introduced in this section on the problem of Example 1, where the goal was to import the score from an MEI file, split the third note (the C) in the top voice into two, and produce a pianoroll representation. This process can abstractly be described solved as: (1) read all durations  $[d_1, d_2, \dots, d_n]$  from the input MEI file; (2) compute the values of the notes under the triplet with Definition 3; (3) split the third note into two notes with duration  $d_{\text{new}} = \text{repeat}(d_3, 1/2)$ ; (4) find each note onset and offset position by  $[d_1 \otimes d_2 \otimes \dots \otimes d_n]$ ; (5) for the last note offset of each voice, compute the maximum with the  $\oplus$  operator; (6) output the pianoroll representations for the two voices, using the start times and durations thus calculated.

## 3 From Abstract Description to Algorithm Implementation

In the previous section, we introduced an abstract formalism to describe algorithms on music sequences. We now discuss cases where it is more efficient to perform such algorithms in ASD units or in RSD units.



### 3.1 Advantages and Disadvantages

The use of ASD units can bring advantages in terms of data types because it can give a prevalence of integers over floating point (and periodic) values. For this to be the case, we need to deal with durations that span a maximum of a whole note. In a 4/4 piece, this will correspond to durations of one measure. This does not mean that algorithms implemented in ASD units cannot handle multiple measures, but rather that they should follow a “divide et impera” principle where every measure is handled independently. This is already quite common in file-parsing systems since scores are encoded measure by measure in file formats such as MusicXML and MEI.

In terms of the number of computations, ASD units are ideal for applications that concern the graphical symbols used in the score, for example, changing the pitch of a note, changing a duration, or segmenting a musical score. Such applications can skip the costly computation of common divisor, and conversion to RSD units altogether. Instead, when the measure is not specified (which could be the case, for example, in handling a MIDI file), or when we want to do operations that don’t follow the measure segmentation (e.g., segmenting a score between measures), the usage of RSD units is preferred.

*Example 3.* Let us consider the problem of Example 1. By considering ASD units, we can parse the input score file simply by copying the values for the note graphical durations. The splitting of the third note of duration 16 in two parts yields two notes of duration  $16 * 2 = 32$ .

A big limit in the efficiency of ASD units is posed by time signatures where the beat is a dotted note, for example, 6/8. A dotted note will make the duration assume noninteger, or even periodic, values. A possible solution to this problem is given in the next section.

### 3.2 The lazy evaluation case

It is common for systems that deal with musical scores to have a generic import function, where the score file is converted to some internal representation. If in this step, the user did not yet specify the set of operations they intend to perform, the choice of whether to use ASD or RSD cannot be performed. In order to let the system choose between ASD and RSD to exploit the advantages described in the previous section, we suggest using a *lazy evaluation* parsing strategy. First, we propose to reduce the domain by considering only the ASD values  $\{2^n \mid \forall n \in [0..7]\}$  (i.e. only single graphical note/rest symbols). Duration modifiers such as dots or tuplets are imported as functions dot or tuple without being computed. Only when the user specifies a task, will these functions be resolved to actual values, and the task is performed either in ASD or RSD units, depending on what would allow for the most efficient computation. From a functional programming perspective, this can be viewed as a Monad transformation [Wad92] of the parsed elements.

### 3.3 Implementation

We provide a proof of concept of the practical utility of the methods introduced in this paper, by implementing some functions in the Python library *Partitura* [Can+22]. The core of this library, i.e., the *Timeline* object, uses RSD units, in particular on the common divisor technique described in the introduction. However, some functions in the file parsing module are modular enough to make it possible to run them in ADS without the need of making major changes to the rest of the library. These are: (1) the functions to compute the common divisor for integer encoding of RSD durations, (2) the function to find the longest voice in a measure, and (3) the computation of the actual duration for a note inside a tuplet. We also implement an alternative (still partial) parser of `**Kern` files that leverages a lazy evaluation approach.

## 4 Conclusions and Discussion

In this paper, we proposed an alternative approach to handling the symbolic music durations from musical scores, that is based on absolute symbolic duration (ASD) units. We formalized ASD, and the (typically used) relative symbolic duration (RSD) units, in a single mathematical framework, and paired them with two operations. The result is two semirings: algebraic structures that enable an abstract description of algorithms on symbolic durations. We then moved to a more practical discussion and described some use cases where one unit is more efficient than the other, in terms of data types (integers vs floating point) and number of calculations. Finally, we advocated a functional parsing of symbolic music formats that can select the most efficient way of performing the various operations in an algorithm and enable considerable speed-up for common use cases.

It is clear that the proposals in this paper are mostly of theoretical interest, and belong to the research branch that formalizes musical elements with mathematical structures [AB19; Maz12; Pop+16]. However, our interest in this topic started from our practical experience with parsing and processing musical score files to use their information as input for music information retrieval (MIR) systems. While the improvement in efficiency that our methods may enable is negligible for a single score, large deep-learning models have to load thousands of scores, thus making each small optimization much more useful. For example, we will probably soon see some general tokenization techniques for musical scores (similar to the multiple ones that have been proposed for MIDI files [Fra+21]); in this context, a tokenization that focuses on the graphical symbols using only ASD, could enable major speedups in computing time.

## 5 Acknowledgements

This work was supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), the Federal State of Upper Austria (LIT AI Lab), and JSPS Kaken 20H04302, 21H03572.

## References

- [Wad92] Philip Wadler. “The essence of functional programming”. In: *Proceedings of the ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 1992, pp. 1–14.
- [GP97] Stéphane Gaubert and Max Plus. “Methods and applications of (max,+) linear algebra”. In: *STACS 97: 14th Annual Symposium on Theoretical Aspects of Computer Science Lübeck, Germany February 27–March 1, 1997 Proceedings 14*. Springer. 1997, pp. 261–282.
- [Pin98] Jean-Eric Pin. *Tropical semirings*. Cambridge Univ. Press, Cambridge, 1998.
- [Moh02] Mehryar Mohri. “Semiring frameworks and algorithms for shortest-distance problems”. In: *Journal of Automata, Languages and Combinatorics* 7.3 (2002), pp. 321–350.
- [Hua08] Liang Huang. “Advanced Dynamic Programming in Semiring and Hypergraph Frameworks”. In: *Int. Committee on Computational Linguistics Conference (COLING)*. 2008.
- [DK09] Manfred Droste and Werner Kuich. “Semirings and formal power series”. In: *Handbook of Weighted Automata*. Springer, 2009, pp. 3–28.
- [CA10] Michael Scott Cuthbert and Christopher Ariza. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.” In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2010, pp. 637–642.
- [DSA10] Wieslaw A Dudek, Muhammad Shabir, and Rukhshanda Anjum. “Characterizations of hemirings by their h-ideals”. In: *Computers & Mathematics with Applications* 59.9 (2010), pp. 3167–3179.
- [Maz12] Guerino Mazzola. *The topos of music: geometric logic of concepts, theory, and performance*. Birkhäuser, 2012.
- [Pop+16] Alexandre Popoff et al. “From K-nets to PK-nets: a categorical approach”. In: *Perspectives of New Music* 54.2 (2016), pp. 5–63.
- [AB19] Giovanni Albini and Marco Paolo Bernardi. “Tropical Generalized Interval Systems”. In: *Mathematics and Computation in Music*. Ed. by Mariana Montiel, Francisco Gomez-Martin, and Octavio A. Agustin-Aquino. Springer International Publishing, 2019, pp. 73–83.
- [Fos+19] Francesco Foscarin et al. “A parse-based framework for coupled rhythm quantization and score structuring”. In: *Proceedings of the International Conference on Mathematics and Computation in Music (MCM)*. Springer. 2019, pp. 248–260.
- [Fos+20] Francesco Foscarin et al. “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.
- [FRT21] Francesco Foscarin, Philippe Rigaux, and Virginie Thion. “Data quality assessment in digital score libraries: The GioQoso Project”. In: *International Journal on Digital Libraries* 22 (2021), pp. 159–173.

- [Fra+21] Nathan Fradet et al. “MidiTok: A Python package for MIDI file tokenization”. In: *Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference*. 2021.
- [SMS21] MK Sen, SK Maity, and KP Shum. “Some Aspects of Semirings”. In: *Southeast Asian Bulletin of Mathematics* 45.6 (2021).
- [Can+22] Carlos Eduardo Cancino-Chacón et al. “Partitura: A Python Package for Symbolic Music Processing”. In: *Proceedings of the Music Encoding Conference (MEC2022)*. Halifax, Canada, 2022.
- [GJ22] Mathieu Giraud and Florent Jacquemard. “Weighted Automata Computation of Edit Distances with Consolidations and Fragmentations”. In: *Information and Computation* 282 (2022).

# Soundscape4DEI as a Model for Multilayered Sonifications

João Neves<sup>1</sup> , Pedro Martins<sup>1</sup> , F. Amílcar Cardoso<sup>1</sup> ,  
Jônatas Manzolli<sup>2</sup> , Mariana Seíça<sup>1</sup> , and M. Zenha Rela<sup>1</sup> 

<sup>1</sup> University of Coimbra, CISUC, Department of Informatics Engineering  
{joaoneves}@student.dei.uc.pt,

{pjmm,amilcar,marianac,mzrela}@dei.uc.pt

<sup>2</sup> Arts Institute - Int. Nucleus for Sound Studies (NICS),  
University of Campinas, São Paulo, Brazil  
{jmanzo}@unicamp.br

**Abstract.** Computation emergence has impacted the development of creative musical systems, as it allows for unprecedented exploration and innovation in the realm of music composition and sound design. As data is becoming more and more complex [1], new information sharing tools and methods arise. In this paper, we present *soundscape4dei*, a system that sonifies data from the daily routine of the Centre for Informatics and Systems of the University of Coimbra (CISUC). The developed system explores and proposes a multilayered approach that succeeds in raising awareness and informally disseminating the (usually invisible) activities at CISUC. We go over the design of the system, we analyse its outputs and we discuss our sonification model.

**Keywords:** Sonification · Sound Design · Sound installation

## 1 Introduction

Humans are equipped with a complex listening system. It is capable of distinguishing sound sources, identifying melodies, recognising patterns even under adverse conditions and, most importantly, ”interpret sounds using multiple layers of understanding” [2], making it a powerful and flexible instrument to explore for portraying data. Our ability to flexibly change the auditory focus and learn and improve discrimination of auditory stimuli [2] creates many possibilities when approaching sound as an information-sharing vehicle.

The study of auditory displays as a scientific field started in 1992, with the foundation of the International Community of Auditory Display (ICAD) [2]. In the past few decades, technological growth and accessibility contributed to the



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

appearance and standardisation of new communication methods and tools. Sonification, “the use of nonspeech audio to convey information” [3], is a powerful auditory data representation based on mapping data attributes to sound parameters for user-friendly comprehension. Its merits include harnessing innate auditory pattern recognition, fostering rapid data interpretation, and aiding visually impaired individuals. Moreover, real-time feedback empowers scientific research, healthcare monitoring, and industry [2][4]. It is currently used in a wide variety of contexts (e.g., bio-medicine, seismology, interfaces for visually disabled people) and its research is associated with a wide list of disciplines, such as physics, perceptual research and computer science [2].

Be it education, research, students’ activities or industry cooperation, most of the activity at CISUC (Centre for Informatics and Systems), University of Coimbra, is mediated through computers and mostly closed networks. This project arose from an attempt to locally raise awareness of those activities and informally disseminate them. By exploring the intersection of music and technology, we developed a sonification system that uses real-time data analysis and a multilayered approach to create musical compositions. The challenge of this sonification is to be able to illustrate four different types of events, each with specific and particular mappings. At the same time, the system must operate continuously, so we aimed for the creation of a non-invasive sound space.

The name *soundscape4dei* refers to DEI, the Department of Informatics Engineering, where CISUC is hosted. The created soundscapes depict events like purchases, scholarship allocations, paper submissions and researchers’ missions. In this paper, we start by overviewing sonification projects that inspired this work, followed by a description of the system and its physical installation. We discuss the multilayered approach and how it can incorporate different techniques while still being successful. We end by discussing how the resulting sonic experiences may be evaluated and then we address future work.

## 2 Related Work

Sonification has been investigated and proven successful when explored in a vast number of contexts. In this section, we briefly describe projects addressing various fields, thus illustrating its multidisciplinary applicability. Furthermore, we envision some of their musical nuances to be either related or likely applied to routine sonifications, the domain of the presented system.

Two Trains [5] is a music composition that emulates a ride on the New York Subway through three boroughs: Bronx, Brooklyn and Manhattan. The number of instruments and dynamics of the song corresponds to the median household income in each location, revealing the economic inequality across the city while exhibiting its energy and the chaos of the subway system.

Sonic Kayaks [6] are musical instruments used to investigate nature. They use a system previously explored by Matthews on Sonic Bikes [7] and allow kayak paddlers to hear real-time water temperature and underwater sounds as they map the marine world data to a generative live composition while navigating.

Seiça *et al.* [8] developed a system that analyses social media (TWITTER) data, estimates the posts emotions and translates them into auditory language. This project explores the subjectivity of human emotions and its relationship with music as a transmedia instrument.

A sonification experience to portray the sounds of Portuguese consumption habits [9] presents a listening experiment that explores the influence of aesthetics in the perception of auditory displays. The system sonifies consumption habits from a portuguese retail company over the course of ten days.

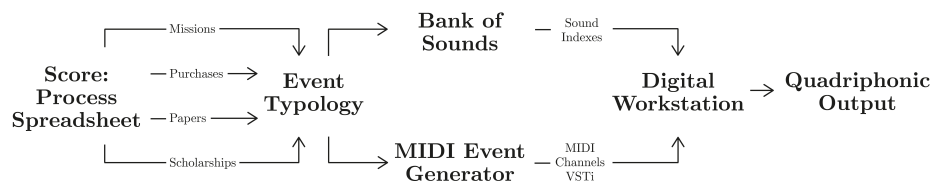
Brian House [10] interprets a continuous year of his location-tracking data to create a recording that sonifies every moment of his daily routine. The 11 minute song, Quotidian Record, suggests that habitual patterns have inherent musical qualities that might form an "emergent portrait of an individual" [11].

Living Symphonies [12] is a sound installation based on the fauna and flora of four ecosystems in the United Kingdom. The designed model reflects the behaviour, movement and daily patterns of wildlife, translating a network of interactions that formed the ecosystem.

This sonification project is based on a developed multilayered model. This model consists of the use of direct parameterization as well as generative techniques, used to enrich the sound experience.

### 3 The approach

Conceptually, we focus on developing a system whose outputs focus on data transparency while taking advantage of sound expressiveness to create immersive sonic experiences. We also had to pay attention to the audio intensity and other components that could disturb students and researchers that often use the installation room to study and work. Figure 1 describes the architecture of the sonification process. In the following sections, we will provide a comprehensive overview of the data and the implemented system, while also explaining the decision-making process throughout the project.



**Fig. 1.** The architecture of *soundscape4dei*: the Spreadsheet contains the data to be sonified, the Event Typology is described in the *Encoding* section, the MIDI events and Bank of Sounds are displayed in Table 1, and the digital workstation and quadriphonic output are detailed in the Instalation section.

### 3.1 Data

In this project, we propose to sonify data describing the activity of our research centre CISUC, which comprises about 150 researchers (faculty members, graduate and post-graduate students) and is organised into six research groups: Adaptive Computation, Cognitive and Media Systems, Evolutionary and Complex Systems, Information Systems, Laboratory Communications and Telematics, Systems and Software Engineering. We found CISUC activities enough diversified to be sonically explored and conceptually rich from a social and technological standpoint since the data is dense, well-structured and periodically refreshed. For the purpose of illustrating our approach, we use a small fragment of the stream of data, manually edited to assure a diversity of situations. We follow a standard visualisation pipeline to handle the data preprocessing — selection, categorisation and validation, before applying any transformations. The manual edition allowed the filtering of spelling inconsistencies.

All the activities at CISUC are chronologically registered and detailed according to their nature. We identified four meaningful categories of events to consider within the sonification: purchases, missions, scholarship allocations and paper submissions. The system must be able to create meaningful sonic scenarios allowing for easier recognition of these categories.

### 3.2 Software Architecture

The implemented pipeline is composed of five main stages (Figure 2). The data is firstly preprocessed by a PROCESSING sketch which checks for spelling typos and standardises every entry (setting the input to lowercase and removing its accents), before transforming and encoding them. This process is based on algorithms described in the next section. The output of this sketch is then sent as osc messages to a MAX/MSP patch, which is responsible for playing organic sounds from WAVE files after handling equalisation, as well as for converting osc messages to MIDI. This messages are sent afterwards through multiple channels to a LOGIC PRO project. LOGIC PRO uses Spitfire LABS VSTI to create the various hybrid soundscapes, composed of both symbolic (human) sounds and organic (nature) sounds [13,14].

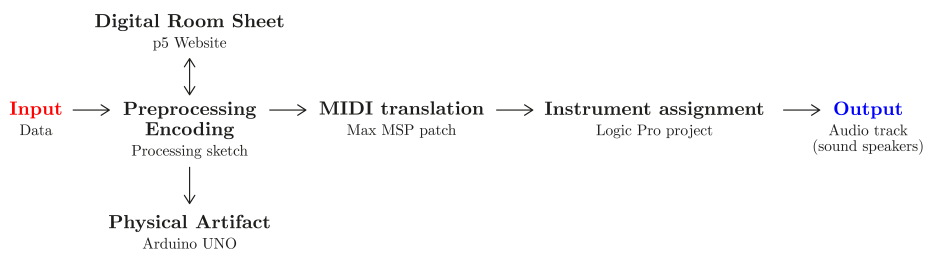


Fig. 2. *soundscape4dei* pipeline.



### 3.3 Encoding

Sound has a multitude of changeable dimensions that allow for many options when mapping data to audio [4][15]. This system relies on algorithms that manage and create different sound layers. They are composed of organic sounds from standard environmental recordings and by symbolic sounds (i.e. triggered by MIDI events) from a selected and specific VSTi bank Spitfire (LABS) to ensure the system creates the experiences we envision (Table 1). The criteria for choosing sounds is based on artistic intuition.

**Table 1.** Nature and symbolic sounds and respective description.

	Recording	Description
Nature Sounds	1. Underwater environment	Used together with a <i>Storm</i> recording in order to create the illusion of submersion in the <i>Purchase</i> category.
	2. Storm	Used together with a <i>Underwater environment</i> recording in order to create the illusion of submersion in the <i>Purchase</i> category.
	3. Sea waves	Used in the <i>Scholarship allocation</i> category to represent the money invested. Money flow translates to water flow.
	4. Crowd	Used in the <i>Missions</i> category to portray the amount of population at the destiny.
	5. Birds	Used in the <i>Paper</i> submissions category. The amount of chirping translates to the importance of the submission, based on the ranking of the targeted conference.
Symbolic Sounds	<b>VSTi</b>	
	6. Electric Piano	Morse code melody used in all sonifications.
	7. Granular Whalesong: Nautilus and Drone	Drone sounds used to represent pollution in the <i>Missions</i> category.
	8. Pedal Pads: Azure Piano	Synth sounds used to establish a major scale mode based on the current weather of a mission destination as well as to create harmony for the <i>Purchase</i> category.
9. Opia: Piano Granular	Synth sounds used to intensity the flow metaphor for the <i>Scholarship allocations</i> category.	

*Soundscape4dei* sonifies every *Score* line (see Figure 1) individually, portraying it for one minute. To allow a easier communication of multiple streams of event data simultaneously, we explore a multilayered approach, resorting not only to standard parameter mapping [2] but also to methods presented in this section and further discussed in section 5. The sonification is composed of specific encodings that differentiate and represent the different categories of events and universal encodings (general components) which aims to provide a unifying character.

Table 2 depicts the structure and sound allocation of our sonification. We name the three layers that compose the different soundscapes as *Melody* (L1), *Harmony* (L2) and *Texture and Signals* (L3). Each layer concerns several symbolic and/or organic sounds, which are selected to each category encoding.

**Table 2.** Layer division and sound allocation for each category. Source numbers according to Table 1. *G* source stands for group instruments which are individually present in all categories but are not exclusive.

Layer	L1		L2		L3					
Source	6	7	8	9	1	2	3	4	5	G
Mission	■	■	■					■		■
Purchase	■		■		■	■				■
Papers	■								■	■
Scholarships	■			■			■			■

**3.3.1 General components** For each entry on the process spreadsheet (1), there is a melody that sonifies its description field. The melody rhythm is generated from a Morse code translation of the input. The scale notes and tonality of the melody, which is always major, are chosen randomly to create more variety and avoid stagnation.

The research group associated with each entry is represented by notes from predefined VST percussion instruments: Glockenspiel for *Adaptive Computation*, triangle for *Cognitive and Media Systems*, jingle bells for *Evolutionary and Complex Systems*, kettledrum for *Information System*, chimes for *Laboratory Communications and Telematics* and cymbal for *System and Software Engineering*. We chose to use instruments from this family because they have defined and precise sounds that stand out sonically.

**3.3.2 Missions** Missions comprise oral and poster presentations at conferences, project meetings and other activities that require travelling. In this type of event the soundscape is composed of elements that vary depending on the destination of the mission. The system uses an API to verify the destination meteorology in real-time, which defines the major mode (for the Morse melody). The brighter the current weather, the brighter the major scale mode [16]. The population number influences the volume of a talkative crowd and the pollution level plays a drone sound, creating a pedal that lasts the entire sequence (the higher the level, the greater the presence).

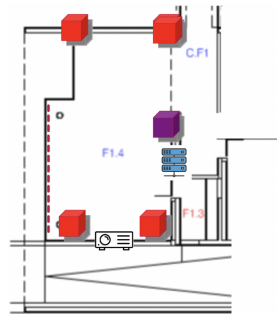
**3.3.3 Purchases** We approached purchases from a more subjective standpoint. This soundscape means to represent how significant the transaction is, since some purchases are cheaper (for example, an arduino compared to a laptop), therefore tendentially less impactful than others. To develop this metaphor, we use water sounds. When combined, they create the illusion of submersion — the more unusual and significant the purchase, the stronger the low-pass filter applied and, consequently, the more depth is simulated. To intensify this encoding, the mode (of the Morse melody) changes depending on the same factors, creating a relationship between its brightness and the proximity to the surface [16]. The VST unifies the various sound components by playing augmented triads that share notes with the tonal center.

**3.3.4 Scholarship allocations** Scholarship allocations always have a certain duration and remuneration associated. The system maps this information into three independent levels and translates the money flow into water flow, overlapping sounds from various water currents. This is intensified by a modal progression that uses more chords if the level is higher. Since the parameter that is being represented (remuneration) translates into the tonality, we randomly select chords from a predefined array. The harmonic movement created provides a tonal center, no matter which chords are chosen. This is a crucial characteristic of the sonification model we are presenting. If the tonal center is established, the harmony is free to fluctuate within the options of the array.

**3.3.5 Paper submissions** The last type of event involves the mapping of the submission into a scale of importance according to the CORE Ranking, for conferences, and Scimago Journal Ranking, for journals. The metaphor links the paper's visibility to the sound of birds chirping as they get louder the more relevant the submission is.

### 3.4 Installation

*soundscape4dei* is installed in a room acoustically studied beforehand. The room is public and accessed by researchers and students. The sound system is composed of four speakers (Genelec 8010a) and a subwoofer (Genelec 7040a Active Subwoofer) as well as an interface (PreSonus Studio 1810c) which is connected to the desktop that runs the software (Figure 3).



**Fig. 3.** Installation mapping: subwoofer as a purple cube, speakers as red cubes.

The graphic identity is inspired both by the physical dimensions of the room and the distribution of the hardware. Each CISUC group is represented by a circle and consequently a colour. In order for the listener to share a deeper understanding of the project, we developed a website that contains live encoding details and an option for a temporary mute. This usability feature allows the installation room to remain a silent place for formal meetings.

The website and the processing sketch communicate via WebSockets within the safe university network. The logo is used both on the website and on a small physical artefact that interacts with the system (Figure 4). We use an **arduino uno** to blink the led that corresponds to the CISUC group responsible for the event being reproduced. This object works as a visual clue that stimulates the audience to investigate about what they are listening.



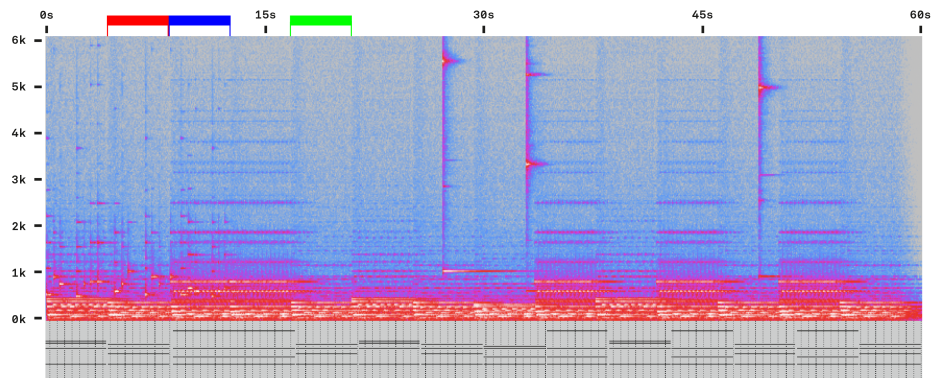
Fig. 4. Physical artefact (left) and *soundscape4dei* website (right).

## 4 Results and Analysis

In this section we analyse the sonification of the *Score* line (Figure 1) represented on the website of Figure 4. We start by analysing the output's spectrogram while referencing its MIDI layer L2. We analyse the harmonic movement and what sort of experience it helps to deliver. We conclude by comparing two recordings of the same *Score* line (Figure 1) by depicting the variations and identifying what differs and stands out in our sonification model.

To get a clear render of the recording we set the spectrogram's scale to linear, the algorithm window size to 8192, we chose the Hann algorithm window type and displayed frequencies between 0 and 6000 Hz. Figure 5 reveals four main incidents:

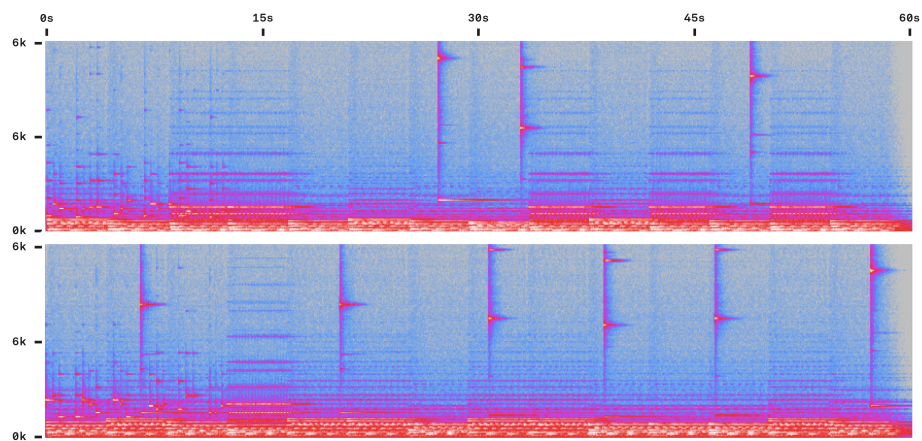
1. The occasional appearance of frequencies in 3k-6k range, caused by the VSTI that represents the assigned CISUC group (glockenspiel);
2. Some initial turbulence due to the Morse melody;
3. A large cluster of low frequencies (up to 1000 Hz). This characteristic is due to the harmony emphasis, more specifically, due to the choice of chord notes;
4. The chords also split the spectrum into temporal sections. These partitions are created by the chord progression (or pulse) like for example between the red and the blue markers. However, we can argue that these divisions do not negatively impact the resonance levels, since it remains similar throughout the recording.



**Fig. 5.** Spectrogram of a scholarship allocation sonification recording (top) and respective L2 MIDI (bottom).

All the described elements reveal that, regardless of whether there are places with more activity than others (such as between the sections comprised in the blue and green marker), L2 blends with the other layers of the soundscape L1 and L3, creating a relaxing, non-intrusive soundscape.

Figure 6 represents the variations of the same *Score* line and displays how the encoding methods affect the composition. The harmonic pulse is still present but progression is different, since the bottom spectrum is more stable after the 30 seconds mark while the top spectrum is not.

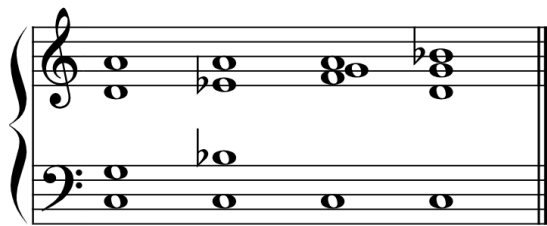


**Fig. 6.** Spectrogram from two recordings or iterations of the same *Score* line sonification.

## 5 Discussion

After presenting the system and analysing the results, we discuss our model. Baxter [17] argues that sonification may be a means to create generative music, but may not the opposite be also true?

Figure 6 revealed that the same *Score* line had two different outputs. One of the main reasons for that to happen is because the progression that is being developed is generated in real time. By having a list of possible chords, the system randomly selects which one is to be played. This does not compromise the sonification accuracy because the chord list guarantees that the outcome is always, in this case, a Dorian progression (Figure 7).



**Fig. 7.** Array of chords that can be selected and played in the recording portrayed in Figures 5 and 6.

We believe that combining standard parameter mapping with encoding algorithms that are partially stochastic creates and, arguably, enhances sonification systems. Our model and contribution works on top of multilayered compositions and is based both on the exploration of those stochastic processes and on testing how they may help to prevent stagnation, may create movement and surprise and may instigate and immerse the listener while they faithfully portray the intended data <sup>3</sup>.

## 6 Future Work

There are several options and approaches for future work related to the development of this model regarding *soundscape4dei*. From the current state we can divide it into three major streams: system evaluation, system enhancement and system design.

**6.0.1 System evaluation** So far, the system's evaluation has been solely qualitative and hasn't incorporated user input. Our aim is to conduct a comprehensive assessment of users' experience and explore whether sonification enhances awareness and insights. This will involve implementing mechanisms for

<sup>3</sup> <https://vimeo.com/824378644>

gathering feedback from individuals present in the room, such as installing a tablet or providing a web page accessible via mobile devices or laptops. Additionally, we plan to collect audio descriptors from the ambient sound in the room and examine potential correlations with user feedback.

**6.0.2 System enhancement** *Soundscape4dei* is currently able to sonify data from a specific dataset. We envision the development of a dynamic API that would feed the implemented system from a larger group of events which take place across the department, such as classes attendance, crowded rooms, masters' and doctors' thesis defences, social network publications, among others.

To avoid repetition and stagnation, the system could also incorporate more diverse sounds. For that matter, we plan to enrich the sound bank by recording original content, not only as a standalone work but also by inviting the community to contribute.

When approaching the project beyond its software components certain enhancements may arise. We plan to build an object that allows the community to locally share their perspective on the soundscapes. By gathering those inputs we can evaluate the system, and consequently the model. Physically, the installation can also be expanded into larger and more populated areas across the campus. Broadening *soundscape4dei* helps to disseminate the sonified activities and evaluate it from a larger community sample.

**6.0.3 System redesign** The implemented system portrays each entry for one minute. Having the activities chronologically separated and melodically identified helps to distinguish them but it also imposes a rigid premise. This trade off hints a new approach where all activities may coexist in the same sonic mist. In this scenario, the system allows for the creation of denser soundscapes. Sounds may be added or removed at any moment since there are no time restraints, which can be used to highlight certain events creating and manipulating the hierarchy. We envision this approach to create an opportunity to further develop our model, since the premise suggests a sort of compositional freedom while raising the challenge to accurately portray data.

## 7 Conclusion

We presented *soundscape4dei*, a system that creates sound compositions through the analysis of data and we discussed the model we propose. Throughout the development of this project, we have focussed on balancing exploration and functionality — the outputs portray data in unconventional manners by exploring sound and musical density without neglecting events recognition through sound. There are multiple options to be explored towards new versions of this system, while extending the reach of our model when applied to multilayered sonifications. Nevertheless, the installation fulfills its purpose, as is able to produce immersive soundscapes and locally share the activities from CISUC.



## References

1. L. Matthews, "The evolution of data visualization: From simple to complex," *Digitalist Magazine*, May 2020.
2. T. Hermann, A. Hunt, and J. Neuhoff, *The Sonification Handbook*. 01 2011.
3. G. Kramer, B. Walker, T. Bonebright, P. Cook, J. Flowers, N. Miner, J. Neuhoff, R. Bargar, S. Barrass, J. Berger, G. Evreinov, W. Fitch, M. Grohn, S. Handel, H. Kaper, H. Levkowitz, S. Lodha, B. Shinn-Cunningham, M. Simoni, and S. Tpei, "The sonification report: Status of the field and research agenda. report prepared for the national science foundation by members of the international community for auditory display," 01 1999.
4. G. Kramer, "Auditory display: Sonification, audification, and auditory interfaces," 1994.
5. B. Foo, "Two trains - sonification of income inequality on the nyc subway." <https://datadrivendj.com/tracks/subway/>. (accessed Apr. 15, 2023).
6. J. Cohen, M. Perteau, and E. Nawrocki, "Reproducible and robust quantification of RNA-seq reads using the Rsubread package," *PLoS Biology*, vol. 16, p. e2004044, mar 2018.
7. K. Matthews, "Sonic bike." <https://sonicbikes.net/sonic-bike/>. (accessed Apr. 15, 2023).
8. M. Seça, R. Lopes, P. Martins, and F. A. Cardoso, "Sonifying twitter's emotions through music," in *Music Technology with Swing: 13th International Symposium, CMMR 2017, Matosinhos, Portugal, September 25-28, 2017, Revised Selected Papers 13*, pp. 586–608, Springer, 2018.
9. M. Seça, P. Martins, L. Roque, and F. A. Cardoso, "A sonification experience to portray the sounds of portuguese consumption habits," Georgia Institute of Technology, 2019.
10. L. Hjorth, J. Silva, and A. Lanson, eds., *The Routledge Companion to Mobile Media Art*. Routledge, 2020.
11. B. House, "Quotidian record." [https://brianhouse.net/works/quotidian\\_record/](https://brianhouse.net/works/quotidian_record/). (accessed Apr. 15, 2023).
12. J. Bulley and D. Jones, "Living symphonies." <https://www.livingsymphonies.com/>. (accessed Apr. 15, 2023).
13. A. Giglio, R. Gorbet, and P. Beesley, "Hybrid soundscape human and non-human sound interactions for a collective installation," 09 2022.
14. E. C. Platt, "Ecological interference: A proposal," 2017.
15. D. J. Levitin, *Memory for Musical Attributes*, p. 209–227. Cambridge, MA, USA: MIT Press, 1999.
16. R. Rawlins, N. Bahha, and B. Tagliarino, *Jazzology: The Encyclopedia of Jazz Theory for All Musicians*. Jazz Instruction Series, Hal Leonard, 2005.
17. I. Baxter, *Sonification as a Means to Generative Music*. University of Sheffield, 2020.



# Interpretable Rule Learning and Evaluation of Early Twentieth-century Music Styles

Christofer Julio<sup>1</sup>, Feng-Hsu Lee<sup>2</sup>, and Li Su<sup>3</sup>

<sup>1</sup> Social Networks and Human-Centered Computing, Taiwan International Graduate Program

<sup>2</sup> Faculty of Creative Arts, University of Malaya

fenghsulee@um.edu.my

<sup>3</sup> Institute of Information Science, Academia Sinica

lisu@iis.sinica.edu.tw

**Abstract.** The paper discusses the classification of four music styles, Serialism, Impressionism, Neoclassicism, and Nationalism, of early-twentieth-century music using interpretable rule learning techniques. Three interpretable rule learning techniques are considered: decision tree, minimum description length (MDL) rule list, and rule set (the skope-rule algorithm). The features of the classifiers are fundamental musical elements based on pitch and interval distributions. Objective evaluation based on the F1 score and subjective evaluation using user study is conducted to understand the result of our classifiers from the musicians' point of view. The results show that a rule set is preferred as the algorithm attained the highest scores for objective and subjective evaluations. The rule set can also generate rules which support music theory and provide new insights regarding the musical characteristics of early twentieth-century music.

**Keywords:** Early twentieth-century music, interpretable AI, rule learning, evaluation, music information retrieval

## 1 Introduction

The studies regarding the classification task of classical music have undergone major development in the last few years. Multiple machine learning classifiers, from transparent models such as decision trees [12] to black-box models such as support vector machines [12, 11, 24] and more sophisticated neural networks [18, 23, 15, 16, 27], have been utilized and have effectively classified classical music across periods. In addition to focusing on good performance, another research endeavor has focused on interpretability, that is, the extent to which the process by which a model arrives at its decision is transparent and understandable by humans [12, 28].

Despite abundant research on classical music classification, few studies include composers from the early twentieth century. Instead of labeling the early twentieth-century compositions based on their respective styles, most researchers label the composers around this period as “modern” [23, 25]. This “modern” label may not be enough



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

to become an accurate representation. The early twentieth-century period in classical music consists of various musical genres – each highly distinctive from the others. Musicologists often categorize these styles as *-isms* [4, 7].

Indeed, early twentieth-century music has a few common concepts. For instance, the composers avoid constructed melodies from the previous periods and do not follow the standard tonal harmony [4, 5]. However, the approaches that each style made differ significantly from one another. Serialism, for instance, focuses on utilizing pitch set series or tone rows [4, 1]. Impressionism does not neglect tonality but maximizes the utilization of timbres, layers, underdeveloped motifs, unresolved harmony, and exotic music scales [4, 5]. Nonetheless, Neoclassicism combines the characteristics of music in the previous periods with modern melody and dissonance treatments.[17, 8]. Conducting the classification tasks over these music styles would be insightful due to the unique characteristics of early twentieth-century music and the scarcity of study for this period.

Instead of focusing on performance alone, this study aims more into the interpretability of the result [21]. We want to see whether there is any new insight regarding the characteristics of early twentieth-century music, which may not be found using conventional music analysis. Our research objectives are motivated by the previous studies that have shown the potential to find new insights into classical music, such as the difference in pitch distribution between Mozart and Haydn’s string quartet works [11] or the differences in interval utilization between Beethoven and the composers before him, such as Haydn and Bach [12]. On the contrary, the interpretable deep learning approaches for music classification and analysis [13, 26] mostly focus on *post hoc* interpretation [21] over the learned representations and still require decision trees, rules, and linear models to explain it under specific situations [10]. Therefore, we take the approach of rule-based, transparent, and *simulatable* (i.e., humans can reason about the entire decision-making process of the model [21]) models instead of black-box ones. Moreover, in this paper, we extensively study various categories of rule-based models, including rule tree, rule list, and rule set [20]. Besides the well-known decision tree, it should be noted that the rule list and rule set models we employed have yet to be considered in music classification problems [22, 9]. For evaluating the result, we perform not only objective evaluation but also subjective evaluation to understand the human’s perceptions regarding the rules generated by the models.

To our knowledge, this paper is the first attempt to machine learning classification of early twentieth-century music. This paper has three major aims. First, we propose a new dataset regarding early-twentieth-century music in symbolic format. Second, we investigate various rule-based machine learning models for music style classification on this new dataset. Lastly, the interpretability of the classification results and the selected rules and features are analyzed and discussed with both objective and subjective aspects.

## **2 Data**

Since the repertoires of early twentieth-century music are wide and complicated, we imposed several restrictions in choosing the works for the dataset. The subject of this research is limited to early twentieth-century composers’ piano works, as we tend to

Table 1: The proposed dataset for classification of the early 20th-century music styles. The number of samples of each composer and each style in the dataset are shown.

Styles	Composers	# of samples	Styles	Composers	# of samples	
Serialism	Arnold Schoenberg	22	Neoclassicism	Maurice Ravel	11	
	Alban Berg	2		Paul Hindemith	86	104
	Anton Webern	5		Béla Bartók	7	
	Hanns Eisler	19		Béla Bartók	247	
Impressionism	Claude Debussy	87	Nationalism	Leoš Janáček	43	304
	Maurice Ravel	23		Manuel de Falla	14	

use homogeneous data to avoid any potential problems related to instrumentation. This approach has also been used in previous studies, where the researchers limited their choice of instruments to only string quartet [11, 14, 24], the melody of the violin [6], piano solo [23, 25], and orchestra [25]. We included the first 20 measures (approximately one page) of every composition to prevent the imbalance of the dataset. Besides, we chose the styles and composers based on the number of piano pieces for each composer and the availability of the scores. Composers who only have a few piano works were not selected. In addition, we only choose the works in the public domain. Hence, the dataset consists of four styles and ten composers, see Table 1.

Except for Béla Bartók and Maurice Ravel, each composer’s compositions are classified in one style. Ravel’s works are divided into Impressionism and Neoclassicism, as Ravel had distinctive styles during his early and late period [4]. In addition, Bartók’s works with Nationalism style are chosen manually based on existing literature due to his unique approaches between the traditional and modernistic style [4]. Besides Nationalism, a few of Bartók’s piano works are also separated into Neoclassicism due to the use of classical forms. Other Bartók’s piano works, which do not fall into these two styles (such as Night music), are not included. Lastly, the pre-serialism works from Serialism composers, such as Schoenberg’s late romantic works, are not incorporated into the dataset.

The dataset of the early twentieth-century music in this study utilizes note events derived from musicXML, as it can save more information compared to MIDI. We collected the data from the Petrucci music library (imslp.org) and manually converted them to the MusicXML format. Note events, including pitch value, onset time, and duration, are then extracted. The dataset will be publicly announced after this paper is accepted.

### 3 Method

#### 3.1 Data representation

We consider pitch-related features and intervals as the data representation, as our pilot study demonstrated that they are more relevant than other music features for classifying our early twentieth-century dataset. [2]. Given a music piece  $\{x_i\}_{i=1}^N$  with  $N$  notes, the pitch value (in MIDI number) of the  $i$ th note being  $p_i$ , the pitch range ( $r_p$ ), pitch mean

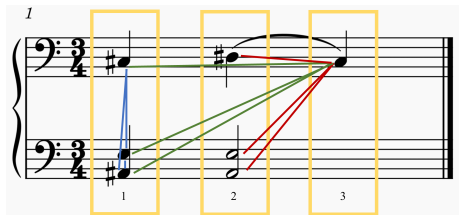


Fig. 1: The example of horizontal and vertical interval feature calculation. Excerpt taken from Bartók’s *9 Little Pieces for Piano no.5*.

$(\mu_p)$ , and pitch standard deviation ( $\sigma_p$ ) over all time steps are

$$r_p := \max_i(p_i) - \min_i(p_i); \quad \mu_p := \frac{1}{N} \sum_{i=0}^N p_i; \quad \sigma_p := \sqrt{\frac{\sum (p_i - \mu_p)^2}{N}}. \quad (1)$$

Then, vertical interval features are calculated to understand the harmony of the repertoire. For simplicity, the positions of notes are grouped based on the beats. By normalizing all the data to 4/4 meter, a *beat* refers to all notes within a quarter note duration. For example, given a music excerpt with  $Y$  beats, the position of a note  $x_i$  is  $y$ ,  $1 \leq y \leq Y$ , if its onset is in the interval  $[y, y + 1)$ , i.e., between the  $y$ th and the  $(y+1)$ th beat of the music piece. For any two notes  $x_i$  and  $x_j$  at the same beat  $y$ , assuming  $p_i \geq p_j$ , the vertical interval between  $x_i$  and  $x_j$  at  $y$  is 12 if  $p_i - p_j = 12n$ ,  $n \in \mathbb{N}$ , and is  $p_i - p_j \pmod{12}$  for other cases. That means a vertical interval is a value ranging from 0 (unison) to 12 (perfect octave). The distribution of the vertical intervals over all time steps is then represented as a 13-dimensional vector, obtained by aggregating the counts of each interval class over all the time steps. The final vertical interval feature (denoted as  $\bar{v}$ ) is a min-max scale normalization over this distribution.

In addition, we employ another feature based on the horizontal interval for understanding the relationship between neighboring notes. For the horizontal features, we consider two groups of notes by  $m$  beats apart from each other, where following the *skip-gram* technique in the field of natural language processing,  $m$  is the number of *skips*, and  $m = 0$  represents no skip. Similar to vertical interval, the horizontal interval of  $x_i$  and  $x_{i+m+1}$  (assuming  $p_i > p_{i+m+1}$ ), is 12 if  $p_i - p_{i+m+1} = 12n$ ,  $n \in \mathbb{N}$ , and is  $p_i - p_{i+m+1} \pmod{12}$  for other cases. Similar to the normalized distribution of vertical intervals, the normalized distribution of  $m$ -skip horizontal intervals (denoted as  $\bar{h}^{(m)}$ ) is also a 13-dimensional vector by aggregating the counts of each interval and min-max normalization.

The straightforward way of calculating the vertical and horizontal interval features is demonstrated by an example in Figure 1. There are three beats (indicated by the yellow boxes) in this example. The vertical interval calculations are shown in the blue lines. At the first beat, for the intervals from the note C#3, we calculate every possible interval in the same time stamp. Here, calculations are made from C#3 to E3 (i.e., a minor third, also denoted as “V-m3”) and from A#2 to C#3 (i.e., minor third or V-m3). The rest of the notes are treated similarly without repetitions; for example, at this beat,

we also have an interval between A $\sharp$ 2 and E3 (diminished fifth or V-d5). Summing up the vertical intervals over all the timestamps in Figure 1, we have in total one V-m2, two V-m3, two V-d5 and one V-P5, so the distribution is [0, 1, 0, 2, 0, 0, 2, 1, 0, 0, 0, 0, 0] and the min-max-normalized distribution is  $\bar{v} = [0, 0.5, 0, 1, 0, 0, 1, 0.5, 0, 0, 0, 0, 0]$ . Meanwhile, the red lines show the calculation of horizontal intervals without any skip. Our example here calculates the horizontal intervals between the second and the third beats, which result in three intervals: D $\sharp$ 3 to C $\sharp$ 3 (major second, or denoted as H-M2), C $\sharp$ 3 to E3 (H-m3), and A $\sharp$ 2 to C $\sharp$ 3 (H-P5). Lastly, the green line indicates the horizontal interval calculation with skips. In this example, we only demonstrate the interval calculation with one skip, i.e., between the first and third beats. The calculation results in three intervals: C $\sharp$ 3 to C $\sharp$ 3 (H-P1), C $\sharp$ 3 to E3 (H-m2), and A $\sharp$ 2 to C $\sharp$ 3 (H-m3). The method of summing up different timestamps is similar to the case of vertical intervals.

In the remainder of this paper, the number of skips is not specified if it is zero. To summarize, we consider the pitch features (pitch range, pitch mean, and pitch standard deviation, totaling three dimensions), vertical interval features (13 dimensions), and horizontal interval features with skips from 0 to 2 ( $13 \times 3 = 39$  dimensions). This results in a total feature dimension of 55.

### 3.2 Classifiers

We consider three categories of rule-based algorithms: rule trees (i.e., decision trees), rule lists, and rule sets. These three are interpretable in that they are all constructed with conditional statements (i.e., if-then-else rules) of the input features and the corresponding outcomes [10]. In decision tree, the if-then-else rules form a tree structure in which the internal nodes represent conditions of features, and each leaf node represents a class label. In rule lists and rule sets, each condition of an if-then clause can incorporate multiple input variables. Specifically, in rule lists, rules are the conditions ordered in nested if-else statements, while in rule sets, rules are unordered and independent from each other in that the else statements do not connect the rules [10]. As for visual representation, rule trees are often illustrated in tree graphs, while rule lists and rule sets tend to have textual or tabular representation. Hence, rule trees, rule lists and rule sets are not equivalent and are different in multiple aspects.

The rule tree classifier we adopt is the decision tree with the optimized CART Algorithm [3], available from the scikit-learn library.<sup>4</sup> For the rule list classifier, we utilize the minimum description length (MDL) rule list, a probabilistic multi-class classifier algorithm. MDL rule list is designed using the minimum description length principle, which chooses the best model based on the ability to compress the data [22].<sup>5</sup> The MDL Rule list requires only a few hyperparameters to work and can acquire competitive accuracy [22]. Lastly, for the rule set classifier, we utilize skope-rules.<sup>6</sup> Similar to Rulefit [20], the rules from Skope-rules are chosen by extracting the path of the tree from multiple decision trees. However, the difference lies in establishing the final rules.

<sup>4</sup> <https://scikit-learn.org/stable/modules/tree.html>

<sup>5</sup> <https://github.com/HMProenca/MDLRuleLists>

<sup>6</sup> Source code available at <https://github.com/scikit-learn-contrib/skope-rules>

Table 2: Classification results using the three rule-based classifiers on the four styles of early 20th-century music. Precision (P), recall (R) and F1-score (F1) values are shown.

	rule tree			rule list			rule set		
	P	R	F1	P	R	F1	P	R	F1
Serialism	0.80	0.59	0.68	0.68	0.68	0.68	0.99	0.73	0.84
Neoclassicism	0.60	0.60	0.60	0.52	0.42	0.47	0.84	0.56	0.67
Impressionism	0.56	0.57	0.57	0.55	0.27	0.37	0.68	0.63	0.66
Nationalism	0.62	0.67	0.64	0.57	0.97	0.71	0.83	0.58	0.68
Average	0.65	0.61	0.62	0.58	0.59	0.56	0.84	0.63	0.71

Table 3: The extracted rule set of four classes

Rules 1	Rules 2	Rules 3	Rules 4	Class
Pitch Range > 45.5	H-M7 (2 skip) > 0.17	V-m7 ≤ 0.39	V-P8 ≤ 0.8	Impressionism
Pitch Range ≤ 52.5	V-P1 > 0.15	V-P8 > 0.23	H-m2 > 0.05	Nationalism
H-P8 ≤ 0.004	H-M7 (1 skip) > 0.05	H-M6 (2 skip) > 0.15	VI 4 > 0.49	Serialism
H-A4 ≤ 0.5	H-P8 > 0.01	V-m7 > 0.4	V-M7 > 0.3	Neoclassicism

Skope-rules filter the rules using out-of-bag (OOB) precision and recall thresholds and the semantic deduplication method for maintaining the diversity of the rules [20].

The hyperparameters utilized in this study are described as follows. For decision tree, we use a maximum depth of four, gini impurity as the criterion, and minimum sample split as two. The rest are followed by the default settings of Scikit-learn’s Decision Tree. Meanwhile, the parameters used for the MDL rule list classifier is elaborated as follows: static data discretization, the maximum size of each rule description being 4, the number of cut point of each variable being 1, minimum support being 0.1, and alpha gain being 0. Lastly, for Skope-rules, we utilize similar hyperparameters with Decision Tree, except that we limit the estimated number of the generated tree to 72.

## 4 Experiments

### 4.1 Experimental settings

Before training the classifiers, data augmentation has to be done since the size of the dataset is considered small. In this case, a normalization process is performed as suggested by [11], converting every sample’s key into C major and a minor. However, these adjustments are strictly for tonal music, and this conversion step is skipped for atonal works. Then, we perform pitch shifting from -5 to 6 semitones for each work. To balance the dataset, we randomly select 35 percent of Nationalism samples due to their larger number. The dataset is then divided into training and testing sets with the 80:20 ratio. Lastly, considering the number of samples, 5-fold cross-validation (CV) is performed for each experiment to obtain stable classification results. The test-set precision, recall, and F1-score values averaged over the 5-fold CV are reported and compared.

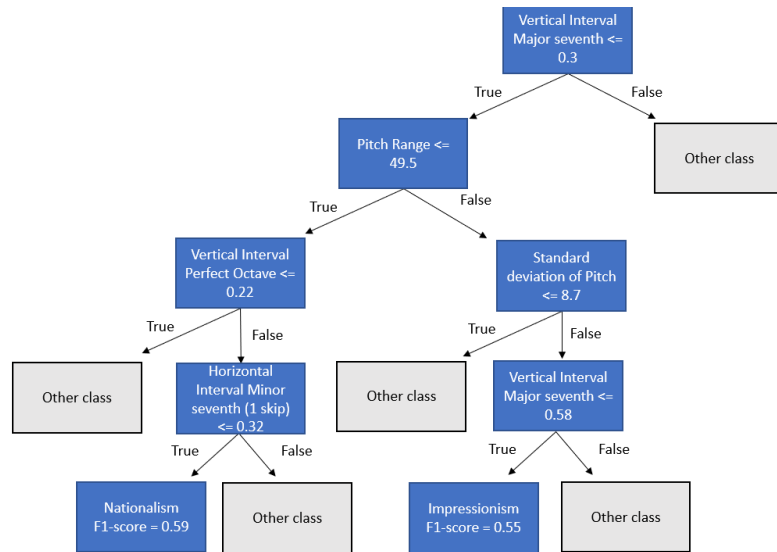


Fig. 2: Partially extracted tree of Nationalism and Impressionism class. "Other class" denotes the weak or irrelevant class which is neither Nationalism nor Impressionism.

- ↳ 19 list of ELSE IF with low impact results
- ELSE IF Pitch Range < 43.0 AND Vertical Interval Perfect Octave >= 0.36 AND Horizontal Interval Perfect Fourth (2 skip) >= 0.4 AND Vertical Interval Minor Second < 1.0 THEN Probability of Nationalism = 1.0
- ↳ 12 list of ELSE IF with low impact results
- ELSE THEN Probability of Impressionism = 0.96; Probability of Neoclassicism = 0.04

Fig. 3: The extracted rule list of Nationalism and Impressionism Class. The hidden lists contain other  $n$  rules which have low impacts to the classification decisions.

## 4.2 Objective evaluation

Table 2 shows the classification result of decision tree, MDL rule list, and Skope-rules. Skope-rules achieves an average F1-score at 0.71, outperforming both decision tree (F1-score = 0.62) and MDL rule list (F1-score = 0.56) by a wide margin. Meanwhile, we have slightly different results for the F1-score of each class. Skope-rules still dominate in Serialism, Neoclassicism, and Impressionism classes, followed by decision tree. However, for Nationalism, the MDL rule list achieves a better result than the other two, with F1-score = 0.71. Lastly, the results of both Neoclassicism and Impressionism of the MDL rule list are underwhelming, with the F1-score less than 0.5.

## 4.3 Subjective evaluation

A user study in the form of a questionnaire is utilized to understand the interpretability of the models for musicologists, musicians, composers, and other music-related

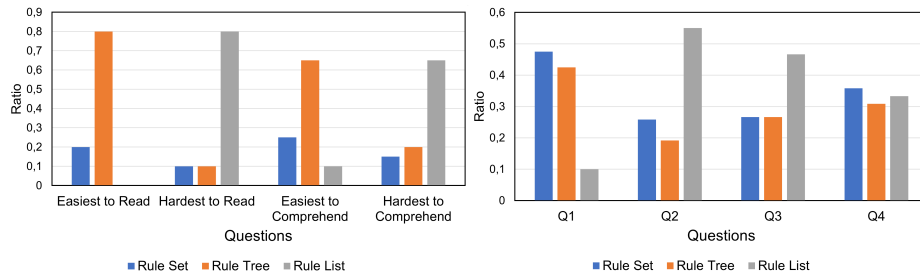


Fig. 4: Results of the subjective tests. Left: visualization test. Right: content test.

researchers. The questionnaire of our subjective evaluation contains two parts, *visualization test* and *content test*. The first part evaluates the subjective response to the *visualization* quality of the rules. Rule tree is visualized with a decision tree graph (e.g., Figure 2) while rule list is represented by text, consisting of the if, else-if, and else rules (e.g., Figure 3). Lastly, rule set is represented by a table (e.g., Figure 3). The questions then aim to understand the most favorable representation of the visualization results based on the opinions of respondents. Since the 5-fold CV generates five different lists, trees, and sets, we decided to take the best graphs or rules based on the best F1-score of the folds for the questionnaire questions.

The second part of the subjective test evaluates the *content* of the generated rules of each class. In the second part of the subjective test, we present  $C_2^4 = 6$  question sets of rule tree, rule list, and rule set based on the binary classification; the six sets contain each of the two styles selected from the four musical styles for pairwise comparison. Each set consists of four questions. They are (Q1) From the three options, which one gives the best result according to current music theory? (Q2) From the three options, which one gives the worst result according to current music theory? (Q3) From the three options, which one gives the most unusual rules? (Q4) From the three options, which one gives the least unusual rules?

20 participants joined the subjective test. 18 of them have a degree in music. Among the participants, 13 have more than 11 years of experience in music. On a scale of 1-5, 7 participants are very familiar with early twentieth-century music (scale 4-5), while 10 participants are familiar with early twentieth-century music (scale 3). Only 3 participants are quite unfamiliar with early twentieth-century music (scale 2).

The left-hand side of Figure 4 shows the result of the subjective test. For the visualization test, rule tree is the model representation that is easiest to read, followed by rule set. Meanwhile, rule list is the hardest to read. Similarly, among the three models, the rule tree is also the most comprehensible, followed by rule set and rule list.

The right-hand side of Figure 4 shows the result of the content. In line with the result of the visualization test, the answers to Q1 and Q2 of the content test show that most participants favor rule set and rule tree over rule list. However, unique results are seen based on the answer of Q3. Even though rule set has the highest ratio in Q1, it turns out rule set also occupies second place in Q3. It means that although rule set has rules strongly similar to current music theory, some are also considered unusual.



## 5 Discussion

### 5.1 The Subjective and Objective Evaluation Analysis

The objective and subjective evaluations conducted in this study show several similar trends. The visualization and content test show identical results regarding the most accurate classifier among the three representations. Skope-rules appears to be the best classifier with the F1-score = 0.71, and the classifier shows the best result for the current music theory based on the *content* test (see Q1 and Q2 in Figure 4). Meanwhile, the rule tree comes second with F1-score = 0.62, with the second-best accuracy towards the current music theory. On the other hand, the rule list becomes the worst classifier among these trees with the lowest F1-score, lowest Q1, and highest Q2 value of *content* test. The Q3 answers show that rule list generates the most unusual rules compared to others. There may be two possible explanations regarding this matter. First, the unusual rules may be the signs of the new possible finding regarding the theory of musicology. Second, the rules from rule list may not be accurate because it occupies the lowest F1-score in objective evaluation. However, at the current state of the study, we are unable to identify whether these found rules from the rule list are truly insightful, and further investigations are required. Lastly, based on the Q4 of the content test, no clear trend was found.

Meanwhile, regarding the interpretability of the rules, we still observe contrasting outcomes in between visualization tests. Based on the result of the visualization test (Figure 4), rule tree offers better comprehensibility and readability compared to rule list and rule set. Rule set comes second despite having the highest precision, recall, and F1-score on the objective evaluation. The result in our case shows that a higher F1-score does not always imply better interpretability. This is possibly due to data representation: the tree data structure in rule tree has the advantage of showing the relationship between the classes. For instance, in Figure 2, the readers can easily notice the distinctions of Nationalism and Impressionism classes directly from the ramification on the first depth onward. Meanwhile, for the rule list and rule set, the readers need to compare each rule one by one. In addition, the rules generated from the rule tree always show at least one related feature of both classes (see Figure 2) since, in the tree model, two child nodes always have at least one shared parent node. On the contrary, in both rule list and rule set, there are possibilities that all features of both classes are distinctive. Readers may be confused in comparing the rules if all the rules between classes are unrelated.

The rule list shows the most inferior performance from both the subjective and objective perspectives: The average F1-score is only 0.56 (although it performs the best in Nationalism), and it is the hardest to read, comprehend, and the worst according to music theory. This might be due to data representation: there is a possibility that even though the rule list may produce reasonable rules, the subjective evaluation participants tend to choose other models due to the unfamiliarity of the respondents with the IF-ELSE concept in Figure 3, which are computer science rather than musical knowledge.

Based on the subjective and objective evaluation results, rule set shows the best accuracy in the F1-score and the content test while the outcome of the visualization test still indicates the potential of the rule tree as a good representation that favors music practitioners and musicologists. Besides, the results of the rule list are least favorable.

Table 4: The features of the four random-chosen excerpts. The green color shows that the feature fits the rule set and the red color shows that the feature unfits the rule set.

Example (Composer)	C. Debussy	B. Bartók	A. Schoenberg	P. Hindemith	
Class	Impressionism	Nationalism	Serialism	Neoclassicism	
Impressionism	Pitch range > 45.5	73	43	63	72
	H-M7 (2 skip) > 0.17	0	0.12	0.13	0
	V-m7 ≤ 0.39	0.22	0.06	1	0.5
	V-P8 ≤ 0.8	1	0.16	0	0.74
Nationalism	Pitch range ≤ 52.5	73	43	63	72
	V-P1 > 0.15	0.007	0	0	0
	V-P8 > 0.23	1	0.16	0	0.74
	H-m2 > 0.05	0.17	0.62	0.99	0.39
Serialism	H-P8 ≤ 0.004	0.74	0	0	0.13
	H-M7 (1 skip) > 0.05	0	0.02	0.3	0
	H-M6 (2 skip) > 0.15	0.46	0.09	0.29	0.12
	V-M3 > 0.49	0.65	0.35	0.53	0.84
Neoclassicism	H-A4 ≤ 0.5	0.21	0.38	0.97	0.11
	H-P8 > 0.01	0.74	0	0	0.13
	V-m7 > 0.4	0.22	0.06	1	0.5
	V-M7 > 0.3	0.03	0.19	0.54	0.39

## 5.2 Case Study

In this part, we perform a case study to see how the learned rules work on real-world music examples. We randomly select four excerpts from our dataset to represent each respective style. The music pieces are the excerpts chosen from Claude Debussy’s *Nocturne*, Béla Bartók’s *Nine Little Pieces for Piano*, Arnold Schoenberg’s *Suite for Piano* and Paul Hindemith’s *Ludus Tonalis*.

The rule set on Table 3 is utilized in the case study since rule set is the most recommended algorithm according to our previous discussion. The details of the chosen excerpts and the values of those features which appear in the rules on Table 3 can be seen in Table 4. Although the statements of feature in the rule set are combined with the logical AND, we discuss the feature separately for convenience. For example, the pitch range of Debussy’s *Nocturne* fits the rule “Pitch range > 45.5” for Impressionism since its value is 73. In addition, the pitch range of Bartók’s piece also fits this rule since it is non-Impressionism, and its value, 43, is smaller than 45.5. As a result, the value with the green color in Table 4 indicates that it fits the rule description of the music style, while the value with the red color shows that it does not fit the rules.

The results in Table 4 indicate promising outcomes. For Debussy’s *Nocturne* (Impressionism) and Bartók’s *Moderato* (Nationalism), two out of the four rules predict the style label correctly. Meanwhile, for Schoenberg’s *Präludium* (Serialism) and Paul Hindemith’s *Preludio* (Neoclassicism), all the rules are correct. It should be noted that current music theory does support certain generated rules on Table 3. For instance, the pitch range of Impressionism is supposed to be larger than 45.5 semitones, and Debussy’s *Nocturne* fulfills the requirements. The requirement of such a wide pitch range

may be correlated with the main characteristics of Impressionism piano works; among them are “open chord, wide spacing, and extreme register” [19, p. 169].

Meanwhile, some other results demonstrate unusual rules based on the perspective of musicology. For instance, the rule from the rule set shows the importance of V-M3 (major third) in Serialism composition. However, Serialism works do not stress the utilization of the major third since Serialism composition utilizes intervals based on the tone rows [4]. Lastly, another issue that concerns us is that some rules generated by the rule set are very weak. For instance, one of the Neoclassicism rules states that the normalized value of H-P8 (perfect octave) needs to be larger than 0.01, a very small lower threshold. Therefore, such a rule may not be as insightful since any music piece that merely utilizes a few numbers of the horizontal perfect octave interval might satisfy it. Hence, in certain parts, such rules do not highlight the important characteristics of the styles but are redundant. However, certain weak rules are still able to show some insights. For example, the rule H-M7 (skip 1)  $> 0.05$  seems weak given that 0.05 is a small lower threshold. However, based on observations, the excerpts in the other three styles have this feature smaller than 0.05, meaning that the horizontal major seventh interval with skip 1 rarely appears except in Serialism.

## 6 Conclusion

To conclude, the systematic study of rule-based interpretable algorithms for classifying the styles of early twentieth-century music indicates that the rule-set-based algorithm, Skope-rules, shows the best performance in the precision, recall, and F1-score of the objective evaluation and also offers decent comprehensibility and consistency of music theory in the subjective tests. In addition, the chosen algorithm with our feature design can find the rules in line with the current music theory, as well as the promising unusual rules which may show new insights regarding early twentieth-century music. Rule tree, on the other hand, is able to provide the best result in visualization test, yet is unable to outperform rule set in other evaluation sections. Thus, while the previous studies on rule-based interpretable music AI mostly considered decision trees, we suggest using rule-set-based algorithms for related research directions.

## References

1. Auner, J.: *A Schoenberg Reader: Documents of a Life* (2003)
2. Author, A.: (Anonymized for double-blind review). Master's thesis, Anonymous school (2022)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *CART: Classification and regression trees*. Routledge (1984)
4. Burkholder, J.P., Grout, D.J., Palisca, C.V.: *A History of Western Music: Tenth Edition*. New York, NY: W.W Norton & Company. (2019)
5. Clark, N.A., Heflin, T., Kluball, J., Kramer, E.: *Understanding music: Past and present*. University System of Georgia, University Press of North Georgia (2015)
6. De Carvalho Junior, A.D., Batista, L.V.: *Composer classification in symbolic data using ppm*. In: 11th International Conference on Machine Learning and Applications. vol. 2, pp. 345–350. IEEE (2012)

7. Frisch, W.: *Music in the Twentieth and Twenty-First Centuries*. New York, NY: W.W Norton & Company. (2013)
8. Gabriela, V.: Baroque reflections in *Ludus Tonalis* by Paul Hindemith. In: 11th WSEAS International Conference on Acoustics & Music: Theory & Applications (AMTA'10). vol. 1 (2010)
9. Gardin, F., Gautier, R., Goix, N., Ndiaye, B., Schertzer, J.M.: Skope-rules. <https://github.com/scikit-learn-contrib/skope-rules>
10. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM Comput. Surv.* **51**(5) (aug 2018). <https://doi.org/10.1145/3236009>, <https://doi.org/10.1145/3236009>
11. Herlands, W., Der, R., Greenberg, Y., Levin, S.: A machine learning approach to musically meaningful homogeneous style classification. In: *AAAI Conference on Artificial Intelligence*. vol. 28 (2014)
12. Herremans, D., Martens, D., Sørensen, K.: Composer classification models for music-theory building. In: *Computational Music Analysis*, pp. 369–392. Springer (2016)
13. Kelz, R., Widmer, G.: Towards interpretable polyphonic transcription with invertible neural networks. arXiv preprint arXiv:1909.01622 (2019)
14. Kempfert, K.C., Wong, S.W.: Where does Haydn end and Mozart begin? Composer classification of string quartets. *Journal of New Music Research* **49**(5), 457–476 (2020)
15. Kim, S., Lee, H., Park, S., Lee, J., Choi, K.: Deep composer classification using symbolic representation. arXiv preprint arXiv:2010.00823 (2020)
16. Kong, Q., Choi, K., Wang, Y.: Large-scale MIDI-based composer classification. arXiv preprint arXiv:2010.14805 (2020)
17. Kostka, S.: *Materials and techniques of twentieth-century music*, 3rd ed. Pearson Prentice Hall (2006)
18. Micchi, G.: A neural network for composer classification. In: *International Society for Music Information Retrieval Conference (ISMIR)* (2018)
19. Miller, H.M.: *History of Music*. Barnes & Noble Books (1973)
20. Molnar, C.: *Interpretable machine learning: A Guide for Making Black Box Models Explainable*. Lulu.com (2019)
21. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* **116**(44), 22071–22080 (2019)
22. Proença, H.M., van Leeuwen, M.: Interpretable multiclass classification by mdl-based rule lists. *Information Sciences* **512**, 1372–1393 (2020)
23. Van Spijker, B.: *Classifying classical piano music into time period using machine learning*. Master's thesis, University of Twente (2020)
24. Velarde, G., Weyde, T., Chacón, C.C., Meredith, D., Grachten, M.: Composer recognition based on 2d-filtered piano-rolls. In: *International Society for Music Information Retrieval Conference (ISMIR)*. pp. 115–121. International Society for Music Information Retrieval (2016)
25. Weiß, C., Mauch, M., Dixon, S., Müller, M.: Investigating style evolution of Western classical music: A computational approach. *Musicae Scientiae* **23**(4), 486–507 (2019)
26. Won, M., Chun, S., Serra, X.: Toward interpretable music tagging with self-attention. arXiv preprint arXiv:1906.04972 (2019)
27. Yang, D., Tsai, T.: Composer classification with cross-modal transfer learning and musically-informed augmentation. In: *International Society for Music Information Retrieval Conference (ISMIR)*. pp. 802–809 (2021)
28. Yu, H., Varshney, L.R., Garnett, G.E., Kumar, R.: Learning interpretable musical compositional rules and traces. In: *ICML Workshop on Human Interpretability in Machine Learning (WHI)* (2016)

# Toward empirical analysis for stylistic expression in piano performance

Yu-Fen Huang<sup>1</sup> and Li Su<sup>1</sup>

Music & Culture Technology Lab (MCTL), Institute of Information Science,  
Academia Sinica, Taiwan

yfhuang@iis.sinica.edu.tw

lisu@iis.sinica.edu.tw

**Abstract.** In the performance of Western art music, musicians apply various strategies to manipulate the performed sound, and communicate their musical interpretations via these subtle acoustic variations. It is a common practice for musicians to use typical conventions to express each compositional style (e.g. Baroque, Classical, or Romantic compositions). However, such stylistic expressive conventions has yet been fully discussed in previous research. In this initial foray, we systematically compare the expressive strategies for different piano compositions. A series of piano performance are recorded with a controlled experimental setting (3 compositions  $\times$  8 pianists  $\times$  3 repeated trials = 72 recordings), and expressive acoustic elements are derived using Music Information Retrieval techniques. In our analysis, we reveal that expressive manners in music performance exhibit stable and systematic features corresponding to each music composition, and those stylistic trends serve as empirical observations for typical performance conventions in different music styles.

**Keywords:** expressiveness, performance style, piano performance, computational musicology, Music Information Retrieval

## 1 Introduction

In the past decades, the way how music audiences approach, appreciate, and get to understand music has been evolved with the revolution of digital technologies. From attending physical concerts, purchasing audio/video medium (e.g. CD, DVD), to getting access to large amount of digitized performance recordings via online streaming services, audiences embrace the opportunities to explore the variety of music performance. In the context of Western art music, musicians have the privilege to interpret the written composition, and to communicate their understanding of the music piece via intricate variations in their performance (e.g. micro-timing, dynamic, timbral, and articulation arrangement). Audiences also enjoy the process to contemplate and compare diverse artistic variations in different performance versions, and through which process to discover potential insights for classical repertoire. The artistic, expressive variations in music performances therefore serve as an essential communicative vehicle to deliver musical ideas in the cultured convention.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Recently, systematic studies for large-scale music performance corpus are advanced with the facilitation of Music Information Retrieval (MIR) techniques. Computational models have been established to map general connections between musical attributes (e.g. note length, note pitch, phrase position) and expressive variations in performance [12] [16]. From musicological aspect, scholars focus on specific genres (e.g. Baroque music) and examine how particular aesthetic styles in music are shaped by artistic variations in performance [9]. Musicians also contribute their creative and unique interpretations in the process of performance execution. [18] [33]. The execution of music performance is therefore a complex and interactive process across aforementioned factors - the composition, stylistic convention, and individual musicians. The relationships between these aspects and how they interact together, however, have yet been fully discussed in previous research, particularly with systematic analysis of individual factors.

In this study, we aim to provide context-valid observations in terms of the interactions between different factors leading to musical expressiveness. A piano performance corpus is collected under a controlled experimental setting, and Music Information Retrieval (MIR) techniques are applied to retrieve expressive variations in tempo and dynamics. The performance variations are analyzed in conjunction with compositional elements through statistical and time-series methods. We identify important factors to induce unique music expression, and then subsequently investigate how those factors interact in different performing contexts. The contribution of this work is threefold:

- To compile a new piano performance dataset with controlled experimental design for comparison;
- To identify different key factors affecting music expression in individual scenario via statistical analysis;
- To provide empirical observations of how different factors interact together in a time-series process.

In the next section, we will discuss previous studies regarding musical expressiveness. The data collection and data processing procedure of this study will be reported in Section 3. In Section 4, we will investigate stylistic expressive trends found in individual compositions, and distinctive expressions bound for different compositional elements. In Section 5, our analysis results will be discussed in conjunction with findings in previous research, and we therefore suggest that our analysis can be implemented as an empirical approach to describe systematic variations for different expressive styles in music performance.

## **2 Related work**

The expressiveness in music performance is shaped by complex interactions among diverse factors. In the context of Western art music, composers follow conventional rules to construct the melodic, rhythmic, and harmonic configurations of music (*compositional factor*) [1] [23], and each composer's work would exhibit distinctive character according to the composer's preference (*stylistic factor*) [6]. During the performance process, musicians have their unique fashion to communicate personal musical interpretations, and control the variations in performed acoustic sound (e.g., micro-timing,

dynamic, and articulation variations) (*musician factor*) [25] [26]. GERMS model systematically categorized different origin of musical expressiveness including: generative rules, emotional expression, random variations, motion principles, and stylistic unexpectedness [19]. We will review related works regarding three different origin of musical expressiveness in this section.

### **2.1 Compositional factors in music performance**

In previous studies, rule-based models are established to describe the connection between compositional elements and expressive variations. The KTH model combines generative rules in melodic, harmonic, rhythmic, and phrase aspects to predict the timing, dynamics, and articulation execution in performance [12]. For piano performance, rule-based models and linear Gaussian models can be applied to jointly predict the tempo, dynamic, and articulation variations in performance according to multiple attributes in melodic, rhythmic, and harmonic aspects of performance [11]. Based on large amounts of jazz performance data, inductive logic rules for expressive elements (note onset deviation, dynamic variation, and ornamentation) are found in jazz music [14]. It is also found that tempo and dynamic variations interact together in music performance, and the tempo-loudness trajectory is an effective description to illustrate distinctive features of performance style [4].

Another cluster of studies apply machine learning approach to explore potential relationships between compositional elements and expressive variations. The connection between expressive tempo variations and musical phrase is mapped using Gaussian Mixture Models (GMMs)[24]. Models with transitional hidden state are applied to predict expressive variations according to score-informed attributes. For instance, Hidden Markov Model (HMM) and Hierarchical HMM are used to estimate the expressive variations in piano performance [17]. Conditional random fields (CRFs) are applied to predict expressive elements based on melodic and harmonic components [20]. Feed Forward Neural Networks (FFNNs) are used to predict the dynamic variations based on local-level score-informed attributes including the pitch, duration, and the note's relative interval with neighboring notes [3]. Linear and non-linear models for musical expression are systematically evaluated in [2], and it is concluded that compared to linear models, non-linear models have better performance to estimate the tempo and dynamic changes in music performance.

### **2.2 Stylistic factors in music performance**

Computational models are also built to explore specific performance styles. Restricted Boltzmann Machines (RBMs) are capable of predicting expressive accentuations in piano performance [36]. For solo violin performance, long-term dynamic variations can be successfully modelled using Random Forest, k-nearest neighbors (k-NN), and Support Vector Machine (SVM) [15]. For string quartet, the timing deviation, dynamic level, and the extent of vibrato in performance can be estimated based on melodic (e.g. relative interval) and rhythmic (e.g. metrical hierarchy) descriptors using model trees, k-NN, and SVM [21]. For jazz music, Decision Tree, SVM, and Neural Network (NN) are developed to formulate the stylistic deviations in jazz performance, and the improvised embellishments can be predicted from attributes including the chord type, note duration, and phrase [14].










(1) Bach:	(2) Mozart:	(3) Beethoven:
Well-tempered Clavier, Book 1, Prelude 1, BWV. 846.	Piano Sonata no. 11, KV.331, mov.1.	Piano Sonata no. 21, Op. 53, mov. 1 (Exposition).
Bar 1-8	Theme A (bar 1-4)	Theme 1 (bar 1-4)
		
	Theme B (bar 24-28)	Theme 2 (bar 35-38)
		
	Theme C (bar 32-36)	Coda (bar 75-77)
		

Fig. 1. The repertoire for data collection: three piano solo works by Bach, Mozart and Beethoven.

### 2.3 Musician factors in music performance

Musicians have their personal expressive manners in music performance. In order to compare different performance versions for the same composition, entropy-based deviation measures are used to describe expressive timing patterns in individual performance versions [25]. Hierarchical clustering is also an useful implement to distinguish different trends of performing styles for orchestral works [26]. For violin performance, individual violinists have their own expressive strategies to convey melodic patterns and phrase structure in the composition [29]. For jazz music, different performance styles by individual musicians can be successfully distinguished based on their intra-note features (e.g. note's attack level, sustain duration, amount of legato, spectral centroid, spectral tilt), inter-note features (e.g. relative pitch and duration to the neighboring notes), and note-to-note transition (pitch contour) [34].

The style of music performance is highly idiosyncratic according to the music genre, the music composition, and the musician. In particular, in Western art music, it is a common practice for pianists to play compositions in Baroque, Classical, and Romantic period following distinctive conventions. Aforementioned studies tends to explore a specific aspect of musical expression (i.e. either compositional, stylistic, or musician factor alone). Yet in music performance, the interactive, dynamical process among different factors work together to shape diverse variations. Based on the foundation of previous research, this work collects a series of piano performance data, and systematically analyze how individual factors interplay together to shape the overall musical expressiveness.

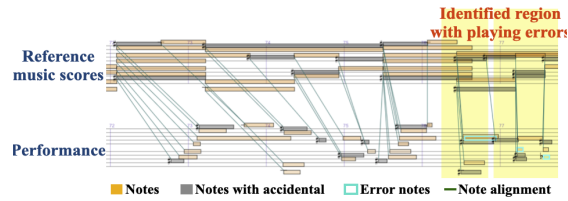
## 3 Method

In order to systematically examine how different factors affect the expressive variations in piano performance, a series of piano performance data are collected in this study. In this section, we report the procedure for data collection, and the data processing methods to extract expressive variations from recorded performance audio.

### 3.1 Data collection

Eight pianists are recruited to participate the recording sessions. The recruited participants are graduate/undergraduate students majoring in piano in music department at university (male = 4, female = 4). We reached out the participant pool via personal contact of music department staff. Participants are all right-handed, with the average age of





**Fig. 2.** The automatic audio-to-score alignment process to extract note timing from performance.

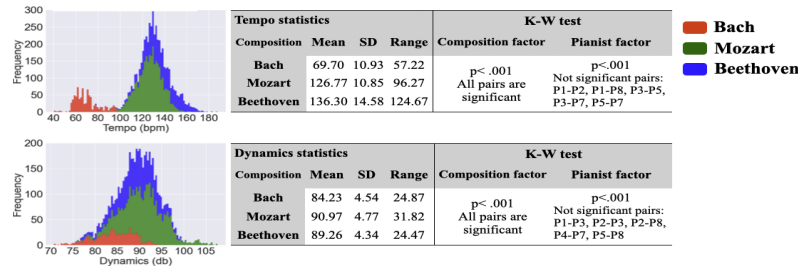
21.63 (SD = 0.70). They have learnt piano for 14.88 years in average (SD = 2.42), and practice for 3.75 hours in average per day at the time of data collection (SD = 1.84).

In order to compare different performance styles, piano solo works written by different composers are selected as materials (Fig. 1). Repeated measures design is used to observe if pianists show stable features when performing the same piece of music multiple times. The three music pieces selected for data collection are Bach Well-tempered Clavier, Book 1, Prelude 1, BWV.846; Mozart Piano Sonata no. 11, KV. 331, mov.1, and Beethoven Piano Sonata no. 21, Op. 53, mov.1 (bar 1-86, Exposition). In our experiment, each pianist played each music piece for 3 times in a random order, which resulted in 72 performance recordings (3 music pieces x 3 performances x 8 pianists). The recording sessions took place at the Motion Analysis Lab (National Yang Ming Chiao Tung University, Taiwan), where the performances were recorded on Yamaha digital piano P-115. In order to accurately align each notes in the performance with the music scores in the subsequent stage of data analysis, the performance is recorded as both midi and audio formats. During recording sessions, participants' body were also attached to optical markers to record 3-d motion capture data for their performance body movement, which will be further analyzed elsewhere.

### 3.2 Data processing

The dynamic and tempo variations in piano performance are the two expressive features to be analyzed in our subsequent investigation. In order to derive our target features, the first step is to perform audio-to-score alignment and to obtain the time stamp for each note in the performance. The audio-to-score alignment for polyphonic music, particularly with casual playing errors and asynchrony between two hands in piano performance, has been considered as a challenging task in Music Information Retrieval [5] [13] [30]. We applied an automatic alignment method to align recorded midi files and music scores files, in which Viterbi algorithms are used to exclude playing error regions and alignment errors in a pre-alignment process, and then hidden Markov models are applied to divide notes playing by two hands and accurately re-align each note based on the merged information [31] (see Fig. 2).

In order to associate expressive variations with the overall compositional structures, we analyze the dynamic and tempo variations at bar-level instead of instant dynamics and tempo. For tempo variations, the timing for each beat is extracted from the audio-to-score alignment data. The note information (e.g. the note pitch, duration, bar and beat position) is extracted from xml files using Python library Music21 [7], and such information from music scores serves as the reference to locate corresponding onset timing for each beat in the performance. In case of absent note on downbeat, linear interpolation is performed based on the timing of neighboring beats; in case of the asynchrony



**Fig. 3.** The data distribution and statistics for two expressive elements (tempo (upper), dynamics (lower)) in piano works written by Bach (red), Mozart (green), and Beethoven (blue).

between two hands, the note with the lowest register is taken as the reference. The *tempo* per bar is then defined as the average bpm (beat per minute) per bar. For dynamic data, the decibel is computed from the input audio (sampling rate = 22050 Hz) using Python library Librosa [28]. In order to eliminate the disturbance of local noise, moving average (window size = 220 samples, roughly 0.01 seconds) is used to smooth the original data. Since each note has a natural ADSR (attack-decay-sustain-release) curve, and the maximum volume on the attack is the main feature concerned, the *dynamic* level per bar is defined as the maximum decibel within a bar duration. Conventional music analysis is performed on the music scores, and structural features of music works including the harmonic progression, phrase and sectional boundaries are analyzed to be compared with features extracted from performance audio.

## 4 Analysis results

Through the data collection and processing procedure, a series of piano performance data are collected, and expressive tempo and dynamic variations are extracted. In this section, we report our analyses and observations for stylistic expression in three aspects: 1) the general expressive manners, 2) the interaction between different compositional elements and expressive features, and 3) the time-series expressive trends found in individual compositions, which can be regarded as individual stylistic expressive strategies attached to the composition.

### 4.1 General expressive manners

In our analysis, it appears that pianists use different strategies to express each composition. In Fig. 3, different tendencies of pianists' expressive manners can be observed in the distributions and statistics of tempo (the upper panel) and dynamics (the lower panel). In order to distinguish the influence from compositions versus from pianists, We perform statistical tests on two factors (composition and performer). Since the distributions of both tempo and dynamic data violate the assumption of homogeneity in Levene's test, non-parametric tests (Kruskal-Wallis tests) are performed instead of regular ANOVA, and Bonferroni correction is applied to post hoc analysis [10]. Regarding the tempo data, the statistical analyses yield significant differences between all three compositions, while the performances for Mozart's and Beethoven's pieces have more similar average tempo (126.77 and 136.30 bpm) compared to Bach's piece (69.70 bpm). Comparing the performances for Mozart's and Beethoven's compositions, Beethoven's

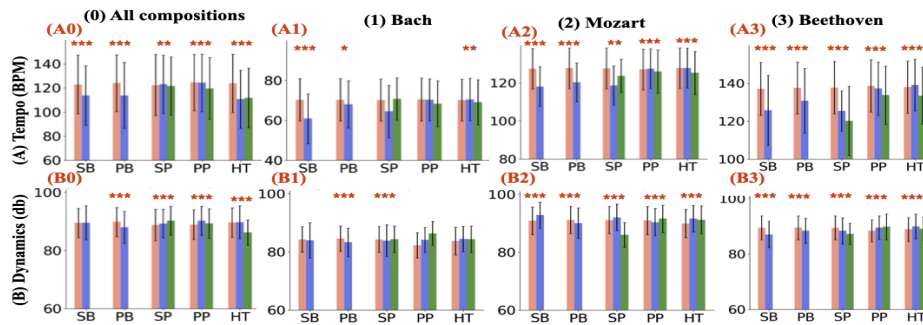
composition possesses higher variations (SD = 14.58, range = 124.67) than Mozart's work (SD = 10.85, range = 96.27). It is worth noted that according to the statistic analyses on the pianist factor, some pianists have distinctive expressive strategies, which can be distinguished from other pianists' expressive trend.

#### **4.2 Interactions between compositional and expressive elements**

Pianists perform individual compositions with diverse expressive manners, and they may use different expressive strategies to communicate each compositional element. In this section, we further analyze the interaction between different compositional elements and expressive features. We focus on the phrase, section, and harmony aspects of composition, and manually-annotate five different features for each musical bar in each composition: 1) section boundary (on section boundary/ non-boundary), 2) phrase boundary (on phrase boundary/ non-boundary), 3) section position (in the first /middle/ last one third of a section), 4) phrase position (in the first/ middle/ last one third of a phrase), 5) harmony (I/ V/ other types of chord). We contemplate both the boundary and relative position for section/phrase, since in our preliminary observation, we found that musicians tend to show different manners at section/phrase boundaries, and their expressive tendencies also vary when they initiate a new section/phrase versus when they are approaching the end of section/phrase. For the harmonic aspect, we only compare three types of chord to simplify the analysis process, since it is not straight forward to observe the overall general tendency in the comparisons of many groups (e.g. comparing all 7 degrees of chord lead to  $C_2^7 = 21$  combinations). In many chord types, we choose tonic and dominant chords to analyze, considering that those two chords take essential position in Western tonal music and are often used to signify structural location in music (authentic or half cadence).

For statistic analyses, we take the expressive measurements (tempo/ dynamics) in each bar, and then split the data into groups according to their compositional elements. Fig. 4 (A0 and B0) shows all the 26 comparison groups for statistic analysis. We first perform normality tests for all groups, and subsequently carry out homogeneity tests for three-group comparisons (section position, phrase position, and harmony type). For groups violating the normality or homogeneity assumptions, the non-parametric counterpart is performed instead of parametric test (i.e. t-test or Mann-Whitney U test for two-group comparisons; one-way ANOVA or Kruskal-Wallis test for three-group comparisons). For three-group comparisons, we further carry out post hoc tests to compare different combinations. Aforementioned procedure is performed three times for the three compositions individually.

In Fig. 4, the general expressive tendencies (column 0) and differences between compositions (column 1 - 3) can be observed. For tempo variations (Fig. 4, A0), musicians generally incline to slow down at section and phrase boundaries, as well as at the bars with tonic and dominate chords, since those chords may coincide with cadence. But different expressions emerge when comparing three compositions. For the section position, in Bach's and Mozart's music piece (Fig. 4, A1 and A2), musicians' tempo variation exhibits a U-shape curve, in which they tend to perform with faster tempi at the beginning and end of section, and slightly slow down in the middle of section, whereas in Beethoven's music piece (Fig. 4, A3), musicians' tempo curve tilts toward



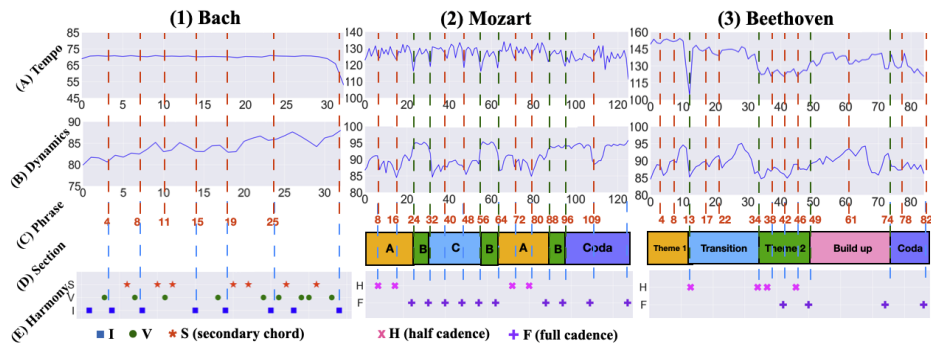
**Fig. 4.** The statistic analyses for 2 expressive elements (tempo (row A), dynamics (row B)), with 5 compositional elements in 3 music pieces (Bach (column 1), Mozart (column 2), Beethoven (column 3)), including the means (bars), standard deviations (error bars), and significance level (stars). Comparison groups are: section boundary (SB), phrase boundary (PB) (non-boundary (pink), boundary (blue)), section position (SP), phrase position (PP) (the first 1/3 (pink), the middle 1/3 (blue), the last 1/3 (green)), harmony types (HT) (others (pink), I (blue), V (green)).

the end of section, in which they apply a ritardando to highlight the end of section. For dynamic variations (Fig. 4, B0), it is shown that pianists tend to perform with softer dynamic levels when they are approaching the end of section or phrase. The softer dynamic level sometimes incorporate with ritardando to be used as the expressive strategy to shape 'the sense of direction toward the end of phrase/ section' in performance. We can observe that most of comparisons yield significant difference between groups, which indicates that musicians generally jointly use the combination of different expressive variations (tempo and dynamics) to deliver compositional traits in music, whereas in Bach's music piece, the section, phrase, and harmonic structures sometimes are not manifest in expressive variations.

### 4.3 Stylistic time-series expressions in compositions

In addition to the comparison between different compositional elements, in music performance, both expressive variations and compositional elements are revealed during the course of time. We therefore take a step further in this section to discuss the time-series connection between expressive and compositional elements. In Fig. 5, the time-series curve of average tempo (Row A) and dynamics (Row B) per bar (for all trials performed by all pianists) are aligned with musical elements in compositions including phrases (Row C), sectional boundaries (Row D), and harmonic progression (Row E). In our analysis, the expression curves show that pianists tend to adopt different patterns of variation in their performance to convey distinctive traits in each composition.

In Bach's work, pianists perform with a steady tempo, except an obvious ritardandos indicating the cadence at the end of the piece (Fig. 5, 1A). The dynamic variation in Bach shows distinct features corresponding to the phrase structure and the harmonic progression (Fig. 5, 1B). For the phrase structure, the dynamic curve exhibits an inverted U-shape matching with phrase boundaries per 4 to 6 bars, which shows that pianists tend to perform a crescendo for the first half of the phrase, and then perform a decrescendo for the second half of the phrase. For the harmonic progression, louder performance dynamics are applied to emphasize harmony with higher tension such as



**Fig. 5. Time-series trends of tempo and dynamic variations in piano performance.**

The expressive curves of mean tempo (bpm) (Row A) and the mean dynamics (db) (Row B), aligned with compositional elements including phrases (Row C), sectional boundaries (Row D), and harmonic progression (Row E) in Bach's (Column 1), Mozart's (Column 2), and Beethoven's (Column 3) compositions.

secondary chords (red markers), whereas softer dynamics associate with tonic chord (blue markers), which often coincides with the boundary of phrase and represents the release of harmony tension.

In Mozart's composition, the tempo and dynamic curves in pianists' performances exhibit different traits compared to Bach's work. In contrast with the smooth tempo curve in Bach's work, pianists apply tempo variations to express the phrase structure when performing Mozart's work, in which their tempi tend to slow down at phrase boundaries (Fig. 5, 2A). For dynamic variations, in Bach's work, pianists use dynamic variations (crescendo-decrescendo patterns) to express phrase structure, whereas dynamic variations in Mozart's work serve as the means to convey higher-level music structure (Fig. 5, 2B). In Mozart's work, the regions with relatively louder dynamic levels often coincide with the appearance of theme B. In Mozart's this composition, theme A and theme C possess contrasting characters compared to theme B. Theme A and theme C mostly consist of rapid sixteenth notes, whereas the main components of theme B are unison chords played by both hands simultaneously. It would be a natural practice for pianists to perform theme B with a louder dynamic level in this case. An interesting observation is that given the contrasting dynamic levels between different themes (theme A, C versus theme B), the dynamic variations in Mozart's work still reflect the harmonic progression at the local-level. As shown in (Fig. 5, 2B), within individual themes, the valleys of dynamic curve at local regions often coincide with the release of harmonic tension, such as half cadences (pink markers) or full cadences (purple markers). Those observations indicate that the dynamic curve in Mozart's work is formed by complex interactions between diverse musical components, including the theme arrangement and the harmonic progression.

In Beethoven's composition, pianists accentuate the musical structure using different strategies compared to the previous two compositions. It appears that pianists tend to focus on the higher-level structure of music rather than local-level details in this composition, and employ combinative strategies to emphasize their interpretation of the overall musical structure. For the tempo variation, Bach's work has a smooth tempo curve, and in Mozart's work, pianists apply tempo variations (accelerando-ritardando

pattern) to express local phrase boundaries. On the other hand, in Beethoven's composition, obvious valleys in tempo curve tend to coincide with the end of structural sections rather than local phrase boundaries (Fig. 5, 3A), which indicates that pianists employ a noticeable *ritardando* to signify the end of the section. The dynamic variations in Beethoven's work exhibit multi-layered musical features in the composition (Fig. 5, 3B). The global trend in dynamic variation shows inverted U-shapes corresponding to the high-level section structure of different themes, which suggests that pianists' performances exhibit the crescendo-decrescendo global pattern for each structural section. In addition, the dynamic curve within local regions still reflects detailed local-level features in the composition, in such a way that occasional fluctuations with limited range match with phrase boundaries, and the curve valleys are usually consistent with the locations with lower harmonic tension (half or full cadences).

To summarize general time-series trends observed in the tempo and dynamic variations, pianists employ diverse strategies to communicate the harmonic, phrase, and sectional structure in the three music compositions. Pianists generally utilize dynamic variations (crescendo-decrescendo pattern) to convey the phrase and harmonic structure in Bach's work. In contrast, in Mozart's work, dynamic variations are the means to communicate higher-level sectional structure rather than local phrase boundaries, and the phrase structure is more manifest in the tempo variation curve (*ritardando* at phrase end). In Beethoven's composition, tempo and dynamic variations exhibit complex influences from diverse musical features. The sectional structure is evident in both tempo (*ritardando* at section end) and dynamic variations (crescendo-decrescendo pattern), and the dynamic fluctuations are affected by global features in sectional structure, as well as by local features in phrase and harmony.

## 5 Discussion

In the previous section, we reported our findings regarding tempo and dynamic variations in performances of three piano pieces, and how pianists apply different expressive variations to communicate distinctive musical structures in the composition. We will further incorporate our findings with previous research in this section.

According to our analysis, musical phrase appears to be one of the main components for musicians to express in their performance. Previous research reported that the arching pattern in tempo curve [9] [35] and dynamic variations [15] are attached to phrase formation. In our analysis, we further found that pianists apply diverse strategies to express phrase structure when they are performing different compositions. For instance, the dynamic variations indicate the phrase structure in Bach's composition (crescendo-decrescendo pattern per phrase), whereas tempo variations are mostly used to express phrase in Mozart's composition (slow down at phrase end). Different expressive strategies also reflect diverse compositional characters in these two music pieces. Bach's composition holds an invariant rhythmic pattern, which is expressed by a stable tempo in pianists' performances. In contrast, pianists are more likely to emphasize the dynamic change in order to highlight the tension-release process for the harmonic progression in Bach's composition, and such harmonic progression usually conforms with phrase structure (e.g. cadence at the end of phrase). The compositional structure in Baroque period mostly focuses on the development of short motives, whereas composi-

tions in Classical period emphasize clear formation of phrase. In Mozart's composition, pianists therefore apply the accelerando-ritardando pattern in the tempo curve to shape the direction of the phrase.

Regarding the combination of local music elements and the global structure of music, previous studies suggest that the expressive manner in music performance is affected by both local elements (e.g. melodic peak, rhythmic grouping) [32] [22] and global structure (e.g. sectional arrangement) [9]. In our analysis, we found that pianists apply different strategies to stress local and global elements. For instance, Beethoven's composition exhibits an interesting combination of local and global factors, in which the general curve in both tempo and dynamic variations remain mostly consistent with global sectional arrangement, while the variations still show small-range fluctuations corresponding to local phrase boundaries. Compared to Bach's and Mozart's works, Beethoven's composition has sophisticated theme transformation accompanied by frequent modulation, and the manifest harmonic tension build-up process is one of the key features in Beethoven's compositions. Pianists may therefore manipulate both tempo and dynamic variations in their performance to communicate this important structural character.

It emerges from our analysis that in piano performance, expressive variations in tempo and dynamics exhibit systematic variations consistent with musical structures. Such systematic variations can be regarded as typical components to shape distinctive performance style, in which we generally expect that pianists should apply different expressive conventions when they are performing different styles of music in Western art music (e.g. Baroque, Classical, Romantic compositions). Our analysis method and results can serve as empirical means and provide observations for diverse performance styles in Western art music. Our current analysis is limited to piano performances for several selected compositions, and this analysis procedure can be further applied to the investigation for wider range of repertoire and for different instrument's performances.

## **6 Conclusion**

In this paper, we show that pianists' performances exhibit systematic expressive variations corresponding to diverse compositional styles in Western art music. We collected 72 piano performance recordings for three compositions, and derived expressive variations in tempo and dynamics using automatic audio-to-score alignment and MIR techniques. Statistical and time-series analyses are performed to clarify the relationship between different compositional and expressive element, as well as their time-series connections during the course of performance. It is found that pianists apply stylistic expressive variations to communicate musical components at both global (e.g. sectional arrangement, harmonic progression) and local (e.g. phrase boundaries) levels, and they choose different expressive strategies according to distinctive traits of each composition. We suggest that those systematic variations in expressive elements constitute the core of distinctive performance style, and the complex interaction among diverse expressive elements (e.g. tempo and dynamic variations) at multi-layered musical structures (local and global levels) can compose an empirical approach to describe and compare idiosyncratic music performance styles.

## References

1. Aldwell, E., Schachter, C., Cadwallader, A.: *Harmony and Voice Leading*. Cengage Learning (2018)
2. Cancino-Chacón, C.E., Gademaier, T., Widmer, G., Grachten, M.: An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. *Machine Learning*, 106(6), 887-909 (2017)
3. Cancino-Chacón, C.E., Grachten, M.: An evaluation of score descriptors combined with non-linear models of expressive dynamics in music. In: *Proceedings of International Conference on Discovery Science* (2015)
4. Cancino-Chacón, C.E., Grachten, M., Goebel, W., Widmer, G.: Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities*, 5, 25 (2018)
5. Chen, C. T., Jang, J. S. R., Liou, W.: Improved score-performance alignment algorithms on polyphonic music. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (2014)
6. Cook, N.: *A Guide to Musical Analysis*. Oxford University Press, USA (1994)
7. Cuthbert, M.S., Ariza, C.: Music21: A toolkit for computer-Aided musicology and symbolic music data. In: *Proceedings of International Society for Music Information Retrieval Conference* (2010)
8. Dodson, A.: Expressive asynchrony in a recording of Chopin's Prelude No. 6 in B Minor by Vladimir de Pachmann. *Music Theory Spectrum*, 33(1), 59-64 (2011)
9. Fabian, D.: *A Musicology of Performance: Theory and Method Based on Bach's Solos for Violin*. Open Book Publishers (2015)
10. Field, A.: *Discovering Statistics Using IBM SPSS Statistics* (4th edition). Sage (2013)
11. Flossmann, S., Grachten, M., Widmer, G.: Expressive performance rendering with probabilistic models. In: A. Kirke, E. R. Miranda (Eds.), *Guide to Computing for Expressive Music Performance*, pp. 75-98, Springer (2013)
12. Friberg, A., Bresin, R., Sundberg, J.: Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3), 145-161 (2006)
13. Gingras, B., McAdams, S.: Improved score-performance matching using both structural and temporal information from MIDI recordings. *Journal of New Music Research*, 40(1), 43-57 (2011)
14. Giraldo, S., Ramirez, R.: A machine learning approach to ornamentation modelling and synthesis in jazz guitar. *Journal of Mathematics and Music*, 10(2), 107-126 (2016)
15. Giraldo, S., Waddell, G., Nou, I., Ortega, A., Mayor, O., Perez, A., Ramirez, R.: Automatic assessment of tone quality in violin music performance. *Frontiers in Psychology*, 10, 334 (2019)
16. Grachten, M., Widmer, G.: Linear basis models for prediction and analysis of musical expression. *Journal of New Music Research*, 41(4), 311-322 (2012)
17. Grindlay, G., Helmbold, D.: Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine Learning*, 65(2), 361-387 (2006)
18. Héroux, I.: Creative processes in the shaping of a musical interpretation: a study of nine professional musicians. *Frontiers in Psychology*, 9, 665 (2018)
19. Juslin, P.N.: Five facets of musical expression: A psychologist's perspective on music performance. *Psychology of Music*, 31(3), 273-302 (2003)
20. Kim, T. H., Fukayama, S., Nishimoto, T., Sagayama, S.: Statistical approach to automatic expressive rendition of polyphonic piano music. In A. Kirke, E.R. Miranda (Eds.), *Guide to Computing for Expressive Music Performance*, pp. 145-179, Springer (2013)
21. Kosta, K., Ramirez, R., Bandtlow, O. F., Chew, E.: Mapping between dynamic markings and performed loudness: a machine learning approach. *Journal of Mathematics and Music*, 10(2), 149-172 (2016)
22. Leech-Wilkinson, D.: Cortot's Berceuse. *Music Analysis*, 34(3), 335-363 (2015)
23. Lerdahl, F., Jackendoff, R.: *A Generative Theory of Tonal Music*. MIT Press (1983)
24. Li, S., Dixon, S., Plumbley, M.D.: Clustering expressive timing with regressed polynomial coefficients demonstrated by a model selection test. In: *Proceedings of International Society for Music Information Retrieval Conference* (2017)
25. Liem, C. C., Hanjalic, A.: Expressive timing from cross-performance and audio-based alignment patterns: An extended case study. In: *Proceedings of International Society for Music Information Retrieval Conference* (2011)
26. Liem, C. C., Hanjalic, A.: Comparative analysis of orchestral performance recordings: An image-based approach. In: *Proceedings of International Society for Music Information Retrieval Conference* (2015)
27. Llorens, A.: Brahmsian articulation: Ambiguous and unfixed structures in op. 38. *Music Theory Online*, 27(4) (2021)
28. McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., Nieto, O.: Librosa: Audio and music signal analysis in python. In: *Proceedings of the 14th Python in Science Conference* (2015)
29. Molina-Solana, M., Lluís-Arcos J., Gomez, E.: Identifying violin performers by their expressive trends. *Intelligent Data Analysis*, 14(5), 555-571 (2010)
30. Nakamura, E., Ono, N., Saito, Y., Sagayama, S.: Merged-output hidden Markov model for score following of MIDI performance with ornaments, desynchronized voices, repeats and skips. In: *Proceedings of International Computer Music Conference* (2014)
31. Nakamura, E., Yoshii, K., Katayose, H.: Performance error detection and post-processing for fast and accurate symbolic music alignment. In: *Proceedings of International Society for Music Information Retrieval Conference* (2017)
32. Ornoy, E., Cohen, S.: Analysis of contemporary violin recordings of 19th century repertoire: Identifying trends and impacts. *Frontiers in psychology*, 9, 22-33 (2018)
33. Payne, E.: Creativity beyond innovation: Musical performance and craft. *Musicae Scientiae*, 20(3), 325-344 (2016)
34. Ramirez, R., Maestre, E., Serra, X.: Automatic performer identification in commercial monophonic jazz performances. *Pattern Recognition Letters*, 31(12), 1514-1523 (2010)
35. Spiro, N., Gold, N., Rink, J.: Musical motives in performance: A study of absolute timing patterns. In E. Chew, G. Assayag, J. B. Smith (Eds.), *Mathematical conversations: Mathematics and computation in music performance and composition*, pp. 109-128. World Scientific (2016)
36. Van Herwaarden, S., Grachten, M., De Haas, W. B.: Predicting expressive dynamics in piano performances using neural networks. In: *Proceedings of The Conference of International Society for Music Information Retrieval* (2014)
37. Windsor, W. L., Desain, P., Penel, A., Borkent, M.: A structurally guided method for the decomposition of expression in music performance. *The Journal of the Acoustical Society of America*, 119(2), 1182-1193 (2006)



# SANGEET: A XML based Open Dataset for Research in Hindustani Sangeet

Chandan Misra and Swarup Chattopadhyay \*

School of Computer Science & Engineering, XIM University  
chandan@xim.edu.in, swarupc@xim.edu.in

**Abstract.** It is very important to access a rich music dataset that is useful in a wide variety of applications. Currently, available datasets are mostly focused on storing vocal or instrumental recording data and ignoring the requirement of its visual representation and retrieval. This paper attempts to build an XML-based public dataset, called SANGEET, that stores comprehensive information of *Hindustani Sangeet* (North Indian Classical Music) compositions written by famous musicologist *Pt. Vishnu Narayan Bhatkhande*. SANGEET preserves all the required information of any given composition including metadata, structural, notational, rhythmic, and melodic information in a standardized way for easy and efficient storage and extraction of musical information. The dataset is intended to provide the ground truth information for music information research tasks, thereby supporting several data driven analysis from a machine learning perspective. We present the usefulness of the dataset by demonstrating its application on music information retrieval using XQuery, visualization through *Omenad* rendering system. Finally, we propose approaches to transform the dataset for performing statistical and machine learning tasks for a better understanding of *Hindustani Sangeet*. The dataset can be found at <https://github.com/cmisra/Sangeet>.

**Keywords:** Hindustani Sangeet, North Indian Classical Music, XML, Music Dataset, Classification, XQuery, Music Rendition

## 1 Introduction

Having access to free, well-maintained databases of music is a crucial resource for researchers. In the case of Indian Classical Music, this is also true since it has been shown to be important for high-quality research in music information retrieval (MIR) [8,11,21,15] and musicological analysis using machine learning [23,22,18], deep learning [12,20,16,19], etc. Several high-quality datasets, [1] and [2] for example, for research in MIR and computational musicology can be found in the published literature.

---

\* We would like to thank the undergraduate students of School of Computer Science & Engineering for helping create the dataset.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Although audio recording-based music corpora are essential in certain types of music applications, studies of existing literature reveal a dearth of substantial research related to the varied domains of *Sound and Music Computing* (SMC), especially in the design and development of interfaces for expressing Indic Music on an electronic medium. One of the domains that interest us is the creation of an Indian music environment through the transcription and rendering of an Indic music piece using Indic notation systems and Indic language script. The ability to compose music electronically entirely in an Indian music environment necessitates the emergence of research in different domains in SMC. This requires a musicological analysis of the grammar and structure of the music sheets presently in use and the consequent development of musical fonts and rendering engines ([5] for staff notation for example). Needless to say, such endeavors would motivate the research community to create models for Indic music notation systems and their language bases [14] and provide ample opportunities to work in building interfaces for music expressions on an electronic medium.

The mere enabling of music practitioners in composing music electronically is insufficient unless we have tools to exchange such musical information seamlessly across applications. Consequently, this establishes the need for the development of a common music exchange format to communicate music independent of any genre, notation system, language script, and music sheet structure [13]. XML-based formats for exchanging musical information have existed for quite some time [13,10,7] and are being adopted by extremely robust and popular notation software like *Finale* [3], *Sibelius* [6], *MuseScore* etc. Additionally, the ability to store musical information in XML solves the problem of archiving our historic musical art form as an electronic database.

One of the most authentic sources of Hindustani Sangeet is the compositions published in the book *Hindusthani Sangeet Paddhati-Kramik Pustak Malika* which comprises of approximately 1900 compositions belonging to North Indian Classical Music penned by Pt. Vishnu Narayan Bhatkhande (1860 - 1936). In order to reach a greater number of music students and scholars, the first volume of *Kramik Pustak Malika* has been translated to Hindi language in 1953 by prominent music scholar Dr. Laxminarayan Garg. This paper introduces SANGEET, arguably the first XML-based music corpora that try to capture comprehensive musical information contained in these rich music sources to apply in various music applications like music transcription, visualization, MIR, computational musicology, etc. We begin the preparation of the dataset with the second volume of *Kramik Pustak Malika* book series and our objective is to store compositions of different genres in a carefully crafted XML database to preserve comprehensive musical information in a single format. This will provide the users to obtain a standard framework for efficient and easy access to the dataset that can be easily transformed to apply to various applications. We refer to three music applications related to visualization, MIR, and machine learning in support of the coverage, quality, and accessibility of SANGEET.

## **2 The Organization and Access of SANGEET**

Pt. Vishnu Narayan Bhatkhande is the pioneer for providing a comprehensive theoretical foundation of Hindustani Sangeet in a published form in his six-volume book series

titled *Hindustani Sangeet Paddhati, Kramik Pustak Malika* written in *Marathi* language in 1920. His book contains a comprehensive description of music symbols for realizing musical components including notes (Svar), time signatures (Lay), beats (Taal), ornaments (Alankar) etc. The dataset created in the current work has been taken from the Hindi translation of the second volume of the series. The second volume of the book series contains a total of 319 compositions belonging to 10 different raags. The present work takes these written compositions as a source of musical information to create the database for Hindustani Sangeet to be used in various applications.

We have taken 116 compositions of the three highest frequent raags i.e. raag *Bhairav* (42), *Todi* (39), and *Poorvi* (35) respectively, from the entire collection of 319 compositions for performing our experimental analysis. Eventually, the entire collection of compositions from all six volumes will be preserved in the dataset for applications related to music information retrieval, music-sheet visualization, etc.

The dataset consists of a number of XML documents that is equal to the number of compositions in the dataset i.e. each XML document represents a single composition of the dataset. The XML documents are equipped with meaningful tags to store all the necessary musical information for the compositions. The format of the XML files is validated against a schema definition document so that the format of the dataset or compositions are preserved. The schema definition document is an XML Schema Definition (XSD) file against which each XML document is checked and validated for legal elements and attributes. The XSD consists of four parts namely *info*, *taal*, *raag*, and *sheet* responsible for storing metadata, rhythmic, melodic, structural, and notational information in the XML files.

The metadata linked to the musical composition is represented by the *info* portion. It contains information on the catalog, the genre, and the notational system as shown in Listing 1.1 describing the first composition of the second volume of the book.

```

1 <INFO>
2 <TITLE>Composition 1 Volume 2 Kramik Pustak Malika</TITLE>
3 <AUTHOR>Pt. Vishnu Narayan Bhatkhande</AUTHOR>
4 <NOTATION_SYSTEM>Bhatkhande</NOTATION_SYSTEM>
5 <DATE_TIME>1923</DATE_TIME>
6 <GENRE>Hindusthani Sangeet</GENRE>
7 <ADDITIONAL>
8 <ENTRY>http://ndl.iitkgp.ac.in/document/
9 R2pPWGRxdkRWn1vOVdPYzdzawpTV0pYYTFIT0VnNTB6V1dnR1dJVW1kUT0</ENTRY>
10 </ADDITIONAL>
11 </INFO>

```

Listing 1.1: Info Part of XML file depicting metadata

The rhythmic foundation of Indian music is provided by *taal*. Indic music has nearly hundreds of Taals, each with its own specific composition that includes the name, *Bibhaga* or measure, *Maatra* or the number of beats, *Avartana* or the number of cycles per line, etc. Additionally, Taal has two designated beat indices, known as *Taali* and *Khali*, to signify stressed or unstressed strokes in addition to a specific beat pattern to uniquely identify a Taal. These patterns, which are required to portray the Taal graphically or as a music sheet accompanied by an Indian percussionist, have been illustrated as a series of numbers (seen in Listing 1.2). Additionally, the regular expression specifies the expression for a beat pattern, making it easier to query the Taal's structure.

```

1 <TAAL>

```

```

2 <TAAL_NAME>Tritaal</TAAL_NAME>
3 <BIBHAGA>4</BIBHAGA>
4 <MAATRA>16</MAATRA>
5 <AVARTANA>1</AVARTANA>
6 <BEAT_PATTERN>4-4-4-4</BEAT_PATTERN>
7 <ALTERNATE_BEAT_PATTERN>NA</ALTERNATE_BEAT_PATTERN>
8 <TAALI_COUNT>3</TAALI_COUNT>
9 <KHALI_COUNT>1</KHALI_COUNT>
10 <TAALI_INDEX>1-5-13</TAALI_INDEX>
11 <KHALI_INDEX>9</KHALI_INDEX>
12 </TAAL>

```

Listing 1.2: Taal Part of XML file depicting Taal and its sub-components

Raag provides the melodic framework to Hindustani Sangeet and each raag can be identified by characteristics like *Arohana* and *Avarohana*, which are ascending or descending movements made up of a series of notes, *Vadi* and *Samvadi*, which are consonant and dissonant notes, and classification forms like *Pakad* and *Jaati*. These characteristics are note sequences and have been encoded using *Ome Swarlipi* [4], the same rendition we use for storing notes in our dataset.

```

1 <RAAG>
2 <RAAG_NAME>Yaman</RAAG_NAME>
3 <THAAT>Kalyan</THAAT>
4 <AROHANA>n-r-g-M-d-n-su</AROHANA>
5 <AVAROHANA>su-n-d-p-M-g-r-s</AVAROHANA>
6 <VADI>g</VADI>
7 <SAMVADI>n</SAMVADI>
8 <JAATI>Sampoorna</JAATI>
9 <PAKAD>nlrgr-s-pMg-su</PAKAD>
10 </RAAG>

```

Listing 1.3: Raag Part of XML file depicting Raag and its sub-components

**Sheet**, which is based on the 2D matrix model *Swaralipi* [14], specifies the layout of the music sheet and the placement of the notation symbols. As a result, it replicates the entirety of the contents as a rectangular row-column arrangement. Even though we haven't yet transcribed the beat markings and lyrics, the model has the provision to include them in the future. The format cleverly transforms row and column models into helpful tags that make it easier to develop various applications, such as real-time note playback, producing music sheets, and retrieving score data. For example, part of the first line of the original composition (shown in Figure 1a) has been converted into the sheet part (shown in Figure 1b and 1c).

### 3 Applications of the Dataset

**Visualization of Music-sheets:** One of the primary applications of any music dataset is to visualize it or render it using a notation system in which it is preserved. We have encountered several difficulties in visualizing the composition in the Bhatkhande notation system since there is no standard font system for rendering Bhatkhande music symbols in any language script. The closest rendition we have found is the *Ome Swarlipi* [4] system which is a compact version of the Bhatkhande notation system and easy to use. In order to visualize in HTML format, the system provides the necessary styling information to render it in *Devanagari* script. Therefore the pre-processing step for this

नि ध प म | ग रे ग म | नि ध म ऽ | प म ग रे  
 ० ३ x २

(a)

```

1 <SHEET>
2 <TOTAL_LINE></TOTAL_LINE>
3 <LINES>
4 <LINE INDEX="1">
5 <ROW INDEX="1">
6 <COL INDEX="1">
7 <NOTE_COUNT>1</NOTE_COUNT>
8 <CONTENT>n</CONTENT>
9 </COL>
10 <COL INDEX="2">
11 <NOTE_COUNT>1</NOTE_COUNT>
12 <CONTENT>d</CONTENT>
13 </COL>
14 <COL INDEX="3">
15 <NOTE_COUNT>1</NOTE_COUNT>
16 <CONTENT>p</CONTENT>
17 </COL>
18 <COL INDEX="4">
19 <NOTE_COUNT>1</NOTE_COUNT>
20 <CONTENT>M</CONTENT>
21 </COL>
22 </ROW>
23 .....
```

(b)

```

1 <LINE INDEX="2">
2 <ROW INDEX="1">
3 <COL INDEX="1">
4 <NOTE_COUNT>1</NOTE_COUNT>
5 <CONTENT>g</CONTENT>
6 </COL>
7 <COL INDEX="2">
8 <NOTE_COUNT>1</NOTE_COUNT>
9 <CONTENT>M</CONTENT>
10 </COL>
11 <COL INDEX="3">
12 <NOTE_COUNT>1</NOTE_COUNT>
13 <CONTENT>p</CONTENT>
14 </COL>
15 <COL INDEX="4">
16 <NOTE_COUNT>1</NOTE_COUNT>
17 <CONTENT>M</CONTENT>
18 </COL>
19 </ROW>
20 </LINE>
21 </LINES>
22 </SHEET>
```

(c)

Fig. 1: Sheet part of the XML file (b) and (c) depicting part of the original music sheet (a).

application is a converter that takes an XML file as a standalone composition and transforms it into equivalent HTML with the *Ome Swarlipi* rendition of the score. The source code of the converter has been given in the online repository link and the corresponding rendition is shown in Figure 2.

**Query and Retrieval of Musical Information:** This is the application where we can appreciate the power of XML as a means to build the music dataset. XML has brought with it a number of tools and technologies to efficiently process the information contained inside it. For the present application, we have used two tools, namely *XPath* and *XQuery*. XPath, the XML Path Language, uses path expressions to parse through the elements and attributes of an XML document and select node elements to extract the contents inside it. This language is also used in another query language XQuery to query an XML database and retrieve required information from it much like the SQL that does the same on a relational database.

The *preprocessing* stage for this application is to create an XML database created from the XML documents. We have used *BaseX* database engine to create the database from our dataset and XQuery to efficiently and easily perform complex queries and retrieve information from it and therefore, can be extremely useful for data-intensive complex web applications. This also provides a single-point query and retrieval system, as opposed to the current search and retrieval platforms [9,17] used for querying and

रु	ग	रु	ग	म	प	ग	म	नी	धु	-	ग	-	म	ग	रु
ग	म	प	ग	म	नी	धु	-	रु	-	सं	गं	-	ग	-	म
नी	धु	रु	-	धु	-	रु	-	ग	म	प	ग	म	ग	रु	नी

Fig. 2: Music-sheet web visualization using Ome Swarlipi

```

1 (: List of compositions having Meend :)
2 for $songs in collection ("
  Bhatkhande-Database")//swarlipi
3 let $title := $songs/INFO/TITLE/text()
4 let $contents := $songs/SHEET/LINES/
  LINE/ROW/COL/CONTENT/text()
5 let $notes := (for $song in $songs
6 return $song/SHEET/LINES/LINE/ROW/COL/
  CONTENT/text())
7 return if (contains(string-join($notes
  , ""),"q")) then
8 $title
1 (: List of compositions having a
  particular Arohana subsequence :)
2 for $songs in collection ("
  Bhatkhande-Database")//swarlipi
3 let $title := $songs/INFO/TITLE/text()
4 let $aroha := $songs/RAAG/AROHANA/text()
5 return if (contains($aroha, "s-R-g"))
  then
6 $title

```

(a)

(b)

```

1 (: Note frequency distribution of each composition :)
2 for $song in collection("Bhatkhande-Database")//swarlipi
3 let $raag := $song/RAAG/RAAG_NAME/text()
4 let $contents := $song/SHEET/LINES/LINE/ROW/COL/CONTENT/text()
5 let $joined_str := string-join(data($contents), ',')
6 let $joined_str := replace($joined_str, "<sup>|</sup>|@|u|l|\)|\(|-|\s+", "")
7 let $notes := (115,82,114,71,103,109,77,112,68,100,78,110)
8 let $code_points := string-to-codepoints($joined_str)
9 let $result := (for $i in $notes
10 return count(index-of($code_points, $i)) )
11 let $result := normalize-space(string-join($result, ","))
12 return $result

```

(c)

Fig. 3: XQuery to retrieve the (a) list of compositions having Meend, (b) List of compositions having a particular Arohana subsequence and (c) Note frequency distribution of each composition

browsing musical data. Figure 3 provides a few interesting and complex queries that satisfy the fine-grained information needs of the user. For example query 3c can be used to generate dataset for raag classification as described in the following section.

**Raag prediction through Machine Learning:** This application refers to the musical analysis of various musical components present in Hindustani Sangeet. It covers statistical and structural analysis, data mining, and inference using machine learning and deep learning techniques. As an example of the application, we apply machine learning techniques on the dataset for the task of raag prediction. The *preprocessing*

Accuracy Score of Classification Models				
Logistic Regression	K-Nearest-Neighbors (KNN)			Decision Tree
	$k = 3$	$k = 5$	$k = 7$	
0.9143	0.9714	0.9428	0.9428	0.9714

Table 1: Performance measure of Logistic Regression, K-Nearest Neighbors with varying values of  $k$ , and Decision Tree. The dataset is divided into 70:30 as training and test set to calculate the accuracy score of different classification models.

step for raag prediction is to convert the XML dataset into a tabular data-frame containing a number of features and a target variable. For raag prediction, we take features as the frequencies of individual notes and the corresponding raag as a target variable for any composition. Instead of taking the note-frequency distribution of 36 notes for a composition spanning across three octaves, we merge the notes to obtain the frequency distribution of 12 notes. Since, the positions of the notes of the *Arohana* and *Avarohana* of any particular composition in different octaves do not affect the raag of the composition, we map corresponding notes of three octaves and make a sum of frequencies of corresponding notes to obtain 12 note-frequency distribution (can be obtained from 3c given in GitHub). Table 1 shows the measure of performance of different machine learning techniques for raag prediction for our dataset. We have transformed our dataset into a three-class classification problem by taking the three most frequent raags i.e. *Bhairav*, *Todi*, and *Poorvi*, and applied the different classification models to generate the accuracy scores. Since each classifier examined shows high accuracy score the dataset can be considered as a robust dataset for raag classification. Table 1 shows that KNN with  $k = 3$  and decision tree classifier gives better accuracy scores than the logistic regression model.

## 4 Conclusions and Future Works

This paper presents SANGEET, a Hindustani Sangeet dataset based on XML to provide easy and efficient access to a music corpora to perform various applications including music visualization, MIR, and Raag prediction using machine learning techniques. Backed by a robust music-sheet framework and a structured XSD, SANGEET provides a comprehensive repository for rich musical information to be shared seamlessly across applications. We have shown that SANGEET is quite efficient for accessing and transforming musical data into a format suitable for various musical applications. Our future objective is to extend SANGEET with the compositions of Bhatkhande’s other five volumes of *Kramik Pustak Malika* and update the structure of the XML dataset with taal markings and lyric information. This will provide better music-sheet rendition and richer queries to fulfill the user’s information needs.

## References

1. Annotated compmusic datasets. <https://compmusic.upf.edu/datasets>, accessed: 2022-05-6

2. Dunya. <https://dunya.compmusic.upf.edu/>, accessed: 2022-05-6
3. Finale—music notation software that lets you create your way. <https://www.finalemusic.com/>, accessed: 2022-05-6
4. Fonts for writing indian music - omenad fonts. <https://omenad.github.io/fonts/>, accessed: 2022-05-6
5. Lilypond... music notation for everyone. <http://lilypond.org/>, accessed: 2022-02-23
6. Music notation software - sibelius - avid. <https://www.avid.com/sibelius>, accessed: 2022-05-6
7. Baggi, D., Haus, G.: Ieee 1599: Music encoding and interaction. *Computer* 42(3), 84–87 (2009)
8. Chithra, S., Sinith, M., Gayathri, A.: Music information retrieval for polyphonic signals using hidden markov model. *Procedia Computer Science* 46, 381–387 (2015)
9. Ghosh, S., Dasgupta, A., Mukhopadhyay, D., Datta, D.: tagoreweb: The complete works of rabindranath tagore. <http://tagoreweb.in/> (2020), accessed: 2022-05-6
10. Good, M., et al.: Musicxml: An internet-friendly format for sheet music. In: *XML Conference and Expo*. pp. 3–4. Citeseer (2001)
11. Kirthika, P., Chattamvelli, R.: A review of raga based music classification and music information retrieval (mir). In: *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*. pp. 1–5. IEEE (2012)
12. Madhusudhan, S.T., Chowdhary, G.: Deepstrgm-sequence classification and ranking in indian classical music with deep learning. In: *Proceedings of the 20th International Society for Music Information Retrieval Conference*. pp. 533–540 (2019)
13. Misra, C.: Sangeetxml: An xml format for score retrieval for indic music. In: *ACM Multimedia Asia*, pp. 1–5 (2021)
14. Misra, C., Chakraborty, T., Basu, A., Bhattacharya, B.: Swaralipi: A framework for transcribing and rendering indic music sheet (2016)
15. Murthy, Y.S., Koolagudi, S.G.: Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Computing Surveys (CSUR)* 51(3), 1–46 (2018)
16. Nag, S., Basu, M., Sanyal, S., Banerjee, A., Ghosh, D.: On the application of deep learning and multifractal techniques to classify emotions and instruments using indian classical music. *Physica A: Statistical Mechanics and Its Applications* 597, 127261 (2022)
17. Society for Natural Language Technology Research, G.o.W.B.: rabindra-rachanabali. <https://rabindra-rachanabali.nltr.org> (1905), accessed: 2022-05-6
18. Patel, E., Chauhan, S.: Raag detection in music using supervised machine learning approach. *International Journal of Advanced Technology and Engineering Exploration* 4(29), 58 (2017)
19. Pendyala, V.S., Yadav, N., Kulkarni, C., Vadlamudi, L.: Towards building a deep learning based automated indian classical music tutor for the masses. *Systems and Soft Computing* 4, 200042 (2022)
20. Sharma, A.K., Aggarwal, G., Bhardwaj, S., Chakrabarti, P., Chakrabarti, T., Abawajy, J.H., Bhattacharyya, S., Mishra, R., Das, A., Mahdin, H.: Classification of indian classical music with time-series matching deep learning approach. *IEEE Access* 9, 102041–102052 (2021)
21. Sridhar, R., Geetha, T.: Raga identification of carnatic music for music information retrieval. *International Journal of recent trends in Engineering* 1(1), 571 (2009)
22. Sridharan, A., Moh, M., Moh, T.S.: Similarity estimation for classical indian music. In: *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. pp. 814–819. IEEE (2018)
23. Ujlambkar, A.M., Attar, V.Z.: Mood classification of indian popular music. In: *Proceedings of the CUBE international information technology conference*. pp. 278–283 (2012)



# JAZZVAR: A Dataset of Variations found within Solo Piano Performances of Jazz Standards for Music Overpainting

Eleanor Row<sup>1</sup>, Jingjing Tang<sup>1</sup>, and György Fazekas<sup>1</sup> \*

Centre for Digital Music, Queen Mary University of London  
e.r.v.row@qmul.ac.uk

**Abstract.** Jazz pianists often uniquely interpret jazz standards. Passages from these interpretations can be viewed as sections of variation. We manually extracted such variations from solo jazz piano performances. The JAZZVAR dataset is a collection of 502 pairs of ‘*Original*’ and ‘*Variation*’ MIDI segments. Each *Variation* in the dataset is accompanied by a corresponding *Original* segment containing the melody and chords from the original jazz standard. Our approach differs from many existing jazz datasets in the music information retrieval (MIR) community, which often focus on improvisation sections within jazz performances. In this paper, we outline the curation process for obtaining and sorting the repertoire, the pipeline for creating the *Original* and *Variation* pairs, and our analysis of the dataset. We also introduce a new generative music task, Music Overpainting, and present a baseline Transformer model trained on the JAZZVAR dataset for this task. Other potential applications of our dataset include expressive performance analysis and performer identification.

**Keywords:** Jazz piano dataset, music generation, transformer model

## 1 Introduction

The growing interest in generative music models has led to the exploration of their potential in specialised music composition tasks. As current trends often focus on generating complete songs or music continuation tasks [2, 3], there is a lack of datasets designed for specialised music tasks. However, these specialised music tasks, such as music infilling [16, 19] and composition style transfer [15, 21], could contribute to the development of artificial intelligence (AI) tools in music composition.

---

\* 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]. J.Tang is a research student also supported jointly by the China Scholarship Council and Queen Mary University of London. Special thanks to Max Graf, for his help in creating the GUI, and to both him, Huan Zhang, and Corey Ford for reviewing our paper.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

We introduce Music Overpainting as a novel specialised generative music task, inspired by the concept of overpainting in fine art and Liszt’s compositional approaches to rearrangement in his piano transcriptions from classical music. Music Overpainting generates variations by providing a rearrangement of a music segment. While the task aims to reframe the musical context by changing elements such as rhythmic, harmonic, and melodic complexity and ornamentation, the core melodic and harmonic structure of the music segment is preserved. Compared to related music generation tasks such as compositional style transfer [4] and music infilling [16, 19], Music Overpainting creates small variations within the same style and retains perceptible similarities in the underlying melodic contour and harmonic structure of the music segment. Outputs from Music Overpainting could be used in AI tools for music composition, to add variation and novelty to desired sections of music.

Our motivation for creating this dataset stems from the lack of available datasets for novel and specialised generative music tasks. Not only did we find that there was a lack of clean and high-quality MIDI data for investigating tasks such as Music Overpainting, but also in the context of solo jazz piano music in general. Most existing jazz datasets consist of transcriptions of improvised “solo” sections within a jazz performance or feature multiple instruments. Few datasets feature interpretations of the “head” section, containing the main musical theme, for solo piano only. Additionally, we found that many jazz datasets do not include performances from female musicians, so we are proud to include several extracts of performances from female jazz pianists within our dataset. Our dataset helps to fill this gap, while also providing insights into how jazz pianists rearrange standards for solo piano from a music information retrieval (MIR) perspective.

Table 1: Overview of *Original* and *Variation* Segments.

Feature	Original	Variation
Segment length	4 bars	misc.
Location	“head” section	“head” section
File format	Manually-transcribed MIDI	Automatically-transcribed MIDI
Musical format	Melody and chords	Two-handed solo piano
Type	Lead sheet of jazz standard	Piano performance of jazz standard
Source	MuseScore	Youtube

The JAZZVAR dataset comprises of 502 pairs of *Original* and *Variation* MIDI segments from 22 jazz standards, 47 performances, and 35 pianists. An *Original* segment is 4-bars long and manually transcribed from a lead sheet of a jazz standard. A *Variation* segment is manually found from an automatically transcribed piano performance of the same jazz standard. We find *Variation* segments by searching for passages that are melodically and harmonically similar to *Original* segments. Figure 1 shows more details of the data curation pipeline. Table 1 provides more information about the *Original* and *Variation* segments. The jazz standards and the piano performances in our dataset

are under copyright, therefore the JAZZVAR dataset cannot currently be made available for direct download. However, researchers will be allowed to access the dataset on request.

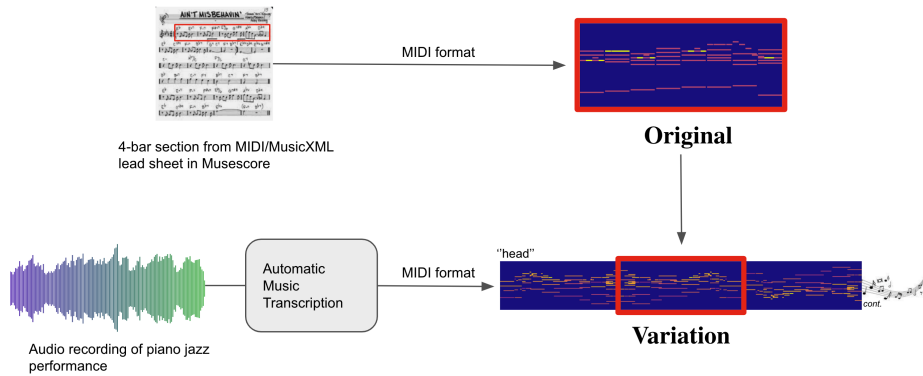


Fig. 1: The process of creating *Original* and *Variation* pairs. *Original* sections are MIDI segments from a lead sheet transcription of a jazz standard. Audio of a piano performance playing the same jazz standard is transcribed automatically into MIDI. A *Variation* is found by manually searching for passages that are melodically and harmonically similar to the *Original* in the “head” section of the piano performance.

The JAZZVAR dataset serves as a foundation for exploring the Music Overpainting task across genres. What we refer to as *Variations* are passages of music from a jazz standard that have been reinterpreted or rearranged by jazz pianists’. However, we can view these reinterpretations as variations on the melody and chords of the jazz standards. We use the *Original* and *Variation* pairs in the dataset to train a Music Transformer model to generate novel passages of variation from a simple MIDI primer. By presenting this novel dataset and introducing the Music Overpainting task, we aim to contribute to the field of generative music research and encourage further exploration of the relationship between composers and AI tools in various music genres.

The remainder of this paper is organised as follows: Section 2 provides an overview of related datasets in the field of generative music and MIR, Sections 3 and 4 present an in-depth description and analysis of the JAZZVAR dataset, Section 5 introduces Music Overpainting as a generative music task and uses the JAZZVAR dataset to train the Music Transformer model for generation.

## 2 Related Works

Existing jazz datasets that can be used for MIR and Generative Music tasks often feature the improvisation or solo section only of the jazz performance. The Weimar Jazz Database (WDB) [17], consists of 456 manually transcribed solos by 78 performers and contains no solo piano performances. The DTL1000 dataset [5] from the “Dig That

Lick” project is a set of 1750 automatically transcribed solos from 1060 tracks. However, it is not clear how many of these tracks are piano solo tracks.

The Million Song Dataset (MSD) [1] is a collection of audio features and metadata for one million contemporary popular music tracks. While the MSD does not specifically focus on jazz, it does include a substantial number of jazz recordings that could be used for comparative analysis. The Lakh MIDI Dataset (LMD) [13] is a collection of 176,581 unique MIDI files that are matched to songs within the Million Song Dataset using Dynamic Time Warping-based alignment methods [18]. Similarly, to the DTL1000 dataset, the MSD and the LMD have no specific focus on solo jazz piano performances.

### 3 JAZZVAR Dataset

#### 3.1 Data Collection

**Repertoire** A jazz standard is a well-known, and commonly played song in the jazz repertoire. Many popular songs composed in the early to mid-twentieth century for film, television, and musical theatre are now prominent jazz standards. Some of the more famous jazz standards include Gershwin’s “Summertime” for the opera *Porgy and Bess* (1935) and “All the Things You Are” by Jerome Kern and Oscar Hammerstein II for the musical *Very Warm for May* (1939). These popular songs have been continually played and rearranged by jazz musicians for decades. Popular songs originating from these times contain a “refrain” section, which was the main theme of the song. In jazz music, the “head” section is often synonymous with these “refrain” sections. Many jazz musicians would learn the songs by ear, or through unofficial lead sheets, such as the ones circulated within the *Fake Real Book*. Some jazz musicians, such as the trumpeter Miles Davis (1926-1991) and Thelonious Monk (1917-1982), composed music themselves and these pieces have also become famous jazz standards.

Within this context, our goal was to find lead sheets of jazz standards and audio recordings of solo piano performances of jazz standards. The first publication dates of the jazz standards in our dataset range between 1918 and 1966, while the performances span from the mid-twentieth to the beginning of the twenty-first century.

**Jazz Standard Lead Sheets** Lead sheets are condensed versions of song compositions that musicians have transcribed and passed through the community. They are presented as a single melodic line with accompanying chords.

We sourced MIDI and MusicXML lead sheets from MuseScore, created by users who often referenced the *Fake Real Book*. Candidate pieces were found using the following criteria:

1. entirely in 4/4 timing,
2. jazz standards mostly consisting of popular songs from the early to mid-twentieth century.

The lead sheets were cleaned and corrected by removing introductions and verses, to retain only the refrain section. Songs with repeated refrains were further edited to

include only the final repeat. We converted any MusicXML files to MIDI and made corrections by referencing the chords in lead sheets. In some cases, we transcribe the chords and melody by ear from early recordings of popular songs or completely rewrite the MIDI, as many of the source files were corrupt. In total, we collected and cleaned 234 jazz standards, of which a subset of 22 appear within the JAZZVAR dataset.

**Audio of Jazz Solo Piano Performances** To compile a list of solo piano performances of jazz standards, we manually searched for well-known jazz pianists' solo performances on Spotify and Youtube that matched the list of 234 MIDI lead sheets we had collected. We also used the *Solo piano jazz albums*<sup>1</sup> category on Wikipedia to help find performances. We gathered Spotify Metadata for these performances, which we used to collect the respective audio data. This approach allowed us to compile a diverse set of performances, including some by female pianists, and to capture the rich history of jazz piano performance.

### 3.2 Automatic Music Transcription of Jazz Audio

Automatic Music Transcription (AMT) algorithms such as [11, 8] enable us to transcribe audio recordings into MIDI representations. According to results from a listening test conducted by Zhang et al. [22], the High-Resolution transcription system proposed by Kong et al. [11] is preferred over the other two systems by participants in terms of conserving the expressiveness of the performances. We used the Spotify metadata to download the jazz audio from Youtube and applied the High-Resolution model [11] to transcribe the downloaded jazz audios into MIDIs. In total, we collected and transcribed 760 audio recordings covering a wide range of performances from 148 albums by 101 jazz pianists, of which a subset of 47 performances appear within the JAZZVAR dataset.

### 3.3 Pair Matching Process

We segmented 4 bar sections from the MIDI lead sheets by taking into consideration the phrases in the main melody. As the jazz standards that we chose were all in 4/4 time, most of the phrases were contained within a 4-bar structure. We labeled these four bar sections as *Original* segments. We segmented 22 jazz standards and collected an average of 6 segments per standard. In order to create our *Variation* segments to form a data pair, we manually searched through the AMT solo jazz piano performances of the jazz standards and found segments that were melodically and harmonically similar to the *Original* segment for each jazz standard. To facilitate the matching process for finding *Original* and *Variation* pairs, we created a Python application with a graphical user interface (GUI), which allowed us to view and listen to individual *Original* segments.<sup>2</sup> We then searched through the AMT jazz performances and saved passages that closely corresponded to the *Original* segments melodically and harmonically.

<sup>1</sup> See Wikipedia: [https://en.wikipedia.org/wiki/Category:Solo\\_piano\\_jazz\\_albums](https://en.wikipedia.org/wiki/Category:Solo_piano_jazz_albums)

<sup>2</sup> We plan to release the GUI for reproducing our dataset. A GitHub page will be released by the publication of the paper.

## 4 Analysis

### 4.1 Experimental Dataset Analysis

We calculated several musical statistics across the dataset to provide insights into the dataset’s musical content and structure according to [6]. We compared the differences between the *Original* and the *Variation* sections and summarise several characteristic features in Table 2.

Table 2: Means and standard deviations for various statistics for combined segments in *Original* and *Variation* sections.

Feature	Originals		Variations	
	Mean	SD	Mean	SD
Pitch Class Entropy	2.94	0.24	3.13	0.24
Pitch Range	36.44	3.60	47.20	10.91
Polyphony	5.30	0.28	5.01	2.08
Number of Pitches	16.08	0.28	29.42	8.05
Pitch in Scale	0.89	0.24	0.83	0.08

**Pitch Class Entropy** The higher mean pitch class entropy in the *Variation* segments (3.13) compared to the *Original* segments (2.94) suggests that jazz pianists tend to introduce more diversity in pitch distribution when interpreting jazz standards. This increased complexity and unpredictability in the variations reflect the improvisational and creative nature of jazz music.

**Pitch Range** The mean pitch range in the *Variation* segments (47.20) is considerably larger than in the *Original* segments (36.44), indicating that jazz pianists often expand beyond the range of pitches used within a jazz standard. This expanded pitch range could contribute to a richer and more expressive musical experience in the variations.

**Polyphony** Polyphony is defined as the mean number of pitches played simultaneously, evaluated only at time steps where at least one pitch is played. The mean polyphony is slightly lower in the *Variation* segments (5.01) compared to the *Original* segments (5.30). This suggests that jazz pianists may use fewer simultaneous pitches on average in their reinterpretations. However, the higher standard deviation in the *Variation* segments (2.08) indicates that the polyphonic structures in these reinterpretations can be quite diverse.

**Number of Pitches** The higher mean number of pitches in the *Variation* segments (29.42) compared to the *Original* segments (16.08) implies that jazz pianists tend to incorporate more distinct pitches when rearranging jazz standards. This increase in the number of pitches adds to the complexity and expressiveness of the variations.

**Pitch in Scale** Pitch-in-scale rate is defined as the ratio of the number of notes in a certain scale to the total number of notes [6]. The slightly lower mean value of pitch in scale in the *Variation* segments (0.83) compared to the *Original* segments (0.89) indicates that jazz pianists may be more inclined to use pitches outside the underlying

scale in their reinterpretations. This tendency could contribute to a more adventurous and explorative musical experience in the variations.

In summary, the analysis of the JAZZVAR dataset reveals that jazz pianists often introduce greater complexity, diversity, and expressiveness when rearranging jazz standards for solo piano. Our findings highlight the dataset’s potential for application in tasks such as Music Overpainting. Not only are these insights valuable for the development of specialised generative music models, but they also provide a better understanding of the creative process in jazz music.

## 4.2 Comparison of Multiple Pianists

Some of the jazz standards featured within the dataset are performed by multiple pianists. Therefore, there are some *Original* segments that are matched to multiple *Variation* segments from different pianists. To further highlight the diversity of variations within the dataset, we present a musical analysis of multiple pianists’ interpretations of the same *Original* segment, from the jazz standard “All the Things You Are”.

**Melody** The melody from the *Original* segment was found and isolated within each *Variation* segment. To obtain accurate representations of the melodies, we manually extracted the melody lines from the *Variation* segments. This manual extraction process involved listening closely to the melody in the *Original* in order to carefully isolate the melody line within the performances note by note, ensuring higher accuracy and fidelity of melodic extraction in comparison to an automatic approach. We then compared the isolated melodies to find their pitch and duration deviation from the ground truth, the melody from the *Original* segment. We applied the Needleman-Wunsch [7, 12] alignment algorithm which aligns melodies by minimizing the differences in pitch class and duration between the corresponding notes. Based on the alignment results, we calculate the average deviation score using the following equation:

$$Average\ Deviation = \frac{1}{n} \sum_{i=1}^n (PC_i + D_i), \quad (1)$$

where  $PC_i$  denotes the deviation of pitch class,  $D_i$  denotes the deviation of note duration, and  $i$  refers to the  $i$ -th note in the melody. We excluded the missing notes in the summation over the note sequences.

This average deviation score provides a measure of how similar the two melodies are, with lower scores indicating higher similarity. The deviation scores of the pianists’ *Variation* from the *Original* melody can be found in Table 3. Our results show that different pianists’ have unique and individual approaches to interpreting the *Original* melody. Some pianists, such as Leslie North, have a closer adherence to the *Original* melody, while others, like Bill Evans, exhibit greater differences.

Table 3: Average deviation from *Original* melody for different pianists

Pianist	Average Deviation
Jim McNeely	1.60
McCoy Tyner	1.04
Roland Hanna	1.50
Lennie Tristiano	0.92
Elmo Hope	1.11
Leslie North	0.65
Bill Evans	2.68

We also mapped the melodic contours of the performances to further explore the differences between the interpretations, using the Contourviz<sup>3</sup> package as shown in Figure 2. The visual representation of melodic contours allowed us to observe the overall structure and direction of the melody as it evolved throughout the performance. By comparing the melodic contours of different pianists, we found that some tended to be more experimental with their melodic choices, while others adhered more closely to the *Original* melody. This variation in melodic contours provides additional evidence of the rich diversity present in our dataset.

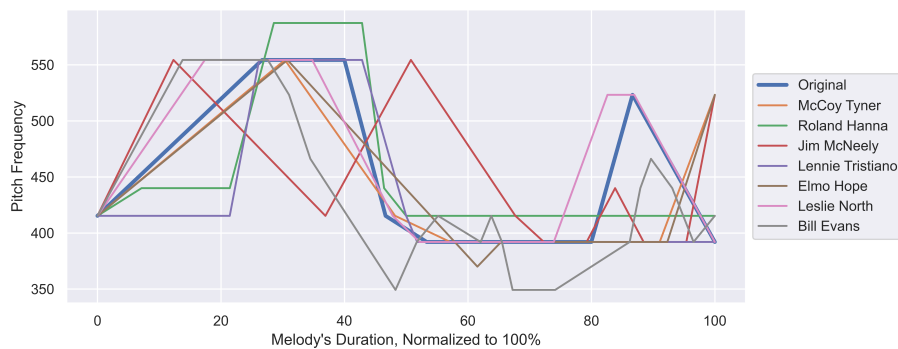


Fig. 2: The melodic contours of the melody taken from the jazz standard “All The Things You Are” (in Blue) and pianists’ interpretations.

**Harmony** The harmonies used within a performance can greatly impact the direction of the music and also the intention of the performer. To analyse some of the harmonic aspects of the dataset, we used Chordino and NNLS chroma [14]. We set out to find the rate of harmonic change across each performance. As shown in Figure 3, we found that some pianists had a higher harmonic rhythm (the rate of chord changes in a chord progression) than others. Other pianists added more chords to the chord progression, which sped up the harmonic rhythm. We observed that most pianists played in the key of the *Original*, however, some transposed keys. Some pianists used the same chord progression as the *Original* but altered specific chords. For example, Jim McNeely used a

<sup>3</sup> Contourviz can be found in: <https://github.com/cjwit/contourviz>



similar chord progression to the *Original*, but modified a minor chord to major, resulting in a significant shift in the performance’s intention and musical direction. We also observed that certain pianists used extended chords more extensively than others who played more closely to the *Original*. Other pianists added more chords to the chord progression, which sped up the harmonic rhythm.

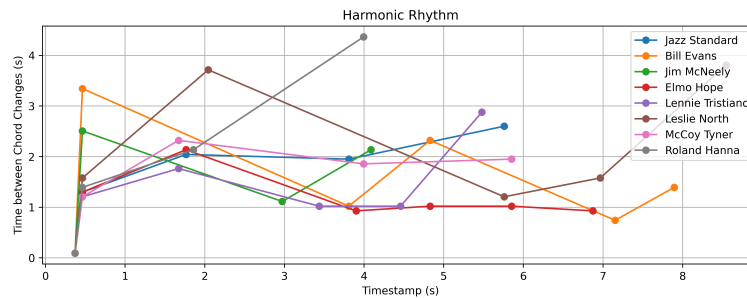


Fig. 3: A line graph comparison of the Harmonic Rhythm of the original melody (in Blue) and pianists’ interpretations of the melody.

Our analysis shows that the dataset contains a diverse range of interpretations, even when playing the same jazz standard. Within jazz, performers are individualistic and can be creative with their musical choices. The differences in melodic deviations, melodic contours, and harmonic rhythms between performances not only demonstrate the artistic freedom of each pianist but also indicates that the dataset could be a useful resource for those interested in expressive performance analysis or performer identification tasks.

## 5 Music Overpainting

### 5.1 Problem Definition

As defined in Section 1, Music Overpainting is a generative music task that aims to create variations on pre-existing music sections. Within the context of the JAZZVAR dataset, we can specifically define the task as generating a *Variation* segment from a given *Original* segment. Given an *Original* jazz standard segment  $O$  from the JAZZVAR dataset, and a *Variation* segment  $V$ , the goal of the Music Overpainting task is to find a reinterpretation  $I(O)$  such that:

$$V = I(O) \quad (2)$$

### 5.2 Generation with Music Transformer

Transformers have been widely applied to generate music in genres such as Pop, Classical, as well as Jazz [10, 9, 20]. Their convincing output demonstrate their capability of modeling musical structures and patterns. In this work, we adopted the design of

Music Transformer [9] which uses music motifs as primers for conditional generation. To train the transformer model, we concatenated the *Variation* segments to the end of the *Original* segments for each pair in the JAZZVAR dataset. In total, we obtained 502 concatenations and used 90% for training and 10% for validation. For the inference process, we treated the *Original* segment as a primer and generated a *Variation* segment following the probability distribution learned by the transformer model.

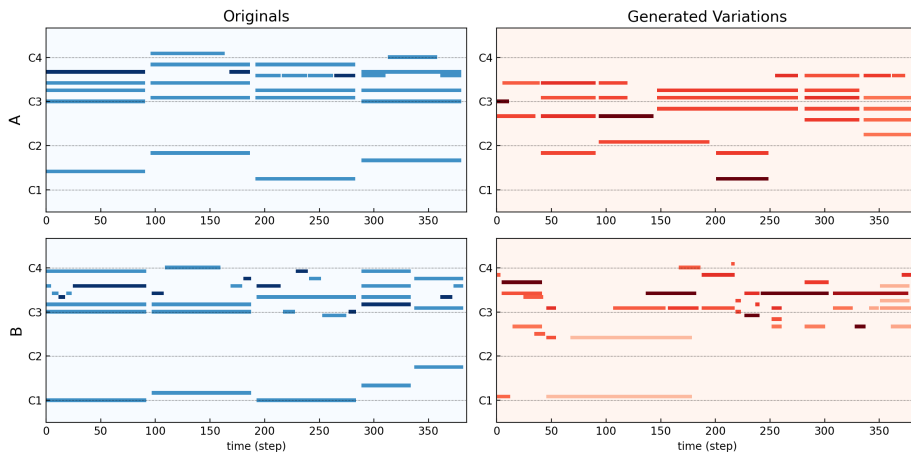


Fig. 4: Piano-rolls of two *Original* (left in Blue) and the corresponding generated *Variation* (right in Red) sections. The *Original* A is from the song “All the Things You Are”, and the *Original* B is from the song “Alfie”.

### 5.3 Results

We present piano-rolls of two *Original* segments, referred to as **A** and **B**, and the corresponding generated *Variation* segments<sup>4</sup> with *Original* segments used as primers to the model in Figure 4. We use the same pitch-related features calculated for the dataset in Table 2 to compare the *Original* segments and the corresponding generations. According to these results, we observe that the generated *Variation* segments are more complex and diverse in terms of the music features presented in Table 4, as well as the articulation and dynamics. By listening to the generations, we find that the model’s ability to accurately preserve the melody and chord patterns of the *Original* segment in the generated output can be improved.

<sup>4</sup> Listening samples of the generations can be found at <https://drive.google.com/drive/folders/13SmiT2AevqP3ma3xWy4LanQwcjyR1LG1?usp=sharing>

Table 4: Comparison of musical features for the *Original* and the generated *Variation* segments.

Feature	Original		Generated Variation	
	A	B	A	B
Pitch Class Entropy	2.73	2.75	2.71	2.86
Pitch Range	28.00	36.00	34.00	36.00
Polyphony	3.98	2.74	4.88	4.68
Number of Pitches	12.00	17.00	13.00	14.00
Scale Consistency	1.00	0.90	1.00	0.98

## 6 Conclusion

We present the JAZZVAR dataset a collection of 502 MIDI pairs of *Variation* and *Original* segments. We evaluated the dataset with regard to several musical features and compared the melodic and harmonic features of *Variations* for different pianists performing the same *Original* jazz standard. Our results indicate the diversity and complexity of *Variation* in the dataset, which is one important component for successfully training a specialised generative music model. We introduced the Music Overpainting task, and trained a Music Transformer using the JAZZVAR dataset to generate *Variation* segments with the *Original* segments as primers.

Having a collection of *Variations* performed by different pianists on the same jazz standard allows us to apply the dataset to explore tasks such as performer identification and expressive performance analysis. We aim to expand the JAZZVAR dataset in the future, using our collection of AMT MIDI data of jazz performances and corresponding jazz standards. This could either be achieved through the manual matching method as shown in Section 3.3, or through an automatic method, which would allow for a greater number of *Original* and *Variation* pairs to be produced. We believe that the deep generative models for the Music Overpainting task will greatly benefit from the increment of dataset size.

## References

1. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., and Lamere, P.: The million song dataset. In: International Society for Music Information Retrieval conference (2011)
2. Briot, J.-P.: From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Computing and Applications* 33(1), 39–65 (2021)
3. Briot, J.-P., and Pachet, F.: Deep learning for music generation: challenges and directions. *Neural Computing and Applications* 32(4), 981–993 (2020)
4. Dai, S., Zhang, Z., and Xia, G.G.: Music style transfer: A position paper. In: The 6th International Workshop on Musical Metacreation (2018)
5. Dixon, S., Crayencour, H., Velichkina, O., Frieler, K., Höger, F., Pfeiderer, M., Henry, L., Solis, G., Wolff, D., Weyde, T., *et al.*: History of Recorded Jazz: DTL1000, 1920-2020. (2022)

6. Dong, H.-W., Hsiao, W.-Y., Yang, L.-C., and Yang, Y.-H.: MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In: AAAI Conference on Artificial Intelligence (2018)
7. Gotoh, O.: An improved algorithm for matching biological sequences. *Journal of Molecular Biology* 162(3), 705–708 (1982)
8. Hawthorne, C., Simon, I., Swavely, R., Manilow, E., and Engel, J.: Sequence-to-sequence piano transcription with transformers. In: International Society for Music Information Retrieval conference (2021)
9. Huang, C.-Z.A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A.M., Hoffman, M.D., Dinculescu, M., and Eck, D.: Music transformer. In: International Conference on Learning Representations (2019)
10. Huang, Y.-S., and Yang, Y.-H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In: Proceedings of the 28th ACM international conference on multimedia. MM '20, pp. 1180–1188. Association for Computing Machinery, New York, NY, USA (2020)
11. Kong, Q., Li, B., Chen, J., and Wang, Y.: GiantMIDI-Piano: A Large-Scale MIDI Dataset for Classical Piano Music. *Transactions of the International Society for Music Information Retrieval* (2022)
12. Kranenburg, P. van, Volk, A., Wiering, F., and Veltkamp, R.C.: Musical Models for Melody Alignment. In: International Society for Music Information Retrieval Conference (2009)
13. Manilow, E., Wichern, G., Seetharaman, P., and Roux, J.L.: Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity. 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (2019)
14. Mauch, M., and Dixon, S.: Approximate note transcription for the improved identification of difficult chords. In: International society for music information retrieval conference (2010)
15. Mukherjee, S., and Mulimani, M.: ComposeInStyle: Music composition with and without style transfer. *Expert Systems With Applications* 191, 116195 (2021)
16. Pati, A., Lerch, A., and Hadjeres, G.: Learning To Traverse Latent Spaces For Musical Score Inpainting. In: International Society for Music Information Retrieval conference (2019)
17. Pfeleiderer, M., Frieler, K., Abeßer, J., Zaddach, W.-G., and Burkhart, B. (eds.): Inside the Jazzomat - New Perspectives for Jazz Research. Schott Campus (2017)
18. Raffel, C., and Ellis, D.P.W.: Optimizing DTW-based audio-to-MIDI alignment and matching. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 81–85 (2016)
19. Wei, S., Xia, G., Zhang, Y., Lin, L., and Gao, W.: Music phrase inpainting using long-term representation and contrastive loss. In: ICASSP 2022 - 2022 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 186–190 (2022)
20. Wu, S.-L., and Yang, Y.-H.: The jazz transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures. In: International Society for Music Information Retrieval Conference (2020)
21. Wu, X., Hu, Z., Sheng, L., and Xu, D.: StyleFormer: Real-time arbitrary style transfer via parametric style composition. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
22. Zhang, H., Tang, J., Rafee, S.R., Dixon, S., Fazekas, G., and Wiggins, G.A.: ATEPP: A Dataset of Automatically Transcribed Expressive Piano Performance. In: International Society for Music Information Retrieval Conference, pp. 446–453 (2022)

# A Live Performance Rule System Informed by Irish Traditional Dance Music

Marco Amerotti<sup>1</sup>, Steve Benford<sup>2</sup>, and Bob L. T. Sturm<sup>1</sup> and Craig Vear<sup>2\*</sup>

<sup>1</sup> Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm Sweden

<sup>2</sup> University of Nottingham {amerotti,bobs}@kth.se, {steve.benford,craig.vear}@nottingham.ac.uk

**Abstract.** This paper describes ongoing work in programming a live performance system for interpreting melodies in ways that mimic Irish traditional dance music practice, and that allows plug and play human interaction. Existing performance systems are almost exclusively aimed at piano performance and classical music, and none are aimed specifically at traditional music. We develop a rule-based approach using expert knowledge that converts a melody into control parameters to synthesize an expressive MIDI performance, focusing on ornamentation, dynamics and subtle time deviation. Furthermore, we make the system controllable (e.g., via knobs or expression pedals) such that it can be controlled in real time by a musician. Our preliminary evaluations show the system can render expressive performances mimicking traditional practice, and allows for engaging with Irish traditional dance music in new ways. We provide several examples online.<sup>3</sup>

**Keywords:** Music performance modeling, traditional music, Irish

## 1 Introduction

The performance of an Irish traditional dance tune involves ornamentation and variation over repetitions. Some practitioners employ small variations where the tune is always recognizable (e.g., Irish accordionist Derek Hickey calls these “microvariations” of the “bones”<sup>4</sup>), while others move far away from the tune (e.g., the fiddler Tommy Potts is well-known as an extreme example). The ornamentation and variation employed in a performance are often guided by the instrument one is playing, which certain choices are made based on the accessibility of pitches, physical constraints, range, and so on.

\* Portions of this work are outcomes of projects that have received funding from the European Research Council under the European Union’s Horizon 2020 research and innovation program: DigiScore (Grant agreement No. 101002086) and MUSAiC (Grant agreement No. 864189).

<sup>3</sup> See this website: <https://www.kth.se/profile/bobs/page/research-data>.

<sup>4</sup> Private communication in a lesson with author Sturm.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Irish traditional music is by and large an aural tradition, where notated music (“the dots on the page”) is passed over in favor of listening to master musicians interpreting tunes and then imitating their creative choices. When Irish traditional dance music is notated, the convention is one of writing what one thinks are the most important notes, leaving ornamentation and variation to the performer. Computer playback of such notated music thus lacks important elements of the traditional music performance: how might we program a performance system so that its rendition is closer to real-life performance?

While there is much research in modeling expressive music performance, e.g., classical piano, we do not find work devoted to traditional music. This paper presents a performance system focused on Irish traditional dance music aiming to render performances that mimic the practice. Our system operationalizes expert knowledge into a set of rules binding musical elements, such as ornamentation and dynamics, to performance parameters for controllable MIDI synthesis. The lack of explicit performance data in the context of Irish music motivates an expert-knowledge-driven, rule-based approach, which is both computationally efficient and sufficient to create at least a baseline model for Irish traditional music performance. Furthermore, since the performance of Irish traditional dance music can involve heterophony (multiple musicians playing their own versions of the same tune together), we make our system real-time and controllable such that one can play *with* it in a live performance scenario. In the next sections we review existing work in music performance modeling, as well as conventions in Irish traditional music performance. We describe our system and how its components operationalize expert knowledge. We then provide some preliminary evaluation of its output, and discuss its use in the context a live performance. Future research is discussed in the conclusion.

## 2 Background

We now review research in the modeling of music performance. We then discuss specific characteristics in the performance of Irish traditional dance music.

### 2.1 Existing work in modeling music performance

Music performance modeling [4] is aimed at making machines perform music in expressive ways. This is accomplished by translating musical elements, such as pitches, phrases, and timing, into expressive parameters, such as articulation, loudness, dynamics, and phrasing. One example is the “KTH rule system” for musical performance [5], which applies a user-weighted rule-based estimation of expressive parameters for each note of a piece. The set of rules has been implemented in the software package *Director Musices* [6], which allows one to inspect the generated expressive contours.

Most work in music performance modeling is aimed at the performance of classical music, but a growing number of studies focus on popular music and jazz performance [4]. While there exists research in the analysis of traditional music practice, we do not find any attempting to generate such performances. For traditional music, computational approaches are usually employed for performance *analysis* rather than *synthesis* [12,16,15,13].

The most commonly modeled parameters among performance systems include loudness, tempo, ornamentation and articulation, and so the MIDI protocol is often used since it allows some amount of modeling of the above through velocity, timing, pitch, and control messages (e.g., pitch bend). Expressive parameters are often modeled jointly since they can be highly related, e.g., tempo and dynamics [17]. Moreover, since human performance can go beyond the written score, such as ornamentation and style-specific musical practices, some work has explored the modeling of such performance conventions, e.g., ornamentation of lead sheets in the performance of jazz standards [7]. Another example is the MusicTransformer system [9], which can generate realistic accompaniments and performances given only melody input. Improvisation and variation are usually ignored when modeling classical music performance, but other styles (jazz and some folk traditions) consider them essential aspects of expressive performance.

## **2.2 Performance of Irish traditional dance music**

Irish traditional dance music has a history going back a few centuries at least [3,18,8]. A dance tune consists of parts, each typically built from simple musical ideas unfolding over two to four beats. These parts are often repeated in performance, as is the whole tune. Common dances are the reel, jig, hornpipe, and polka, each executed with characteristic rhythms. Tunes are modal, most often in major, mixolydian, dorian or minor, and typically involve melodic motion that combines stepwise movement with arpeggiated chords. Ornamentation is an essential aspect of traditional performance, contributing to the rhythmic drive of a dance tune.

Irish traditional music is an aural practice, the expert performance of which does not involve playing tunes “as written”. Figure 1 notates the A part of the well-known jig, *The Connachtman’s Rambles*, as printed in “O’Neill’s 1001” [14], along with a transcription of one of its repetitions performed by master musician Máirtín O’Connor. This shows his variation of the jig rhythm, playing with the timing of quavers within each beat. He uses a variety of “cuts” (a grace note ornament emphasizing the attack of the following note), some of which provide tonal value to establish a counter melody (bars 11–12).

O’Connor’s performance of this tune demonstrates how the practice of the music involves “microvariations”, which lends itself well to performance in “sessions” where musicians of varied abilities gather informally to play tunes together. While varying greatly, sessions tend to exhibit some common characteristics including [2] performers joining and leaving throughout, numerous and diverse melody instruments playing in unison (often accompanied by a few guitars, citterns and bouzoukis), musicians with different skills – from beginners to seasoned experts – playing alongside one another, and playing and learning by ear more often than playing from printed music. Furthermore, sessions feature tunes linked together in “sets” of two or more, each repeated a number of times. This structure allows tunes to be learned by ear or recalled to the fingers before then being embellished on subsequent repeats. There is also a degree of improvisation in selecting tunes that fit well together as sets and in guessing which tunes other players might or might not know and/or be able to pick up.

*Double jig #218 in ``O'Neill's 1001''*

*Máirtín O'Connor*

*Ornamented by our performance system*

The image displays a musical score for a double jig in 6/8 time, titled "The Connachtman's Rambles". It consists of seven staves of music. The top staff is the original printed version from "The Dance Music of Ireland: O'Neill's 1001". The middle three staves represent a performance by Máirtín O'Connor on accordion in 1979, transposed down to D from Eb. The bottom three staves show the performance as interpreted by a system, which includes various ornaments (trills, grace notes) and triplet markings (indicated by the number '3' above or below groups of notes) that are not present in the original score.

**Fig. 1.** The A part of *The Connachtman's Rambles*. Top: as printed in "The Dance Music of Ireland: O'Neill's 1001". Middle: as performed by Máirtín O'Connor on accordion in 1979 (transposed down to D from Eb). Bottom: as interpreted by our performance model. The performance hyperparameters were set to create a performance similar to Máirtín O'Connor's.

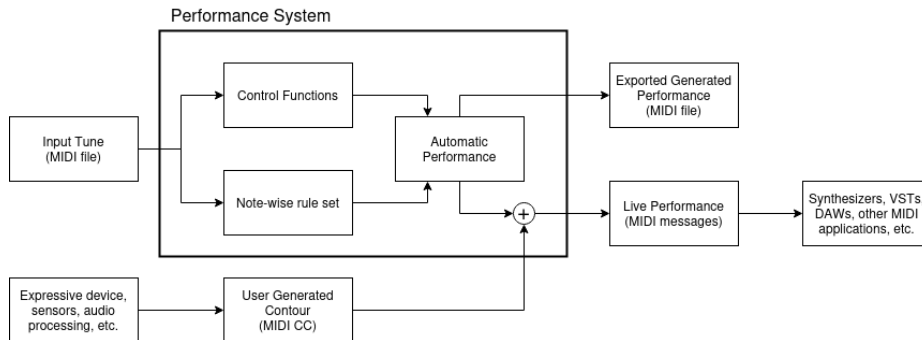
### 3 Performance System

We now present our performance system, which processes MIDI input and outputs control parameters to synthesize an expressive MIDI performance that can be exported on its own without human interaction, or input in real-time to any software or hardware with MIDI input capabilities. The resulting performance incorporates style-specific ornaments, time deviations, and dynamics to reflect conventions of Irish traditional dance music practice. The performances of each of these three aspects (ornamentation, dynamics, tempo) are modeled with expert-knowledge-based rules and functions, user-specified performance parameters, and metadata in the MIDI file itself (e.g., the key signature MIDI meta-message).

The expert-knowledge-based rules and functions are motivated not only by our own practical knowledge of Irish traditional dance music performance, but also that of noted Irish musician and theorist Tomás Ó Canainn [18]. In his analysis of Irish music, Ó Canainn presents a formalism of note importance in which he assigns points to each note appearing in a tune:

- 1) a note frequency count giving a point for each appearance of the note; 2) the addition of a further point (a) to a note which occurs on a strong beat, (b)





**Fig. 2.** The performance system pipeline. An input MIDI tune is processed by computing control functions to guide the performance and by evaluating rule-sets for each note (e.g. to generate ornamentation). The performance can be live, and optionally steered in real-time, or otherwise exported to a MIDI file.

to the highest note on its first appearance, (c) to the lowest note on its first appearance, (d) to a note preceded to by a leap greater than a fifth, (e) to the first stressed note, (f) to a long note (e.g., a dotted crotchet in a jig).

Inspired by Ó Canainn our system assigns five scores to each note in a tune, and then uses these to derive and apply control functions to guide the resulting performance. For each note-on event of a MIDI source, we compute the following scores:

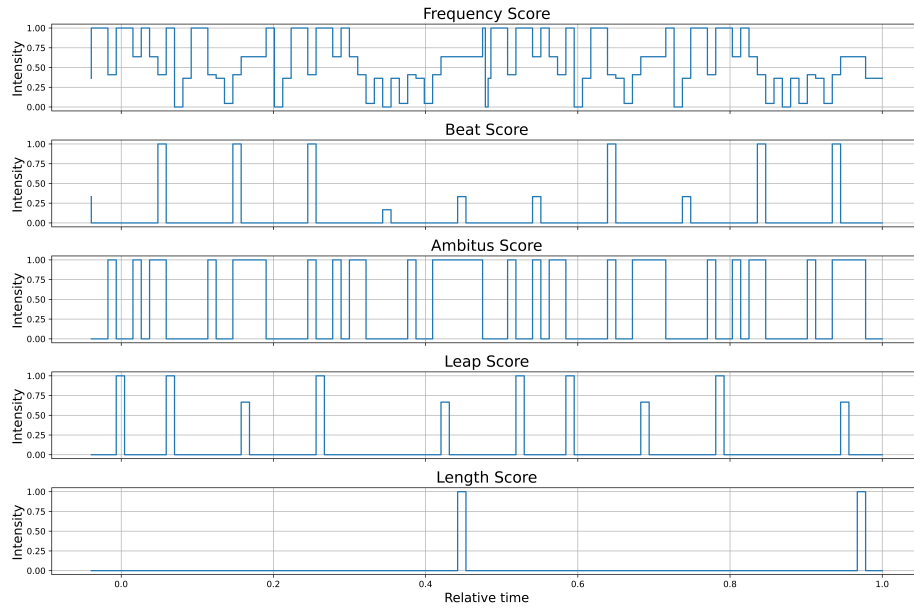
1. a number of points equal to the number of occurrences of that pitch class in the entire tune (*frequency score*);
2. if the note occurs on a strong beat, it gets a number of points equal to the number of times the pitch occurs on a strong beat; otherwise zero (*beat score*);
3. a point if it is either the highest or the lowest pitch of the tune (*ambitus score*);
4. a point if the interval leading to it is greater or equal to a fifth (*leap score*);
5. a point if its duration is longer than the mode of the note durations of the tune (*length score*).

To generate control functions the system normalizes these scores and linearly combines them to manifest particular musical qualities relevant to the three modeled performance aspects. We hand-craft these linear combinations through a combination of formalizing our musical experiences and expectations, as well as trial and error, e.g., that a cut is more likely to occur between a repetition of a pitch and on a strong beat. Table 1 shows the weightings involved, which have proven to be sufficient at this preliminary stage, but work is required to determine their sufficiency for modelling real performance. Finally, we apply smoothing to these functions to reduce extreme sudden variations that make the performance erratic. In particular, we employ a third-order Savitzky–Golay filter with a window size of 15 notes, and use mean-value padding at the edges.

To illustrate the procedure of generating control functions, consider the A part of *The Connachtman’s Rambles* from Fig. 1. Figure 3 show the five series of scores derived

Control function	frequency	beat	ambitus	leap	length
Ornaments	0.2	0.3	0.15	0.15	0.2
Dynamics	0.1	0.25	0.25	0.2	0.2
Tempo	0.25	0.1	0.3	0.25	0.1

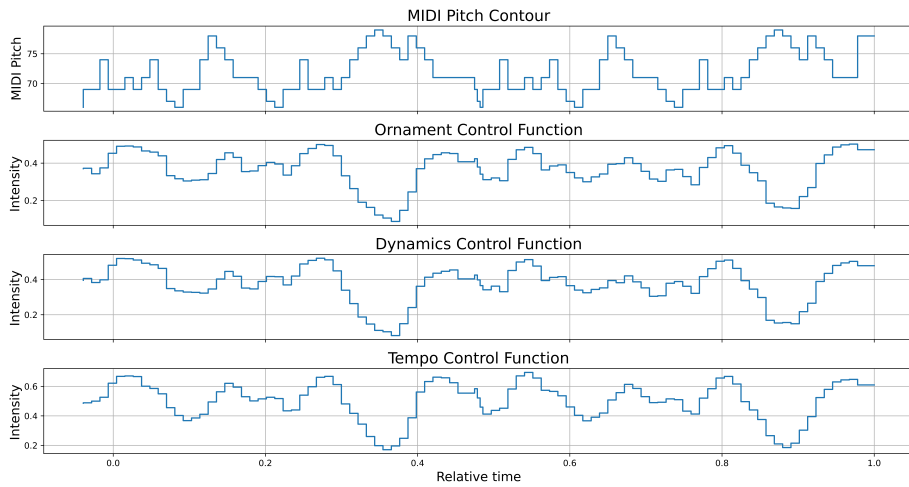
**Table 1.** Score weighting for computing control functions. For instance, the ornament score is given by the sum  $0.2 \cdot \text{frequency score} + 0.3 \cdot \text{beat score} + 0.15 \cdot \text{ambitus score} + 0.15 \cdot \text{leap score} + 0.2 \cdot \text{length score}$ .



**Fig. 3.** Individual scores used to generate control functions for each modeled aspect for the A part of *The Connachtman's Rambles* (Fig. 1). The single note-wise scores are summed and weighted according to Table 1 to generate the control functions.

using the above method. Figure 4 show the control functions resulting from the linear combination and smoothing for each performance model aspect. For each note in the input melody, the system uses the corresponding value of the control functions to affect its performance according to each modeled aspect.

Being intended not only as a performance generator, but also as an agent capable of performing live with another musician, we make our system responsive to an external user-defined control signal to steer the performance, e.g., an accompanying guitarist with an expression pedal. The following subsections describe how our system models each of the three performance aspects and how each of those is influenced by a control function.



**Fig. 4.** From the top, MIDI Pitch contour and generated ornament, dynamics, and tempo control functions for the A part of *The Connachtman’s Rambles* (Fig. 1). The ornament control function is directly mapped to the probability of ornamenting a note; the dynamics control function is mapped to note velocity in the range [0, 127]; the tempo control function is mapped to a tempo drift percentage with 0.5 being the original tempo, higher/lower values meaning a faster/slower tempo.

### 3.1 Performing Ornamentation

Ornaments in Irish traditional music are partly dependent on the physical characteristics of the traditional instruments [18] (e.g., sliding between pitches on the fiddle, “tapping” on the tin whistle to articulate repeated notes) and musical characteristics (e.g., to accentuate beats in accordance with traditional dancing). On top of that, musicians may impose personal stylistic preferences and improvisational elements. We focus on the modeling of three of the most common ornaments which can be easily modeled with MIDI, and are not exclusive to particular instruments by and large. The particular ornaments performed by our system are slides, cuts and rolls:

- A slide entails approaching a note from a lower pitch. We create a slide ornament by using a series of MIDI pitch-bend messages between a notes and next lowest scale degree.
- A cut is like a grace-note that emphasizes the attack of a note, or separates repeated notes. We create a cut by adding a short note one scale degree above the note.
- A roll is a decoration of three quavers that involves separating each with cuts, similar to the classical “gruppetto” or “turn”, and consist in approaching the pitch from above and then from below. We create a roll by adding the appropriate notes.

In our system, the pitches of rolls and cuts are drawn from the mode of the tune.

For each note of a MIDI file, the system determines if it is a candidate for ornamentation with a probability computed from the ornament control function and user-

specified parameters. If it is a candidate, the system selects one of the ornaments, or possibly drops the note (a “humanizing” of the performance), at random.

### 3.2 Performing with dynamics and tempo deviations

Our systems models dynamics via MIDI note velocity using the dynamics control function, which is scaled to the range  $[0, 127]$  and applied directly to the note velocity parameter. Notes falling on a beat are further accentuated by increasing their velocity. To humanize the performance, the system implements tempo deviations around a user-specified tempo by locally warping the performance tempo using the tempo control function. The motivating idea is that musicians will tend to speed up or slow down at times the melody has certain characteristics, e.g., ornamentation, repeated notes, and melodic leaps.

### 3.3 Human Interaction

Our system is capable of continuously reading an external MIDI control change signal on a specified MIDI control number during a performance with an accompanying musician. This signal is scaled to  $[0, 1]$  and interpolated with the control functions with a user-defined weight. For instance, the musician can make the performance system play without ornamentation at first, and then gradually make it more adventurous. While basic, this approach can be quite versatile since any kind of user-generated control function can manipulate the MIDI control signal processed by the system. Possibilities include conventional controls, e.g., MIDI pedals and knobs, but also unconventional ones such as body sensors.

## 4 Preliminary Evaluation

We now conduct a preliminary evaluation of our performance system to determine its effectiveness and chart future work for development. We first compare an expert performance of *The Connachtman’s Rambles* (Fig. 1) with that of our system to determine acceptable parameters. We then apply the performance system to a novel tune generated by Folktune-VAE [1] – a model trained on Irish traditional music – and gauge its plausibility with respect to traditional practice. Audio examples of generated performances are available on our website.

Figure 1 shows transcriptions of two performances of the A part of *The Connachtman’s Rambles*, one by an expert and another generated by our system. We see that the system is able to mimic some of the ornamentation of the expert, with a general difference in the pitch used to cut. Most of O’Connor’s cuts are pitches a third above – which are convenient to do on the accordion because of the physical distribution of its pitches. Our system at this time is not instrument-specific and produces “generic” cuts within the mode of a tune. In terms of rhythm, our model accentuates the beats using cuts as in the human performance. Rhythmic swing is clear in the human performance (emphasizing the jig rhythm), while the performance system shows none since it is only subtly adjusting tempo at this time. The performance system stays with a steady



**Fig. 5.** A tune generated by the model *FolkTune-VAE* [1]. Top: original tune as generated by the model. Bottom: as interpreted by the performance system. A rolled note is notated with a tilde.

tempo but introduces subtle drifts. While still far from the human performance, our system generates stylistically coherent elements that are more expressive than basic MIDI synthesis.

We now analyze how our system performs the machine-generated tune shown at top of Fig. 5. There is of course no reference performance for this tune, but we can gauge the stylistic coherence of our system’s performance. The transcription, shown at the bottom of Fig. 5, shows the system mainly employs cuts and occasional slides and dropped notes. Cuts are placed appropriately, e.g., on the start of bars or between repeated notes. The system generates a roll at the start of the B part on a dotted crochet, which is also consistent with the practice. Listening to the performance, the reel rhythm is clear and the dynamics are varied throughout.

While preliminary, this evaluation shows our system shows some success in rendering a performance of melodies in ways that are consistent with the practice of Irish traditional dance music. Linking probabilistic decisions with higher-level control signals derived from the music content make expressive and varied renditions that are more interesting than straight MIDI playback. Furthermore, adding interactivity makes for a dynamic playing partner.

## 5 Discussion: The system as a musician

The inspiration for our system arose from an ongoing project to explore how human and AI musicians might perform together. Our distinctive focus was on how humans might accompany AI, specifically on how a human guitarist might improvise an accompaniment to AI-generated and performed tunes. First, a human musician (Benford) used the

*FolkRNN* system<sup>5</sup> to generate around twenty tunes, from which eight were selected and segued to form two sets – a set of four reels and a set of four jigs. While this yielded sets of tunes potentially interesting to accompany, the automated playback of the resulting MIDI files through a standard digital audio workstation was, unsurprisingly, flat and immediately striking for its lack of variation when tunes were repeated. There was no sense of the system pushing the performer or vice versa, and the human accompanist was left to do all of the work in making the performance dynamic and interesting.

Our preliminary exploration involved a series of scripts generating ornamentation, micro timing, and pitch errors<sup>6</sup> based on random chances alone; with those, a human musician generated twelve variations of each tune with different parameters (three levels of ornamentation, note pitch error, micro timing, and three combinations of all of them with the scripts applied in sequence). The human musician listened to and compared these, selecting three versions of each tune to be sequenced together as part of the overall set. This process already led to some immediate insights and inspirations for further developments.

The traditional ornamentations of rolls, cuts, and slides worked to introduce aesthetically pleasing variation. Timing errors and dropped notes on the other hand gave a sense of the agent struggling to play the tune, or being tentative, giving the impression that it was learning it or trying to recall it back to its fingers. Pitch errors were sometimes heard as mistakes, but sometimes as more as attempted ‘jazzy’ (chromatic) improvisations, especially if the system was otherwise performing fluidly (*e.g.*, was introducing odd pitched without obvious timing errors).

These observations led to the idea that the system might have a persona that would support a narrative through the performance that would make sense of their variations. For example, they might be a learner struggling to learn new tunes and/or to master their instrument, or alternatively, a skilled and proficient player quickly trying to recall ‘out of practice’ tunes. The latter felt particularly appealing, as a skilled musician might conceivably play with various embellishments, but also errors depending on their situation, and might even be expected to vary these through a performance. The journey through such a narrative should be interactive, *i.e.* the human should be able to influence it. This inspired the idea of a simple control based on an expression of musical intensity; that the human musician should be able to signal that they would want their skilled AI collaborator to play with more or less intensity. This might be interpreted in various ways. Lowering intensity might signal the AI to back off, perhaps playing more solidly (if boringly) and ultimately more tentatively (as if trying to recall a tune). Raising intensity might cause it to introduce more traditional embellishments, and eventually introduce jazzy improvisations or even take risks that would lead to mistakes. Intensity might be signaled explicitly (*e.g.* through the expression pedal) or perhaps detected automatically from the accompanist’s own playing. In this case, our performance system allows us to hear a tune that is foreign to the tradition and which would have to be physically learned and played by a musician to be heard otherwise.

---

<sup>5</sup> <https://folkrrnn.org>

<sup>6</sup> The pitch errors were not included in the system presented here as we felt that more work was needed in modeling this aspect.

In summary, the variations introduced by our performance system felt potentially productive in inspiring accompaniment, but might benefit from the creation of an underlying musical persona and narrative for the system that would help make sense of them and enable them to be influenced during the performance. Our goal was to generate a performance an authentic performance, more than an ideal one: such a performance cannot be modeled without having an underlying musician, or an idea of them.

## 6 Conclusion

To the best of our knowledge, this is the first work explicitly modeling the performance of folk music, and Irish traditional dance music in particular. We have operationalized expert knowledge to form a rule-based performance system that is able to render a melody expressed in MIDI as an expressive and dynamic performance that exemplifies conventions of the practice. Performance aspects that we have modeled include various ornaments, tempo variations and dynamics. Our preliminary evaluation demonstrates the effectiveness of our system and points to its usefulness in human-AI co-performance. The system has clear limitations, however. For example, its parameters and hyperparameters are currently set by trial and error, but could be estimated from expert performances. Another limitation is the performances it generates are not instrument specific, possibly rendering a performance on a synthetic instrument that would not be typical or even possible on a real instrument. More work should thus be conducted to improve this baseline system, e.g., further humanizing the performance, introducing melodic variation and extemporization, and modeling specific instruments. Another aspect deserving of further work is the rendering of a *session* performance, where two or more artificial performers play a melody together.

While the task of creating a performance system presents common elements across styles, datasets, and approaches [4], a main difference in the modeling of folk music performance is *performance creativity* [11], not limited to creativity in planning the performance, but ranging from adding ornamentation and micro timing to explicitly playing wrong notes and drifting in tempo. This is in contrast to the performance of classical music, where one must play a score as written without mistakes, but taking liberties with phrasing, dynamics, articulation, and tempo. It might seem at first that modeling the performance of just a melody is a trivial matter, but the conventions of the practice of Irish traditional dance music bring subtle and interesting challenges. Folk music is more complex than it appears at first. The modeling of folk music performance presents challenges that are different from those when modeling other styles, and in our opinion should be pursued to enrich opportunities engaging with music traditions.

Of the performance systems dealing with performance creativity listed in [11], most of them present very limited evaluation, if any at all. An obvious way of evaluating a performance system is through listening tests, RENCON [10] being a key example. Our preliminary evaluation analyzes generated performances from an expert-knowledge-based perspective, but future work can conduct listening tests as done at RENCON, both of the system's output and its application in the context of human-AI co-performance. This latter aspect brings ambiguity, however: if a performance system is difficult to play with, it could be a bad performance system, or a good emulation of a bad musi-

cian. Nonetheless, we aim for rendering performances that are expressive and faithfully reflect practical conventions.

## References

1. AMEROTTI, M. *Latent representations for traditional music analysis and generation*. Tesi di laurea, Università di Bologna, Oct. 2022.
2. BENFORD, S., TOLMIE, P., AHMED, A. Y., CRABTREE, A., AND RODDEN, T. Supporting traditional music-making: designing for situated discretion. In *Proc. of the ACM 2012 Conf. on Computer Supported Cooperative Work* (2012), pp. 127–136.
3. BREATHNACH, B. *Folk Music and Dances of Ireland: A comprehensive study examining the basic elements of Irish Folk Music and Dance Traditions*. Ossian, 1971.
4. CANCINO-CHACÓN, C. E., GRACHTEN, M., GOEBL, W., AND WIDMER, G. Computational Models of Expressive Music Performance: A Comprehensive and Critical Review. *Frontiers in Digital Humanities* 5 (2018).
5. FRIBERG, A., BRESIN, R., AND SUNDBERG, J. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology* 2 (2006), 145–161.
6. FRIBERG, A., COLOMBO, V., FRYDÉN, L., AND SUNDBERG, J. Generating Musical Performances with Director Musices. *Computer Music J.* 24, 3 (Sept. 2000), 23–29.
7. GIRALDO, S., AND RAMÍREZ, R. A machine learning approach to ornamentation modeling and synthesis in jazz guitar. *J. of Mathematics and Music* 10, 2 (May 2016), 107–126.
8. HALLMHURÁIN, G. O. *A pocket history of Irish Traditional Music*. The O’Brien Press, 1998.
9. HUANG, C.-Z. A., VASWANI, A., USZKOREIT, J., SHAZEER, N., HAWTHORNE, C., DAI, A. M., HOFFMAN, M. D., AND ECK, D. Music transformer: Generating music with long-term structure. *arXiv preprint arXiv:1809.04281* (2018).
10. KATAYOSE, H., HASHIDA, M., DE POLI, G., AND HIRATA, K. On evaluating systems for generating expressive music performance: The RENCON experience. *J. of New Music Research* 41 (12 2012), 299–310.
11. KIRKE, A., AND MIRANDA, E. R. *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*. Springer, 2021, ch. Performance Creativity in Computer Systems for Expressive Performance of Music, pp. 521–584.
12. LERCH, A., ARTHUR, C., PATI, A., AND GURURANI, S. An interdisciplinary review of music performance analysis. *arXiv preprint arXiv:2104.09018* (2021).
13. MÜLLER, M., GROSCHE, P., AND WIERING, F. Automated analysis of performance variations in folk song recordings. In *Proc. Int. Conf. on Multimedia Information Retrieval* (Mar. 2010), pp. 247–256.
14. O’NEILL, F. *The Dance Music of Ireland: O’Neill’s 1001*. Chicago, 1907.
15. RAMIREZ, R., MAESTRE, E., PEREZ, A., AND SERRA, X. Automatic performer identification in celtic violin audio recordings. *J. of New Music Research* 40, 2 (2011), 165–174.
16. REN, Y., KOOPS, H. V., BOUNTOURIDIS, D., VOLK, A., SWIERSTRA, W., VELTKAMP, R. C., HOLZAPFEL, A., PIKRAKIS, A., ET AL. Feature analysis of repeated patterns in Dutch folk songs using principal component analysis. In *Proc. of the 8th International Workshop on Folk Music Analysis* (2018), pp. 86–88.
17. TODD, N. P. M. The dynamics of dynamics: A model of musical expression. *The J. of the Acoustical Society of America* 91, 6 (June 1992), 3540–3550.
18. Ó CANAINN, T. *Traditional Music in Ireland*. Routledge and Kegan Paul Ltd., 1978.



# VERSNIZ - Audiovisual Worldbuilding through Live Coding as a Performance Practice in the Metaverse

Damian Dziwis<sup>1,2</sup> \*

<sup>1</sup> TH Köln - University of Applied Sciences

<sup>2</sup> Technische Universität Berlin

damian.dziwis@th-koeln.de

**Abstract.** Even before the circumstances the global pandemic forced, a diverse ecosystem of technologies and artistic practices for performances in digital and virtual media was raising. Thus, not only is there a sustained interest in transferring existing performance practices into said media, but it also enables the emergence of new practices and art forms. In particular, immersive, networked, virtual multiuser environments (summarized under the term "metaverse") offer many possibilities for creating new art experiences that need to be explored. In this paper, we present VERSNIZ, a system for audiovisual worldbuilding, the spatial shaping of virtual environments, as a collaborative real-time performance or installation practice. It combines gamification concepts, known from popular sandbox video games, with the performance practice of live coding based on the esoteric programming language IBNIZ. We describe the technical implementation of the system, as well as the resulting artistic concepts and possibilities.

**Keywords:** Live Coding, Metaverse, Networked Music Performance, Virtual Installations, IBNIZ

## 1 Introduction

The term "metaverse" has become a real hype, and today all kinds of virtual and augmented reality (VR, AR) applications are often promoted as "metaverse". Used as a marketing term, the question of how these called metaverse applications differ from other VR/AR applications often remains unclear. Derived from the dystopian science fiction novel "Snow Crash" [1], the definition of the metaverse as "an interconnected web of social, networked immersive environments in persistent multiuser platforms" ([2], p. 1) seems to be gaining acceptance. According to this definition, the term metaverse environments is suitable for classifying online, multiuser, interactive, and interconnected virtual worlds, in contrast to other virtual applications. As virtual,

---

\* I thank Christoph Pörschmann (TH Köln), Stefan Weinzierl and Henrik von Coler (TU Berlin) for supervision and support in carrying out the research and writing this paper.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

freely experiential environments, metaverse worlds offer a high degree of immersion and many creative possibilities. The manifold ways of multimodal interaction via text, speech, and movement allow for diverse social exchange and self-expression [3]. Thus, they have many properties that make them suitable for art experiences [4], also in the context of music practice [5]. A particular challenge in the context of metaverse environments for art expression is real-time performances with multimedia content of music and visuals. Although there is a long history of realizations and concepts for live performances in metaverse environments [6], in recent years, they have come into focus for a broader audience due to the limitations imposed by the global pandemic. But as all related areas of the metaverse continue to grow and advance technologically, we can expect to see continued interest in these topics. Metaverse environments can provide an immersive environment for telematic/networked music performances (NMPs) [7] and are an environmentally friendly and barrier-free alternative for bringing together audiences and artists from around the world.

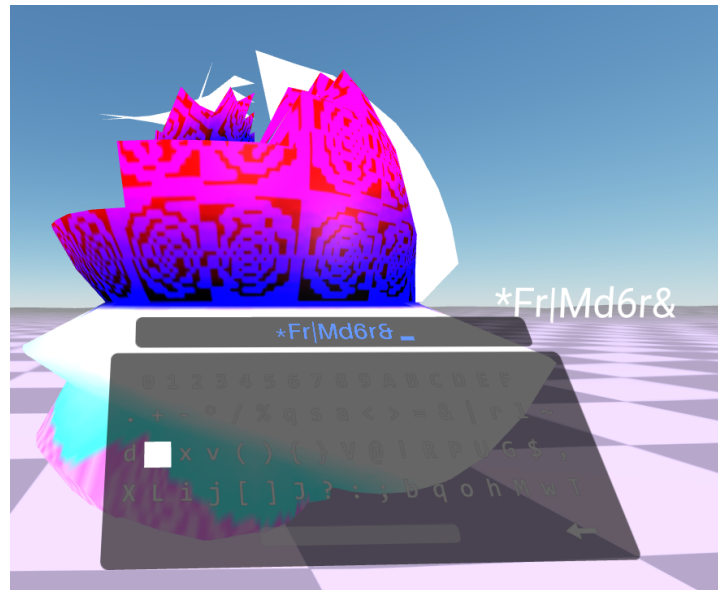
Here, not only the issue of suitable streaming technologies is in the foreground [8], but also the resulting artistic possibilities. It is not only about how to transfer existing performance practices into such virtual environments but also about which new practices can emerge from these environments. Virtual environments, such as in VR, already allow for expanded possibilities in terms of virtual instruments, composition, and performance practice [9,10,11]. They also offer the possibility of creating entire worlds under the concept of worldbuilding/worldmaking [12,13]. The technological overlap with video games also allows for the incorporation of gamification elements for composing or performing in such environments [14,15].

In comparison to the above-mentioned virtual instruments or systems for performances in enclosed virtual environments, we present a system for real-time performances and installations incorporating the audience in online, multiuser, metaverse environments. Because they are shared virtual environments that are also accessible to the public via the Internet, they enable audience participation in virtual performance and composition processes and thus new art practices. How can the role of the performer, the stage, and the audience be redefined in this process? To explore the emerging possibilities of such environments, we combine the gamification concept of worldbuilding with the practice of live coding. As a result, we propose a performance and composition system that is only feasible in metaverse environments.

In the following, we describe the development of the metaverse environment "VER-SNIZ" for audiovisual live coding using the IBNIZ programming language [16]. The implementation integrates the above-mentioned concepts, to dissolve conventional ideas of performer, audience, and stage and enable a new performance practice in metaverse environments.

## **2 Background**

A long and wide-ranging history of virtual environments as a medium can be found in video games. The perspective of the recipient is significant here. Usually, the user does not take the passive role of a spectator, but is an active protagonist in narrative scenarios or collaborative games. In concepts like sandbox games, such as the popular



**Fig. 1.** A screen capture from the VERSNIZ environment, showing a live coding object placed in the default virtual world. It is rendering music and an animated visual from an example algorithm.

Minecraft [17], the player also takes on the role of the creator, creatively shaping the environment and the game experience. Here, players build virtual worlds by placing static or interactive elements as building blocks that shape the entire environment, referred to here as the concept of worldbuilding.

Thus, especially in the medium of virtual environments, the long practice of video games has led to a familiar blending of the recipient and the performing actor. The surrounding world is thereby both, the material as well as the stage for the creation of these experiences. When it comes to live performances of audio/video content, there are a growing number of musical live performances in multiplayer games like Fortnite or metaverse environments such as VRChat or Mozilla Hubs [18]. In its most basic application, these are concerts with 2D audio and video live streams on such platforms [19,20]. The experience, with the audience viewing a large virtual screen, is more akin to a public screening than an actual live performance. More complex approaches allow artists to perform as virtual avatars, partly based on motion capture [21,22]. High-quality, pre-recorded live performances with 3D sound and video are another form of virtual live performances [23].

This century has seen a rise in the concept of the composer-programmer, which manifests itself especially in the performance practice of live coding [24]. A practice in which the on-the-fly programming of algorithms for the generative composition of music and/or visuals is performed live in front of an audience [25,26]. As programming languages, these can also be embedded well in new technologies, such as web-based applications in the browser [27,28,29]. Therefore also into web-based metaverse envi-

ronments. Embedding live coding into such environments allows live performances to be executed within the platform without the need to stream audio and video from local computers. For VERSNIZ we implemented an appropriate programming language for audiovisual live coding in metaverse environments and present a performance practice that combines the advantages of metaverse environments with the gamification concept of worldbuilding.

### **3 Concept**

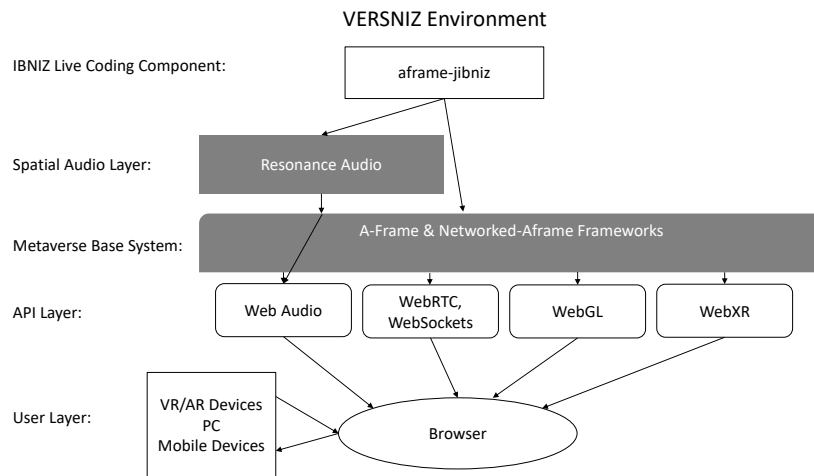
Our objective for VERSNIZ was to create a metaverse environment where multiple users can create a transforming virtual world as a collaborative performance or installation through audiovisual live coding. For this purpose, as in many worldbuilding video games, objects can be placed in virtual space by the user (see Fig. 1). These objects have a programming terminal and can be live-coded in the IBNIZ programming language. The algorithms simultaneously create the visual form, the animation, and the music. The placement and movement of the live coding objects, as well as the design of the acoustics of the virtual environment, allow for the additional application of spatial composition techniques [30].

Through movement in six degrees of freedom (6-DoF), the individual selection of what is seen (field-of-view) and heard is influenced by the head orientation and the position in space. Different visual details and a different "mix" of auditory components have a significant and individual influence on the perceived art experience.

Various criteria were considered during the implementation of a suitable system to meet the definition of a metaverse environment: along with the fundamental requirement of being a multiuser virtual platform, particular focus was put on the resulting immersion. In addition to the 3D rendering of the visual environment, special attention was also paid to the spatial audio rendering. Here, three-dimensional 6-DoF audio reproduction is realized with binaural synthesis for headphone-based auralization [31]. An adequate auditive room simulation was additionally implemented. Compliance with the WebXR [32] standard ensures the use of common VR/AR end devices. While compatibility with VR and AR devices using head-mounted displays (HMDs), controllers, hand or room tracking enables immersive experiences, the ability to use a conventional computer or mobile device ensures a low barrier to entry for the widest range of users. As a web-based application, users do not need to perform platform-specific installations, and the environment is automatically networked as the metaverse concept intends. Integrating audio and video streaming or text-based chats enables additional forms of interaction and enhances the social component of the experience. In addition, the possibility of modifying the virtual environment allows new types of stage and design concepts that can be realized with low effort compared to physical reality, and would otherwise be difficult or impossible to realize.

### **4 Implementation**

Implementing a system with the mentioned features requires a complex interchange of various programming languages, frameworks, libraries, and interfaces (see Fig. 2).



**Fig. 2.** The architecture of the VERSNIZ metaverse environment. The graphic shows the involved frameworks, libraries, and programming interfaces.

To comply with the described requirements for a metaverse environment, a selection of suitable web technologies was made. Thus, the virtual environment was developed based on the A-Frame framework [33]. Together with the library Networked-Aframe [34], shared, multiuser virtual environments can be implemented. This combination is well established, being the basis of the popular Mozilla Hub metaverse systems. The binaural rendering of A-Frame was extended with an implementation of the Resonance Audio spatializer [35]. The most critical component for the audio-visual composition is the programming language for live coding. The programming language IBNIZ unites various features that are particularly beneficial for programming in VR or AR and enables simultaneous programming of audiovisual algorithms. The following describes the development of these elements in more detail.

**Metaverse Base System** The web-based multiuser virtual environment was mainly realized with the A-Frame framework. A-Frame is a framework developed in JavaScript that enables the programming of virtual environments for VR and AR abstracted in HTML-like syntax. It is based on the JavaScript library Three.js [36] for programming WebGL applications. The HTML-like abstraction allows it to start designing 3D virtual worlds easily, while the access to Three.js makes it still very powerful. Embedding the WebXR application programming interface (API) enables integration with common VR/AR hardware and mobile devices - with stereoscopic rendering on HMDs and interaction via controllers and tracking. A-Frame applications can still be used with conventional computer hardware via screen, mouse, and keyboard. This allows most users to experience it without requiring special hardware. A-Frame provides various so-called "components" that can be used to program 3D geometries and models, mate-

rials, lights, shadows, and multimedia content such as images, videos, and sounds into a virtual environment. The framework is also arbitrarily extendable by programming custom A-Frame components in JavaScript.

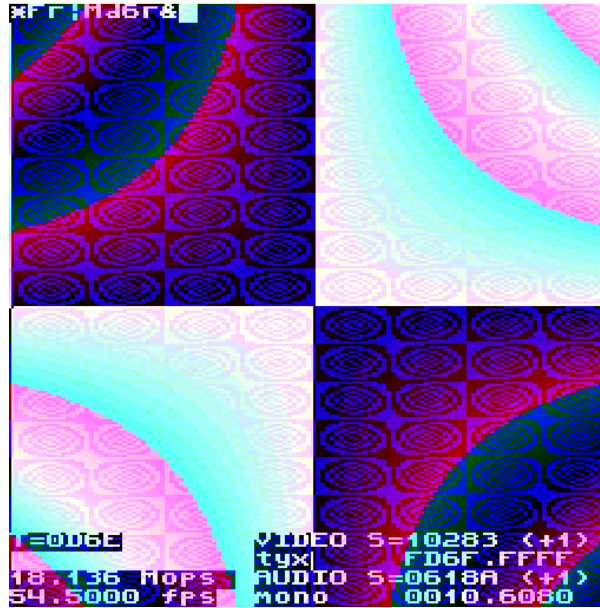
The Networked-Aframe library builds on A-Frame and enables the programming of multiuser environments for VR and AR. Networked-Aframe offers various adapters for data exchange via WebRTC with a WebSockets fallback. The data is transferred from user to user in a so-called peer-to-peer (P2P) network. The integration of the WebRTC standard [37] enables the low-latency transmission of audio and video streams, so that video and audio chats can be realized. Text can also be transmitted in-between users, as can the parameters of all A-Frame components, including custom-developed ones. In addition to sharing data, Networked-Aframe also provides templates for implementing avatars and synchronized interaction with the environment to ensure interactivity and persistence of the virtual environment.

For VERSNIZ, Networked-Aframe was integrated using the default EasyRTC adapters. Besides the synchronization of avatars and user interaction, the IBNIZ source code of the audiovisual live coding objects is shared. The virtual world unfolds its immersive potential when immersive end devices are used [38]. Stereoscopic rendering on HMDs and tracking head and movement in the room, create the feeling of presence in virtual environments. Not only is the image rendered to match the user's perspective, but so is the sound. A-Frame, in combination with the Web Audio API [39], already creates a dynamic three-dimensional sound experience through binaural rendering for headphone playback using real-time convolution with head-related transfer functions [31]. Sound sources are rendered at the appropriate location depending on the head orientation and the user's position in the room. To increase acoustic plausibility, a spatial room simulation for reverberation that considers the material properties of reflective surfaces, was added. This makes it possible to match the visually designed environment acoustically and to increase audiovisual coherence [40]. As there is already an A-Frame port [41] of the Resonance Audio spatializer, and Resonance Audio can be considered an appropriate choice for web-based applications [42], the existing port was extended and implemented in conjunction with the IBNIZ live coding component [43]. Based on these technologies, any desired immersive world can be created to represent the stage in VERSNIZ.

**IBNIZ Live Coding Component** IBNIZ, "Ideally Bare Numeric Impression giZmo", is a virtual machine for low-level programming of audiovisual algorithms [16] which is closely related to the Bytebeat concept [44]. It was developed by Ville-Matias "Viznut" Heikkilä and is linked to ideas present in the Demoscene. This is reflected in the minimalistic design of the language, resulting in the reduced instruction set, consisting of only one character per instruction.

Through this minimalist approach, IBNIZ is often considered an esoteric programming language [45]. A kind of programming language developed out of the motivation to implement experimental, weird, or sometimes artistic concepts rather than to pursue a practical use. Here many results can be already considered software art themselves.

The language design of IBNIZ also has artistic characteristics in it, and it is at the same time a domain-specific language [46] for the programming of further 2D video and au-

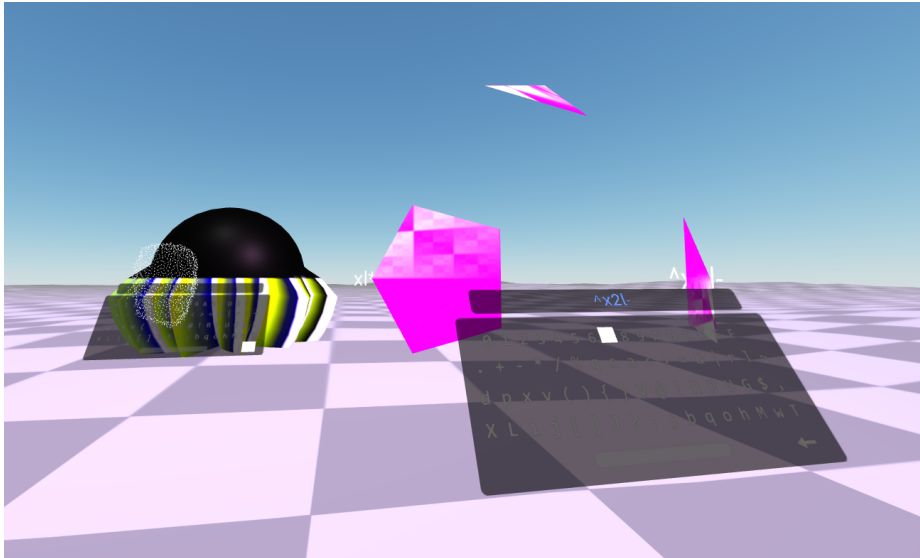


**Fig. 3.** A screen capture from the original IBNIZ code editor. It shows an example of the same algorithm from Fig. 1.

dio art (see Fig. 3).

The language's minimalism is not only ideal for live coding in real-time performances, allowing the artist to program expressive algorithms quickly and with few characters only, but also for typing with virtual keyboards on VR/AR devices. While the use of virtual keyboards in VR is often limited to short texts due to the difficulty of using them with the available input devices [47], the advantage of IBNIZ as the chosen language is particularly evident here. The entire instruction set can be placed on a single virtual keyboard view to code the desired algorithms quickly, even with controllers, the mouse, or hand tracking. Building on a web-based port 'jibniz' [48] in JavaScript, we realized an implementation of IBNIZ as an audiovisual live coding language for meta-verse environments. The IBNIZ virtual machine is implemented as an A-Frame and Networked-Aframe compatible component 'aframe-jibniz' [49] using the Web Audio API. The resulting audio is linked to a Resonance Audio source and can be rendered spatially depending on its position in relation to the listener. The visual output serves as a displacement texture for arbitrary 3D geometries; in this way, using algorithms, an animated, constantly changing object is generated. In combination, three-dimensional audiovisual objects are created that continuously evolve in real-time. Each object can be programmed independently with a programming terminal and a virtual keyboard, implemented using the Aframe-Super-Keyboard component [50]. The written code and the objects' position are also synchronized P2P using Networked-Aframe with all other users. Since the rendering is client-based for each user, neither audio nor video needs

to be streamed, only the text of the programmed code. This way, low bandwidth is used to transmit only text, enabling low-latency, real-time performances.



**Fig. 4.** A screen capture of a collaborative live coding scenario in VERSNIZ. Two people represented as a head point-cloud avatar performing together in the default environment setup.

## 5 Use Cases

With VERSNIZ we provide an exemplary template implementation of the 'aframe-jibniz' live coding component into an A-Frame/Networked-Aframe metaverse base system. Using this template, users can create arbitrary virtual worlds and art experiences. Within the possibilities offered by A-Frame, artists can freely design virtual environments, also specifically for use with VR or AR systems. It is intended for performances or installations using the proposed worldbuilding concept (see Sec. 3): the placement and live coding of multiple IBNIZ audiovisual objects in this environment, to enable the algorithmic composition of a constantly changing virtual world. The composition can then take place in real-time within the virtual experience, rather than being produced in advance. The resulting music and the 3D visualization of the objects change constantly depending on the algorithm. The location of their placement adds a spatial component to the composition and allows for extensive integration of spatial composition techniques. Placement and live coding can be done by any user or restricted to specific performers, allowing for different levels of audience engagement. The audience can not only simultaneously take on the role of the performer by live coding themselves, but they can also have their individual experience as passive spectators by interacting with the



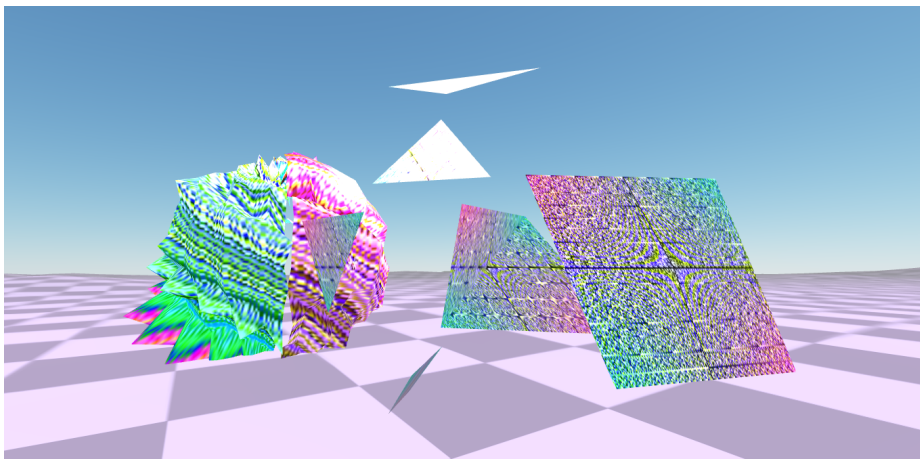
environment, for example by freely roaming around the world of various audiovisual objects. The recipient is free to move around the virtual environment at any time; the artist has no control over the time and place where the objects are experienced. This adds a spatial component to a time-continuous performance whose control is entirely in the hands of the recipient.

The above-mentioned properties allow a performance practice specifically tailored to the possibilities of metaverse environments, where all actors (performers and audience) come together in an immersive virtual world. The spatial aspect is important here; multiple IBNIZ live coding objects allow an audiovisual composition to be placed as individual fragments in space. This concept, inspired by worldbuilding videogames, allows different composition and performance concepts in combination with audiovisual live coding:

1. virtual worlds can be used as a shared playground for free-form audiovisual creation by different users
2. as a multiuser environment, it can be used for networked music performances in the form of collaborative improvisations or rehearsed compositions (see Fig. 4)
3. artists can create immersive, persistent audiovisual installations in virtual worlds (see Fig. 5)

With VERSNIZ, a new dimension is added to live coding as a performance practice, while the medium of virtual environments offers a high degree of freedom in designing artistic experiences. The gamification concept of worldbuilding allows for a new way of spatially distributed collaboration. A demo of the default VERSNIZ template implementation can be found at: <https://versniz.glitch.me/>

A video with a brief demonstration of the concept and mechanics is available at: <https://youtu.be/O4TmE1-bth4>



**Fig. 5.** A screen capture of multiple audiovisual objects composed into a sculpture, as an example for an audiovisual installation.

## 6 Conclusion & Future Work

With VERSNIZ we have created a metaverse environment for networked, collaborative live coding of audiovisual worlds. It allows for various novel performance and composition concepts characterized by their spatial aspects. This makes it possible to create art experiences that would be difficult or impossible to realize in physical reality. As an open-source environment, it allows artists a high degree of freedom in design, customization, and expansion. With the incorporation of the worldbuilding concept inspired by video games, we have described a specific performance practice that particularly benefits from the advantages of metaverse environments. The primary constraint is the programming language IBNIZ. While its minimalist design has significant advantages for programming in VR/AR, it is limited in stylistic variety. The properties of the IBNIZ virtual machine in terms of visual and audio resolution or the lack of external media integration, such as images and audio samples, limit the results to a lo-fi 8-bit-like aesthetic.

The current implementation of VERSNIZ works best for Chromium-based browsers and can be found at: <https://github.com/AudioGroupCologne/VERSNIZ>. However, it is still under continuous development to increase compatibility with browsers and end devices, improve the user experience, provide additional features for artists, and improve stability and performance. Referring to the limitations of IBNIZ mentioned above, also other systems for performances in metaverse environments [51] are being developed parallel to VERSNIZ. This includes additional programming languages for live coding, systems for programming and performing with virtual instruments, and performances using real-time streaming of volumetric audio and video [52].

## References

1. Neal, S.: Snow Crash. New York: Bantam Books (1992)
2. Mystakidis, S.: Metaverse. *Encyclopedia* 2(1), 486–497 (2022)
3. Park, S.M., Kim, Y.G.: A Metaverse: Taxonomy, Components, Applications, and Open Challenges. *IEEE Access* 10, 4209–4251 (2022)
4. Lee, L., Lin, Z., Hu, R., Gong, Z., Kumar, A., Li, T., Li, S., Hui, P.: When creators meet the metaverse: A survey on computational arts. *CoRR abs/2111.13486* (2021), <https://arxiv.org/abs/2111.13486>
5. Turchet, L.: Musical Metaverse: vision, opportunities, and challenges. *Personal and Ubiquitous Computing* (2023)
6. Elen, R.: Music in the metaverse. *Journal of the Audio Engineering Society* (April 2008)
7. Oliveros, P., Weaver, S., Dresser, M., Pitcher, J., Braasch, J., Chafe, C.: Telematic music: Six perspectives. *Leonardo Music Journal* 19, 95–96 (2009)
8. Rottondi, C., Chafe, C., Allocchio, C., Sarti, A.: An overview on networked music performance technologies. *IEEE Access* 4, 8823–8843 (2016)
9. Turchet, L., Hamilton, R., Camci, A.: Music in Extended Realities. *IEEE Access* 9, 15810–15832 (2021)
10. Loveridge, B.: Networked Music Performance in Virtual Reality: Current Perspectives. *Journal of Network Music and Arts* 2(1), 2 (2020)
11. Men, L., Bryan-Kinns, N.: Supporting Sonic Interaction in Creative, Shared Virtual Environments, pp. 237–267. Springer International Publishing, Cham (2023), [https://doi.org/10.1007/978-3-031-04021-4\\_8](https://doi.org/10.1007/978-3-031-04021-4_8)

12. Wakefield, G., Ji, H.: Artificial nature: Immersive world making. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 5484 LNCS, pp. 597–602 (2009)
13. Wakefield, G., Smith, W.: Cosm : a Toolkit for Composing Immersive Audio-Visual Worlds of Agency and Autonomy. In: Proceedings of the International Computer Music Conference (2011)
14. Hamilton, R.: Collaborative and competitive futures for virtual reality music and sound. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 1510–1512 (2019)
15. Ciciliani, M.: Virtual 3D environments as composition and performance spaces\*. Journal of New Music Research 49(1), 104–113 (jan 2020)
16. V. Heikkilä, IBNIZ. <http://viznut.fi/ibniz/>, accessed: 2023-02-15
17. Microsoft: Minecraft (2023), <https://www.minecraft.net/>, accessed: 2023-02-15
18. Foundation, M.: Mozilla Hubs (2023), <https://hubs.mozilla.com/>, accessed: 2023-02-15
19. Toplap Berlin: Mozilla Hubs Concertspace (2023), <https://hubs.mozilla.com/oUQRigk/toplap-berlin/>
20. Champlin, A., Knotts, S., Chicau, J., Xambó, A., Saladino, I.: Community Report : Live-coderA. In: 7th International Conference on Live Coding (ICLC). pp. 1–5 (2023)
21. T. Scott, Epic Games: Travis Scott and Fortnite Present: Astronomical (Full Event Video) (2023), <https://www.youtube.com/watch?v=wYeFAlVC8qU>
22. VRROOM: Jean Michel Jarre - ZERO GRAVITY - IN VR - Welcome to the Other Side (2023), <https://vimeo.com/547145530>, accessed: 2023-02-15
23. VRROOM: Jean Michel Jarre, Oxymore 2 - Overview (2023), <https://vimeo.com/767098450/6fcf797c5a>, accessed: 2023-02-15
24. Nilson, C.: Live coding practice. In: Proceedings of the 7th International Conference on New Interfaces for Musical Expression. p. 112–117. NIME '07, Association for Computing Machinery, New York, NY, USA (2007), <https://doi.org/10.1145/1279740.1279760>
25. Collins, N., McLean, A., Rohrhuber, J., Ward, A.: Live coding in laptop performance. Organised Sound 8(3), 321–330 (2003)
26. Juan Romero, Borgeat, P.: Live-Coding – programming masterly music. In: TEDx KIT (2018), [https://www.ted.com/talks/juan\\_romero\\_patrick\\_borgeat\\_live\\_coding\\_programming\\_masterly\\_music\\_jan\\_2018](https://www.ted.com/talks/juan_romero_patrick_borgeat_live_coding_programming_masterly_music_jan_2018), accessed: 2023-02-15
27. Roberts, C., Kuchera-Morin, J.A.: Gibber: Live coding audio in the browser. ICMC 2012: Non-Cochlear Sound - Proceedings of the International Computer Music Conference 2012 pp. 64–69 (2012)
28. Ogborn, D., Beverley, J., Navarro Del Angel, L., Tsabary, E., McLean, A.: Estuary: Browser-based Collaborative Projectional Live Coding of Musical Patterns. In: International Conference on Live Coding (2017)
29. Lan, Q., Jensenius, A.R.: QuaverSeries : A Live Coding Environment for Music Performance Using Web Technologies. In: Web Audio Conference WAC-2019, (2019)
30. Baalman, M.A.: Spatial composition techniques and sound spatialisation technologies. Organised Sound 15(3), 209–218 (2010)
31. Møller, H.: Fundamentals of binaural technology. Applied Acoustics 36(3-4), 171–218 (1992)
32. W3C: WebXR (2023), <https://immersiveweb.dev/>, accessed: 2023-02-15
33. A-Frame: Homepage (2023), <https://aframe.io>, accessed: 2023-02-15
34. Networked-Aframe: Git Repository (2023), <https://github.com/networked-aframe>, accessed: 2023-02-15

35. Resonance Audio: Git Homepage (2023), <https://resonance-audio.github.io/resonance-audio/>, accessed: 2023-02-15
36. Three.js: Homepage (2023), <https://threejs.org/>, accessed: 2023-02-15
37. Developer, G.: WebRTC (2023), <https://webrtc.org/>, accessed: 2023-02-15
38. Slater, M., Lotto, B., Arnold, M.M., Sanchez-Vives, M.V.: How we experience immersive virtual environments: The concept of presence and its measurement. *Anuario de Psicología* 40(2), 193–210 (2009)
39. W3C: Web Audio (2023), <https://www.w3.org/TR/webaudio/>, accessed: 2023-02-15
40. Werner, S., Klein, F., Mayenfels, T., Brandenburg, K.: A summary on acoustic room divergence and its effect on externalization of auditory events. In: 2016 8th International Conference on Quality of Multimedia Experience, QoMEX 2016. pp. 1–6. IEEE (2016)
41. Kungla, M.: A-Frame Resonance Audio Git Repository (2023), <https://github.com/mkungla/aframe-resonance-audio-component/>
42. McArthur, A., Tonder, C.V., Gaston-Bird, L., Knight-Hill, A.: A survey of 3D audio through the browser: Practitioner perspectives. In: 2021 Immersive and 3D Audio: From Architecture to Automotive, I3DA 2021. Institute of Electrical and Electronics Engineers Inc. (2021)
43. Dziwis, D.: A-Frame Resonance Audio Git Repository (2023), <https://github.com/AudioGroupCologne/aframe-resonance-audio-component>
44. Heikkilä, V.M.: Discovering novel computer music techniques by exploring the space of short computer programs pp. 1–8 (2011), <http://arxiv.org/abs/1112.1368>
45. Temkin, D.: Language without code: Intentionally unusable, uncomputable, or conceptual programming languages. *Journal of Science and Technology of the Arts* 9(3 Special Issue), 83–91 (2017)
46. Voelter, M., Benz, S., Dietrich, C., Engelmann, B., Helander, M., Kats, L., Visser, E., Wachsmuth, G.: DSL Engineering - Designing, Implementing and Using Domain-Specific Languages (2013)
47. Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., Kristensson, P.O.: Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In: 25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018. pp. 159–166. IEEE (2018)
48. L. Escot, jibniz. <https://github.com/flupe/jibniz>, accessed: 2023-02-15
49. Dziwis, D.: aframe-jibniz. <https://github.com/AudioGroupCologne/aframe-jibniz>, accessed: 2023-02-06
50. Supermedium: Aframe-Super-Keyboard (2023), <https://github.com/supermedium/aframe-super-keyboard>, accessed: 2023-02-15
51. Dziwis, D., von Coler, H., Pörschmann, C.: Orchestra: a Toolbox for Live Music Performances in a Web-Based Metaverse. *Journal of the Audio Engineering Society* pp. 1–11 (2023), (accepted for publication)
52. Dziwis, D., von Coler, H.: The Entanglement – Volumetric Music Performances in a Virtual Metaverse Environment. *Journal of Network Music and Arts* 5(1), 1–12 (2023)

## **Spatial Sampling in Mixed Reality An Overview of Ten Years of Research and Creation**

Dr. Grégory Beller<sup>1</sup> and Pr. Dr. Jacob Sello<sup>1</sup> and  
Pr. Dr. Georg Hajdu<sup>1</sup> and Pr. Thomas Görne<sup>2</sup>

<sup>1</sup> HfMT Hamburg, Ligeti Zentrum, Hamburg, Germany

<sup>2</sup> HAW Hamburg, Ligeti Zentrum, Hamburg, Germany  
contact@gregbeller.com  
[0009-0007-3553-4650]

**Abstract.** Spatial Sampler XR is a new musical instrument linking gesture capture to sound production. In the same way that a sampler is an empty keyboard filled with sounds, Spatial Sampler XR uses gesture capture to transform the surrounding physical space into an area of keys for, recording, indexing and playing back samples. Spatial Sampler XR let the musician arrange the sound around him or her through gesture, creating a spatialized and interactive soundstage. A virtual reality headset adds to the instrument the ability to visualize the layout of sounds. The 3D immersion greatly facilitates their organization and increases the precision of the interaction. Several modes of play are possible and the interaction modalities vary according to the type of performance and the number of performers. This article first introduces the Synekine project, a 10-year research project from which the concept of spatial sampling is derived. It presents the technical devices used, the instruments created, the different modes of play in performative situations. The instrument relies on movement to link time and space. Thus, the Spatial Sampler XR is particularly suitable for movement artists as well as for extra-musical applications.

**Keywords:** Spatially Situated Media, Spatial Sampling, Gestural Interaction, New Musical Instrument, Gesture and Sound Processing, Interactive Performance, Movement Computing

### **1 Introduction**

For the past ten years, through residencies and artistic creations, *the Synekine Project* has invited performers to question the intimate relationship between vocal gestures and manual gestures, through the manipulation of scenic devices based on new technologies. Metaphorically, the preeminent neuromotor link between voice and gesture is closed by “creative prostheses” joining the capture of movement to the transformation of the voice by artificial intelligence. Linking space and time through movement then transforms the search for sound into a scenic exploration.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

This paper presents the Synekine Project according to several dimensions. After the exhibition of the scientific basis and the genesis of the project, the new organology developed is described in chronological relation with the artistic works produced. Although it generates devices that can be used in other fields of application such as therapeutic rehabilitation, commercial signage or even dance and music education, these corollary results are not presented in this article for the sake of clarity of presentation.

Researches on the augmented voice in the theater [Beller11e, Beller12a, Beller14c, Beller17a] led to the instinctive, expressive and pleonastic relationship between vocality and gestural. The faculty of speaking with the hands would not only result from cultural origins, but would also result from a deep neuronal relation connecting the speech to the gestures of the hands [Iverson98]. By analogy to synesthesia, a phenomenon by which two or more senses of perception are associated, “synekinesia” would reflect our ability to associate two or more motor senses.

There is ample evidence for the ubiquitous link between manual motor and speech systems, in infant development, in deictic pointing, and in repetitive tapping and speaking tasks [Parrell14]. So, it is obvious to most people to clap their hands with every syllable spoken when we speak. Metaphorically, the Synekine Project proposes to “complete” this neuronal loop by artificially establishing an “external” link between vocal gesture and manual gesture. The voice is conventionally captured by a microphone, while the positions and dynamics of the hands are informed by different movement capture processes, which have changed from one residence to another (glove-accelerometers, Genki Wave, LeapMotion, Kinect, Optitrack and Metaquest2, see table 1.). Between the two, different computer programs based on artificial intelligence, allow the direct manipulation of sound by gesture or the learning of temporal relationships between vocal and gestural. In both cases, the result is a new organology made up of different intangible musical instruments offering the user to spread her.his voice in space, to multiply it, to segment it or to manipulate it.

Taking advantage of the instinctive nature of the voice - gesture link, and taking advantage of new technologies allowing the joint capture of the sound of the voice and the coordinates of the gesture in real time, the Synekine Project has deployed across artistic productions and residencies, a new organology of sound, invisible and intangible instruments belonging to new Human-Computer Interactions. Motivated by artistic practice, the development of this organology is presented here in a genealogical form mixing innovations and creations.

**Table 1.** Chronology of the works realized within the framework of the Synekine project presenting the various instruments elaborated from various technical devices.

Year	Work	Instrument	Gesture Sensor	Hand Button	Transmission	Hand tracking	Sound Recording	Visualisation
2011	Luna Park	SpokHands	XBEE V1	X	Wifi	X	Wireless headset	X
2013	Babil-on	Body Choir	XBEE V1	XBEE V1	Wifi	Kinect V1	Wireless headset	activity only
2014	TIIME	Hand Sampling, Hyper Ball, Wired Gestures, Gesture Scape, Body Choir, Sound Space	XBEE V2	XBEE V2	Wifi	Kinect V2	Wireless headset	video for gesture scape
2015	Babil-on V2	Hand Sampling, Hyper Ball, Body Choir, Sound Space	XBEE V2	XBEE V2	Wifi	Kinect V2	Wireless headset	X
2017	TEDx Paris	Body Choir	X	X	X	Kinect V2	Wireless headset	X
2016	The Memory Palace - Installation	Sound Space	X	Kinect Gesture Recognition	X	Kinect V2	Ambient mic	Projected 3D space
2016	The Memory Palace - Performance	Sound Space	XBEE V2	XBEE V2	Wifi	Kinect V2	Wireless headset	X
2017	Fissures	Hand Sampling, Body Choir, Sound Space	XBEE V2	XBEE V2	Wifi	Kinect V2	Wireless headset	X
2020	Birth of a Tree	Spatial trigger, Spatial looper, Sound Space	Genki Wave	Genki Wave	BLE	Kinect V2	Wireless headset	X
2020	Air Sampling #001	Spatial trigger, Spatial looper, Sound Space	Genki Wave	Genki Wave	BLE	Kinect V2	Static mics	X
2021	Residency UME	Spatial trigger, Spatial looper, Sound Space	OptiTrack	Genki Wave	BLE	OptiTrack	Wireless headset	X
2022	Air Sampling #002	Spatial Sampler VR	Metaquest 2	Metaquest 2	Airlink	Metaquest 2	Static mics	VR + video
2022	The Vanishing Mirror	Spatial Sampler VR	Metaquest 2	Metaquest 2	Airlink	Metaquest 2	Static mics	VR only
2023	The Fault	Spatial Sampler VR	Metaquest 2	Metaquest 2	Airlink	Metaquest 2	Static mics	VR only
2023	Air Sampling #003	Spatial Sampler VR	Metaquest 2	Metaquest 2	Airlink	Metaquest 2	Wireless headset	VR + video
2023	Macht macht macht							
2024	TBA	Spatial Sampler XR	Hololens2	Hololens2	Wifi	Hololens2	Wireless headset	XR + video



**Fig. 1.** Different hardware used to realize the instruments: Top left an XBEE sensor, bottom left a Genki Wave ring, top right a Kinect V2, bottom right two Optitrack cameras.

## 2 Genesis: Expressivity, Sensors and Luna Park

On generative models of expressivity and their applications for speech and music, an artificial intelligence algorithm based on a corpus of expressive sentences, has been used to generate an “emotional” speech, by modulating the prosody of a “neutral” utterance [Beller09a, Beller09b, Beller10]. During the development of this synthesizer of the emotion in the voice came the desire to control the prosody by the gesture. At the same time, instrumental gesture sensors were developed allowing the measurement of the dynamics of a bow by integrating small accelerometers and gyroscopes. The data related to movement is transmitted in real time by WIFI to a computer which triggers sounds and modulates effects according to the dynamics of the gesture [Bevilacqua06]. In 2010, as part of the creation of *Luna Park*<sup>1</sup>, a musical theater work by Georges Aperghis, these sensors have been integrated into gloves and a first instrument called *SpokHands* has been developed, which literally made it possible to speak with the hands [Beller11a, b, c, d].

*SpokHands*<sup>2</sup> allows the triggering and modulation of voice samples by aerial percussion and hand elevation. Like a vocal Theremin, the instrument offers the performer the option of three-voice polyphony (her/his own and both hands) or control of text-to-speech parameters.

In this case, the natural division of a conductor's brain is used, the left brain (right hand) for the segmental part, and the right brain (left hand) for expressivity. The percussive gestures of the right-hand trigger pre-selected syllables whose pitch and intensity are modulated by continuous gestures of the left hand. A particular aerial percussion technique aimed at triggering sounds in a temporally precise manner without hurting the self has been practiced by a percussionist. Indeed, the absence of haptic feedback from a physical object could cause, in the long run, pain in the handles which acted as a stop to the percussive movement. The research work on gestural control of speech synthesis has been pursued as part of an artistic research residency at IRCAM entitled *The Synekine Project* [Beller14a].

## 3 *Babil-on*: From speech to time

*Babil-on*, for solo and electronic voice is an augmented musical theatre performance. The composition benefited from IRCAM's artistic research residency program and the piece was premiered by Richard Dubelski in Marseille, at the Théâtre des Bernardines, in 2013, as part of CMMR 2013. Like a close-up, a “Speech” character discovers his own voice, cuts it up, superimposes it, spreads it around him, multiplies it, and reveals the emotional charge intrinsic to the language. A pair of button-rings have been added to the sensor-gloves allowing for the picking and erasing of voice samples on the fly.

---

<sup>1</sup><http://www.gregbeller.com/2011/06/luna-park/>

<sup>2</sup><http://www.gregbeller.com/2011/06/spokhands/>



Thus, *SpokHands* and the triggering of pre-made sounds evolved into *Hand Sampling*, in which the vocal flow is cut and recombined in percussive gestures.

*Hand Sampling*<sup>3</sup> allows the performer to cut her/his voice in real time, and recombine immediately by the gesture. It involves percussive gestures that will segment and trigger vocal fragments.

The length of these fragments can vary from syllable to sentence. The order of the re-played segments can be sequential, random or palindromic, which allows different playing modes. In addition, the quality of the gesture influences the quality of the sound perceived, making the instrument expressive.

To the fast capture of the dynamics of the gesture by the accelerometer gloves has been added the relatively slow capture of the absolute position of the hands in space, by the use of depth cameras of the Kinect type [Kean2011]. This made it possible to obtain, in addition to the fine temporal precision of the percussive type triggering, the continuous control of sound processes according to the posture and the spatial position of the hands. On the other hand, the sensor brought other constraints such as a reduction in the playing area, a single performer possible, the need for a phase of calibration and detection of the skeleton, the risk of infrared disturbance by lights. The *Body Choir* uses the hand position to control a choir effect and the *Hyper Ball* to control a granular synthesizer.

*Body Choir* transforms a singer into a choir. This virtual choir accompanies the singer according to her/his gestures and the postures s/he adopts.

Singing involves movement of the body. This movement is captured and used to magnify the singer's musical intentions. The posture of the body and the sung note modulate in real time the harmony, the number of voices, or the spatial density of the choir.

*Hyper Ball* takes the form of a virtual sound ball, which the participant waters with her/his voice and modulates with her/his gesture.

The position, size and orientation of the ball influence the height, density and volume of the sound generated. This type of musical activity, by its constitution, causes choreographic movements.

In 2016 in Vancouver, Simon Fraser University, during ISEA2016, a new version Babil-on V2 has been premiered. The Kinect V2 replaced the V1 offering better acuity in capturing movement, greater flexibility of use and the possibility of following the hands of several people at the same time. Another pair of button-rings have been added to the sensor gloves.

From a compositional point of view, this second pair of button-rings offers free navigation in the structure of a work whose duration of each scene is flexible according to performer's own perception of time. From there, from the table to the stage, a change of writing paradigm takes place and we evolve in an open form that can break with the linearity of the pre-defined musical structure. Now, an improvised form can emerge from

---

<sup>3</sup><http://www.gregbeller.com/2014/02/hand-sampling/>

the dynamic choices made by the performers in a situation of improvisation with these instruments.

#### **4 *TIIME*: From time to memory:**

*TIIME* stages three performers who play gestural, sound, visual and temporal mirror effects, in an apparent collective improvisation of which emerge from temporal themes: perception, memory, movement. The temporal relationship between vocal and manual gestures is modelled by artificial intelligence. On stage, a performer feeds a machine with his own vocal and gestural catalogue, then *Wired Gestures* restores fragments of this vocality in a way that is synchronous with the recognition and tracking of new gestures.

*Wired Gestures* dynamically links voice to gesture, in an artificial way. The machine simultaneously records a voice gesture and a manual gesture [Françoise2014]. It learns the temporal relationship between the two. Then it reproduces the voice, when the performer repeats the same gesture.

The nuances of timing in the gesture are then heard as prosodic variations of the voice, and it becomes possible to break down the expressivity.

Visual extension, *Gesture Scape* records jointly and categorizes voice, gesture and video. Then new gestures, either manual or vocal will activate the visual and auditory archive. Video capture is introduced as a referential element of sound time. Not only does the manual gesture reproduce the sound of the voice, it can now also reproduce the image of the performer at the time of recording. *Gesture Scape* can be seen as the visual extension of *Wired Gestures*. The performer dances with her/his double, in a dialogue made of unison and counterpoint with the past, finding her/his inspiration in the lapsus of memory.

*Gesture Scape* jointly records voice, gesture and video. Then, new gestures will activate this memory. The performer animates the video, by reproducing the same gesture, or by repeating the same associated sound.

The performer dances with her/his double, in a dialogue made of unison and counterpoint, inspired by lapsus of memory. From the development of these two devices, based solely on the dynamics of the gesture and not on its location, was born the desire to be able to arrange and organize it in space. Symbolically, a wave of the hand, placed above the head, to say goodbye, differs from a refusal, however expressed by the same gesture, but located below the shoulder. From a performative point of view, the staging of learners manipulating learning machines has necessarily questioned the situation of memory.

## **5     *The Memory Palace: From memory to the process***

Three installations and a choreographic performance, all entitled *The Memory Palace*, were created during the FACTS - Bordeaux and EXPERIMENTA - Grenoble festivals in 2017, with the support of IRCAM and ADAMI. This work cycle benefited from an artistic research residency at IDEX - University of Bordeaux, in scientific companionship with the LaBRI.

The Memory Palace is a mnemonic device practiced since antiquity allowing for memorizing long lists by arranging the elements of these in imaginary places [Yates66]. The construction of an interior architecture sensitive, has been realized with the *Sound Space*, a choreographic musical instrument linking space and time through movement. The surrounding space becomes a key zone in which the voice can be deposited and awakened by the gesture.

*Sound Space is a choreographic musical instrument which links space and time, through movement. It transforms the physical space surrounding the performer, into a zone in which s.he can place her.his voice and awaken it by gesture.*

By drawing her.his voice, s.he creates a unique soundstage, while evolving within it, in a creative process. The space then vibrates with a sound quality in line with the quality of movement. The *Sound Space* won the prize for technical excellence during the Guthman competition for new musical instruments.

Parable of the mnemonic, the installation transforms the place in which it is exhibited into a collective sound sculpture. Each participant is involved in the creation of a work of which s.he constitutes one of the many voices. Her.His gestures act as a reveler of the invisible sculpture. S.He can contribute to it by depositing elements of stories, sounds, or even songs that he can immediately recall in a creative process. A virtual but yet very audible forum, *The Memory Palace* acts as an indicator of the borders of the intimate, confronting the participant with the direct and immediate use of her.his voice imprint by others. In this mediation, the physical space, however empty, resonates with the different memories delivered by the participants in a temporal polyphony. Two sound installations based on the principle of *Sound Space* were presented.

The first uses a Kinect V2 for the localization and recognition of gestures, an ambient microphone as well as a video projection which materializes the sound traces of the participants on a screen. Two people were able to record sounds and play back the sounds of the others. The recording and erasing of sounds were controlled by gesture recognition (stone, leaf, scissors). The feedback showed that the main difficulty for the participants was to synchronize in order to avoid feedback (one playing while the other was recording). Finally, the representation on a 2D screen of the 3D space was a great help for the audience and made us want to use a 3D visual representation using holography or mixed reality.

The second uses the microphone and an ad hoc system of geo-localization in the confined space of mobile phones. This second version allows everyone to leave audio

messages for others by simply wandering through the space with their phone, after installing a small dedicated application.

Within the former installation, the dancer Valencia James delivers a musical choreography involving personal memories, ancestral traces and imaginary characters. The choreographic process consists of the progressive materialization of a memory palace by the deployment at different points of the stage of characters combining sound quality and quality of movement. Then, a free wandering generates by interpolation a new sound space which in turn provokes new states of the body. Everything happens as if the dancer were making an improvisation with herself and the traces of memory that she has just placed on the stage.

This dramaturgical structure in three stages (discovery of a new device; development of an ad hoc language; expression of “something else” with this language) has the capacity to attract an audience and to take it somewhere to finally surprise it there. If “the something else” refers to the discovery, the device or the elaboration of language, this linear structure resonates and loops in *mise en abyme* generating meaning. This is the case, for example, in the musical theater solo *Fissures*, in which fragments of a text on amnesia are arranged in a spatial and repetitive cut-up.

## **6 Birth of a Tree: From Process to Language**

As part of a *Scientiarum Musicae* doctorate in the framework of the KiSS program - Kinetics in Sound and Space - at the Hochschule für Musik und Theater Hamburg, Germany, new technical gesture capture devices have been gradually being tested and integrated, such as the Genki Wave accelerometer rings, the Optitrack Motion Capture system or the Meta Quest 2 virtual reality headset. Spatial Sampling paradigm has been developed [Beller15a]. The constantly renewed interest in the device *Sound Space* in different artistic configurations stems from its adaptability. Indeed, it is an empty and silent box at the start, just like the musical sampler. So other instruments were derived from the hybridization of these two concepts and tested in the creation of *Birth of a Tree*. They are the basis of the creative improvisation process of the *Air Sampling* series.

*Air Sampling* is a series of improvised performances in which a sound source is sampled and distributed in space in real time. In the first performance #001, the sound source is given by Lin Chen on percussion and vocals. The author, playing the *Sound Space*, *Spatial Trigger* and *Spatial Looper* instruments, records and plays the samples in space. They perform with the percussionist an improvised musical choreography of which percussions are the only acoustic source.

In the 1970s, the sampler revolutionized music production. This electronic music instrument, whose memory is empty at the beginning, allows the musician to create her/his own universe from percussion samples, ensembles, groups of instruments or orchestras and can also serve as a platform for musical creation.

The paradigm of the *Spatial Sampler* is to substitute the midi keyboard which classically indexes the samples, with spatial coordinates describing the positions of the hands.

*Spatial Trigger* allows the automatic segmentation of a sound and its distribution in space by the gesture, then the selection of one of these fragments according to its position and its triggering by aerial percussion.

The *Spatial Looper* allows the mixing of several sound loops according to the proximity of the hands, as well as the generation of music from the fragments distributed in space.

Many musicians are now familiar with the "looper" or live looping. One of the main difficulties in this art is dealing with multiple layers of samples (usually dealt with a guitar pedal). The *Spatial Looper* transforms the space surrounding the performer into a spatial sequencer and makes it easier to not only access the different layers but also to remember them.

## 7 *Air Sampling: From Language To 3D Sculpture*

For Air Sampling #002, still using the percussion sounds played by Lin Chen, VR versions of the Sound Space, Spatial Trigger and Spatial Looper instruments were created and tested in a performance situation. An OpenGL representation allows the performer to visualize the position of the recorded sounds in a Meta Quest 2 headset, which greatly facilitates the organization of the session and allows for greater spatio-temporal accuracy.

The graphical representation of the recorded sounds also offers the possibility to draw 3D structures in the virtual space. The other performers and the audience can see the structure emerging from the gestures through video projection. Several performance situations have been explored with or without sharing the representation of the structure with the audience through video projection. Making it visible facilitates the understanding of the instrument by the audience but unbalances the performance if other musicians contribute. The other musicians only show the scores and their interpretation choices in front of them through the sound produced. In Vanishing Mirror, the performer is the only one to visualize the recording of the sounds produced by a piano-cello-vocal trio. Different modes of interaction with the sound sources have been explored, from the duet with a percussionist in AirSampling #002, to the live sampling of an ensemble of seven instruments in The Fault (see Figure 2).

While Virtual Reality allows for a better organization of the sounds recorded and produced, it has the unfortunate side effect of visually cutting the performer off from the other musicians as well as the audience. Not only does it reduce the facial expression of the performer whose eyes are no longer visible, but the rupture of the visual contact of the performer with the others can generate complex situations for the synchronization in a improvisation situation. In AirSampling #003 - Macht macht, the strolling public interacts with the performer, who has a microphone. A protocol must then be established

between the participating visitor and the performer in order to synchronize the production of sound on the one hand with the gesture of the other.

The total obstruction of the visual contact in situation of improvisation can cause an interesting situation of performance, but can also harm the connivance between the public, the musicians and the performer made "blind". As holographic video is not yet mature enough, mixed reality seems to be the way to reduce the loss of visual contact. The evolution of the Spatial Sampler VR to the Spatial Sampler XR is currently being developed with the HoloLens 2 mixed reality technology. This evolution implies the substitution of controllers by gesture recognition, thus reducing the playability of the instrument by decreasing its response time. Apart from the Optitrack system which operates at 120 frames per second, the other systems based on video or depth sensors do not offer a small enough latency to allow aerial percussion (latency lower than 20ms). Just as we added acceleration sensors to the data from the Kinect, the fusion of gesture recognition data with data from on-board accelerometers (Xbee sensors or Genki Wave rings) is being considered to improve gameplay in mixed reality.



**Fig. 2.** The *Fault*, Opera composed by the g. Beller, 2023, Hamburg, Germany. In the background, a representation of a 3D sculpture elaborated from the sounds of the instrumentalists.

## 8 Conclusion

The nexialist approach of *the Synekine Project* aims to establish protocols that generate creative processes in a holistic approach mixing Arts and Sciences. The path oscillates between meetings with scientific researchers, technical development phases, experimentation residencies of new scenic devices and crystallization of performative situations in shows or installations.

This article relates 10 years of research and artistic production accompanied by technological development. In chronological form, it exposes the development of the research theme of spatial sampling, the new musical instruments elaborated through the evolution of the technologies of gesture capture as well as the artistic stakes approached in different works. From the link between gesture and voice, the research has evolved towards the manipulation of spatially situated media. From Hand Sampling or spatial sampler to sound space, different instruments make movement the link between time and space. The technical developments are based on the evolution of technologies for capturing gestures. The fusion of data from the capture of the dynamics of the gesture, the position of the hands in the space and the sound, allows the elaboration of an interactive sound scene. In performance, improvisation, comprovisation or composition situations, different modes of play are presented. The creation of about fifteen works with these instruments gives a context to the artistic stakes of spatial sampling.

Virtual reality allows to represent and manipulate "sounds located in space" in an environment comparable to that of a 3D painting software. The custom arrangement of media whose recording and playback is done on the fly inaugurates the possibility of "spatially situated media" editing. Compared to modern sequencers whose organization of windows and sounds is constrained by 2D, the possibility of freely organizing media content in 3D space seems to accelerate the work of the editor, who joins spatial memory to content memory. The use of the memory palace in the audio-visual editing activity can greatly facilitate the access to the contents and accelerate the work steps without requiring any particular cognitive dispositions of the user.

## **Acknowledgements**

The authors would like to thank the supervisors of the KiSS program, Prof. Dr. Georg Hajdu (HfMT Hamburg), Prof. Thomas Görne (HAW Hamburg), Prof. Dr. Jacob Sello (HfMT Hamburg) Nina Noeske (HfMT Hamburg), Prof. Sabina Dhein (HfMT Hamburg), Prof. Dr. Julius Heinicke (HfMT Hamburg), Dr. Rama Gottfried (HfMT Hamburg), as well as its coordinator Dr. Benjamin Helmer (HfMT Hamburg). This work is partially supported by the project "Stage\_2.0: Alsterphilharmonie. Die Bühne als Ort des künstlerischen Wissenstransfers und der gesellschaftlichen Teilhabe" and by the TransferBüro of the HfMT Hamburg. We would like to thank all those who have contributed to this artistic research.

## **References**

- [Beller09a] Beller, Grégory (2009). Analyse et Modèle Génératif de l'Expressivité: application à la parole et à l'interprétation musicale », Paris 6 – IRCAM
- [Beller09b] Beller, Grégory (2009). Transformation of Expressivity in Speech », The Role of Prosody in the Expression of Emotions in English and in French, ed. Peter Lang. (Peter Lang)

- [Beller10] Beller, Grégory (2010). *Expresso: Transformation of Expressivity in Speech*, Speech Prosody, Chicago
- [Beller11a] Beller, Grégory, Aperghis, Georges (2011). *Contrôle gestuel de la synthèse concaténative en temps réel dans Luna Park: rapport recherche*
- [Beller11b] Beller, Grégory, Aperghis, Georges (2011). *Gestural Control of Real-Time Concatenative Synthesis in Luna Park* », P3S, International Workshop on Performative Speech and Singing Synthesis, Vancouver, pp. 23-28
- [Beller11c] Beller, Grégory (2011). *Gestural Control Of Real Time Concatenative Synthesis*, ICPhS, Hong Kong
- [Beller11d] Beller, Grégory (2011). *Gestural Control of Real-Time Speech Synthesis in Luna Park*, SMC, Padova
- [Beller11e] Beller, Grégory (2011). *Arcane d'Un mage en été*, Théâtre Public, n° 200
- [Beller12a] Beller, Grégory (2012). *In-vivo: laboratoire de recherche et d'expérimentation autour du son pour le théâtre*, Towards a History of Sound in Theatre, Montreal
- [Beller14a] Beller, Grégory (2014). *The Synekine Project*», MOCO 2014, IRCAM, Paris
- [Kean2011] Kean, S., Hall, J.C., Perry, P. (2011). *Microsoft's Kinect SDK*. In: *Meet the Kinect*. Apress. [https://doi.org/10.1007/978-1-4302-3889-8\\_8](https://doi.org/10.1007/978-1-4302-3889-8_8)
- [Françoise2014] Jules Françoise, Norbert Schnell, Riccardo Borghesi, Frédéric Bevilacqua. *Probabilistic Models for Designing Motion and Sound Relationships*. *Proceedings of the 2014 International Conference on New Interfaces for Musical Expression*, Jun 2014, London, UK, United Kingdom. pp.287-292. (hal-01061335)
- [Beller14c], Beller, Grégory (2014). *L'IRCAM et la voix augmentée au théâtre: Les nouvelles technologies sonores au service de la dramaturgie*, L'Annuaire théâtral, Numéro 56-57, p. 195-205
- [Beller15a], Beller, Grégory (2015). *Sound Space and Spatial Sampler*, MOCO 2015, SFU, Vancouver
- [Beller17a], Beller, Grégory (2017). *Spectacle vivant: des voix imaginaires aux monstres vocaux*, InaGlobal, Paris, France
- [Bevilacqua06] Bevilacqua Frederic, Rasamimanana Nicolas, Fléty Emmanuel, Lemouton Serge, Baschet Florence (2006). *The augmented violin project: research, composition and performance report*. In *6th International Conference on New Interfaces for Musical Expression (NIME 06)*, Paris
- [Iverson98] Iverson Jana M. & Goldin-Meadow Susan (1998). *Why people gesture when they speak*, Nature 396, 228
- [Laukka13], Laukka, Petri, Eerola, Tuomas, Thingujam, Nutankumar S., Yamasaki, Teruo, Beller, Grégory (2013). *Universal and Culture-Specific Factors in the Recognition and Performance of Musical Affect Expressions*, Emotion, American Psychological Association, Vol 13(3), 434-449
- [Parrell14] Benjamin Parrell, Louis Goldstein, Sungbok Lee, Dani Byrd (2014). *Spatiotemporal coupling between speech and manual motor actions*. *Journal of Phonetics*, Volume 42, Pages 1-11



## **Networked performance as a space for collective creation and student engagement**

Hans Kretz

Department of Music, Stanford University  
kretz@stanford.edu

**Abstract.** This article takes a practice-based approach to exploring the specific issues and problems of distributed networked performance, in the light of the various aesthetic categories directly affected by this practice. It considers how traditional categories of aesthetics, such as the notion of presence, are called into question by the virtualisation of sonic space. Distributed performance also casts the notion of space in music in a new light. Another essential contribution of online practice is that it allows participants to decentre the question of auctoriality – or authorship – as it is anchored in a metaphysics of presence.

**Keywords:** #Networked performance #Aesthetics of distributed performance #Philosophy of technology.

### **1 Introduction**

This article takes as its starting point my experiences working with the Stanford New Ensemble (henceforth SNE) over the period 2020–21, as global circumstances pushed us to transition from in-person to online rehearsing and performing. This shift was accomplished thanks to the invaluable commitment and energy of the staff at CCRMA (the Center for Computer Research in Music and Acoustics at Stanford University, led by Chris Chafe), who made it possible for us performers to maintain our musical activities under the best possible online conditions. In the course of growing familiar with JackTrip – a software developed by Chris Chafe and Juan Pablo Caceres at CCRMA for high-quality, uncompressed audio in networked performance – the ensemble participants and I were able to develop our understanding of the particulars of online music-making, on both a musical and an aesthetic level. Questions of student engagement and of community-making in this particular context were also critical to us. The article presents some of the outcomes of this reflective thought as it emerged through and beyond my and our engagement in network performance.

I begin by discussing the necessity for an explicit aesthetic investigation into distributed performance, some of its particular problematics and stakes, and seek to show how this practice provides an opportunity to reassess and reevaluate some of the traditional questions of aesthetics. But the scope of networked performance extends beyond a reflection confined to the sphere of musical practice. Indeed, questioning the nature of the phenomena that occur in a virtual acoustic space calls for a rethinking of



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

certain assumptions in the light of the philosophy of mind. I explore in detail the implications for both thought and practice of the accentuation of musical space, a notion that in recent practice gains an unprecedented autonomy. I also consider how the knowledge gained through these contemplations can lead us to rethink the notion of the work of art as well as that of the musical object, and possibly the technical object – that is to say, the question of the modes of existence of works as well as of technical objects. All these elements culminate in the question of the work considered natively as an object that expresses itself online, in a plural way, namely as a collective creation that renews student commitment. I end by taking up again the notion of space, now understood as the public space of both music-making and philosophical discussion, and consider the potential for an ‘ethics of discussion’ seen through the lens of technological artefacts.

## **2 The aesthetics of networked performance**

The literature devoted to distributed performance has blossomed over the past twenty years. A significant number of these publications have, unsurprisingly, been primarily concerned with technical questions as to the ways in which real-time interactions could be achieved, starting with video telephone with images updating every five seconds in the 1990s and, with the more recent development of JackTrip, achieving near-zero or ultra-low latency. If publications from the early 2000s such as Craig Saper’s *Networked Art* and Anna Munster and Geert Lovink’s ‘Theses on Distributed Aesthetics’ began to engage with the challenge posed by distributed performance to the very nature of artworks, it nonetheless remains common for performers to consider online performance, however high the quality, as always a substitute: the ideal scenario is presumed always to be on-site presence shared with audience members.

Since 2019, the *Journal of Network Music and Arts*, a peer-reviewed open-access digital research journal published by Stony Brook University, has been offering a transdisciplinary approach to a diversity of questions raised by network performance, beyond the already existing literature of technological import. The questions addressed involve network arts technologies such as JackTrip, LoLa (a low-latency, high-quality audio/video transmission system for network musical performances and interaction), virtual reality, OBS Studio (OBS = Open Broadcaster Software) and other software. The journal also engages with thematic approaches, among which aesthetic issues have begun to find their place. For example, the latest issue focused on the notion of ‘distance’ and its transformations, whether ‘physical, emotional, societal, environmental [or] dimensional’.[1]

In addition to the innovations and technological developments in networked performance since the beginning of the pandemic, this performance practice has thus lent itself to the emergence of a vast corpus of research questions. This context calls for the progress of a philosophical and aesthetic inquiry that could potentially inform performers, researchers and technologists – a perspective I would compare to Hubert

L. Dreyfus' phenomenology-based approach to technology in his critique of Artificial Intelligence, which led to an enhanced dialogue between philosophers and AI engineers that culminated in technological developments founded precisely upon the results of this dialogue.

In particular, I believe that thinking about networked performance can lead to a re-framing of traditional questions in aesthetics. Such common notions as the concept of presence, from Plato to Derrida through Heidegger, need to be rearticulated or reinvestigated in the light of the notion of *telepresence*, which has assumed such critical importance since the development of VR and augmented reality in the arts. Jack Loomis's 1992 article 'Distal Attribution and Presence' is foundational here, as is Stephen Jones's 'Towards a Philosophy of Virtual Reality'; a more recent publication of primary interest focuses on the UnStumm | Augmented Voyage mobile app and server infrastructure, an artistic vehicle for the realization of telematic live performances (video art, music, and dance in augmented reality).[2] If, traditionally, the notion of presence in art has been linked to that of truth, the ontology of the work of art must question the way telepresence challenges its fundamental assumptions.

### **3 From aesthetics to philosophy of mind**

Because of the focus on technical aspects, there was a lack of development of aesthetic thought in early thinking/writing about networked performance. Thus, an opportunity was missed to renew or reframe canonical aesthetic questions in the context of a performance practice that, by virtue of the communities it serves and reaches out to, paradigmatically associates art and technology. Some of the questions raised by distributed performance, such as the nature and qualities of the virtual sonic space in which the performers 'meet', extend beyond a solely aesthetic inquiry (for example in the notion of presence – *phanesthai* – or telepresence), and touch also upon questions of philosophy of mind. (Are musical mental phenomena internal or external, are they to be found 'within' the mind or are they only real in as much as they are actualised in the public sphere? What if this reality is enacted in a virtual space? Etc.) These are some of the questions that would benefit from being confronted to philosophical approaches other than aesthetics under the paradigm of subject philosophy (a paradigm under whose influence Heidegger remains, even though he seeks to distance himself from it).

### **4 Accentuation of the musical concept of space**

One concept that has received recent interdisciplinary attention, bringing together artists and scientists to consider the notion in both its aesthetic and cognitive dimensions, is the question of space. For example, this topic was a focus of discussion in a 2021 event in Aalto University's LASER Talks series (LASER = *Leonardo* Art Science Evening Rendezvous).[3]

Thinking about the concept of space has merits also on a strictly musical level. A number of questions and assumptions of compositional and/or theoretical significance can be profitably reassessed in the light of the experiences to which online jamming exposes participants. One of the most obvious is the question of reconciling improvisation and composition (improvisation in writing, or notions such as musical discourse in improvisation and written music, etc.), since many contexts in which networked performance is produced call for improvisation. Another obvious area of questioning is the concept of space and how musical space can be elaborated compositionally; how it differs from one composer to the next; how different compositional approaches entail a particular relation to the notion of space, compositionally speaking (whether tonal, polytonal, atonal, metatonal, concrete, stochastic, repetitive, etc.). Thus, internet acoustics and audio panning systems could go hand in hand with a reflection of space as an intraspecific category of compositional practice. In that context, it would be particularly interesting to question whether or not the technological means used entail a predetermination of certain aesthetic aspects, or whether the technology employed has no aesthetic qualities, in the same way Langdon Winner in a canonical essay from 1980 investigated whether, beyond mere efficiency, “technical things” (as he calls them) were embodying “specific forms of power and authority”.<sup>[4]</sup> I would like to pursue this reflection by investigating more thoroughly what a reevaluation of space as a musical category, or parameter, entails, both in terms of sonic and acoustic qualities, but also as a notion that appears to be relatively little looked into compared to, say, the notion of time in music.

When referred to the use of technological means that enable ultra-low latency in sharing sound for collective music-making, one cannot fail to wonder whether these tools do not implicitly call for a reevaluation of notions that had previously received little attention. A virtual space whose sonic qualities are not predetermined as they would be in a physical space – i.e. a space whose morphology is dependent upon factors that can be largely acted upon, such as latency, spatialisation of sound sources, panning, reverb, loopback, etc. – underlines the fact that sonic space is as much the result of a deliberate compositional decision as musical figures themselves are the result of a compositional strategy. Different archetypal harmonic patterns or distinctive musical figures, the relations different sounds have with one another in general, convey for each composer a particular image of a sonic space that is dependent on idiomatic syntactical features. The exceptional breadth and diversity of musical approaches to organising sound and material after World War II (from constructivist procedures to indeterminacy, microintervals to a mathematical approach to sonic space (Xenakis), composition using algorithms to metatonicity, etc.), concurrent with key developments in electronic music, pushed to the foreground compositional concerns about space and the localisation of a sonic source as an intraspecific component of sound itself, along with pitch, duration, timbre and dynamics. It is of utmost importance at this point in the discussion to make a clear distinction between two different aspects covered by space as a musical phenomenon. As I referred to different styles and composers having their idiosyncratic signature as to what a space is, I intend to highlight an understanding of musical space as a sonic space, i.e. a space dependent on pitch organisation. In that regard, space enables a certain phenomenality

of sonic perception that is different from one typology of sonic space to another. This phenomenality is accompanied by certain physical effects the music has on the listener. On the other hand, the concern with spatialisation has to do with the notion of acoustic space – a separate notion from sonic space in that the spatialisation of music and the constitution of a sonic space specific to music are two different things. Thus, no music can escape dealing with space, as it develops concurrently with the sonic organisation of the musical phenomenon itself. We could therefore think of sonic space as a notion entirely defined by pitch organisation. Another way of characterising both notions is to think of sound space as an *intrinsic* space, dependent on spatial configurations generated by the relationships sounds have with one another. Acoustic space, on the other hand, can be thought of as an *extrinsic* space that deals with the physical spatialisation of sound in the space in which the music is being performed and heard. Of the latter, Stockhausen says that it constitutes a “new dimension of musical experience”.<sup>[5]</sup> An appropriate way to summarise the specifics of both notions while maintaining in the listener’s mind their conceptual proximity is to say that *acoustic space has to do with spatialising the music, while sonic space musicalises compositional space*. By the musicalisation of compositional space, I mean the characterisation of a space proper to music, an intrinsic component, as opposed to space in other artistic media, such as sculpture in its making or painting in its making.

Before turning to broader philosophical and political considerations, what provisional conclusions can be drawn from the previous reflections concerning the diversity of musical and aesthetic investigations to which the reevaluation of the notion of sonic space lends itself? First and foremost, it seems to me that the increased sensibility to space that distributed performance calls for, and which it helps shape as a musical parameter equal in importance to the four traditional parameters (timbre, duration, pitch, dynamics), highlights the need to question the idea of space as a ‘given’, as a compositional *a priori* – as a void component, deprived of any intrinsic qualities, that merely needs to be filled with sounds – or as a domain of music creation in and of itself that calls for an active elaboration. I have sought to indicate already that if I consider space to be a valuable means of questioning our auditory sensibility, this is precisely because it results from a deliberate compositional strategy or decision. If I do not think of musical space as an *a priori*, a void to be filled, I nevertheless consider it to be an *a priori* of our sensibility, as any trained musician will necessarily perceive its plasticity differently, from one composer to another, from one principle of sound organisation (tonal, atonal, etc.) to another. Music therefore does not ‘happen’ in a given metaphorical space, but it gives that metaphorical space its specific form or *Gestalt*. There is an expressive plasticity to music as much as there is a plastic expressivity to it. The category that we can deduce from these considerations is that of morphology: morphology of the musical figures and morphology of the sound space that results from these figures.

The particularity of being part of a networked performance is that two different spaces, that of the musical figures created by the composer(s)/improvisers and that of the virtual shared space, become spheres of expression that can be acted upon in such a way that the performers are hearing an actual polyphony of spaces: the space immanent to the pitch organisation, and the shared space of the performance that becomes

audible as such through the headset. This experience is particularly acute when using the software JackTrip.

Distributed performances call for an ‘augmented’ approach to musical composition, one in which sonic space is dealt with as a parameter of equal importance to the other parameters – both on a metaphorical level as well as on an acoustic level, and their mapping. Latency, too, can be turned into a compositional constraint from which imagination can flourish, rather than an impediment to real-time interactions. In a similar fashion, we must reconsider how we can make sense of the sonic organisation of these pieces on an analytical level. The question that arises is how to formalise new analytical models that would facilitate a taxonomy of the different approaches that composers and improvisers in distributed performances take as they develop a musical approach based on the particularities of the software. At stake is the possibility of giving an account of the phenomenality of sound and its physical aspects in a virtual acoustic space. In addition, the fact that the virtual acoustic space is the space perceived by the performers, not the one perceived by the audience members, who experience the rendition of the piece or improvisation on their computer, creates a disparity between the performer’s and the listener’s experience of the music.

## **5 Rethinking the artwork, decentering the composer**

Network performance calls for reevaluating the notion of the artwork itself. As mentioned earlier, if traditionally the notion of presence in art has been linked to that of truth, the ontology of the work of art has to take into consideration the modifications that it has undergone since the emergence of the concept of telepresence.

The theory of telematics is rooted in questions pertaining to the philosophy of technology, notably that of the articulation of the social sphere of cultural practices and of the technological sphere. Concretely, this means that questions are raised about technological determinism applied to ensemble music – questions of ‘reverse adaptation’ (Langdon Winner) and of the social practices linked to the traditional practice of music in Western societies. To take a concrete example: how can telematics help call into question or reformulate a fundamental assumption of Western classical music such as the distinction between the categories of improviser and composer? While this question is not specific to telematics, network performance poses it with particular insistence. Besides asking whether network performance bears predetermined aesthetic attributes or whether it is a ‘transparent’ environment on which technological constructivism has no hold, then, the issue that I would like to address here is that of the decentering of the figure of the composer. In doing so, I hope to establish which aspects of the discussion are dependent upon the technological artefact, or made possible by the artefact, or whether this decentering is cultural in nature, i.e. emerges from the supportive and collaborative nature of the community of music practitioners who work with technology.

If telematics does not entail the death of the author in the structuralist sense of the term, it leads to what I would call a ‘decentring and redistribution’ of the role of the composer, concomitant with the reevaluation of the traditional distribution of roles in Western musical practices to which it also leads. Telematics has the ability to reformulate the spatial distribution of instruments, as instruments can be remixed and respatialised in real-time diffusion. This leads to a metaphorical democratisation of access to sound, as the musicians can reassign their placement in the virtual space, while heterogeneous timbres can be remixed and rebalanced. The hierarchy of roles assigned to the different instrumental groups in a classical ensemble thereby becomes scrambled and recoded. As for the role of the composer, it is in large part determined by the culture of the musicians participating in telematics concerts, as is exemplified by pioneering figures such as Pauline Oliveros: this culture by its nature and history encourages collaborative practices. The programme notes of the pieces *PicYour-Score 2020 – Pandemic Edition* by Hassan Estakhrian and *Whose turn is it anyway* by Michele Cheng exemplify this tendency, as the composers position their works as the product of a collective effort. Both works were performed by the SNE in the period 2020–21, after its shift to online activity, and were composed specifically for this online performance environment.

## **5 Collective creation and student engagement**

The question of international community-making is at the heart of networked performance. A significant example of this is a collaboration whose results, both artistic and musical, are still vividly remembered by the community of performers and musical technologists involved. In 2008, musicians from Beijing and Stanford universities were able to perform Pauline Oliveros’ *The Tuning Meditation* with audio that the composer subsequently described as ‘beautifully clear’.[6] This kind of collaboration highlights the importance of thinking of collective practices as a way to enhance a sense of community in music-making, precisely Oliveros’ project in the aforementioned piece. At the same time, the technological means used make it possible for the participants to identify as a community, despite the almost 6,000 miles that separate the two campuses.

As artistic and musical director of the SNE, I have placed great value, during the pandemic, on cultivating a sense of community, of which students risked being deprived. The outreach initiatives extended way beyond the usual students who register for the ensemble, as online music-making with uncompressed audio and near-zero latency was made available to virtually anyone interested, regardless of their location at the time the pandemic started. CCRMA and the Department of Music sent tens of JackStreamer kits that contained a mic, cabling and a digital audio interface. The intimate sonic rendition of JackTrip, once the initial setup was done, made it possible for all SNE participants to maintain an ensemble musical activity in a virtual acoustic space that made them feel as if they were in the same ‘room’, even though some of them were thousands of miles away from the Bay Area, where CCRMA’s servers are located. Because of the long distance, we frequently had to use a larger window size

with more latency, so as to avoid glitches and packet loss. That technological aspect itself determined musical and compositional strategies, in terms of what was possible and how.

Reflecting on my own experience using these tools in pedagogical settings, the community- building potential of networked performance is clear, especially in the context of the pandemic era. As a sense of belonging to a learning community was made very difficult for many students, the impression of shared audio space provided by networked performance – as opposed to videoconferencing software, which is designed for turn-taking in audio rather than simultaneity – conveyed the impression of being in the same room or space, even though the ‘room’ was the internet. This creation of a shared virtual space allowed for the formation of musical ensembles which connect across geographical distances, allowing students who spent significant periods of the pandemic in other states or other countries to remain connected with their classmates in a unique space for sonic sharing.

The question of technology (latency, quality of service, etc.) is not simply a question of the milieu in which music is being performed, a milieu which one hopes would provide optimal sonic rendition. The question of technology here is intramusical. Reevaluating the notion of sonic space in the light of its becoming an online virtual space implies reevaluating both the notions of sonic object and musical object, i.e. the notion of artwork itself. Rethinking the notion of the artwork implies questioning its genealogy. In the context of products of the mind, it means questioning the notion of authorship as it was inherited from modern philosophy, centred around the notions of subject and consciousness.

Highlighting the erasure of the authorial presence seems to involve a paradox. I have argued in favour of distributed performance as a musical practice that has the potential to dehierarchise the traditional roles of music-making as they are conventionally delineated by different specialisms, and thus to help marginalised practices by underrepresented artists and technologists gain visibility and audibility. The decentering of the authorial presence remains a paradox for as long as the discussion remains informed by subject philosophy. But a more eloquent and potentially fecund approach to the disappearance of the author may come from reconsidering the notion in a different philosophical context. For if we set aside the philosophy of consciousness, a holistic approach to philosophy of mind reveals itself to be a suitable analogy to the way musical minds interact with each other in a virtual space. In Barthes’ text, the death of the author had “the birth of the reader” as its corollary. Now, to the question of whether the mind is ‘inside’ or ‘outside’, the performer can respond, with Wittgenstein or C. S. Peirce: outside, within the public sonic sphere.

## **6 Towards a conclusion: aesthetics and politics**

Having presented what I believe to be some of the most salient aspects of networked performance, particularly in relation to the kind of use we made of JackTrip, and as I reflect upon these aspects, not only as a sequence of separate considerations, but as a bundle of problematics to be made sense of together, I would like in closing to offer



an outline of how distributed performance might help us think about the relation between aesthetics and the political. The work of Jacques Rancière provides a particularly eloquent account of this interrelation, with his notion of ‘artistic regimes’; as does that of Jean-Louis Déotte, who in addition to politics and aesthetics managed to develop – through the notion of *appareil*, inherited from Walter Benjamin – a polyphonic dialogue at the crossroads of art, politics, the sciences and philosophy.

Rancière refers to aesthetics as:

the system of *a priori* forms determining what presents itself to sense experience. It is a delimitation of spaces and times, of the visible and the invisible, of speech and noise, that simultaneously determines the place and the stakes of politics as a form of experience.[7]

For this reason, according to Rancière, the “distribution of the sensible” is what “is at stake in politics”.[8] The redistribution of the attributed roles to which I referred above, made possible by networked performances conceived *natively* as distributed performances, therefore fosters a reconsideration of the political and ethical import of a sense experience that is made possible by the mediation of computers – that is to say, it questions computing ethics directly through the lens of aesthetics and politics. As it redistributes sense experience, networked performance thus gives form to communities that become conscious of themselves *as* communities as they display “what is common to the community, the forms of its visibility and of its organization”.[9]

I only briefly mention these aspects that are currently central to my research on communicational activity, technology and philosophy of culture, so as to indicate perspectives that in my view live up to the task of thinking in the context of liberal democracies. In conclusion, I will just hint at potential further steps for my research that I think are the corollary of some of the ideas I have exposed in the present article.

The conceptual framework in which we can think freshly of an ethics of discussion has necessarily to be informed by the computing breakthrough of recent decades. Such a dimension was noticeably lacking in the attempts of philosophers in the late 1970s and early 1980s such as Karl-Otto Apel and Jürgen Habermas – a lack all the more disconcerting when one considers that information technology was at that time a blooming topic that directly impacted the philosophy of communication.[10] The idea of a virtual and metaphorical space for a community that defies attributed roles and rearranges the sense experience of the singular and the collective, the near and the far, entails the idea of an unlimited communication community, and of a reconfiguration of the sensible within an ethics of discussion that acknowledges the mediating role of the computer.

It is the question of ‘community’ that makes dealing with the notion of an ethics of discussion a necessity; and the particular modification that ‘community’ undergoes in the context of networked performance, as I have argued, is that of a redistribution of sense experience. JackTrip thus offers an analogy to the ideal community of communication, without needing to anchor it in an *a priori* that seeks to absolutely found the moral requirement in a transcendental pragmatics. From phrase to musical phrase, from proposition to philosophical counter-proposition, an ethics of discussion requires a continual exercise of judgment, without any guarantee of communicational felicity or infelicity, without searching for the consensus that precisely inhibits our philosoph-

ical faculty of judgment. Networked performance allows this reconfiguration of sensible experience, and hints at a way of approaching computer ethics in which human-computer interactions can help us imagine a potentially unlimited macro-ethics of communication.

## References

1. Weaver, S 'Editorial', *Journal of Network Music and Arts* 4/1 (2022), <https://commons.library.stonybrook.edu/jonma/vol4/iss1/1>, last accessed 2022/11/04
2. Loomis, J. M., 'Distal Attribution and Presence', *Presence: Virtual and Augmented Reality* 1/1 (1992), 113–119; Jones, S., 'Towards a Philosophy of Virtual Reality: Issues Implicit in "Consciousness Reframed"', *Leonardo* 33/2 (2000), 125–132; Hein, N., Schmitz, C., and Hahne, S., 'UnStumm | Augmented Voyage: A Platform for Telematic Live Performances in Augmented Reality', *Journal of Network Music and Arts* 4/1 (2022), <https://commons.library.stonybrook.edu/jonma/vol4/iss1/6>, last accessed 2022/11/04.
3. Miltiadis, C., Singh, S., Landau, F., and Gencoglu, O., 'How Can We Define Space?', LASER Talk at Aalto University (29 April 2021), <https://www.youtube.com/watch?v=v9RyiObRM3U>, last accessed 22/11/28.
4. Winner, L., 'Do Artifacts have Politics?', *Daedalus* 109/1 (Winter 1980; special issue titled 'Modern Technology: Problem or Opportunity?'), 121–136 (here p. 121).
5. Stockhausen, K., 'Musik im Raum', in Stockhausen, *Texte zur Musik*, Vol. 1 (Cologne: DuMont, 1963), pp. 152–175, (here p. 153). My translation.
6. Oliveros, P., 'Networked Music: Low and High Tech', *Contemporary Music Review* 28/4–5 (2009; special issue 'Network Performance'), 433–435.
7. Rancière, J., *The Politics of Aesthetics*, transl. by Gabriel Rockhill (London: Bloomsbury, 2004), p. 8.
8. Rancière, p. 3.
9. Rancière, p. 13.
10. I would particularly mention here Hiltz, S. R., and Turoff's, M. pioneering digital sociology study *The Network Nation: Human Communication via Computer* (Reading, MA: Addison–Wesley, 1978).

# eLabOrate(D): An Exploration of Human/Machine Collaboration in a Telematic Deep Listening Context

Rory Hoy and Doug Van Nort

DisPerSion Lab - York University  
hoy@yorku.ca, vannort@yorku.ca

**Abstract.** *dispersion.eLabOrate(D)* is a networked performance system which augments and supports Deep Listening workshop experiences through an environment that integrates human and machine collaboration. The sonic materials for this co-performance/creation are seeded by vocal activity of human participants, which continually contribute to an audio corpus of past content used for resynthesis of machine voices. Each participant experiences their own spatial sonic reality within a shared virtual audio space, as relative placement to other collaborative sources provide a unique vantage point via an accompanying virtual acoustics system. Responses from public play sessions are analyzed using a grounded theory approach to report on salient qualitative data resulting from performances with the system.

**Keywords:** Telematic Performance, Networked Audio, Interactive Agents, Deep Listening

## 1 Introduction

Network-based communal activity and connection in the area of telematic music grew dramatically throughout the early lockdowns and cancellations caused by the global COVID-19 pandemic [13], as musicians turned to software solutions in order to continue their regular performance sessions at a distance. For example, in this period via the DisPerSion Lab we produced over forty telematic performance events involving more than forty performers. Research and development in our current “post-pandemic” context continues to foster distributed musical practice and telepresence, for ourselves and many others, through various systems capable of very low latency and high quality audio. One distinct performative practice that was impeded by the lack of in person events, both in our own local lab context and more broadly, was Deep Listening – described as “a practice that is intended to heighten and expand consciousness of sound in as many dimensions of awareness and attentional dynamics as humanly possible” [18], by its creator Pauline Oliveros. Public engagement with the typically in-person and group-based Deep Listening workshop events were therefore put on hold until restrictions had lightened.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Continuing this trajectory of research-creation work conducted on telematic performance, Deep Listening practice and human/machine co-creation, the system outlined in this paper was developed to explore the results of augmenting a well-known Sonic Meditation-style [20] text piece by Oliveros from the Deep Listening literature called The Tuning Meditation. In particular, this augmentation introduces non-human agents directly performing alongside human participants. Entitled *dispersion.eLabOrate(D)*, this project is a conceptual reconstruction and extension of a previous system by the authors, entitled *dispersion.eLabOrate* [10] - with the updated name reflecting that the system, and the practice that it fosters, is now *(D)istributed*. Vocalizations made by human performers during this piece are captured in an audio corpus which is then used both as raw material for the synthesis of new tones by the machine agents, as well as functioning as their running memory of past sonic events. The population of agents is variable and each acts autonomously according to the score for The Tuning Meditation (which will be outlined in section 2). While physically dispersed, players and machine agents are placed within a singular virtual acoustics space to be heard within the same environmental conditions.

One challenge for this project, something faced by most telematic-based performances, is the collapse of spatial qualities to a (typically) stereo mix of all performers. This flattens any variance in positioning of local sound sources one would find within an in-person event, which can provide contextual information or sonic material to react to. Systems that can support spatial representations of source placement typically orient sound to a particular “sweet-spot” in the centre of the virtual space, which all sources are placed relative to. For *dispersion.eLabOrate(D)*, we develop and present a spatial audio setup which allows for unique sonic perspectives tied to relative placement within the virtual acoustics environment.

Following the completion of the system, Deep Listening workshop sessions were held to gather qualitative feedback on various elements of the experience. These responses are presented and analyzed with a grounded theory approach in section 5. Key categories of responses are discussed, which were found to focus on immersion and communal space, diversity of machine voices, and strategies for human/machine collaboration.

## 2 Related Work

Our previous work on augmenting Deep Listening practices emerged over the course of a 12-week DisPerSion Lab seminar that posed the question: “Can we imagine ways that interactive systems might synergize, entangle with, and augment – but not distract from – Deep Listening practice?” This resulted in our performance system *Dispersion.eLabOrate*, created for collective listening and sounding in a shared physical space. This system can detect and react to player vocalizations as well as ambient sound within an environment. Like this newer project, *eLabOrate* was designed to engage and augment group performances of The Tuning Meditation (TM), placing the output of the system as a machine agent which engages with the vocal and collaborative dynamics of the human participants. The TM asks participants to focus on their breath – inhale deeply, exhaling on a tone of their choice for one full breath. On the following exhalation,

tion, match a tone currently being sung by another player. Then on the next vocalization, sing a new tone that hasn't been sung yet. This cycle continues until a natural end point is reached where each player has stopped.

Building upon this past work, we once again begin from the position that the machine voice/participation is not an element which should conceptually or perceptually dominate the piece, but rather should work in tandem alongside human performers to facilitate broadened sonic potentials. We investigated the past system through the lens of "Sonic Ecosystems", foregrounding resonance, feedback, and autonomous behaviour inside/outside the direct influence of human action. Building upon related works in the field, Sonic Ecosystems are framed here as performative contexts – ones which establish environments that in turn adapt to agents, which define their own self-regulating populations and their own ambience. This ambience may be naturally-occurring within the acoustic space facilitating the system, or could be generated as a result of the sonic ecosystem's behaviour. These systems rely on self-monitoring techniques both virtually and physically, often including microphones or other sensing devices within the space in order to enact and react to these recurrent activity loops [6]. Musick (2016) [16] provides a thorough look at the theory and practice of the field within their Sonic Space Project, and includes assessment strategies of sonic ecosystems within their 2014 paper [15].

In addition to *eLabOrate*, another recent DisPerSion Lab project by Maraj and Van Nort [14] also focused on developing an agent-based system for interactive performance, building upon rules found in a Sonic Meditation-style piece. In this case the text piece *Interdependence* was used as a starting principle for structuring agent interaction, and the focus was on gestural performance with an interactive system rather than collective vocalization in a workshop setting. Both projects emerge from and sit at the intersection of two larger DisPerSion Lab projects that engage this broader area, entitled *Deeply Listening Machines* and *Deep Listening Entanglements* [7], both of which build upon past work on intersecting machine improvisation and Deep Listening principles [24].

### 3 System

*eLabOrate(D)* allows for more complex, refined, and flexible agent behaviour to that of its predecessor *eLabOrate*. Where *eLabOrate* created a pervasive and mirroring-like behaviour for all vocalizations and ambient sound, *eLabOrate(D)* more closely follows the cyclical behaviors of new tones, matching, and most importantly active listening as is requested within the context of *The Tuning Meditation*. A key concept discussed within the Deep Listening community of practice is the distinction between directed and focused active listening, as opposed to the passive physiological process of hearing [19]. The behaviour for each machine agent in the system enacts this active listening as opposed to the more passive and reflexive hearing and sounding which occurred in our past work. Situating this performance system as a telematic piece, we also investigate the viability of a remote virtual shared space as a facilitator for the characteristics and behaviour of sonic ecosystems.

In practice, the *eLabOrate(D)* system is comprised of modules to capture human signals and generate machine voices created in Max/MSP, which are detailed in the following subsections. These are instantiated at the beginning of a session, with an individual module for each human/machine participant being scripted once the total number of participants is selected. These modules allow captured and generated sound to be sent to an accompanying patch to be spatialized appropriately for each human participant. All participants (both human and machine) are placed in a circle within this virtual acoustics space, which is relevant for machine voice behaviour and will be outlined in subsection 3.2.

### **3.1 Audio Corpus - Sonic Memory**

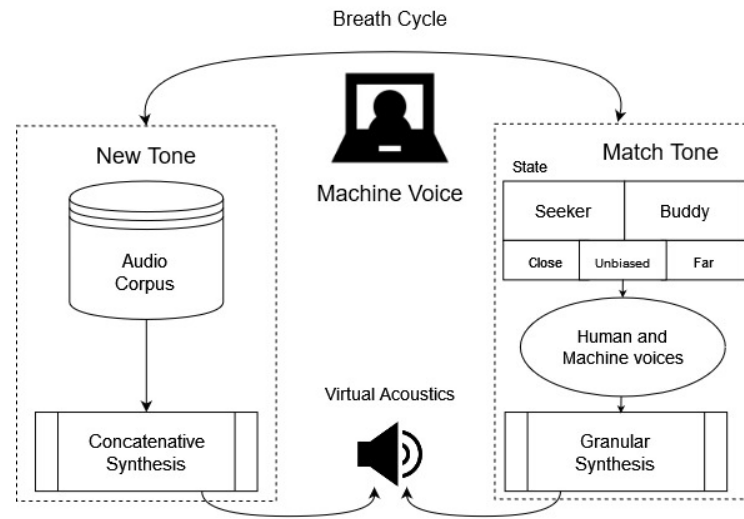
The audio corpus is implemented through the Mubu package [21] in Max, and is used within the system as a running memory of all vocalized tones. Input signals from human players are segmented into audio buffers based on detected onsets, with recording taking place until the signal falls below an established threshold. The system is limited to the last  $n$  (default 50) segmented tones in order to avoid memory & processing issues, but could be extended depending on the hardware capabilities of the host computer. This places the corpus as a short term memory of sonic events, as the oldest events are erased when a new buffer is saved to the corpus beyond the limit. Buffer input is held for a short time after vocalization ends, to allow machine voices the opportunity to match without the target voice being explicitly active. This behaviour is similar to human participants matching another tone briefly after a given vocalization has stopped, which happens often in practice. The buffer content is analyzed and segmented through Mubu and is made accessible within the corpus as separated grains.

### **3.2 Machine Module**

Each machine module consists of separate logic sub-modules inside. Controlling the movement between matching and new tone behaviors is the breath control module, which mimics a range of human time scale breath cycles (Fig. 1). This approximation of human breath allows the machine agents a voicing and breathing alternation, so as to both avoid continuous output and to better align with the time scales present within a typical performance of the Tuning Meditation.

New tones are made up of grains from the collective audio corpus populated by human participant voices, scrubbed through and resynthesized using a concatenative synthesis [22] method within Mubu. A target frequency area is scrubbed to playback these grains as a continuous voice, with the resulting tone constituting a new voice comprised of grains from various participant sounds.

Machine voices are assigned one of 6 possible states upon instantiation, which define either their matching (Seeker or Buddy) or their spatial biasing (Close, Far, Unbiased) behaviors. Matched tones result from copying a desired target's current voice buffer content into the acting machine voice's buffer. This buffer is then played back with a granular synthesis method to mimic the held tone by another player (both human and machine). The *Seeker* state implements a spatial encoding neural net using



**Fig. 1.** Machine voice module depicting breath cycle alternating between both vocalization states and synthesis engines

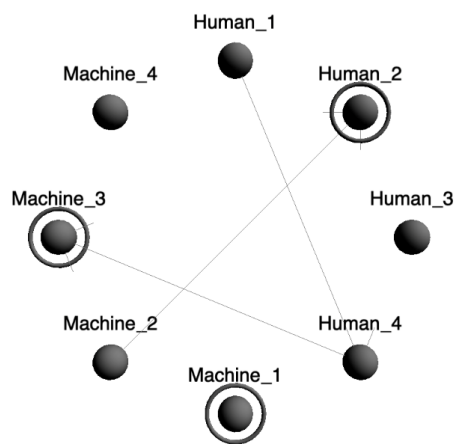
ml.spatial [5] to keep track of recently-matched participants. A recent match of a participant would mean that on the next match behaviour, they become the least likely to be chosen. The *Buddy* state chooses a small subset of the total population to use as potential matching targets. A spatial bias of near/far/unbiased for both *Seeker* and *Buddy* skews the likelihood to match a specific subgroup based on angular proximity within the established virtual circle. Participants within a  $\pm 90^\circ$  degree arc from a given voice or less are deemed “close”, while beyond this places a voice as “far”. Through these added logic modules, the system possesses possible “listening behaviours” that are above and beyond the basic instructions of the TM piece, thereby providing differing kinds of performative identities for each machine voice.

### 3.3 Human Module

Human participants are connected to the system and to one another by utilizing Jack-Trip [1], a software tool for low latency and high quality audio. Depending on the microphone setup of each participant, acoustic characteristics of each player’s environment may be sent along with their vocal input allowing for a perceived commingling of ambient sounds within the virtual acoustics space. These various local sonic realities are relayed along with one’s intended vocal utterance of a new tone or matched tone, and may therefore become sonic material that a machine voice may pull from to generate its own new tone. In this way, the expanded sense of shared acoustic environments remains present as a point of focus during these sessions - in the spirit of Deep Listening practice that emphasizes attention to one’s sonic environment. These vocalizations can be conceived of as “the performer-instrument articulation”, theorized by Waters as

“[...]result[ing] not only from the physiology of the player, but also from the complex feedback into that player’s body of vibrating materials, air, room, and the physiological adaptations and adjustments in that body and its ‘software’ which themselves feed back into the vibrating complex of instrument and room.” [25]

This feedback is additionally facilitated by the networked nature of the system, and in the individual behaviours of each machine voices. Resynthesis of vocal material of participants continually shapes the timbral range of the machine voice, and in turn has influence over the potential tones a human participant may match through the score. This also comes about through the matching behaviour of the machine voices, as they are able to copy another vocalizing machine.



**Fig. 2.** Layout of human and machine voices (top-down) within the accompanying virtual acoustics space. Lines denote matching behaviour and rings denote new tones being vocalized.

### 3.4 Multi-Spat - Virtual Acoustics

All output is processed and positioned spatially within an accompanying multi-listener spatialization patch developed for this system. Audio from each human and machine participant is placed at a corresponding source location distributed evenly in a circular pattern around the virtual space. Relative spatial listening mixes are achieved through separate instances of IRCAM’s Spat 5 [2], with one virtual space model per human player. In practice, this allows a hypothetical listener 2 to be placed to the left of listener 1, and be heard from that direction by listener 1. The same is true for listener 2; their relative position to 1 would perceptually place listener 1 to their right. Each instance of Spat is run using a binaural panning mode for a stereo output, as headphone based monitoring is encouraged for the performance. Two virtual audio drivers [12] [9] are employed to allow for routing to and from JackTrip for each participant and allows a



separate and accurate binaural mix to be passed individually back to players in relation to their virtual orientation. An accompanying visual layout of the spatial positions and behaviour of each participant can be displayed to depict current activity of sources (Fig. 2). New tones are depicted as rings around a given source, and matching behaviors point to the target source a voice is matching.

#### **4 Study/Play Sessions**

Telematic Deep Listening workshop sessions were held with various sized groups of players in order to explore multiple factors of engaging with machine agents in this setting. The sessions were facilitated by the second author, a certified Deep Listening instructor, and placed focus on performance of the Tuning Meditation in the context of also drawing attention to the shared sonic environment, one's local environment, one's body and to inner listening - all common elements of a Deep Listening workshop session.

Participants were invited through calls sent out to online email lists and social media groups focused on computer music, Deep Listening, sound art/studies and listening more broadly. Based on scheduling alignments, we arrived at 8 total participants who connected from disparate locations in North America and Europe. Multiple sessions were held, which included an equal number of human and machine players at a given time (eg. 4 human players, 4 machine players in one session). Sessions were an hour in length and included two different performances of the TM, each with a different active state for the machine voices. States were decided randomly (without duplicates) in order to have at least one response to each of the varied matching behaviors across all of the sessions. To recap, these states include:

- Far Seeker, Close Seeker, Non-biased Seeker
- Far Buddy, Close Buddy, Non-biased Buddy

These states introduce a bias towards spatial positioning of participants (far, close, unbiased), and a matching behaviour (Seeker or Buddy) which alters how the machine voices attempt to match another vocalizing participant.

Participants were not primed on several factors of the experience, as we were interested in gathering undirected qualitative data for analysis using a grounded theory approach. Grounded theory is a qualitative methodology aimed at uncovering key information from responses and allowing for central themes to emerge via multiple stages of coding - extracting relevant data and ultimately "Crystallizing the significance of the points" [3]. Our grounded theory-based approach for this study consisted of separate open coding steps each done individually by both authors. After this initial coding pass, cross-checking of codes occurred followed by focused coding - creating larger categories of responses that were synthesized from the resulting codes and will be presented in the following section.

After engaging in two runs of the meditation, players were invited to complete an online form in the (approximately) fifteen minutes remaining within their given session. The questions were designed to allow for open reporting on the experience with the goal of allowing key areas of personal interest and thought processes to come to the

forefront. That said, once the session was completed it was clear to participants that there were both human and machine voices present, and so we explicitly asked about the experience relative to these distinct entities. The questions provided were as follows:

1. What are your general thoughts about the session?
2. How would you characterize the various voices (both human and machine)?
3. Could you compare and contrast your experience of the two (human/machine)?
4. What was your strategy for following the TM piece?
5. Could you characterize your relationship to space and describe if (and how) it might have influenced your experience?

To clarify demographic and background information, we also asked participants if they had any previous experience with Deep Listening as a practice, and if they had ever performed The Tuning Meditation.

## **5 Analysis**

Three key categories of focus emerged through our process of coding participant responses and subsequent analysis, which we will discuss in the following sub-sections.

### **5.1 Experience of Immersion in a Shared Communal Space**

A recurring theme that was prevalent throughout participant responses was a sense of immersion, with this being tied to a characterization of the session as a shared communal space. “I lost all sense of my local space. With eyes closed I was entirely in a shared space with everyone. I forgot we were not physically together”, noted one participant with an extensive background in Deep Listening practices. As one might expect, specific mention of the term “sonic ecosystems” was not present in responses, however those characteristics we previously identified as belonging to sonic ecosystems were indeed reported, with one player explicitly noting “I noticed a bit of a back and forth between being influenced by the machines and the other humans in the session”. In response to a subsequent question, they evoked metaphors of physical ecosystems:

“I was visualizing the sound itself a lot more. I felt like I was contributing to a moving stream. I didn’t know how loud I was, and so it felt like I was occasionally throwing a bucket of dyed water into the stream as the water flowed by, changing it in ways I wasn’t aware of.”

This ambiguity of outcome also gestures to the lack of direct control over the system. While each performer is an active participant in seeding the amalgamated voices generated by the audio corpus, there is a blurring of causal human action to machine reaction. This is congruent with the concept of a “floating phenomena/floating piece of art” from Weibel & Dinkla, described by Dixon as “[...] no longer the expression of a single individual. Neither is it the expression of a collective, but it is the state of a ‘connective’ - a web of influences that are continually reorganized by all participants.” [8] This is further emphasized by another participant, in commenting on the influence that the space had on their interaction with others:

“The relationship with space was expansive, I was traveling across the space to meet the tones. It highly influenced my experience particularly in the second tuning. I was able to go with more ease and pleasure, as the space expanded for me, on the possible tones and voices to tune in.”

These performer statements, representative of the broader viewpoints expressed, depict experience within the session as taking part within a shared space which affected their perception of the inter-relational action present within the Tuning Meditation.

## **5.2 Diversity of Machine Voices Expanding Timbral Content of Meditation**

The sense of a diversity of voices from machine agents was reported, and this was characterized as allowing for extended timbral content beyond the human. One participant stated: “[...] it was a great experience to listen to both human and machine voices and respond to them in real time, reflecting my own impression on them. Analyzing various notes of multiple voices was not an easy task but I enjoyed finding atonal harmony in inharmonious sounds.” Similarly, conception of collective space was also addressed via perception of the machine voice character, as one participant articulated: “The expansion of tones was really helping me to expand my sense of space and time, and my connection with others, and with my body tuning in.” This reinforces sentiment from the previous subsection (5.1) while here being expressed in relation to the “expansion of tones” offered by the machine voices.

One participant characterized the voices taking part within the piece as “diverse, some calmer than others. Most of them steady, but some evolving and agile”, which highlights the varied approaches that both human and machine voices took in either matching or new tone vocalizations. Another participant expressed that the character of the machine voices was “[...] radical, refined, with a different atmosphere, pleasant too in a different way.” These statements cause us to question if this evolutionary/radical character was a product of the cyclical matching behaviours that are capable within the system, depending on the nature of the machine voices’ behavioural states. As one voice matches another, a chain-like effect can occur between both human and machine players that either match that same voice, or a voice that is already matching another (visible in Fig. 2). Once established, this type of chain may build upon and subtly (or not so subtly) alter the timbres at play and seed new material into the audio corpus which defines the machine voices. If and when this is established, such diversity and “refined” nature of the machine voices may also be a direct result of the concatenative synthesis technique used to derive the new machine tones via the “raw material” of the captured player’s voices.

In this complex human/machine network we can only speculate on the specific causalities - though we do note the above as affordances of the *eLabOrate(D)* system that we know to be at play in establishing a sense of evolution and refinement over time. What we can say with more certainty however, is that the machine voices were characterized as diverse, refined and evolving, both in terms of their timbral character and in their behaviours of interaction. This was articulated as a central influence in moving the dynamics of human attention forward in time.

### **5.3 Reflexive Engagement with Machine Voices and Influence Upon Performance Strategies**

An openness towards the involvement of the machine agents was clear in the previously-mentioned responses. For some participants, the overall experience was that all audio was collapsed into a cohesive sounding body: “At a few points, I couldn’t tell which were human and which were not.”

When they are clearly recognized, the incorporation of accompanying machine agents introduces such Deep Listening sessions to sonic material and gestures which would not occur within a human-only performance. It was reported by some participants that this coaxed out playful transgressions upon the score itself, with one player noting: “I tried to make sound[s] that were machine-like myself. I tried to ‘chop’ my voice, by tapping my cheek or throat, as I was influenced by the other sounds that were made.” Such transgressions certainly can (and do) come about in general during pieces such as the TM by participants who would like to push the limits of what constitutes a held tone, a pitched sound, etc. – and this is an important aspect of the social dynamics of this and similar pieces. Through the interjected behaviours of the machine voices, new timbral and rhythmic components are often introduced into the palette of materials which participants are engaging with. This added dimension of uncertainty and dynamism from this hybrid context is captured by Waters, who states “One of the benefits of hybrid (physical/virtual) systems is their very impurity: their propensity to suggest or afford rich unforeseen behaviors which engage the player (and the listener) at a variety of levels: sonic, tactile, and dynamic.” [25].

For some respondents, the incorporation of machine voices changed the conception of their own voice in relation to others. One participant stated, “The machine tones bring a strength from me, fearless voicing. The human voices invite me to listen more, and to engage with care for them, trying to explore the soft voices I haven’t listen[ed] to yet.” In contrasting human and machine contributions (Q3), another respondent expressed, “The human is easier to follow accurately, the machine leaves more room for interpretation of the note and timbre (which I enjoyed!)”, further relating these characteristics to one’s own strategy for realization of the score’s instructions.

## **6 Conclusion & Future Work**

We have presented our system *dispersion.eLabOrate(D)*, a set of telematic autonomous participants who engage the Tuning Meditation in the context of a Deep Listening workshop setting. An overview of our previous research and a system overview were presented, providing context and structure for relating the design of the system to a set of broader reflections, which were informed by a set of qualitative data that emerged from a series of workshop sessions with the system. Through these sessions, we investigated the potentials for augmenting group Deep Listening practices in a telematic setting via machine participation, and presented an accompanying virtual acoustics system allowing for unique sonic vantage points into the collective virtual performance space.

Responses to play sessions were approached and processed through a grounded theory methodology to parse out key “codes” and broader themes, resulting in the three

salient response categories outlined in section 5. These revealed a sense of immersion that was tied to “space”, understood as dual conception that was both social and sonic, a sense of evolving diversity that was carried forward by awareness of machine actors, and a set of strategies that articulated human responses towards positioning themselves within this sonic-communal engagement.

In future work, we will look to further explore and assess these perceived dimensions in this Deep Listening performative context, iterating our design (both computational and workshop structure) in light of what we’ve learned this far. This includes an examination of the concepts of immersion intensity [4] & presence [23]). From a design perspective, future considerations include new and varied implementations of virtual acoustic parameters, and expanding timbral possibilities concerning voices of the machine agents. Both realistic representations of in-person performance spaces and extended potentials for virtual space (physically impossible listening orientations, source positions, room qualities, etc.) offer new potentials for facilitating Deep Listening practices. This virtualized potential for Deep Listening practice is a key component of the *eLabOrate(D)* project in particular.

More broadly, this work contributes to the larger Deeply Listening Machines and Deep Listening Entanglements lab projects. These sister projects seek to transform and augment the kinds of listening and sounding practices found within the Deep Listening literature, such as that expressed by the Tuning Meditation (or Interdependence, in the case of Intergestura). These existing text scores act as starting points - seed ideas - for an evolving set of structured approaches to collective listening and sounding, both in public workshop and improvised performance settings. Each system such as *eLabOrate(D)* is part of this ecosystem of human/machine engagement in a Deep Listening context. Thus future work for this system is focused on diversity of approaches, such as exploring different methods for machine voice synthesis based on sound analysis and machine learning from human vocal inputs, and on modularity such that these particular machine voices might evolve new listening/sounding rules, and interact with other agents that emerge from the larger project.

## References

1. Cáceres, J., Chafe, C.: JackTrip: Under the Hood of an Engine for Network Audio. *Journal of New Music Research* 39 (3): 183—187. <https://doi.org/10.1080/09298215.2010.481361>. (2010)
2. Carpentier, T.: A New Implementation of Spat in Max. In: 15th Sound and Music Computing Conference (SMC2018), 184–191. <https://hal.science/hal-02094499>. (2018)
3. Charmaz, K.: *Constructing Grounded Theory*. Thousand Oaks, Calif: Sage Publications. (2006)
4. Colman, A., Aspă, L.: Development of a Questionnaire to Investigate Immersion of Virtual Acoustic Environments. In: DAGA, Aachen, Germany. (2016)
5. Day Smith, B.: *ML\* - Machine Learning Toolkit in Max*. <https://www.benjamindaysmith.com/ml-machine-learning-toolkit-in-max>. (2017)
6. Di Scipio, A.: Sound Is the Interface: From Interactive to Ecosystemic Signal Processing. *Organised Sound* 8 (December): 269—277. <https://doi.org/10.1017/S1355771803000244>. (2003)

7. DisPerSion Lab.: DisPerSion Lab Research Areas. <https://dispersionlab.org/research/>. (2023)
8. Dixon, S.: *Digital Performance : A History of New Media in Theater, Dance, Performance Art, and Installation*. Cambridge, Massachusetts ; London, England : The MIT Press. <http://archive.org/details/digitalperforman0000dixo>. 561. (2007)
9. ExistentialAudio.: BlackHole. C. <https://github.com/ExistentialAudio/BlackHole>. (2019)
10. Hoy, R., Van Nort, D.: Augmentation of Sonic Meditation Practices: Resonance, Feedback and Interaction through an Ecosystemic Approach. In: Kronland-Martinet R., Ystad S., Aramaki M. (eds) *Perception, Representations, Image, Sound, Music*. CMMR 2019. LNCS, vol 12631. pp. 591–599, Springer, Cham. (2021)
11. Hoy, R., Van Nort, D.: A Technological and Methodological Ecosystem for Dynamic Virtual Acoustics in Telematic Performance Contexts. In *Audio Mostly 2021, virtual/Trento Italy*: 169–74. <https://doi.org/10.1145/3478384.3478425>. (2021)
12. Ingalls, M.: SoundFlower. Objective-C. <https://github.com/mattingalls/Soundflower>. (2014)
13. Keller, D., Costalonga, L., Messina, M.: Editorial: Ubiquitous Music Making in COVID-19 Times. In: *Proceedings of the 10th Workshop on Ubiquitous Music (UbiMus 2020)* <https://halshs.archives-ouvertes.fr/halshs-03035034>. (2020)
14. Maraj, K., Van Nort, D.: Intergestura: A Gestural Agent Based on Sonic Meditation Practices, In: *13th International Conference on Computational Creativity, Bozen-Bolzano, Italy*. (2022)
15. Musick, M.: Examining the Analysis of Dynamical Sonic Ecosystems: In Light of a Criterion for Evaluating Theories. In: *ICMC 2014, Athens, Greece*: 154–161. (2014)
16. Musick, M.: *Practice-Led Research / Research-Led Practice Identifying the Theory and Technique of Sonic Space Ecosystems*. Ph.D., New York University. <https://www.proquest.com/docview/1871306234/abstract/9000417E723B4A90PQ/1>. (2016)
17. Neidhardt, A., Schneiderwind, C., and Klein, F.: Perceptual Matching of Room Acoustics for Auditory Augmented Reality in Small Rooms - Literature Review and Theoretical Framework. *Trends in Hearing* 26 (January): 233121652210929. <https://doi.org/10.1177/23312165221092919>. (2022)
18. Oliveros, P.: *Deep Listening: A Composer's Sound Practice*. iUniverse, Lincoln. (2005)
19. Oliveros, P.: Quantum listening: From Practice to Theory (to practice practice). *Culture and Humanity in the New Millennium: The Future of Human Values*, 27–41. (2002)
20. Oliveros, P.: *Sonic Meditations*. Baltimore, MD: Smith Publications. (1974)
21. Schnell, N., Röbel, A., Schwarz, D., Peeters, G., Borghesi, R.: MuBu & Friends - Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP. In: *ICMC 2009, Montreal, Canada*: 423–426 (2009)
22. Schwarz, D.: Corpus-based concatenative synthesis. *IEEE signal processing magazine*, 24(2), 92–104. (2007)
23. Støckert, R., Bergsland, A., Xambó, A.: The Notion of Presence in a Telematic Cross-Disciplinary Program for Music, Communication and Technology. In *Music Technology in Education*, 77–101. Cappelen Damm Akademisk/NOASP. <https://doi.org/10.23865/noasp.108.ch3>. (2020)
24. Van Nort, D., Pauline, O., Braasch, J.: Electro/Acoustic Improvisation and Deeply Listening Machines. *Journal of New Music Research* 42 (4): 303–324. <https://doi.org/10.1080/09298215.2013.860465>. (2013)
25. Waters, S.: Touching at a Distance: Resistance, Tactility, Proxemics and the Development of a Hybrid Virtual/Physical Performance System. *Contemporary Music Review* 32 (2–03): 119–134. <https://doi.org/10.1080/07494467.2013.775818>. (2013)

## Estimating Interaction Time in Music Notation Editors

Matthias Nowakowski<sup>1</sup> and Aristotelis Hadjakos<sup>1</sup> \*

Center for Music and Film Informatics (CeMFI),  
University of Music Detmold, Germany  
matthias.nowakowski@hfm-detmold.de

**Abstract.** Modern music notation software is extensive and so can be a comparative analysis. Since they employ a lot of different interactions to write scores, due to the mass of different symbols and their combinations, we developed a method to estimate the time spent by the user in interacting with the software interface in order to perform fundamental operations. For this we applied and extended the Keystroke-Level Model by analyzing interaction percentages in MusicXML files. Our findings contribute to modeling interaction and usability/ user experience research about interaction in music notation editors. These findings can be then transferred to analyze other editors and we expect to use the method in formative analyses to reduce user studies and thus development time in the long run.

**Keywords:** Human Computer Interaction · Music Notation Editor · Keystroke Level Model

### 1 Introduction

Score editors allow users to create, edit and play musical scores. They are widely used by composers, musicians, teachers and students for various purposes, such as composing music, arranging songs, transcribing audio, or learning music theory. However, developing score editor interfaces with good usability and user experience is hard. User interface design is a complex task in general, as evidenced by the documents of the extensive ISO standard 9421 [1], which provides various guidelines for interface design. In the context of score editors, it is necessary to identify the needs and context of use for the score editors, to specify the design criteria as well as functional and non-functional requirements, to produce design solutions, create prototypes and test them with users or experts. Implement the design solutions and evaluate the use of the score editors in real or simulated situations.

---

\* We would like to thank Claudia Cecchinato, Árpád Kovács and Juan Sebastian Mora Lopez for creating the KLM encodings and giving valuable input for task description and selection.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Another important challenge is media specificity by which we mean the purpose for which a software is created and by which means the (editable) medium can be accessed. For example, a word processor is created for the purpose of writing and editing text documents. In the case of text editing, consensual interactions and modalities with mouse and keyboard were established which are based on the metaphor of the type writer. These interactions are familiar and so feel intuitive to most users. Transferring the input modality to another medium—namely sheet music, music creation and music editing in general—may cause conceptual dissonance simply because of mismatch of input and output symbols and gestures. This means that users may find more difficult to match interactions that rely on a text-based keyboard to interact with musical symbols. In lack of consensual metaphors, it is symptomatic that widely used notation editors such as Sibelius, MuseScore, Dorico and Finale often employ vastly different interaction paradigms combining mouse and keyboard interaction in idiosyncratic ways. We could already show that standard questionnaires yield mostly low ratings for usability and user experience [2]. Moreover, we found in that study that most users used the score editors mainly for practical purposes and tend not to require features that are specifically supporting the creative process.

By visualizing a distribution profile of task-specific interaction times, we can gain insight into how long different tasks take and which ones are particularly time-consuming. With this approach we analyze six widely used score editors as well of our web-based score editor for Learning Management Systems, which uses the Verovio<sup>1</sup> engraving packet to render musical scores.

## 2 Related Work

To our knowledge there is no systematic and comparative work which deals with usability and user experience of notation editors. Nevertheless, Human Computer Interaction (HCI) is a prevalent topic in music research especially in the context of New Instruments for Musical Expression (NIME) [4] and Education [5]. Dealing with music production specifically Nash et al. [6][7][8] were interested in the creative involvement of the user with trackers and sequencers based on the concepts of Cognitive Dimensions [9] and Flow [10]. Although score editors are not explicitly analyzed they employ the same feedback loops as trackers, albeit by using different symbols. Based on this we also applied these metrics to analyze characteristics of usage of different score editors and isolate significant items by which they can be best described and most efficiently assessed [2]. Peterson et al. [11] approached the quality of creative outcome in digital media and on paper respectively by measuring interaction times.

A good starting point to research interaction times in general is to refer to the earliest of HCI research relating to text editors which can be adjusted and applied to music notation editors which we could view as text editors with special requirements. First wave HCI methodology in the 1980s was concentrated on operations which could be modeled as simple reactions in order to operate a system, ignoring such factors as emotions or the personality of the user. Card et al. [3] introduced the Keystroke-Level model which

---

<sup>1</sup> <https://www.verovio.org/index.xhtml>



was intended to model expert user interactions with text editing software on a low level, consisting of operations such as “Keystroke”, “Button Press” and “Pointing”. However, mental operations were also introduced and with higher complexity of software core tasks had to be defined [12] to focus on the most relevant interactions, i.e. tasks which every software with the same purpose should have implemented. We will discuss how we defined these operators and core tasks for our research in Section 3.

In contrast to this, the GOMS (Goals, Operators, Methods, Selection rules) Model tries to explain a users behavior from a top-down perspective. It takes the actual goal of the action into account and fragments it into the actions that have to be taken to accomplish it [13]. This is useful in analyzing the procedural knowledge users and why they might use a certain interaction path. It is also useful to model new tasks around the given goals, since it is not based on an existing system [14].

Today the Keystroke-Level model remains a viable tool for fundamental research with new input modalities and situations, e.g with touch screens [15][16], in virtual reality [17], device interaction while driving [18] or exploring interactions with non-western writing systems [19]. Of course, the list of operators was adapted, where necessary to accommodate for new input devices and gestures [20].

### **3 Method**

#### **3.1 Program Selection**

To decide for which music notation editors to compare we referenced to a previously conducted study in which we analyzed the most used ones [2]. From the 29 mentioned programs six were viable for statistical analysis, being Capella, MuseScore, Dorico, Finale, Sibelius and Lilypond. For the paper at hand we were not able to make a KLM analysis for Lilypond, since it is entirely text based and cannot be adequately compared to graphical user interfaces (GUI) we implicitly had in mind for the study. Since MuseScore had a major update during the preparation of the data, we also decided to integrate the KLM of MuseScore Versions 3 and 4, giving us the opportunity to discuss recent changes in their interaction design.

#### **3.2 Data Collection & Evaluation**

First we agreed on a set of unit tasks, meaning any atomic tasks that can be accomplished with a music notation editor. Encoding these tasks then was then performed by three people according to Card et al. [3]. We did not expect the encoders to know every program, but they must have worked with music score editors in the past. We are aware that each encoder might have more experience with a certain program and to ensure an average view on the multiple interaction paths we have taken the following measures:

- Multiple people explored each software for the same unit task. This accounts for different ways to solve a task in the case the software has different paths.
- Encodings were taken for different modalities, i.e. major keyboard and major mouse use respectively. This represents people with different ways of working. Although many people might prefer a mix of both, we have a potential range and a basis to interpolate between those values.

- The resulting times of all encoders were averaged for each software and modality.

The KLM provides encodings and already fixed times for series of actions (operators) as “methods” that are necessary to perform a certain task. The operators can be mental preparation (M), keystroke (K), button click (B), homing (H), pointing (P) and selecting from a pull down menu (pd) in different combinations which are empirically determined and applied in our study. The encoders worked on their own computers. We regard the variability of screen sizes as negligible, since the KLM already provides times which are insensitive to this factor.

We also aim to incorporate actions involving multiple inputs that lead to valid changes in the score, such as composing compound elements like tempi or chords. To achieve this, in the following section we calculated average sequence lengths, which we utilized as factors for encoding interaction methods.

As we cannot anticipate every potential context in which a task might arise, the encoding process may result in tasks being coded with slightly slower execution times than they would exhibit in actual scenarios. For instance, consider the scenario where a specific palette must be accessed before adding an articulation, and typically, the palette remains open when multiple articulations are added in sequence. However, in our encoding approach, the act of opening the palette is always included. Consequently, it’s important to acknowledge a margin of error, which could extend up to 20% according to previous research [3].

### **3.3 Task Selection**

KLM describes existing systems on a single level by taking inventory of interaction durations and so making tasks comparable between systems. Since not all tasks are used with equal frequency, we decided to perform four steps that helped us to access interaction times for relevant tasks:

1. Define all unit tasks that are found in at least one music notation editor.
2. Analyze MusicXML data to find frequencies of all elements that result from interactions.
3. Apply the frequencies from step 2 as weights to compute distributions for all unit tasks.
4. Filter unit tasks with the help of step 2 that account for 95% of interactions. Include interactions that are necessary to write a valid music score to get a more manageable number of tasks to discuss. All these tasks we will be denoted as “core tasks”.

In total we defined 234 unit tasks first. These are actions that lead to a visual and/or sound change in the GUI (including score and menus). This effectively filters out all subordinate system interactions which only indirectly contribute in visual outcome. Unit tasks do not have to be solely tasks that change sound events such as notes, chords or articulations. This can be annotations of every kind, as well as lyrics, but also the act of selecting elements, since they add highlighting to the score, and playing the music which adds automatic highlighting to the currently sounding events.

According to Roberts et al. [12] core tasks consist of a cross product of the following operations and objects as seen in Table 1. We had to do some accommodations

for musical syntax, since some operations like “transpose” have different meanings in music. We also omitted “swapping”, “splitting” and “merging” for which we found no scenarios in the described GUIs. Also the number of objects is much larger than in linguistic text, so that we had to group symbols in a similar hierarchical manner (Table 2).

**Table 1.** Operations and objects according to [12].

<i>Operations</i>	<i>Objects</i>
insert	
delete	
replace	character
move	word
copy	line
transpose ( $\approx$ swap)	sentence
split	paragraph
merge	section

**Table 2.** Adjusted operations and objects for music notation editors.

<i>Operations</i>	<i>Objects</i>
add	
delete	
replace	primitives (notes, rests, lines, clefs, marks, etc.)
move/displace	diacritic signs (beams, articulations, ornaments, etc.)
rebind	compounds (chord, measures, key signatures, tempo, etc.)
copy	semantic structures (parts, voices, lyrics, annotations, etc.)
paste	

Some operations are only applicable to some objects such as transposing can only be applied to chords and notes while rebinding (bind an anchor to a new event and so making also a change in the synthesized sound) is mostly associated with elements which modify the sound on larger time scales such as crescendo/ decrescendo, tempi, slurs, dynamics, etc.

To get a more concise view of the frequencies of occurrence of all elements we analyzed freely available MusicXML files by simply counting the elements and mapping them to their corresponding tasks. We also counted sequences of inputs to account for unit tasks that require multiple consecutive inputs, like writing a sequence of notes with the same duration, writing a chord, writing chord symbols or textual tempo instructions. For durations this value lies at 1.4, word length is 5.7 on average. The mean of all found sequences in the analyzed pieces is 3.6 which we will use as a multiplier to compute individual task related times.

As a base for our model we took four pieces from different time periods, with different instrumentation to cover a wide range of quantities of used symbols:

- Johann Sebastian Bach: Orchestral Suite in D Major (BWV 1068)  
20897 elements
- Wolfgang Amadeus Mozart: Clarinet Concerto in A Major (KV 622)  
81844 elements
- Frédéric Chopin: Three Waltzes (Op. 64)  
13209 elements

- Frederik Pfohl: Symphonic Phantasy for great Orchestra *The Sea*, Movement 5 *Frisian Rhapsody* (PWV 24)  
92519 elements
- Gabriel Fauré: Piano Quintet No. 2 (Op. 115)  
75591 elements

Table 3 shows the most used elements in the MusicXML that account for  $\approx 95\%$  of interaction according to the weighted means. These percentages are representing the weights which we will apply to the tasks resulting in the distribution in Figure 1.

In the table “type” is referring to the symbolic duration (quarter, 16th, etc.), which by itself accounts for 35.72% of interactions in a notation program, followed by pitch with 29.25%. Slurs and articulations are child elements of “notations” element and can include further symbols that modify the note such as ornaments, arepeggios etc. “Dot” represents the prolongation of a note.

We decided to base our evaluation on the weighted mean, since we have wide differences in element numbers per piece. By this we assume that the selected pieces are somewhat representative for scores produced for music of this period.

The ranks of the elements for each piece follow the ranks of the arithmetic and the weighted mean in general. The arithmetic mean has some shifted numbers, only the ranks for “slur” and “accidentals” are swapped. Higher shares of accidentals are found for Chopin, Fauré and Pfohl, whose pieces may include extended harmonic development which can likely occur in 19th century pieces. Rests have higher percentages in the large orchestra pieces (Mozart and Pfohl), where entire instruments could stop playing for long times which results in a relatively high standard deviation of 4.8%.

**Table 3.** Percentages of MusicXML elements that are found in all of the analyzed pieces and account for  $\approx 95\%$  of the interaction with the score. sd = standard deviation, mad = mean absolute deviation, wt ... = weighted ...

<i>element name</i>	<i>BWV 1068</i>	<i>Chopin Op.64</i>	<i>Fauré Op.115</i>	<i>PWV 24 (Mvt 5)</i>	<i>KV 622</i>	<i>mean</i>	<i>sd</i>	<i>mad</i>	<b><i>wt mean</i></b>	<i>wt sd</i>	<i>wt mad</i>
type (= symbolic duration)	38.98	41.24	34.01	34.54	36.92	37.14	3.03	3.52	<b>35.72</b>	1.96	1.77
pitch	33.88	39.44	29.76	27.49	27.94	31.70	5.01	3.37	<b>29.25</b>	2.83	1.20
rest	5.10	2.03	5.68	13.20	11.98	7.60	4.78	5.42	<b>9.73</b>	3.75	3.74
beam	14.16	2.99	7.06	1.63	7.32	6.63	4.89	6.03	<b>5.70</b>	3.50	3.41
slur	0.58	2.94	5.16	4.72	5.48	3.78	2.04	1.12	<b>4.67</b>	1.27	0.51
accidental	1.38	5.65	7.93	2.82	2.39	4.04	2.69	2.13	<b>4.08</b>	2.44	0.83
articulations	0.22	0.11	0.99	8.06	3.73	2.62	3.37	1.30	<b>3.98</b>	3.09	4.15
dot (= prolongation)	2.94	0.91	3.22	2.86	1.92	2.37	0.95	0.52	<b>2.60</b>	0.63	0.44

However, these elements do not encompass the entirety of essential functionalities found in music notation editors. The editor must include specific features for initializing and managing information necessary for reading and playing from a score, as these aspects are imperative for its validity. We have meticulously selected these features and refer to them as “essential tasks”. Table 4 summarizes the number of relevant tasks over the 12 most central unit task areas according to the combination scheme of operations and objects mentioned in Table 2 and Table 3.

**Table 4.** Number of core tasks accounting for most relevant interactions in music notation editors.

<i>unit task area</i>	<i>correspondences in XML elements</i>	<i>number of core tasks</i>
duration	type, dot, rest, chord	16
pitch	pitch	6
accidental	accidental	5
beam	beam	3
notations	slur	5
	articulations	6
initial score configuration		2
time signatures		5
key signatures	essential tasks comprising various compounds of XML elements	6
tempo		7
clefs		6
playback		1
staff/ measure		11
<i>tasks total</i>		101

## 4 Results

### 4.1 General

In Figure 1 we show all accumulated KLM values that we encoded for all accessible functionalities, separated by using (if possible) only keyboard or only mouse. It is not surprising that mouse interaction is much slower than pure keyboard interaction in general. The distributions are already weighted according to Table 4. Mouse modality has less outliers in general which points to more equally distributed data. The violin plots now mostly remind hi-hats, meaning that interaction times of core tasks cluster around different regions with few tasks in between. Despite some variations between the graphs one can clearly identify peaks in the lower portions which mostly represent core tasks. Tasks that could be subsumed under “notations” as well as pitch and duration related tasks have usually similar speeds within the software and modality and so forming distinguishable peaks.

The descriptive statistics in Table 5 show that most of the tasks are performed in very similar speeds. Overall Sibelius is slowest for mouse interaction with 6.89 seconds. MuseScore4 is the fastest in key interaction with 3.22 seconds. Dorico, Finale and MuseScore 3 are significantly faster in mouse interaction than the rest. In general most tasks are performed in between 3 to 9 seconds.

Comparing MuseScore 3 and 4 we can see, that the later Version tends to make some interactions slower especially with mouse interaction. Pitch and duration related tasks are clearly visible in the peaks. For mouse interaction the times in both editors are similar, but MuseScore4 having a higher median despite having a similar interquartile range. This indicates that non-essential tasks have become faster, which are not heavily weighted. Comparing the peaks around 7 seconds with Sibelius we can find mostly tasks for “notations” like in MuseScore 3 and 4 but Sibelius also includes many tasks about various changes about staves and element displacement which is usually faster in other editors. Many similar peaks over a wide range resulting in a mostly symmetric

**Table 5.** Descriptive statistics for Figure 1.  $q1$  = first quartile,  $q3$  = third quartile,  $iqr$  = interquartile range,  $mad$  = median absolute deviation,  $sd$  = standard deviation,  $se$  = standard error,  $ci$  = 95% confidence interval.

<i>software</i>	<i>modality</i>	<i>min</i>	<i>max</i>	<i>median</i>	<i>q1</i>	<i>q3</i>	<i>iqr</i>	<i>mad</i>	<i>mean</i>	<i>sd</i>	<i>se</i>	<i>ci</i>
Capella	key	2.08	13.58	3.56	2.96	4.46	1.51	1.3	4.14	1.66	0.01	0.01
	mouse	2.55	14.89	6.86	5.34	8.1	2.75	1.84	6.6	2.17	0.01	0.02
Dorico	key	1.95	15.17	3.58	2.95	4.01	1.06	0.74	3.81	1.33	0	0.01
	mouse	1.95	15.17	4.46	3.75	8.01	4.26	1.48	5.88	2.56	0.01	0.02
Finale	key	1.75	16.26	3.91	3.7	4.66	0.96	0.78	4.24	1.08	0	0.01
	mouse	2.52	19.18	4.94	3.76	8.32	4.56	1.76	5.96	2.85	0.01	0.02
MuseScore 3	key	1.68	15.62	3.47	2.86	4.04	1.19	0.85	3.64	1.15	0	0.01
	mouse	2.42	16.36	4.93	3.75	7.86	4.11	2.79	5.96	2.66	0.01	0.02
MuseScore 4	key	1.75	15.15	3.22	2.72	4.15	1.43	0.94	3.54	1.33	0	0.01
	mouse	2.15	14.25	6.58	3.75	7.79	4.04	3.8	5.99	2.51	0.01	0.02
Sibelius	key	1.68	13.67	3.7	3.2	4.2	1	0.75	4.05	1.42	0	0.01
	mouse	2.35	15.92	6.89	4.56	8.67	4.12	3.46	7.17	3.07	0.01	0.02

shape can be an indicator that some core tasks may be inconsistently modeled, also more coherent plots over a wide range can show special treatment of some methods that are less consistent with similar tasks and should be examined in more detail.

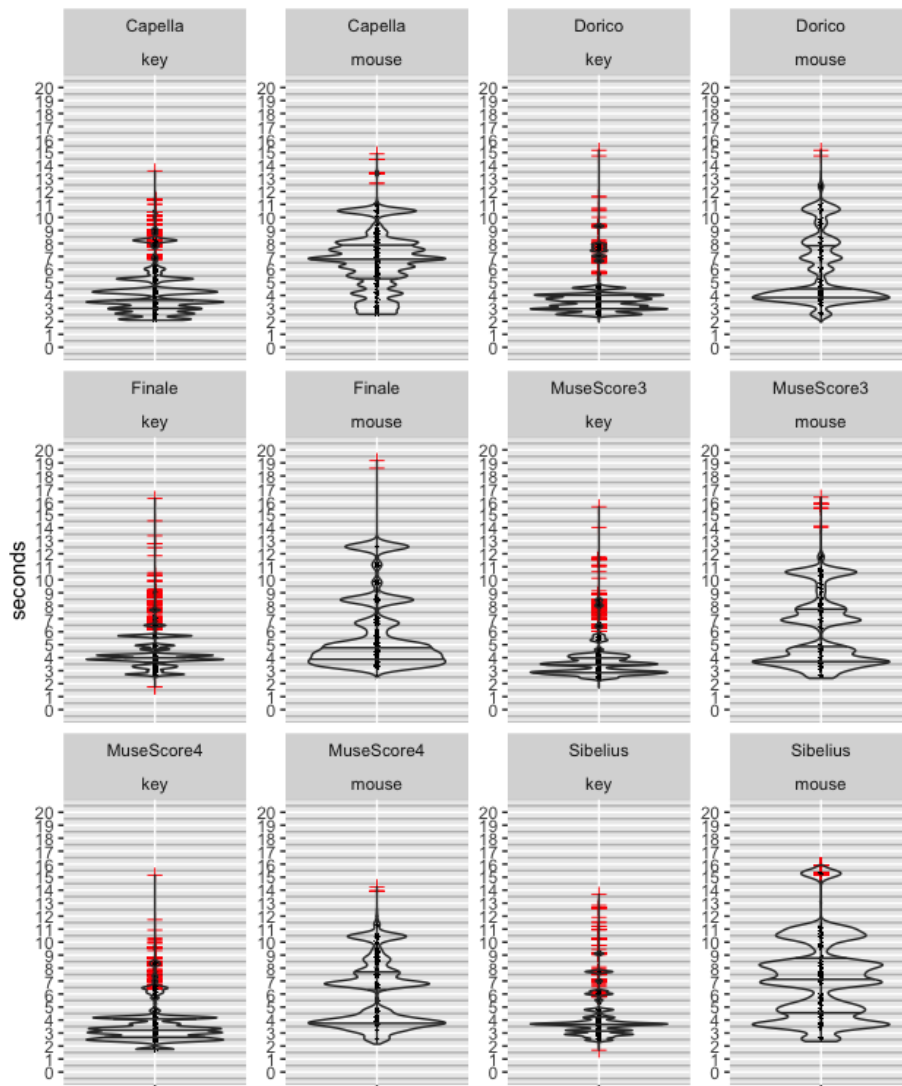
## 4.2 Outliers

As a rule it is not problematic having many outliers in the set of interactions. Since most of the editors have low third quartile boundaries in the key modality, interactions with lengths of 5 to 7 seconds can already count as outliers in these cases.

From the perspective of the software designer this might point to concentration on faster speeds in interaction design and addressing specific problems that have to be solved in order to appeal to a certain user group. Also, this does not mean, that one program is more preferable over the other due to interaction speed differences. As mentioned our previous study [2] Capella has the best usability and user experience ratings in our experiments, while from a KLM perspective there are more peaks in higher regions in both modalities. With this method it is more important to analyze different peculiarities, like for example having mostly core tasks hidden behind slow or dissimilar interactions.

In Dorico we can see, that the data is much wider distributed using a mouse than using solely keys. Only changing the instrument for a specific staff, transfer notes between voices and creating multiple bars at the end were considered to be very slow. In contrast we have 38 outliers for key interaction, most of them including tasks that immediately result in a different layout, especially adding and deleting measures. But we can also find frequently used objects as described in Table 3, like beams and staves, while 101 Tasks can be completed between 3 to 4 seconds.

Finale has a very characteristic peak at 12.5 seconds for the mouse modality, which consist of some layout and MIDI operations. Also there seems to be no simple way to



**Fig. 1.** Violin plots with quartiles of the weighted Keystroke-Level Model of the six music notation score editors. The red '+'-Symbols mark outliers. The width on the x-axis is not

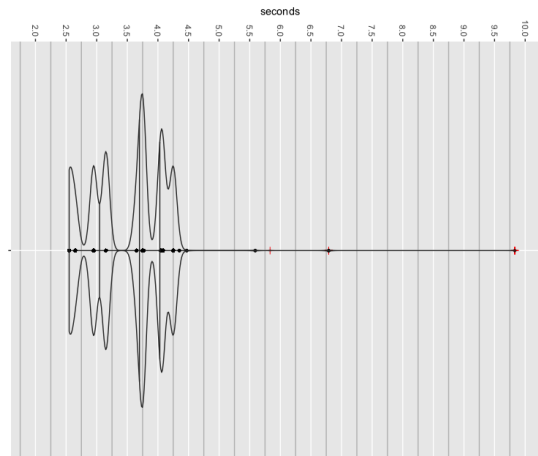
paste notes, chords or rests—objects which are highly weighted—by mouse which in turn is better handled by keyboard interaction.

In general keyboard interactions across all editors are especially slow making major changes to the layouts or creating scores. Most outliers consist of these since there are seldom adequate methods so that here are no options despite using the mouse. These are actions that one might access rarely are not among outliers in any mouse interaction.

This also applies for more fine grained interactions considering changes around staves and notes, like beams, meter changes or barline related tasks like creating repetitions. It is debatable, if fast keyboard access is necessary if such actions account for less than 2% of the total. Outliers in mouse interactions are mostly idiosyncratic and revolve around elements outside the staff like tempi and charts. Here especially Sibelius, MuseScore3 and Finale seem to have deficits.

### 4.3 Application

The results shown above provide guidelines for monitoring the ongoing development of our music notation interface called **VIBE** (Verovio Interface for Browser-based Editing)<sup>2</sup>, as well as for assessing its performance in comparison to other solutions. While the method presented above entails a summative analysis, we are confident that it can also be adapted to formative scenarios. These scenarios can then be employed at different stages throughout the development process.



**Fig. 2.** Violin plot of VIBE. The red '+'-Symbols mark outliers.

VIBE currently implements 41 of the 101 listed core tasks (see Table 4), most of them use mouse interactions which are better explorable visually when using the program for the first time. Mostly “notations” and “dynamics” have to be implemented yet, as well as several actions of copying, pasting and rebinding. Main development dealt with actions around adjusting durations, interacting with the identity of a note directly and creating a valid score. Additionally we were also interested in handling annotations and chord symbols since these are important features for analyzing a score and make information accessible for other persons in a teaching environment, as required by the

<sup>2</sup> Source Code: <https://github.com/mnowakow/VerovioScoreEditor>  
Demo: <https://mnowakow.github.io/>



underlying project for which it is developed. This added possible 14 unit tasks of which 10 are implemented, combining to 51 implemented unit tasks in total.

In Figure 2 we can see, that most of the interactions are around 3.75 seconds (with the median at this point and a very narrow third quartile) which belong to the relevant core tasks handling durations and pitch. Although we concentrated on mouse interaction first, the results can keep up with other editors sometimes even with fast keyboard inputs. In our case especially creating time signatures is slow, since it is currently required to choose always from two drop down menus to create a combination of count and unit which then has to be dragged to the intended position. Generally actions that include dragging and dropping items (clef, key, time) are found among the outliers, as well as actions which have to be performed multiple times to accomplish the intended result, like deleting or adding multiple measures at the end of the score.

## **5 Discussion**

In this paper we presented an approach to evaluate music notation editors objectively by simulating and comparing their interaction times. We oriented our research on the original publications about the KLM and applied them to editors by several annotators. By defining unit tasks and model their weights after element occurrence in MusicXML we can find slow and fast interactions and especially locate relevant ones in the resulting distributions. Some speeds usually point to similar sequences of operators. Horizontally symmetric plots over a wide range might point to inconsistencies in the interaction modeling. This helps us to evaluate different music notation editors and model interactions in new interfaces according to access times. These times do not represent rigid metrics but a way do identify potential shortcomings fast.

By listing and ranking core tasks, this paper also contributes to monitor the process of development and functional completeness of notation editors as shown in section 4.3. New editors will at least have to implement the core tasks presented here, but might have different requirements for working more creatively or making elaborate editions. In these cases the list of unit tasks can be extended as presented in section 3.3. KLM is flawed when it comes to evaluating user responses. In our case we approached the topic by modeling methods which are not informed by the user manual, but by exploration and restriction to input modalities which might represent a user with average skills. We did not expect a perfect and efficient user, but did assume a perfectly set score. So still questions remain about the system and user behavior in case of errors: How fast can users correct their errors? What methods does the system provide to make corrections? That is why most research using KLM is concerned with user tests to verify interaction times with new interaction modalities such as touch, pen or VR. In our case, we adopted an established model, as we focused on mouse and keyboard interactions. We then extended its application to a domain within HCI that has received limited scientific attention until now. However, since we base our method on informed assumptions, we still would like to verify these results with actual user tests. This will also help us to bring the results from this paper closer to the field of user experience. When conducting user tests it will be also fruitful to combine it with discussions and questionnaires to evaluate specific usability issues, which could not be represented by the KLM directly.

## References

1. Bevana, N., Kirakowski, J., Maissela, J.: What is Usability? Proceedings of the 4th International Conference on HCI, pp. 1–6 (1991)
2. Nowakowski, M., Hadjakos, A.: Online Survey on Usability and User Experience of Music Notation Editors. Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR'23) (2023), in press
3. Card, S. K., Moran, T. P., Newell, A.: The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM*, 23(7), 396–410 (1980)
4. Fasciani, S., Goode, J.: 20 NIMES: Twenty Years of New Interfaces for Musical Expression. Proceedings of the International Conference on New Interfaces for Musical Expression, (2021), <http://doi.org/10.21428/92fbeb44.b368bcd5>
5. Reppenning, A., Basawapatna, A.R., Escherle, N.A.: Principles of Computational Thinking Tools. Emerging Research, Practice, and Policy on Computational Thinking. *Educational Communications and Technology: Issues and Innovations*. Springer, pp. 291–305 (2017)
6. Nash, C., Blackwell, A.: Tracking Virtuosity and Flow in Computer Music. Proceedings of the International Computer Music Conference (ICMC) (2011)
7. Nash, C., Blackwell, A.: Liveness and Flow in Notation Use. Proceedings of the International Conference on New Interfaces for Musical Expression (NIME) (2021), [http://www.nime.org/proceedings/2012/nime2012\\_217.pdf](http://www.nime.org/proceedings/2012/nime2012_217.pdf)
8. Nash, C., Blackwell, A.: Flow of Creative Interaction with Digital Music Notations. Oxford University Press (2014), <https://doi.org/10.1093/oxfordhb/9780199797226.013.023>
9. Green, T., Petre, M.: Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. *Journal of Visual Languages Computing*, vol. 7, no. 2, pp. 131–174 (1996)
10. Csikszentmihalyi, M.: Flow and the Psychology of Discovery and Invention. Harper Perennial, vol. 39 (1997)
11. Peterson, J., Schubert, E.: "Music Notation Software: Some Observations on its Effects on Composer Creativity, Proceedings of Intercational Conference on Music Communication Science (ICoMCS), vol. 127-130 (2007)
12. Roberts, T. L., Moran, T. P.: The Evaluation of Text Editors: Methodology and Empirical Results. *Communications of the ACM*, 26(4), pp. 265–283 (1983)
13. Card, S., Moran, T., Newell A.: Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology*, Volume 12, Issue 1, p. 32–74 (1980)
14. Kieras, D., Butler, K.: Task Analysis and the Design of Functionality. *The computer science and engineering handbook*, 23, pp. 1401–1423 (2014)
15. Holleis, P., Otto, F., Hussmann, H., Schmidt, A.: Keystroke-Level Model for Advanced Mobile Phone Interaction. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07). pp. 1505–1514 (2007), <https://doi.org/10.1145/1240624.1240851>
16. Abdulin, E.: Using the Keystroke-Level Model for Designing User Interface on Middle-sized Touch Screens. CHI'11 Extended Abstracts on Human Factors in Computing Systems, pp. 673–686 (2011)
17. Guerra, E., Kurz, B., Bräucker, J.: An Extension to the Keystroke-Level Model for Extended Reality Interactions. *Mensch und Computer*, (2022).
18. Pettitt, M., Burnett G., Karbassioun D.: Applying the Keystroke Level Model in a Driving Context. *Contemporary Ergonomics 2006*, Taylor Francis (2006)
19. Myung, R.: Keystroke-Level Analysis of Korean Text Entry Methods on Mobile Phones. *International Journal of Human-Computer Studies*, 60(5-6), pp. 545–563 (2004)
20. Al-Megren, S., Khabti, J., Al-Khalifa, H. S.: A Systematic Review of Modifications and Validation Methods for the Extension of the Keystroke-Level Model. *Advances in Human-Computer Interaction*, 1–26 (2018)

# Human-Swarm Interactive Music Systems: Design, Algorithms, Technologies, and Evaluation

Pedro Lucas and Kyrre Glette

RITMO Centre for Interdisciplinary Studies in Rhythm, Time, and Motion  
Department of Informatics  
University of Oslo  
Oslo, Norway  
pedroplu@uio.no

**Abstract.** This paper presents considerations for developing Human-Swarm Interactive Music Systems (IMS), based on previous work in the field. We discuss design principles, algorithms, technologies, and evaluation methods for creating user-centred Human-Swarm IMSs using architectural approaches, swarm strategies, and levels of embodiment in implementation. Our contribution aims to establish a framework for future applications and research studies on swarm-based music platforms.

**Keywords:** Interactive Music Systems, Digital Instruments Design, Swarm Intelligence, Multimodality

## 1 Introduction

Sending, processing, and response are three stages that form a concise and straightforward model to represent Interactive Music Systems (IMS). However, the different contexts in which an IMS can be developed give rise to several levels of complexity, demanding a critical cross-disciplinary investigation. This expands the model to more concrete representations and design considerations for innovative applications [9].

This paper focuses on a specific instance of an IMS related to a Human-Swarm system. This type of IMS refers to improvisational systems that allow a user to interact with a swarm of artificial agents that are self-organized (working locally without a central controller) and exhibit emergence (interaction between agents in the swarm produces higher-level patterns and structures) [32]. These and other properties are commonly based on the theory of *Swarm Intelligence*, which can be found in nature and has been modelled in computational simulations.

This type of IMS is important in its potential to develop various levels of representation of sonic and/or musical units, ranging from micro sounds for granular synthesis to the embodiment of individual artificial musicians capable of collaborating with human performers to achieve complex music improvisations.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Depending on the levels of representation, modelling an IMS as a Human-Swarm system can have benefits. In the case of music improvisation, musical elements can be highly interactive and uncertain. Therefore, swarm strategies are a good fit for a process that can reproduce such behaviour for real-time music composition [1]. Additionally, embodied representations of artificial musical agents with the role of additional musicians can lead to collaborative and enjoyable human-machine experiences [14].

To advance the development of Human-Swarm systems, we contribute with the proposal of a framework that includes four relevant areas: *design considerations, algorithms, technologies, and evaluation methods*. This proposal is based on previous work on swarm intelligence applied to IMS, theoretical explorations of multi-agent systems that use swarms in music, and musical agents. A thematic analysis approach was used to extract information from these works, having these four areas as central themes. Our contribution is intended to support applications and research studies concerning music platforms that use swarm approaches, as well as provide a foundation upon which creativity can be effectively channelled.

This paper is organized as follows: Section 2 presents a background of Human-Swarm IMSs. Section 3 presents the framework including a detailed discussion focused on the four areas mentioned above. Finally, Section 4 provides conclusions and future directions for Human-Swarm IMSs.

## **2 Background and Related Work**

The interest in musical interaction with artificial swarms began with Blackwell and Bently's work [2], where they proposed the first application of swarm intelligence to music. They related music features to swarm descriptors, such as attraction and repulsion, suggesting that improvised music is a self-organized system that can lead to complex musical structures. This self-organization is carried out by local interactions between individuals and the environment, which can be direct or indirect. Indirect interactions are mainly focused on in some works [4] [1] [32], considering the concept of *Stigmergy*, which is a mechanism that manifests when an individual modifies characteristics of the environment so that other individuals respond to it later.

In most swarm applications, the elements that participate in self-organization interactions can vary in terms of the size of the musical material. Blackwell [1] presented a classification based on perceptual time-scales, which can be seen as musical material elements organized by size. The elements as events are: *micro* (small-scale times like tenths of a millisecond), *mini* (musical notes or sound objects), *meso* (phrases or groups of mini-events), and *macro* (time encompasses form and lasts several minutes or more). This classification is also useful to determine the level of embodiment that the agents from swarms can have regarding their interaction with human performers, which is reflected in the works described below.

The usual strategies that utilize swarm intelligence in music systems are focused on mappings of sonic or musical features over spatial properties in swarms. The musical interaction is given by the swarm dynamics, which commonly has led to interactive solutions in which the agents from the swarm are hidden elements with a low embodied perception. This concept is portrayed by *Swarm Music* [2] [4], which is based

on flocking algorithms and a process of capturing, updating, and interpretation so that users can modify the dynamics of the swarm for influencing the musical input. Another relevant work is *Musebots* [6], which explores the concept of Musical Metacreation (MuMe) related to the automation of aspects regarding musical creativity to model a musician more than an instrument, and thus closer to working on music improvisation. A higher embodiment can be achieved through visual feedback and gestures in a 3D environment to display agents, as the work of Unemi and Bisig [30], which shows an interactive installation where the user acts as a conductor for influencing flock's musical activity; moreover, agents can also perceive aspects of musical outputs and operate in a 3D space as virtual sound sources, as shown in [7] and [23]. Physical implementations develop mappings with spatial or sonic properties from entities as robots, as described in works such as [31], [33], and [13]. Other approaches include using quantum physics simulations [16] and physical-virtual environments that portray full embodiment with agents as musicians [14].

Theoretical frameworks that support swarm applications have been explored for Human-Swarm IMS. In this case, we have the concept of *Musical Agents*, which are entities as computer programs that generate music autonomously or in collaboration with human musicians [25]. These entities can be part of Multi-Agent systems, such as the *Virtual Musical Multi-Agent System (VMMAS)* [34] and the *Mobile Musical Agents* project based on the *Andante* project, which deals with musical agents that decide to migrate and react to changes in the environment [29]. Architectures under these theoretical structures have been proposed, such as *MAMA* [19] [18], which is grounded on the theory of communicative acts and enables agents to reason about intentionality, or the *MASOM* architecture [24] that works with *Self-Organizing Maps* based on musical agents, that has been used in works such as REVIVE [26] [27], and Spire Muse [28]. Additionally, an approach that involves improvisation with human interaction was elaborated and presented as a concept called *Live algorithms* [3] for representing analysis, process, and synthesis modules for IMSs in the human-machine domain.

When it comes to Human-Swarm interaction and collaboration, it is essential to consider how agents can work together, which can be achieved through *negotiation behaviours* to satisfy the interests of the individual agents, such as in [10]. Synchronized works, as in the case of those based on pulse-coupled oscillators inspired by fireflies and implemented as fireflies [21] [20], or self-synchronization with percussive robots that achieve equilibrium [13], are also examples of collaboration. Interaction and collaboration can be conceptualized in terms of influence and motion, as seen in the system *Swarm Lake* [12], which also uses a game development approach for its design and considers environmental features to conceptualize a theme in a hypothetical world presented to the user. Moreover, it is possible to have higher levels of control for swarm collaboration considering *swarm dynamics* (e.g. swarm-wide; that is, control over a group more than an individual) instead of *direct control* of sound parameters (e.g. audio volume). Control regarding swarm dynamics is present in most related works and significantly affects the resulting music [32].

In summary, most of the previously cited works that involve systems use a swarm representation to map sonic or music features, which can be based on different musical material sizes. The complexity for some of them rises in a final musical piece that

can be achieved in an improvisation musical session together with a human performer, but others can reach a higher level and become actual artificial musicians interacting with each other and with the user, which demands more sophisticated ways to develop and represent agents. We are mainly interested in this last type of system to remark the embodiment of agents in a swarm, but without discarding the possibility of building solutions with more abstract representations for lower levels of embodiment. As the human is part of the system, this work intends to provide means to increase the understating of a swarming process in human-machine music performances.

### 3 Human-Swarm IMS Framework

The section presents a framework to enhance the creation process of a Human-Swarm IMS. The developer can start to look at the general considerations described below to create a unique solution, then specify the architecture to use and check if the solution complies with the swarm design properties listed later. Moreover, the sound generation can follow mapping strategies according to the nature of the designed swarm, and suitable algorithms can be implemented to support that design. Finally, the technologies to choose would depend on the design and the available resources.

All these considerations are presented and discussed below.

#### 3.1 Design Principles

The design and development of IMSs have been explored in a variety of works for several years [8] [9] [15], emphasizing user interaction, system design, and mapping strategies. In this work, we want to provide a more specific scenario for Human-Swarm IMSs which have used implicitly or explicitly the design approaches explored before. In consequence, we present in this section a set of design principles based on previous work related to Swarm Intelligence applied to IMSs.

##### 3.1.1 General Considerations

The following sections focus on specific considerations regarding architectures, swarm design, and sound mappings. On top of this, other considerations are recommended to develop a Human-Swarm IMS as illustrated commonly in literature, such as:

**-Idiosyncratic Approach:** Design is mostly a personal choice [1], and that is reflected in IMSs that want to achieve specific goals which are recommended to be primarily related to artistic intentions and creative process more than technological-driven motivations. This is also called a practice-driven approach [17]. However, guidance in this process is relevant for a solid structure that supports those personal choices, and this paper intends to suggest such guidance.

**-Representation and Dynamics:** Two significant decisions are required to design a swarming system: *representation* and *dynamics* [1]. The *representation* has to do with inputs and outputs and how they are processed, and *dynamics* is the swarm algorithm that interacts with the representation. These decisions are based on an architectural approach that is explained in Section 3.1.2.

**-Novelty:** We can achieve novelty through self-organized approaches considering three aspects: music representation, music style definition, and music style evolution [11]. These aspects can be explored in the results obtained from the system. Finding ways to have a fast switch between instances of these aspects helps to fine-tune our musical intentions.

**-External Inspiration:** The design of a Human-Swarm IMSs can approach several levels of embodiment, which require integrating multiple disciplines in complex cases. Thus look at other areas such as game design (e.g. *Swarm Lake* [12]) and human-robot interaction (e.g. *Dr.Squiggles* [13]) can enable several possibilities to enhance the experience.

### 3.1.2 Architectures

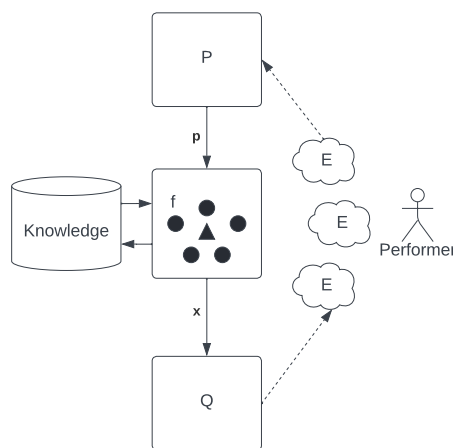


Fig. 1:  $PQf+K$  Architecture. This structure is based on Blackwell's work [1] with the addition of a knowledge base for swarm dynamics.

Works on Human-Swarm IMSs usually depict specific architectures based on the particular problem they are solving. However, especially in theoretical works, there are proposals where system modularization can lead to a clear design base. We depart from the simplest *sending, processing, and response* stages to model an IMS as in [9], which can be seen as the traditional minimized structure *input, process, output*. This model can be expanded to a complex set of units to describe a system to *capture, update, and interpret* information in the environment in which a human performer is a participant [2] [4], and if we want to see them closer to the human process of improvising music, we can portrait them as *perception, cognition, and musical execution* [34].

For Human-Swarm IMSs, we need structures that encourage coexistence between the human performer and the artificial entities, thus considering the models mentioned above, the concept of *Live Algorithms* developed by Blackwell et al. [3] is a suitable choice of representation. A Live Algorithm is “an autonomous music system capable of human-compatible performance... the Live Algorithm listens, reflects, selects, imagines, and articulates its musical thoughts as sound in a continuous process”, hence a Live Algorithm works with collective human-machine musical improvisation. This concept is structurally represented by the  $PQf$  architecture proposed in [1], having  $P$  for analysis,  $Q$  for synthesis, and  $f$  for patterning supporting the two major decisions mentioned previously: representation ( $P, Q$ ) and dynamics ( $f$ ).

We propose to add an explicit module to this representation called *Knowledge* since there are applications that require a knowledge base for the dynamics depending on the algorithm that is being used as in [34] that applies a fuzzy mechanism, or in [19] that

uses a knowledge base for musical agents based on communicative acts. Fig. 1 illustrate this proposal as the  $PQf+k$  architecture.

The knowledge can itself be modelled with high complexity; however, it is sensible to take into account limitations and the trade-off of using a knowledge base in a real-time setup since IMSs are improvisational systems and potential problems like *latency* can affect the user experience significantly.

The advantage of this modularization is the flexibility to change among strategies so that system properties are adjusted in real-time if needed (e.g. change the knowledge base or swarm algorithm in the middle of the performance); that is why a particular emphasis on this architecture is given for the interfacing between modules.

Another useful approach is using a *Finite State Machine (FSM)* to model an individual agent behaviour or the external influences of the human performer, which can exhibit different states when the performer interferes in the environment [23]. Moreover, FSM can help to minimize the complexity of designing multimodal systems, which is relevant, especially for Human-Swarm IMSs that target higher embodiment [5].

### 3.1.3 Swarm Design

Commonalities found in previous work referenced in Section 2 related to Human-Swarm IMSs lead us to propose the following design principles:

- Decentralization:** Even though most swarm systems are developed over a centralized platform, the nature of a swarm should target decentralization, which implies looking for local communication methods and rules between individuals and the environment to portray independence from global management. Inspiration of decentralized behaviours can be found in animal swarms.
- Emergence:** Emergent behaviour allows a swarm to create dynamic and unpredictable musical outcomes. This is also known as *self-organization*, which arises from the collective actions of individuals. The system should allow the emergence of complex and adaptive behaviours from the interactions of individuals, resulting in unique and creative musical compositions that are co-created by the swarm.
- Stigmergy:** As mentioned earlier, this mechanism manifests when an individual modifies characteristics of the environment so that other individuals respond to it later. Modelling stigmergy can be useful for indirect control through the environment and limit direct interaction with agents when it is not entirely possible (e.g. interaction with a swarm of physical drones).
- Scalability:** The design should support the accommodation of various agents, ranging from small groups to large crowds. This feature is the system's scalability in terms of technical infrastructure and user experience, which ensures that the system handles different swarm sizes and that the interaction remains meaningful and enjoyable, regardless of the group size when the design allows it.
- Stability:** For some swarm systems in which agents can fail individually (e.g. each agent can be a physical robot that could potentially withdraw), the musical task should continue with the rest of the participants and the consequences of losing some of them should not impact, at least, the essence of the performance. Consideration of this aspect results in a more stable swarm system.



- Flexibility and adaptability:** Systems should be flexible and adaptable to different musical styles, genres, and contexts, as the goal is music improvisation. The system should allow for customization and configuration to suit different musical perspectives and should be able to adapt to changes regarding the swarm's size, behaviour, or musical preferences over time.
- Time-scale of Material:** Depending on the system's focus, the sonic or musical material in terms of duration can be framed as *micro*, *mini*, *meso*, or *macro*, as described earlier. The solution's complexity level could rise as time increases since more sophistication is required for higher levels like *macro*, which deals with complex musical structures.
- Level of Embodiment:** The swarm individuals can be conceived as mere abstract units that contribute to a musical solution, which can be hidden from the user to a certain extent. However, if these individuals are closer to artificial musicians to collaborate with, it is necessary to provide a level of embodiment that transcends into the spatial domain. In that sense, multimodal approaches through 3D environments are helpful, which could require spatial audio solutions and visualization strategies.
- Environmental Perception and Actuation:** The swarm system should sense the environment to respond accordingly with actions through direct interaction or by stigmergy. Thus it requires defining and designing sensing capabilities according to the level of embodiment and decentralization as well as suitable output mediums. For instance, a robot swarm can be equipped with microphones and speakers for music sensing and actuation in the environment.
- Level of Control:** Human-Swarm IMSs require a certain level of control from a human operator, in which the designer should define how much of this control is provided from a fully manual operation to a completely autonomous system. Allowing a real-time definition of these levels could increase the diversity of the music material produced by human-machine improvisations. Additionally, controls can act over the swarm dynamics, sound parameters, or higher descriptors as commands.
- Feedback and Transparency:** To support decision-making during music improvisation, it is essential that the actions performed by the swarm are *transparent* to the user. This can be achieved through adequate feedback from the artificial agents and any human operators involved in the system. Auditory feedback is particularly important in an IMS, but visualization and haptic feedback can also be useful for confirming actions. However, designers need to be careful not to overwhelm the user with too much information and consider whether certain types of feedback might go against the artistic purposes of the system.
- Accessibility and Inclusivity:** The design can consider an inclusive and accessible system for diverse participants, including individuals with different abilities, backgrounds, and musical skills. If the intention is to cover a wide variety of performers, the design should consider multiple modes of participation and accommodate different levels of physical, cognitive, and musical abilities, ensuring that everyone can participate and contribute to the music-making process.
- Trust:** Building trust with a non-human agent requires calibration between a person's expectations of the agent and the agent's capabilities. Exploration of trust at different levels might significantly enhance the musical result.

**-Room for Failure:** We can design a system with a high amount of constraints, but it could restrict potential interesting results that can emerge from the music improvisational process; thus, to encourage the element of surprise in the results, we can leave some room for failures and user exploration in that context.

Several of these principles overlap and belong mostly to the swarming nature of the solution, which mainly deals with spatial properties.

### 3.1.4 Mapping Strategies

The most common mapping strategies for Human-Swarm IMSs relate sonic or music parameters to spatial properties; for instance, amplitude and pitch from a specific sound sample could be associated with coordinates X and Y of an agent, and music can emerge from the swarming behaviour. These associations can be simple and direct, as the example provided, or use non-linear or probabilistic approaches; it depends on personal choices and the designer's goals.

The previous example considers *swarm-sound/music* mapping; however, the interaction with a user demands establishing *human-swarm* mapping strategies. In that sense, apart from usual ways to feed musical input (e.g. using MIDI controllers), motion capture techniques for gestures, or other sensing solutions, can be used to manipulate swarm parameters to have a *human-swarm-sound/music* mapping; nevertheless, an option of *human-sound/music* mapping can be combined with swarm dynamics depending on the design.

As we deal with swarms, mappings can also focus on the dynamics of collective actions and general descriptors. For instance, as the swarm explores the spatial environment, the centre of mass can be a parameter that influences higher musical features, like the global panning or a general reverb effect, which can also have more complex interaction in terms of the behaviour of every individual, leading to a dense music result.

For certain applications, especially in a physical domain, there could be noises with a significant effect on the sonic result (e.g. motors, propellers, etc., from robot swarms); in that case, we can include these sounds as part of the performance by processing them through mapping strategies that allow their inclusion to the musical result.

Consequently, we can create a rich and engaging musical experience through a mapping design that encourages the participation of all actors and situations while allowing individual expression and creativity from the user, according to adjustable levels of autonomy in the system.

## 3.2 Algorithms

Based on the works listed in Section 2, we can identify common strategies for handling the *input*, the *processing algorithm*, and the *synthesis of sonic output* or other useful feedback, as described below.

**-Input:** The audio stream of a music performance is a typical source of input. It can be analyzed using signal processing techniques to extract features for further usage, such as loudness, pitch, and onsets. Musical material can also be collected directly from

human performers through common interfaces like musical keyboards or traditional instruments. However, complex control mediums like gestures and image recognition require sophisticated capture strategies. In such cases, machine learning algorithms for real-time data collection can be useful for these tasks by applying classification techniques to identify discrete states and regression strategies for continuous values.

**-Process:** Common *Swarm intelligence* approaches use flocking strategies based on the *Reynolds's boids algorithm*, in which agents have attraction and repulsion rules concerning neighbours as well as velocity matching. These rules can be structured on reasoning mechanisms that take advantage of descriptive parameters through algorithms such as *fuzzy logic* or *language processing through communicative acts*. Other proposals consider mathematical models that define acceleration or velocities for the agents' position calculated from local individuals and the performer's spatial features. Additional techniques used in this category include *Particle Swarm Optimization (PSO)*, *Ant Colony Optimization (ACO)*, and *Genetic Algorithms*. However, in some cases, the goal is not to optimize specific parameters but to fulfil musical intentions that take advantage of the algorithm's mechanics. Other strategies, such as *Self-Organizing Maps*, can be used for sound organization and pattern recognition. Music generation through real-time input and pre-loaded knowledge as *Markov Chains*, can lead to interesting results. Synchronization techniques, such as *Pulse-Coupled Oscillators* inspired by the behaviour of fireflies or custom strategies based on the analysis of temporal events in the audio stream, can be applied to rhythm.

Switching between algorithms requires that they share similarities in a swarm. The selection of behaviours determines the overall structure of the swarm, while the weighting of different behaviours affects the current dynamics of the simulation.

**-Output:** The output depends on the mapping between the swarm's spatial properties and the sonic and musical result. Possible mapping strategies include *additive synthesis*, *granular synthesis*, *control based on agents' proximity*, *procedural patching from swarm dynamics*, *modulation synthesis*, and *sound physical modelling*. The choice of mapping depends on the specific musical goals.

The designer can decide the suitable technique to use, and there is plenty of room for applications and research studies regarding algorithms that can be explored at different levels of embodiment, so the user experience has to be taken into account as a centre point of departure to develop a system that characterizes the nature of the musical interaction between human and machine.

### 3.3 Technologies

We classify potential technologies to use into three categories according to their level of embodiment, as described below.

**-Virtual:** In this category, agents exist solely in a virtual environment implemented through software on a central device, such as a computer. Input is received via integrated peripherals, MIDI keyboards, or sophisticated devices such as cameras with image recognition algorithms. Sonic output is played through loudspeakers, ranging from a simple mono configuration to multiple channels for spatial audio. While complexity

can increase in terms of input and output devices, processing remains centralized, and agents are virtual objects that can produce music as a hidden process or with a higher representation visualized on a screen.

**-Physical-Virtual:** This category builds on the previous virtual category, but agents reach a higher level of embodiment by sharing the physical space with the performer and being aware of the real environment. Extended reality technologies, such as mixed reality headsets or augmented reality systems, can support this configuration. For a more immersive experience, it may require additional complexity in terms of motion capture, visualization, and audio playback to portray a virtual 3D world that overlaps the physical space where the performance is happening.

**-Physical:** In this category, agents exist as actual entities, such as robots, which can interact with the human performer. Design principles for human-robot interaction can be applied, and additional considerations such as trust, safety, and treatment of noises are considered. Each agent requires its own input and output capabilities and capacity for local communication, as this category can be approached as a decentralized system.

As technology advances, we can improve the response time for the interaction, integrate better ways to reach transparency, and potentially extrapolate to the participation of larger groups to the performance (e.g. audience with no musical skills).

### 3.4 Evaluation Methods

Evaluation methods have been proposed before, such as in the work of O’Modhrain [22] that describes methodologies depending on the stakeholder and recommends clearly understanding of what to apply and to whom depending on the interest of the study. In that sense, Human-Swarm systems are focused on the *performer/composer* and the *designer*. The following types of evaluations can be considered to assess these systems.

**-Autoethnography:** The designer can evaluate the system by using it and reflecting on the music creation process to improve the design.

**-Observation:** The system can be used by different users in different settings, such as a controlled environment like a laboratory or a concert. The designer can observe the advantages and limitations in those environments to understand how different users can approach the system.

**-System Measurements:** The designer can measure sections of interest in the system to discover limitations that can impact the user experience, such as latency or jitter.

**-Physical and Physiological measurements:** For user studies, data can be captured while participants use the system. Physical data, such as positions in space, can be useful for higher embodiment applications, and physiological measurements can give insights into the user’s state while performing. An important consideration is that the measurement methods should not interfere with the performance.

**-Surveys:** We can evaluate the user’s response to the system by applying surveys before to gather expectations and, commonly after, to collect points of interest that help to improve the user experience.

We suggest integrating these methods in alignment with the system and the designer’s goals. It is important to prioritize the user’s experience and the quality of the music created during the evaluation process.

## 4 Conclusions

This paper presents previous work on Human-Swarm Interactive Music Systems to distil design principles, algorithms, technologies, and evaluation methods to establish a framework for swarm-based music platforms. We organize this information so that designers can explore novel solutions for performers, and researchers can have additional support to contribute to this field.

We do not intend to provide a strict recipe for Human-Swarm IMSs but a starting guide to propose specific principles that work for particular projects, which can increase and optimize the definition of new approaches for future applications.

For future work, we plan to use this framework to create multiple music platforms and enhance these suggestions through research and data analysis.

## References

1. T. Blackwell. Swarming and Music. In E. R. Miranda and J. A. Biles, editors, *Evolutionary Computer Music*, pages 194–217. Springer London, London, 2007.
2. T. Blackwell and P. Bentley. Improvised music with swarms. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, volume 2, pages 1462–1467, Honolulu, HI, USA, 2002. IEEE.
3. T. Blackwell, O. Bown, and M. Young. Live Algorithms: Towards Autonomous Computer Improvisers. In J. McCormack and M. d'Inverno, editors, *Computers and Creativity*, pages 147–174. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
4. T. Blackwell and M. Young. Self-organised music. *Organised Sound*, 9(2):123–136, 2004.
5. M.-L. Bourguet. Designing and Prototyping Multimodal Commands. *Human-Computer Interaction (INTERACT'03)*, pages 717–720, 2003.
6. O. Bown, B. Carey, and A. Eigenfeldt. Manifesto for a Musebot Ensemble: A platform for live interactive performance between multiple autonomous musical agents. *International Symposium on Electronic Art 2015*, 2015.
7. P. Codognot and O. Pasquet. Swarm intelligence for generative music. In *ISM 2009 - 11th IEEE International Symposium on Multimedia*, pages 1–8. IEEE, 2009.
8. P. R. Cook. Re-Designing Principles for Computer Music Controllers: A Case Study of SqueezeVox Maggie. *NIME 2009*, 2009.
9. J. Drummond. Understanding Interactive Systems. *Organised Sound*, 14(2):124–133, 2009.
10. A. Eigenfeldt and P. Pasquier. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. *Sound and Music Computing Conference (SMC2012)*, 2012.
11. M. Gimenes, E. R. Miranda, and C. Johnson. Musicianship for robots with style. In *NIME '07*, page 197, New York, New York, 2007. ACM Press.
12. M. Kaliakatsos-Papakostas, A. Floros, K. Drossos, K. Koukoudis, M. Kyzalas, and A. Kalantzis. Swarm Lake: A Game Of Swarm Intelligence, Human Interaction And Collaborative Music Composition. *ICMC 2014*, 2014.
13. M. Krzyżaniak. Musical robot swarms, timing, and equilibria. *Journal of New Music Research*, 50(3):279–297, 2021.
14. P. Lucas. A Human-Machine Music Performance System based on Autonomous Agents. Master's thesis, University of Oslo, 2022.
15. J. Malloch, J. Garcia, M. M. Wanderley, W. E. Mackay, M. Beaudouin-Lafon, and S. Huot. A Design Workbench for Interactive Music Systems. In *New Directions in Music and Human-Computer Interaction*, pages 23–40. Springer International Publishing, Cham, 2019.

16. M. Mannone, V. Seidita, and A. Chella. Quantum RoboSound: Auditory Feedback of a Quantum-Driven Robotic Swarm. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 287–292, Napoli, Italy, 2022. IEEE.
17. J. McCormack, P. Hutchings, T. Gifford, M. Yee-King, M. T. Llano, and M. D’Inverno. Design Considerations for Real-Time Collaboration with Creative Artificial Intelligence. *Organised Sound*, 25(1):41–52, 2020.
18. D. Murray-Rust. *Musical Acts and Musical Agents: theory, implementation and practice*. PhD thesis, University of Edinburgh, 2008.
19. D. Murray-Rust, A. Smaill, and M. Edwards. MAMA: An architecture for interactive musical agents. In *Frontiers in Artificial Intelligence and Applications*, volume 141, pages 36–40, 2006.
20. K. Nymoen, A. Chandra, K. Glette, and J. Torresen. Decentralized harmonic synchronization in mobile music systems. *2014 IEEE 6th International Conference on Awareness Science and Technology, iCAST 2014*, 2014.
21. K. Nymoen, A. Chandra, and J. Torresen. The challenge of decentralised synchronisation in interactive music systems. *Proceedings - IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops, SASOW 2013*, pages 95–100, 2013.
22. S. O’Modhrain. A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal*, 35(1):28–42, 2011.
23. J. C. Schacher, D. Bisig, and M. T. Neukom. Composing with swarm algorithms - creating interactive audio-visual pieces with flocking behavior. In *International Conference on Mathematics and Computing*, 2011.
24. K. Tatar and P. Pasquier. MASOM: A Musical Agent Architecture based on Self-Organizing Maps, Affective Computing, and Variable Markov Models. *The 5th International Workshop on Musical Metacreation (MuMe 2017)*, 2017.
25. K. Tatar and P. Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):56–105, 2019.
26. K. Tatar, P. Pasquier, and R. Siu. REVIVE: An Audio-visual Performance with Musical and Visual AI Agents. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, volume 2018-April, pages 1–6, New York, NY, USA, 2018. ACM.
27. K. Tatar, P. Pasquier, and R. Siu. Audio-based Musical Artificial Intelligence and Audio-Responsive Visual Agents in Revive. In *Proceedings of the International Computer Music Conference and New York City Electroacoustic Music Festival*, 2019.
28. N. J. W. Thelle and P. Pasquier. Spire Muse: A Virtual Musical Partner for Creative Brainstorming. In *NIME 2021*. PubPub, 2021.
29. L. K. Ueda and F. Kon. Mobile musical agents. In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications - OOPSLA ’04*, page 206, New York, USA, 2004. ACM Press.
30. T. Unemi and D. Bisig. Playing Music by Conducting BOID Agents - A Style of Interaction in the Life with A-Life. *Conference on the Simulation and Synthesis of Living Systems*, 2004.
31. Y. Uozumi, M. Takahashi, and R. Kobayashi. A Musical Framework with Swarming Robots. In R. Kronland-Martinet, S. Ystad, and K. Jensen, editors, *Computer Music Modeling and Retrieval. Sense of Sounds*, volume 4969, pages 360–367. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
32. D. Waghorn. Controlling emergence in an interactive multi-agent musical system. Master’s thesis, Monash University, 2016.
33. J. Wong, C. Williams, and V. Mirecki. The Application of Swarm Robotics in Music. *Archives of Worcester Polytechnic Institute*, 2020.
34. R. D. Wulffhorst, L. Nakayama, and R. M. Vicari. A multiagent approach for musical interactive systems. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems - AAMAS ’03*, page 584, New York, USA, 2003. ACM Press.

## Improving Instrumentality of Sound Collage Using CNMF Constraint Model

Sora Miyaguchi<sup>1</sup>, Naotoshi Osaka<sup>1</sup>, Yusuke Ikeda<sup>1</sup>

<sup>1</sup>Tokyo Denki University  
23fmi32@ms.dendai.ac.jp, {osaka,yusuke.ikeda}@mail.dendai.ac.jp


**Abstract.** In this study, the improvement in a new audio effect called sound collage, whereby one sound waveform (target sound) is synthesized using another sound waveform (element sound), is investigated. We propose a new model of convolutional NMF (CNMF) with constraints. And we compared the performance of three methods: the original CNMF, the new CNMF constraint model, and modified of Driedger's NMF (non-negative matrix factorization method). Sound collage sounds are synthesized using a combination of animal calls as the target sound and several instrumental sounds as the element sounds. Psychological experiments are conducted to evaluate the extent to which the target sound and instrumental character, namely reproducibility and instrumentality, are demonstrated. The results confirm that the instrumental nature of the synthesized sounds for both models improve compared with CNMF.

**Keywords:** CNMF, NMF, Audio mosaicking, sound collage, instrumentality

### 1 Introduction

Audio effects have applications in various domains, such as game music and animation; furthermore, new effects are desired to achieve richer expression. Previously, we have analyzed an effect called “sound collage” or “audio mosaicking” whereby one sound waveform (target sound) is synthesized using another sound waveform (element sound). Our interest here is a case of an environmental sound as a target sound and instrumental sound as an element sound. Furthermore, we have studied several methods to improve the performance of this effect. Additionally, we have defined two indices for evaluating this effect: (1) reproducibility, which is the degree to which the target sound is represented, and (2) instrumentality, which is the degree to which the sound is perceived to be instrumental.

Previously, we proposed a method for sound collage based on nonnegative matrix factorization (NMF) for sound source separation [1]. This method reproduces the sound by fitting a very short frame of the element sounds, and it has very high reproducibility;

 This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

however, it has low instrumentality owing to the destruction of the temporal structure of the element sounds. To overcome this limitation, Ikeda et al. proposed a method that improved on Driedger's NMF method with three constraints [2] by adding one more constraint (NMF\_DM) [3] and a method using convolutive NMF [4]. Although NMF\_DM improved the instrumentality, the convolutional NMF (CNMF) method at that time was not guaranteed to be an optimal solution and was impractical.

Later, a new optimal solution was reported [5], and a revised method based the new CNMF was proposed by the authors [6]. As the CNMF method can treat all part of element sound as a single basis, the temporal structure of element sounds is preserved; however, the same sound is repeated multiple times in a short period of time, thus rendering difficulty in perceiving instrumentality. In this study, a horizontal proximity restriction was added to the temporal activation of the CNMF method to create a CNMF constraint model (CNMF\_C), and a sound collage was synthesized.

Herein, we compared the three methods, including the new method, and conducted psychological experiments to improve both the instrumentality and reproducibility to the greatest extent possible.

## 2 Sound collage

### 2.1 Sound collage with NMF

The NMF algorithm decomposes matrix  $V$  into the product of matrices  $W$  and  $H$ , with error matrix  $C$ , as follows.

$$V = W \times H + C \quad (1)$$

To estimate  $W$  and  $H$ ,  $C$  is minimized with various criteria, such as by using Frobenius norm.  $W$  and  $H$  are not estimated by an analytical method but rather as an optimization problem, wherein the error  $C$  with the original data is reduced through iterative computation.

In audio signal processing,  $V$  is a spectrogram. Therefore,  $W$  consists of spectra of the target sound (basis matrix) and  $H$  is the temporal activation corresponding to the basis matrix. The original NMF is a supervised algorithm, which simultaneously estimates  $W$  and  $H$ . However, in sound collage, we adopted unsupervised algorithm, where the spectrogram is synthesized, considering the target sound to  $V$  and spectra of element sounds as the basis matrix to  $W$ , and only the time-axis activation  $H$  is estimated.

This method can represent the target sound with significantly high reproducibility because it fits a very short frame of the element sound; however, the temporal structure of the element sound is destroyed, which renders difficulty in perceiving the instrumentality of the element sound.



## 2.2 Sound collage with NMF modified model

To improve the instrumentality, a model which preserves the temporal structure is necessary. Driedger proposed an improved NMF, NMF\_D [2], which imposes constraints on the estimated activation matrix with respect to

1. Horizontal Repetition Restriction: This constraint limits the repetition of spectra within a certain interval along the horizontal direction of the activation matrix.
2. Polyphony Inhibition in Activation Matrix: The proposed polyphony-restricted activation matrix suppresses the presence of multiple sounds within a single frame.
3. Enhanced Element Sound Continuity: Another constraint aims at enhancing the continuity of element sounds, which is manifested as diagonal patterns in the activation matrix.

However, the NMF\_D is insufficient to improve instrumentality as it synthesizes only a portion of the element sound, rather than the whole. Moreover, this modification results in degradation caused by the synthesis of sound solely from the power spectrum, devoid of phase information.

To improve the instrumentality, we modify Driedger's third proposal with the addition of the following constraint referred to as NMF-DM:

1. Instead of utilizing a portion of an element sound, the entire sound is employed from start to finish.
2. To prevent any compromise in sound quality, the original waveform is retained, while the amplitude is drawn from the activation result, which is different from normal Griffin-Lim method [7].

Fig.1 depicts the correspondence between element sounds as basis and synthesized signal using modified activation (inverted upside down). Both are drawn in wave instead of spectrogram in convenience.

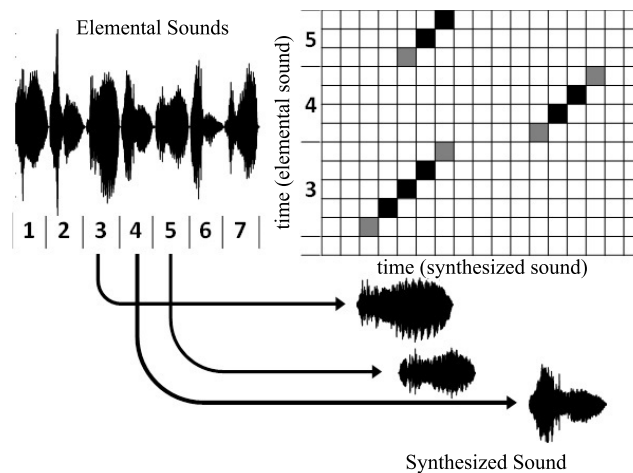


Fig. 1. Modification of activation and preservation of temporal structure in NMF-DM.

### 2.3 Sound collage with CNMF

The CNMF algorithm does not differ from NMF in its basic structure of decomposition in the form of a product of matrices; however, the basis matrix is decomposed into a third-order tensor. For the length of the sound, the prescribed matrix  $W$  is provided, thus allowing for the preservation of temporal ordering. The structural schematic is shown in Fig. 2 and is expressed as follows.

$$V \approx \sum_{t=0}^T W(t) \times H^{t \rightarrow} \quad (2)$$

where the right arrow ( $\rightarrow$ ) indicates that the matrix is shifted  $t$  to the right and 0 is assigned to the vacant space.

Previously, we studied sound collage using CNMF [8]; in this CNMF version, the value of the evaluation function did not decrease monotonously, and the result could not be guaranteed as an optimal solution. However, in 2019, Dylan Fagot et al. proposed a new method to explore the optimal solution of the evaluation function [5]. We applied this method to implement a new sound collage synthesis method. Despite the mathematical optimization, this method has a problem in that the same sound is played multiple times in a short period of time, which causes stuttering and degrades instrumentality.

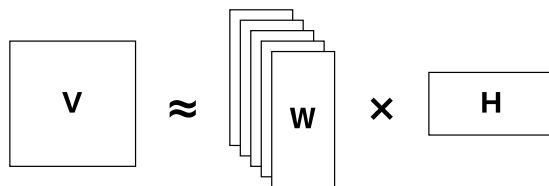


Fig. 2. Structural schematic of CNMF.

### 2.4 Sound collage CNMF with constraint model (CNMF\_C)

We proposed a new model CNMF\_C that prevents temporal proximity caused by an estimated activation. The algorithm that imposes a constraint to the activation is shown in Fig 3. It modifies the activation as a post-processing of CNMF such that only a dominant (local maximum) value survives in a fixed interval and the rest approaches zero as the iteration continues, as shown in the most outside loop.

**Fig. 3.** Constraint algorithm for  $H$  in CNMF\_C (constraint part only).

```

w = appropriate frame length
for i = 1 → N_iteration
  for k_number = 1 → number of element sounds
    R = frame length of the element sound
    for j = 1 → R
      j0 = argmax_[j-w, j+w] (H(j))
      if j == j0
        H(j) remains unchanged.
      else
        % Damping H(j)
        H(j) = H(j) * (1 - (i + 1/N_iteration))
      end
    end
  end
end
end

```

### 3 Experiments

We considered three models: CNMF (as baseline), CNMF\_C, and NMF\_DM, and executed psychological evaluation test. In the experiment, sound sources were played back randomly; furthermore, six male and four female experimental collaborators in their 20s were asked to rate the reproducibility and instrumentality of the two items in an opinion test (five-category test).

#### 3.1 Experiment details

For the experiment, animal calls were used as the target sound and instrumental sounds were used as the element sounds. The experimental parameters are listed in Table 1. Furthermore, element sounds of each number are presented in Table 2. For the experiment participants, the target and instrument of the synthesized sound to be heard were written in advance on an evaluation sheet. In addition, each original sound was also demonstrated in advance. The synthesized sound is also available at the following website.

*<https://acl.im.dendai.ac.jp/index.php/team/sora-miyaguchi/>*

Regarding the instrumentality, the participants were asked to evaluate the degree to which the synthesized sound resembled an instrument sound on a 5-point scale (1~5). They were instructed to give a score of 5 if it felt very similar. For reproducibility, the participants were asked to evaluate how close the synthesized sound was to the target sound on a 5-point scale (1~5). They were instructed to give a score of 5 if it felt very close.

**Table 1.** Experimental parameters.

Target sound	Frog, cicada, horse, elephant
Element sound	Marimba, Accordion, Metallophone, violin (single note, glissando, trill, pizzicato)
Evaluation method	MOS
Test participants	6 men and 4 women

**Table 2.** Element sound of each number.

	Target Sound	Element Sound
1	Cicada	Violin (Glissando), Metallophone
2	Frog	Marimba
3	Frog	Violin
4	Frog	Accordion
5	Elephant	Marimba
6	Elephant	Violin (Glissando), Accordion
7	Elephant	Violin (Single note)
8	Elephant	Violin (Glissando, Trill), Accordion
9	Horse	Marimba
10	Horse	Accordion
11	Horse	Violin

### 3.2 Comparison of CNMF and CNMF\_C with CNMF

The instrumentality and reproducibility results of the experiment are shown in Figs. 4 and 5, respectively. The 95% confidence intervals are indicated on the bar graph. Compared with the CNMF, both CNMF\_C and NMF\_D exhibited higher instrumentality, as shown in Fig. 4; thus, the instrumentality improved. By contrast, for reproducibility, the evaluation changes significantly depended on the target sound, as shown in Fig. 5. In particular, when the elephant was used as the target sound, the results for CNMF\_C were significantly lower.

Comparing CNMF and CNMF\_C, the evaluation was higher for instrumentality except for conditions #6 and #7. On the other hand, for reproducibility, the evaluations were low except for conditions #3 and #11.

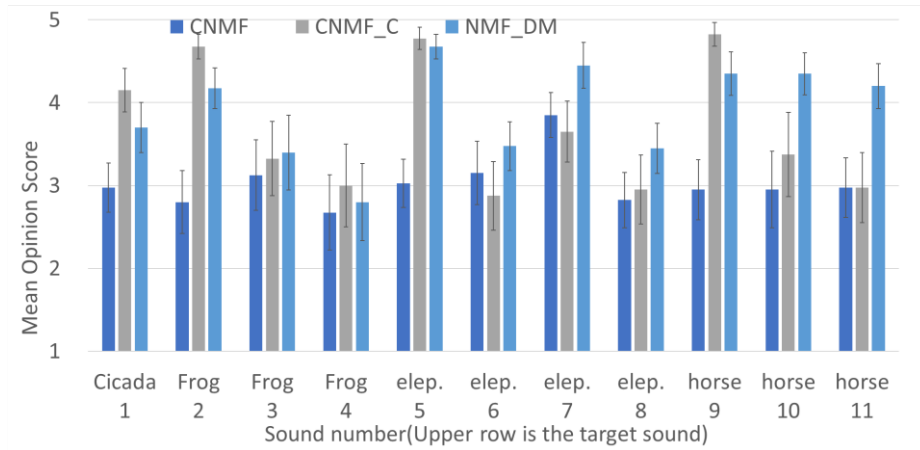


Fig. 4. Evaluation results of the three models for instrumentality.

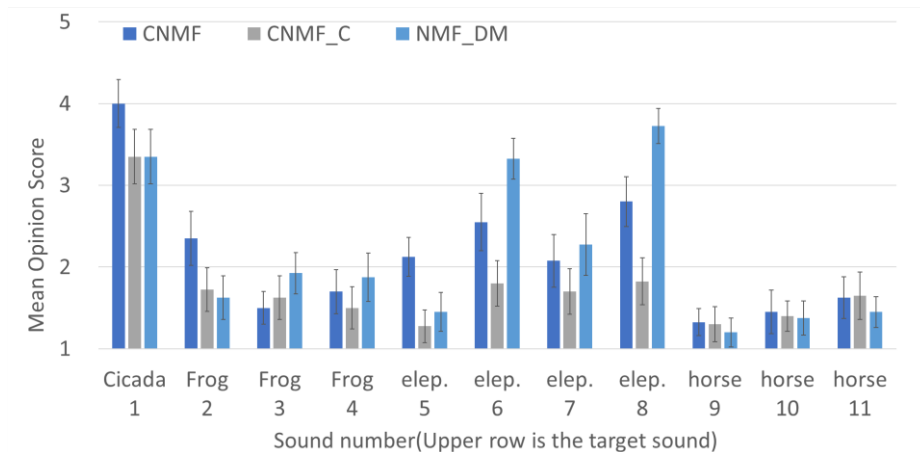


Fig. 5. Evaluations results of the three models for reproducibility.

### 3.3 Discussion

Compared to CNMF, CNMF\_C showed improved instrumentality in almost all conditions, except when the target sound is an elephant. On the other hand, the reproducibility results were lower under almost all conditions. This demonstrates that limiting the temporal proximity of element sounds affects instrumentality. Additionally, combining CNMF with CNMF\_C makes it possible to control the trade-off between instrumentality and reproducibility. Instrumentality and reproducibility are a trade-off: when one rises, the other falls.

Certainly, NMF\_DM showed higher instrumentality than CNMF\_C depending on the combination of element sounds and target sounds. This indicates that NMF\_DM has the potential to show higher instrumentality than CNMF\_C with the appropriate combination of timbres. However, In terms of controllability, CNMF\_C, a generalization

of CNMF, is higher because of its wider control over reproducibility and instrumentality. Therefore, in designing sound collages, CNMF\_C, which allows moderate control of the two, is desirable and will better meet the user's needs.

## 4 Conclusions

In this paper, we compared three methods: CNMF\_C, CNMF, and NMF\_DM. Through experimentation, it was demonstrated that, by adding constraints in CNMF\_C, the instrumental quality could be improved in most cases compared to CNMF, although the reproducibility decreased. This suggests that when CNMF\_C is defined as a generalization of CNMF, there is a trade-off between instrumental quality and reproducibility, and that this trade-off can be controlled. Since users of Sound collage should be able to synthesize at their preferred level of reproducibility, we can say that our research was successful in improving the performance as Sound collage. Certainly, in some conditions, NMF\_DM showed higher instrumental score than CNMF\_C. However, controllability is important in sound collage, so CNMF\_C can be considered more suitable in this study than NMF\_DM.

In the future, we plan to explore ways to improve the performance of CNMF\_C, such as finding more appropriate combinations of sounds, and examining finer control over instrumental quality and reproducibility. Furthermore, we will attempt to define a comprehensive evaluation in scalar values, incorporating both subjective evaluation and yet to be defined physical evaluation.

## References

1. Tanaka, M., Osaka, N.: Sound quality evaluation of sound collage using NMF 2022 Autumn meeting of the Acoustical Society of Japan, 1-1-19, (2022). (In Japanese)
2. Driedger, J et al.: LET IT BEE-Towards NMF-Inspired audio mosaicking, Proc. of the 16<sup>th</sup> ISMIR, Malaga, Spain (2005).
3. Masaya, I., Osaka, N.: Synthesis of sound collage using NMF, IPSJ, Vol. 2020-MUS-126, No. 8, 1-6 (2020). (In Japanese)
4. Paris Smaragdis, Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs, ICA 2004: Independent Component Analysis and Blind Signal Separation, pp 494-499 (2004).
5. Fagot, D. et al.: Majorization-minimization Algorithms for Convolutional NMF with the Beta-divergence, ICASSP 2019, Brighton, UK.
6. Miyaguchi, S., Osaka, N.: Improvement of instrumentality for sound collage using CNMF, 2023 Spring meeting of the Acoustical Society of Japan, 1-9-20, (2023). (In Japanese)
7. D. W. Griffin and J. S Lin, Signal Estimation from modified Short-Time Fourier Transform, ASSP-32, April 1984, pp. 236-242 (1984)
8. "nmf - toolbox", <https://github.com/colinvaz/nmf-toolbox>, last accessed 2023/05/03.

# Quantum Circuit Design using Genetic Algorithm for Melody Generation with Quantum Computing

Tatsunori Hirai

Komazawa University  
thirai@komazawa-u.ac.jp

**Abstract.** In this paper, we explore the potential of quantum computing for music generation, particularly for generating melodies. We propose a method of designing quantum circuits with genetic algorithms for melody generation. Our method allows for the generation of subsequent musical notes for arbitrary input notes and the production of melodies of varying lengths based on the transition distribution between melodies in the training data. We compared the accuracy of a quantum computer in predicting the subsequent note based on the training data with that of a classical computer. Our results demonstrate the potential of quantum computing for melody generation.

**Keywords:** Melody generation; quantum computing; genetic algorithm

## 1 Introduction

The first instance of music being automatically generated by a computer was the “ILLIAC Suite, for String Quartet,” composed by ILLIAC I in 1957 [1]. Since then, researchers have been investigating music generation techniques, including automatic composition, from the early days of computing to the present day

The realization of quantum computing, expected to be the next-generation computing paradigm, is becoming increasingly feasible. As of May 2023, Noisy Intermediate-Scale Quantum Computers (NISQ), a quantum computer designed with the assumption of the inclusion of various types of noise, have been realized with hundreds of qubits and are available on the cloud.

To discuss the need for a quantum computer, it is important to explore its potential applications. The extent and fields in which quantum computers will be useful remain uncertain at present. It is also unclear whether quantum computers can be considered superior to classical computers.

In this paper, we investigate the potential applications of quantum computing for the task of music generation, a domain that has been extensively studied using classical computing methods. We specifically assess the feasibility of music generation using current gated quantum computer architectures and propose a novel approach for designing quantum circuits by employing genetic algorithms.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

## 2 Related Work

The utilization of quantum computers for musical expression, termed “quantum music,” has been the subject of multiple investigations in recent years, reflecting a growing interest in this interdisciplinary field. For example, Kirke et al. proposed Q-MUSE, a live performance music system where output sound is modulated by altering input parameters of a quantum computer through a button controller and gesture controller [2]. Additionally, Clemente et al. introduced a keyboard, termed “qeyboard,” in which sound parameters are controlled by quantum circuits [3].

The QuTune project [4] is a research project focused on generating music through quantum computing, culminating in the organization of the first international symposium on quantum music in 2021. The QuTune team has produced several technical papers and comprehensive reviews on this subject. In previously published review articles [5], [6], the authors discuss the fundamentals of quantum computers and computer music, introducing specific applications such as the Quantum Vocal Synthesizer and the Quantum Walk Sequencer. Notably, the Quantum Walk Sequencer is a sequencer that facilitates note-to-note transitions using a quantum random walk [7]. This approach, which utilizes quantum circuits to represent note transitions, offers valuable insights for melody generation. Furthermore, the QuTune team is developing a music generation system that incorporates a quantum natural language processing (QNLP) approach, integrating quantum computing within a natural language processing framework [8].

Kirke proposed the hybrid music generation system qGEN, which integrates a gated quantum computer and a quantum annealing machine [9]. qGEN produces music by combining GATEMEL, a melody generator utilizing a gated quantum computer, with qHARMONY, a system that generates accompaniments for given melodies using a quantum annealing machine. Souma proposed quantum live coding, a method for generating improvised music based on gated quantum algorithms [10]. This approach involves producing melodies by connecting consecutive notes through quantum entanglement.

The recent efforts applying quantum computing to musical expression outlined in this section signify the emerging development of this research domain.

## 3 Possibilities of Quantum Computing in Music Generation

A key feature of quantum computers is the superposition state of qubits. Upon measurement, the superposition collapses, yielding a 0 or 1 state similar to classical bits. Quantum algorithms leverage superposition states, representing all possible inputs, to increase the likelihood of obtaining the desired outcome through measurement.

In this context, we explore the application of quantum computers to the task of music generation. It is essential to recognize that in music, there is no definitive “correct” answer, and pursuing a singular answer might not be ideal, especially in the context of music generation. The pursuit of a single correct answer in generative tasks is unlikely to be accomplished, regardless of the efficiency of search algorithms developed. In music, there are many sequences and combinations of sounds that are considered undesirable by many people. Therefore, it is desirable to develop an algorithm that can avoid such results when generating music.



**Table 1.** A binary representation of a note name.

note name	C	D	E	F	G	A	B	R
binary digits	000	001	010	011	100	101	110	111

The superposition state of qubits in a quantum computer enables the representation of all possible melody combinations simultaneously when generating melodies. As there are numerous undesirable note combinations included in all possible melodies, designing the quantum algorithm such that the probability of measuring such an undesired combination is reduced is a potential approach. However, the final measurable result is just one of all possible combinations, meaning that even though the quantum computer considers all combinations simultaneously, the resulting melody is only one.

Due to the distinct features of quantum and classical computers, it may be possible to achieve efficient results by incorporating a quantum computer, depending on the algorithm being used. An example of a successful application of quantum computing is the generation of random numbers. Quantum computing can successfully generate true random numbers, thanks to the probabilistic nature of qubits. However, since precision is not crucial in music generation, the significance of introducing a quantum computer remains debatable. In this paper, we propose a melody generation algorithm that utilizes quantum circuits to replicate the note transitions observed in training data, taking advantage of the characteristic of true random number generation in quantum computers.

## 4 Data Representation for Quantum Circuit Design

In this chapter, we introduce the data representation for melody generation with a quantum computer. Generating melodies using quantum circuits necessitates determining an appropriate method to represent notes and quantum gates numerically.

### 4.1 Data Representation of Note Names

Here, we have simplified the problem to its core elements to enable clear observation of the behavior of the quantum circuit. The problem is set up by considering only the essential elements that compose a melody, namely note names. To handle melodies as simply as possible, we represent notes using a 3-bit binary number that only represents the note name. Specifically, the note names are represented as a 3-bit binary number, with C being represented as 000. A total of 8 note names are used, including seven types of notes from C (000) to B (110) and a rest represented as R (111). Note durations are not considered, and all notes are assumed to have a fixed duration of one quarter note. The binary representation and corresponding note names are presented in Table 1. When applying this data representation to a quantum computer, the problem is set up such that a 3-qubit quantum circuit generates the next note based on the input note.

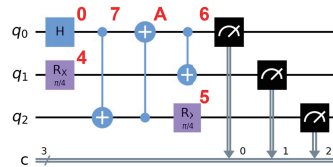
### 4.2 Data Representation of Quantum Gates

There are various types of quantum gates that compose quantum circuits. To realize melody generation in the simplest problem setting, we selected 4 basic types of gates

**Table 2.** Numerical representation of quantum gates.

quantum gate	numerical representation
H gate (register 0)	0
H gate (register 1)	1
H gate (register 2)	2
X-axis rotation gate : $\pi/4$ (register 0)	3
X-axis rotation gate : $\pi/4$ (register 1)	4
X-axis rotation gate : $\pi/4$ (register 2)	5
CX gate (register 0 to 1)	6
CX gate (register 0 to 2)	7
CX gate (register 1 to 2)	8
CX gate (register 1 to 0)	9
CX gate (register 2 to 0)	10 (A)
CX gate (register 2 to 1)	11 (B)
CCX gate (register 0,1 to 2)	12 (C)
CCX gate (register 0,2 to 1)	13 (D)
CCX gate (register 2,1 to 0)	14 (E)
no gate	15 (F)

and numerically represented 16 different gate placement patterns. These patterns include variations in which qubits the gates act upon among the 3 input registers. The 16 possible gate arrangements include a state with no gates, and each arrangement is represented by a unique hexadecimal number. For each quantum circuit representation, we use 8-digit hexadecimal numbers to represent the gate arrangement. This enables representation of quantum circuits with 0-8 gate combinations. Table 2 shows the numerical representation of quantum gates and their corresponding gate arrangement patterns.



**Fig. 1.** Example of quantum circuit represented by “04F7A6F5.”

Using the hexadecimal numerical representation of quantum gates in Table 2, a quantum circuit can be represented by an 8-digit hexadecimal number. For example, the circuit diagram represented by “04F7A6F5” is shown in Fig.1. According to the correspondence shown in Table 2, F represents no gates, so in this case, the quantum circuit consists of 6 gates. The 0 represents the H gate (Hadamard gate) applied to register 0 (the register corresponding to  $q_0$  in Fig.1), which puts the qubit in a superposition. The 4 and 5 are rotation gates around the X-axis applied to registers 1 and 2, respectively, with rotation angles of  $\pi/4$ . The 7, A, and 6 are all CX gates (controlled-NOT gates), which enable interactions between the registers.

**Table 3.** Measurement results (for 200 shots) when  $|000\rangle$  (C) is input to the quantum circuit shown in Fig.1.

output	000(C)	001(D)	010(E)	011(F)	100(G)	101(A)	110(B)	111(R)
measurement count	80	0	17	0	93	0	10	0

There exist many more varieties of quantum gates beyond those listed here. While more complex quantum circuits can be expressed in the same numerical framework by assigning numbers to other gates, this paper prioritizes simplicity, and only the basic gates listed here will be utilized. Experiments with more complex quantum circuit configurations are also possible, but are left as future work, as current quantum computers are highly susceptible to errors in constructing such circuits.

To ensure the usefulness of quantum circuits, it is necessary to take advantage of the superposition of quantum states. If superposition is not utilized, the quantum circuit operation can be reproduced using a classical computer, making the use of quantum circuits meaningless. Hence, we operate the H gate once for all inputs of registers 0 to 2, and then add other gates represented by 8-digit hexadecimal numbers to utilize superposition.

### 4.3 Generation of subsequent Note with Quantum Circuit

The problem of generating a subsequent note for an input note can be represented by the input/output of data to/from a quantum circuit, achieved by combining the data representation of note names and quantum gate representation. To generate a melody, the initial note is determined and input to the quantum circuit to generate subsequent notes by measuring the output qubits. The process is repeated by inputting the generated subsequent note as input data to the quantum circuit to generate further notes, thus completing the melody.

Table 3 shows the measurement results (for 200 shots) obtained when the input note C, represented as  $|000\rangle$ , is used as the input to the quantum circuit shown in Fig.1. If this quantum circuit were used to generate the subsequent note, it would only output notes corresponding to the states of C, E, G, or B. The results presented in Table 3 are obtained when  $|000\rangle$ , corresponding to C, is directly input to the quantum circuit in Fig.1 without using the H gates at the beginning.

We attempted two methods for generating quantum circuits:

- **Training-A:** train one circuit for each type of input note.
- **Training-B:** train a single circuit for all input-output combinations.

In a case circuit is trained for each input note (Training-A), the C circuit generates the subsequent note from the input C, and the D circuit generates the subsequent note from the input D. The design process of each quantum circuit will be presented in the next chapter.

## 5 Designing Quantum Circuits with Genetic Algorithms

Designing a quantum circuit is equivalent to determining a quantum algorithm. It affects the quality of the generation results. There are several possible approaches to design

quantum circuits, and one example is to use a H gate in every register to represent a superposition of all combinations, resulting in a circuit with completely random outputs.

Manual gate determination in quantum circuits can lead to inflexible designs with fixed outcome patterns. Thus, this paper investigates a flexible circuit design method using training data. We prepare arbitrary melodies as training data and search for optimal quantum gate combinations that reproduce the desired output distribution based on that training data. This approach establishes a well-defined criterion for designing quantum circuits that accurately emulate the input-output relationship in the training data.

We utilize a genetic algorithm to explore quantum gate combinations, aiming to minimize the discrepancy between the quantum circuit's output and the training data's note transition distribution. The genetic algorithm runs on a classical computer, while the quantum computer generates subsequent notes based on input notes, making our approach a hybrid method. Moreover, training processes are conducted using a quantum circuit simulator on a classical computer, while the actual quantum computer is used for generating melodies with the resulting quantum circuit. Using a quantum computer during circuit design is feasible, but the extensive trials needed for training make it challenging within a realistic time frame, given current capabilities. Future advancements in quantum computing may address these challenges.

### **5.1 Preparation of Training Data**

The training data is created from the note sequences present in pre-existing musical compositions. The types of notes that can be handled by the algorithm proposed in this study are limited to eight types, from C to B and R. Therefore, the melody used for training is composed solely of simple quarter notes in the key of C major. In this study, melodies from three pieces, namely "Twinkle, Twinkle, Little Star," "Tulip," and "Froggy's Song," were chosen for the purpose of training.

For instance, when creating training data based on the initial melody of "Twinkle, Twinkle, Little Star," the melody can be represented as "C→C→G→G→A→A→G→R." Consequently, the note following C is exclusively either C or G, with no transition to other notes. To replicate this pattern, an ideal quantum circuit would measure C (000) and G (100) with a 50% probability each for an input of C (000). This input-output relationship can be achieved using a quantum circuit with a single H gate in the second register. For the actual training process with a more intricate output distribution, we automate quantum circuit design using a genetic algorithm instead of manual configuration.

The training data in this study is composed of transition probabilities between note names. As a result, the generated quantum circuit serves as a model for generating subsequent notes utilizing a bi-gram approach. The actual training data consists of a single matrix representing the transition probabilities between note names found in the three selected pieces. The transition probabilities for the note names utilized as training data are presented in Table 4. According to these transition probabilities, when note A is input, the likelihood of G being generated as the subsequent note is the highest at 0.57, followed by A at 0.43, while the probabilities for the other notes are 0. Notably, none of the melodies in the training data included the note B.

The training data can be readily expanded by incorporating a greater number of musical pieces. However, the melody generation approach presented in this paper is limited

**Table 4.** Training data (transition probability of note names).

	subsequent note							
	C	D	E	F	G	A	B	R
input note C	0.10	0.38	0	0	0.10	0	0	0.43
D	0.33	0.14	0.38	0	0	0	0	0.14
E	0.04	0.40	0.20	0.12	0.04	0	0	0.20
F	0	0	0.58	0.33	0.08	0	0	0
G	0	0	0.17	0.17	0.28	0.22	0	0.17
A	0	0	0	0	0.57	0.43	0	0
B	0	0	0	0	0	0	0	0
R	0.50	0	0.11	0.11	0.28	0	0	0

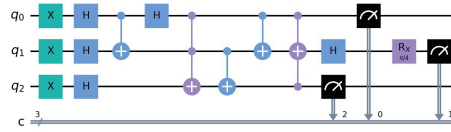
to eight note types. Given that all notes are quarter notes, the range of applicable pieces is somewhat restricted. While it is feasible to learn more complex melody distributions within the same framework by eliminating constraints on note value and increasing the diversity of manageable note names, such considerations are beyond the scope of this paper. In order to validate the efficacy of automating quantum circuit design, it is crucial to initially establish a simplified problem framework to the greatest extent possible.

## 5.2 Details of Genetic Algorithm

The genetic algorithm is utilized to learn a sequence of 8-digit hexadecimal numbers representing quantum gate combinations, as introduced in Section 4.2. A randomly initialized sequence of 8-digit hexadecimal numbers is treated as an individual within the genetic algorithm, with each digit corresponding to a gene. The process was repeated for 100 generations, with 1000 individuals in each generation subjected to tournament selection, two-point crossover, and mutation steps, ensuring a preference for individuals exhibiting high fitness. The fitness value is computed by taking the mean squared error between the output distribution of 200 shots from a quantum circuit, as shown in Table 3, and the target output note distribution (Table 4) used for training. The crossover probability was set to 0.5, the probability of individual mutation was set to 0.2, and the gene mutation probability was set to 0.05.

In the case that no gates are present to compose a quantum circuit, the gene sequence is represented as “FFFFFFF.” In this state, each H gate operates once on every input qubit, resulting in a superposition of all possible states, which implies that the output may consist of any of the eight notes. Building upon this initial state, the quantum circuit’s gate configuration is trained by altering the gene sequence and incorporating quantum gates corresponding to the sequence modifications, thereby generating outputs that more closely align with the training data. The more closely the distribution of outputs aligns with the training data, the more desirable the design of the quantum circuit can become to achieve the desired output distribution.

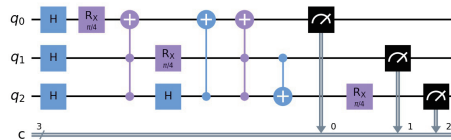
One of a quantum circuit designed by the genetic algorithm utilizing the training data (Table 4) is depicted in Fig.2. Given that the quantum circuit in Fig.2 is designed for the input note R (rest), X gates are placed at all the registers before the remaining gates to set the input bit value to 1. Consequently, the input state  $|000\rangle$  transitions to



**Fig. 2.** A quantum circuit designed by genetic algorithm (“60C86D14”), with the input note R (Training-A).

$|111\rangle$ , followed by the H gates operating on all qubits and the subsequent operations, as represented by the 8-digit hexadecimal numbers. For Training-B, where a common quantum circuit is used for all note inputs, the configuration of the initial X gates varies for each input note, while the circuit represented by the 8-digit hexadecimal number remains consistent.

Although the quantum circuit in Fig.2 contains redundant gate arrangements, such as consecutive utilization of CX and CCX gates, its output distribution closely approximates the training data  $[0.5, 0, 0.11, 0.11, 0.28, 0, 0, 0]$  (bottom row of Table 4). The output result will be described in Section 5.3.



**Fig. 3.** A quantum circuit designed by genetic algorithm (“3D24ADB5”), for all input/output note combinations (Training-B). X gates for distinguishing the types of input notes are omitted here.

In the Training-B setting, an integrated version of a quantum circuit capable of handling all input notes within a single circuit was trained. The training result for all input notes in a single quantum circuit are presented in Fig.3. This quantum circuit accommodates all input notes by inserting X gates that correspond to each input note name at the beginning of the quantum circuit. For input notes other than C, X gates are applied to modify the input qubits to the corresponding input state.

### 5.3 Generation of subsequent Note by Trained Quantum Circuits

A subsequent note corresponding to the input note was generated using a quantum circuit that had been trained on the given data through a genetic algorithm. First, as a result of training distinct quantum circuits for each input note (Training-A), Table 5 displays the outputs obtained by executing 100 shots for each of the eight different circuits corresponding to each input note respectively. It is important to note that, during the actual generation step, only one shot is executed, and the measured output serves as the generated subsequent note. Table 5 displays the distribution of outputs obtained by

**Table 5.** The results of generating subsequent notes with the Training-A setting, using quantum circuits trained for each input note (100 shots). Each row corresponds to a trained quantum circuit.

	Generated subsequent notes (measurement counts)								
	C	D	E	F	G	A	B	R	
quantum circuit	C	29	25	0	0	25	0	0	21
	D	27	22	28	2	0	6	0	15
	E	4	30	15	4	12	10	2	23
	F	0	0	55	45	0	0	0	0
	G	6	1	26	20	19	23	3	2
	A	0	0	0	0	49	51	0	0
	B	8	11	12	16	6	18	17	12
	R	53	0	4	0	36	0	7	0

**Table 6.** The results of generating subsequent notes with the Training-B setting, using a single quantum circuit (Fig.3) trained for all input notes (100 shots).

	Generated subsequent notes (measurement counts)								
	C	D	E	F	G	A	B	R	
input note	C	28	21	3	3	2	3	24	16
	D	24	35	15	11	0	0	5	10
	E	21	22	4	3	5	9	16	20
	F	14	15	26	22	9	14	0	0
	G	2	4	25	21	20	19	5	4
	A	0	0	15	10	26	28	10	11
	B	4	2	16	25	25	21	3	4
	R	7	9	0	0	19	12	29	24

executing 100 shots. Although the results vary with each execution, the distribution of subsequent notes closely resembles that presented in the training data (Table 4).

Subsequently, the output obtained by executing 100 shots for each of the eight distinct input notes on a single quantum circuit trained for all notes (Training-B) is presented in Table 6. In this case, the quantum circuits, represented as 8-digit hexadecimal numbers (i.e., “3D24ASB5”), remain the same for all inputs. As a result, the subtle bias in the distribution for each input note could not be trained as effectively as when employing different circuits for each input note.

A comparison of Table 4 and 5, or Table 4 and 6, reveals the extent to which the output distribution of the quantum circuits resembles the training data distribution. In the training data, the transition probability to note B was zero for all input notes; however, the resulting quantum circuits did allow for transitions to note B. This was particularly noticeable when training a single quantum circuit for all input notes (Training-B). For the input note B, all outputs were measured in both the results of Table 5 and 6, since the distribution of outputs to be trained comprised only zeros. Therefore, this result is not an erroneous.

It should be noted that exact replication of the output distribution is impossible for a quantum computer, as the outcome varies with each execution.

**Table 7.** Comparison of error rates in reproducing note transitions. Comparison of random number generation with classical computers, quantum circuit for each input (Training-A), and quantum circuit for all input (Training-B).

Random number generated with classical computer	Quantum circuit for each input note (Trainig-A)	Quantum circuit for all input notes (Trainig-B)
$6.90 \times 10^{-4}$	$4.63 \times 10^{-3}$	$1.91 \times 10^{-2}$

#### 5.4 Comparison of Note Transition Reproduction

In this section, we investigate the extent to which the generation of subsequent notes by quantum circuits can accurately reproduce the note transitions in the training data. To make a comparison, we also include results obtained from a classical computer that generates subsequent notes according to the distribution of training data using random numbers. For generating random numbers with a classical computer, we utilized the random module in the Python standard library.

The error rate  $R$  of reproducing subsequent note is defined as follows:

$$R = \sqrt{\frac{1}{64} \sum_{i=1}^8 \sum_{j=1}^8 (t_{ij} - x_{ij})^2} \quad (1)$$

where  $t_{ij}$  denotes the note transition probability of the training data from note  $i$  to note  $j$ , and  $x_{ij}$  denotes the probability of transition with each method to be compared. The  $x_{ij}$  is calculated based on the distribution obtained from 100 generated subsequent notes for each input note (i.e. Table 5 and Table 6). The error rate of subsequent note reproduction corresponds to the mean square error (MSE).

We generated  $x_{ij}$  for the classical computer by generating 100 random numbers for each input note. The subsequent notes are then determined by comparing the generated random numbers with the distribution of the training data. In other words, subsequent notes are directly generated from the note transition probabilities of the training data.

Table 7 shows a comparison of error rate  $R$  in reproducing subsequent notes using each circuit/computer, where smaller  $R$  values indicate higher reproducibility. The classical computer achieved high accuracy by directly using the training data's transition probability distribution. On the other hand, in the case of quantum circuits, the error rate was lower when using different circuits for each input note (Training-A setting), suggesting that varying circuits yield better output distribution. Due to the randomness of all methods, the error rate of reproduction varies to some extent with each execution.

Note that the training data did not include note B, so the values for the note transitions from note B were excluded from the calculation in Table 7 ( $x_{7j}$  and  $x_{i7}$  were set to 0). This is because including note B would simply increase the error rate for all methods, thus it was excluded from the analysis.

This comparison was verified using the Qasm simulator, a quantum circuit simulator that uses a classical computer. It should be noted that in the case of an actual quantum computer, errors in the quantum circuit must also be considered.



## 6 Melody Generation Results

We utilized the quantum circuit designed using the above methodology to generate musical melodies. The process starts with selecting the first note and obtaining the output from the corresponding quantum circuit specifically designed for the input note. The next note in the sequence is determined by inputting the initial output note into the quantum circuit that corresponds to it, and this process is repeated iteratively. The choice of the first note and the number of iterations can be arbitrarily determined, contingent on the desired length of the melody.

Fig.4 shows scores of melodies generated by the quantum computer. Two 4-measure melodies were generated for each quantum circuit design pattern (Training-A and Training-B), starting with C and G. We employed the `ibm_bogota` quantum computer, provided by IBM Q, which can operate up to five qubits. It should be noted that when using an actual quantum computer, errors may occasionally arise, leading to note transitions that exhibit zero probabilities in Table 5 ( $C \rightarrow A$  and  $F \rightarrow C$  in Fig.4 bottom-left) and Table 6 ( $D \rightarrow G$  in Fig.4 top-right).



**Fig. 4.** Examples of melody generation utilizing an actual quantum computer. Left: results employing eight different trained quantum circuits for each input note (Training-A). Right: results employing single trained quantum circuit for all input note names (Training-B).

The execution time required for generating a melody utilizing an actual quantum computer is dependent on the waiting time experienced by the quantum computer when performing the task at a particular instance. In our experiments, generating a 16-note melody necessitated approximately 15 to 30 minutes. This observation does not necessarily suggest that the process inherently requires an extended execution time; instead, it reflects the current limitations of quantum computing resources accessible through cloud services, which result in the increased execution time.

## 7 Conclusions

In this paper, we explored the potential application of quantum computers in music generation and proposed a genetic algorithm-based method for designing quantum circuits to generate melodies. We compared the accuracy of reproducing subsequent notes between quantum and classical computers. The results of new melodies generated by quantum circuits trained on specific musical pieces are presented. Although a similar melody generation can be achieved with a classical computer, the distinct difference lies in the randomness of the outcome. Nevertheless, we demonstrated that the process of generating subsequent notes, as performed by classical computers, can also be implemented with a quantum computer without manual design of quantum circuits. The

input/output relationship of notes is successfully represented using quantum circuits, indicating that music generation algorithms on classical computers may also be feasible in quantum circuits. We aim to further investigate the possibilities of quantum computing in the domain of music generation in future research.

In order to maintain a simple problem setup, the melody generation approach proposed in this study employed a limited number of available note types and lengths, as well as restricted types and quantities of quantum gates. Although these constraints can be easily mitigated, the primary focus of this paper was not to increase the complexity of the results. Instead, the central objective of this work was to explore the potential of quantum computing for music generation.

Quantum computing is currently in the early stages of development. Future quantum programming paradigms may not rely on quantum circuits. Moreover, the optimal design of applications may experience considerable transformations in response to changes in the utilization of quantum computing. We will persist in investigating the potential application of quantum computers for music generation. In our future research, we aim to utilize quantum computing to enhance the expressiveness of music generation in ways that have not been achievable with classical computers.

## References

1. Hiller, L. Isaacson, L.M.: Illiac suite, for string quartet. Vol.30, No.3, New Music Edition (1957)
2. Kirke, A., Shadbolt, P., Neville, A., Antoine, A., Miranda, E.R.: Q-Muse: A quantum computer music system designed for a performance for orchestra, electronics and live internet-connected photonic quantum computer In: Proceedings of the 9th Conference on Interdisciplinary Musicology (2014)
3. Clemente, G., Crippa, A., Jansen, K., Tüysüz, C.: New Directions in Quantum Music: Concepts for a Quantum Keyboard and the Sound of the Ising Model. Quantum Computer Music: Foundations, Methods and Advanced Concepts, Cham: Springer International Publishing, pp.433–445 (2022)
4. Eduardo R. M. Bob C., et al.: QuTune Project. <https://iccmr-quantum.github.io/>.
5. Miranda, E.R.: Quantum Computer: Hello, Music!. Handbook of Artificial Intelligence for Music, pp.963–994 (2021)
6. Miranda, E.R. Bask, S.T.: Quantum Computer Music: Foundations and Initial Experiments. Quantum Computer Music: Foundations, Methods and Advanced Concepts, Cham: Springer International Publishing, pp.43–67 (2022)
7. Aharonov, Y., Davidovich, L., Zagury, N.: Quantum random walks. Physical Review A, Vol.48, No.2, pp.1687–1690 (1993)
8. Miranda, E.R. Yeung, R. Pearson, A. Meichanetzidis, K. Coecke, B.: A Quantum Natural Language Processing Approach to Musical Intelligence. Quantum Computer Music: Foundations, Methods and Advanced Concepts, Cham: Springer International Publishing, pp.313–356 (2022)
9. Kirke, A.: Programming gate-based hardware quantum computers for music. Physical Review A, Vol.48, No.2, pp.1687–1690 (1993)
10. Souma S.: Exploring the Application of Gate-type Quantum Computational Algorithm for Music Creation and Performance. Quantum Computer Music: Foundations, Methods and Advanced Concepts, Cham: Springer International Publishing, pp.88–103 (2022)

# Automated Arrangements of Multi-Part Music for Sets of Monophonic Instruments

Matthew McCloskey<sup>1</sup>, Gabrielle Curcio<sup>1</sup>, Amulya Badineni<sup>1</sup>, Kevin McGrath<sup>1</sup>,  
Georgios Papamichail<sup>2</sup>, and Dimitris Papamichail<sup>1</sup> \*

<sup>1</sup> The College of New Jersey, Ewing, NJ 08618, USA

<sup>2</sup> New York College, Athens, Greece

papamicd@tcnj.edu

**Abstract.** Arranging music for a different set of instruments that it was originally written for is traditionally a tedious and time-consuming process, performed by experts with intricate knowledge of the specific instruments and involving significant experimentation. In this paper we study the problem of automating music arrangements for music pieces written for monophonic instruments or voices. We designed and implemented an algorithm that can always produce a music arrangement when feasible by potentially transposing the music piece to a different scale, permuting the assigned parts to instruments/voices, and transposing individual parts by one or more octaves. We also published open source software written in Python that processes MusicXML files and allows musicians to experiment with music arrangements. Our software can serve as a platform for future extensions that will include music reductions and inclusion of polyphonic instruments.

**Keywords:** music arrangement, music algorithms

## 1 Introduction

Music arrangements involve the adaptation of a piece of music for different instruments or ensembles. This allows the music to be performed in a variety of settings, enhances the repertory of musicians, and can also help to bring new life to a piece that may have been composed for a specific instrument or ensemble [16]. Additionally, arrangements can help to showcase the unique strengths of different instruments or even create entirely new interpretations of a piece. The process of arranging a piece of music can be a creative endeavor in itself, giving the arranger the opportunity to put their own spin on a familiar work, greatly enhancing the listening experience for audiences [1, 5, 12].

The computational complexity of arranging music written for a set of instruments toward a target single instrument, often employing reasonable reductive constraints, has

---

\* The authors acknowledge use of the ELSA high performance computing cluster at The College of New Jersey for conducting the research reported in this paper. This cluster is funded in part by the National Science Foundation under grant numbers OAC-1826915 and OAC-1828163.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

been examined in the work of Moses and Demaine [4]. Complexities of dealing with polyphonic instruments, such as piano and guitar, include the need of considering possible fingerings as well as reductions, the elimination of certain notes for playability of even feasibility. Most research in automating music arrangements has concentrated on the piano, primarily concerning orchestral pieces [2, 8, 10, 11, 13, 14]. Much of that work involves reductions to enable feasibility. Other work in the field has examined arrangements for the guitar [6, 7, 15], wind ensembles [9], and other orchestral instruments [3].

Despite its obvious benefits, we are not aware of any published algorithm or widely available software that allows for the automated arrangement of a given music piece to a different set of instruments that it was originally written for in the general case. Working toward filling that need, we designed and implemented an algorithm that arranges music written for monophonic instruments and guarantees a successful outcome when an arrangement is possible without score reduction. Our recursive backtracking algorithm exhaustively examines all feasible assignments of parts to available instruments and all possible transpositions of the piece, including independent octave transpositions of individual parts, to determine a successful arrangement that minimally affects the musicality of the piece. The use of memoization, storing partial results for reuse, further enhances the time efficiency of our software and allows processing of most music pieces in a matter of seconds.

## 2 Methods

### 2.1 Definitions

For the purposes of our research, a music piece is written in a chromatic scale and notes are separated by the interval of a semitone. We will assume that all notes fall within a total range of 88 semitones, the notes of a traditional piano, from A0 to C8. We will assign an integer to each note in the range, such that all notes can be represented by an integer from 1 to 88. For our discussion, a monophonic instrument is one that can only play one pitch at a time, such as the flute, the oboe, or a voice. Polyphonic instruments can play multiple notes simultaneously, such as the piano, guitar, or harp. A polyphonic instrument can always play a monophonic part within its range.

For our study an input music piece will consist of  $n$  parts, each being assigned to a single monophonic instrument or voice. Such parts are presented in the sheet music representation of the piece in an equal number of staves each. Our algorithm preserves the rhythm, rhythmic values of notes and rests, as well as bar lines of the music piece. Clefs, key signatures and accidentals are adjusted based on the scale of the transposed music and the instruments/voices that parts are assigned to. Our algorithm does not control for instrument timbre that may be expected in any part of the music; similarly, the thickness of the piece is not being necessarily maintained.

We will assume that an input music piece is originally written for  $n$  instruments  $I_1, I_2, \dots, I_n$ , each assigned to play a part  $P_i$  of the piece, with  $1 \leq i \leq n$ . We seek to arrange the music for  $n$  output instruments  $O_1, O_2, \dots, O_n$ . The range of each part  $i$  is an integer interval  $R_i = \llbracket a_i, b_i \rrbracket$ , where  $a_i$  is the integer value corresponding to the lowest frequency note and  $b_i$  to the highest frequency note played by instrument  $I_i$  in

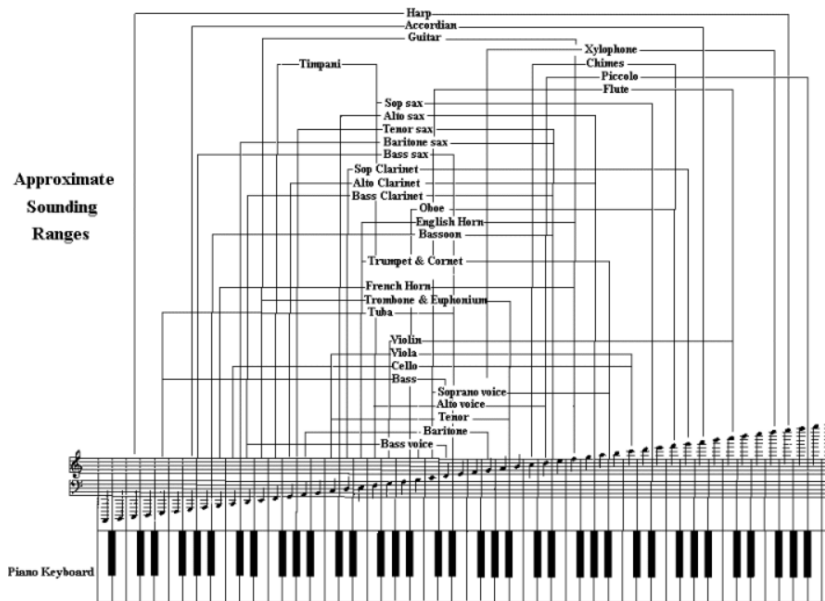


Fig. 1: Approximate sounding ranges of instruments and voices. Figure reproduced with permission from Dr. Brian Blood (dolmetch.com)

part  $P_i$ ,  $1 \leq i \leq n$ . Likewise, the playing range of each output instrument  $O_i$  will be denoted by  $OR_i$ ,  $1 \leq i \leq n$ , indicating the integer interval of values corresponding to the notes the instrument is able to play. Approximate ranges for a set of instruments and voices can be seen in Figure 1.

## 2.2 Monophonic instrument set arrangement algorithm

Our Monophonic Music Arrangement (MMS) algorithm performs a nearly comprehensive search of possible permutations of parts. The music is transposed to all twelve keys, and the algorithm runs on each key, unless a solution has been found so far that results in fewer sharps/flats over all keys for each part. This is designed to prevent the "ideal" transposition from having a complex key signature if not necessary. Other than that, the search is fully comprehensive. For each part, the algorithm finds all possible transpositions of each part in the source piece that can be played by at least one available instrument. All permutations of these possible transpositions are then examined. If all parts can be played by at least one instrument, the algorithm then checks if there exists a set of part assignments that is valid. This is performed by a recursive function that is memoized to improve performance. If a transposed key yields valid permutations, the transposition with the least total deviation from the original composition is selected. Once all twelve keys have been checked, all permutations are tried using the selected transposition, unless there is no selected transposition, in which case the algorithm fails. All permutations are checked, and for those that are valid in the given

transposition, the best arrangement is selected based on how closely the average pitch of each part matches the median pitch of the instrument's range.

The MMA algorithm implementation consists of four main function described in pseudocode below.

---

**Algorithm 1** Find Transposed Options

---

```
procedure FINDTRANPOSEDOPTIONS(originalStream, arrangementParts, semitones)  
  stream  $\leftarrow$  originalStream transposed by given semitones  
  parts  $\leftarrow$  new list  
  for part in stream do  
    choices  $\leftarrow$  new list  
    for each transposition do  
      set  $\leftarrow$  the subset of arrangementParts that can play at this transposition  
      add (semitones + transposition, set) to choices  
    end for  
    if choices is empty then  
      return null  
    end if  
    add choices to parts  
  end for  
  return parts  
end procedure
```

---

---

**Algorithm 2** Run Transposed

---

```
procedure RUNTRANPOSED(stream, parts, semitones)  
  selections  $\leftarrow$  new list  
  for option in all possible transpositions from FINDTRANPOSEDOPTIONS(stream, parts, semitones) do  
    partsCovered  $\leftarrow$  new list  
    selection  $\leftarrow$  new list  
    for transposition in option do  
      add set of parts covered to partsCovered  
      add deviation of transposition to selection  
    end for  
    allPartsCovered  $\leftarrow$  the union of all sets in partsCovered  
    if allPartsCovered contains all parts and ValidateArrangement(parts, partsCovered, allPartsCovered) then  
      add selection to selections  
    end if  
  end for  
  return selections  
end procedure
```

---

---

**Algorithm 3** Find Best Choice

---

```

procedure FINDBESTCHOICE(stream, parts)
  bestChoice  $\leftarrow$  null
  bestSharps  $\leftarrow$   $\infty$ 
  for semitones from  $-6$  through  $5$  do
    sharps  $\leftarrow$  the total number of sharps/flats that would appear in the key signature for
    each part
    if sharps  $\leq$  bestSharps then
      thisBestChoice  $\leftarrow$  element from RUNTRANS-
      POSED(stream, parts, semitones) with the least deviation
      if thisBestChoice  $\neq$  null and either sharps  $<$  bestSharps or deviation of
      thisBestChoice  $<$  deviation of bestChoice then
        bestChoice  $\leftarrow$  thisBestChoice
        bestSharps  $\leftarrow$  thisBestSharps
      end if
    end if
  end for
  return bestChoice
end procedure

```

---



---

**Algorithm 4** MMA Algorithm

---

```

procedure MMA(stream, parts)
  bestChoice  $\leftarrow$  FINDBESTCHOICE(stream, parts)
  if bestChoice = null then
    return null
  end if
  transpose each part by the resulting transposition
  bestFit  $\leftarrow$   $\infty$ 
  for each permutation of newParts do
    if all parts are valid in the given permutation then
      fit  $\leftarrow$  the total absolute difference between the average pitches and the median
      pitch of each part
      if fit  $<$  bestFit then
        bestFit  $\leftarrow$  fit
        bestPermutation  $\leftarrow$  this permutation
      end if
    end if
  end for
  return bestPermutation
end procedure

```

---

### 2.3 Implementation

The MMA algorithm was implemented in Python utilizing the Music21 library and the MuseScore software. Our program requires two input files and produces a single output file with the music arrangement. The required input files consist of the original piece of music in MusicXML format and a TOML file listing the instrument set to arrange for, where an assigned value of  $k$  to an instrument indicates  $k$  parts should be arranged for that instrument. An example of a TOML file with an input instrument set consisting of one clarinet, two tenor saxophones, and two alto saxophones is shown in Figure 2a. Metadata about each instrument, consisting of its key in notation and a reasonable note range, is defined in a separate TOML file which is loaded separately by the program and is populated with common music instruments. An example of an entry for the alto saxophone in the instrument metadata file is shown in Figure 2b.

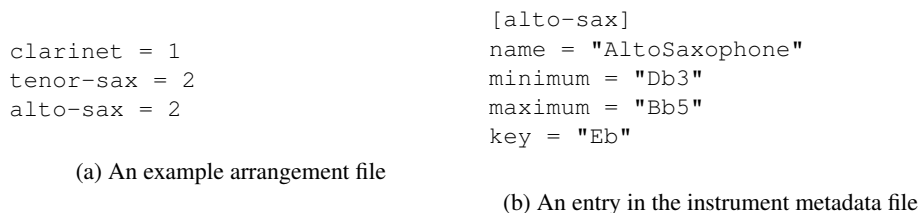


Fig. 2: Examples of input instrument set and instrument information files

During execution our program checks whether the number of input instruments matches the number of parts in the piece, and then attempts to arrange for the given instruments as previously described. If arrangements are found, the best arrangement based on the criteria described in section 2.2 is output as a MusicXML file. If no feasible arrangement is found, or if the number of instruments does not match, then an error message is displayed and no output file is produced.

## 3 Results

We tested our software on a variety of music pieces written for monophonic instruments. In Figure 3 we show three measures, starting at measure 16, of the *Puttin' on the Ritz* song by Irving Berlin. Part (a) shows the input score composed of four monophonic parts. Part (b) displays the arranged piece for saxophone quartet, consisting of a soprano, alto, tenor, and baritone saxophones. Similarly, in Figure 4 we display three measures of *Carol of the Bells*, as arranged and performed by the Pentatonix voice group, starting at measure 18 of the piece.

Complete input/output files for three test cases of our software, including the *Puttin' on the Ritz* and *Carol of the Bells* above, can be examined at:  
<https://owd.tcnj.edu/~papamicd/music/mma/examples/>

The software repository for this project can be found at: <https://github.com/spazzylemons/music-arrangement/>



Figure 3 shows two musical scores for 'Puttin' on the Ritz'. Part (a) is the 'Original Score' for piano, featuring four staves with various dynamics like *mf* and *ff*. Part (b) is the 'Arranged score' for saxophone quartet, featuring four staves for S Sax, A Sax, T Sax, and Bar Sax, with dynamics like *mf* and *ff*.

Fig. 3: Three measures from an arrangement of 'Puttin' on the Ritz' from piano to saxophone quartet

Figure 4 shows two musical scores for 'Carol of the Bells'. Part (a) is the 'Original Score' for piano, featuring four staves with dynamics like *ff*, *mf*, and *f*. Part (b) is the 'Arranged score' for saxophone quartet, featuring four staves for S Sax, A Sax, T Sax, and Bar Sax, with dynamics like *ff*, *mf*, and *f*.

Fig. 4: Three measures from an arrangement of 'Carol of the Bells' from voices to saxophone quartet

#### 4 Conclusions and future work

Our monophonic music arrangement algorithm and its software implementation create a platform for automating music arrangements with minimal user input. Although currently basic in its functionality, it is now being extended in a number of different directions. For accommodating arrangements for a smaller sets of instruments than the number of parts in the music, we are examining score reduction techniques to eliminate certain parts or at least reduce the number of simultaneous notes that are played throughout the piece, while maintaining faithfulness to the original. To allow for the inclusion of polyphonic instruments in the arrangements, we are looking into analyzing and decomposing polyphonic parts into monophonic ones and inversely, while adhering to constraints related to fingerings and other instrument and player restrictions.

## References

1. D. Baker. *David Baker's Arranging & Composing: For the Small Ensemble, Jazz, R & B, Jazz-rock*. Alfred Publishing Company, 1988.
2. Shih Chuan Chiu, Man Kwan Shan, and Jiun Long Huang. Automatic system for the arrangement of piano reductions. In *ISM 2009 - 11th IEEE International Symposium on Multimedia*, 2009.
3. Léopold Crestel and Philippe Esling. Live orchestral piano, a system for real-time orchestral music generation. In *Proceedings of the 14th Sound and Music Computing Conference 2017, SMC 2017*, 2019.
4. Erik D. Demaine and William S. Moses. 364 Computational Complexity of Arranging Music. In *The Mathematics of Various Entertaining Subjects: Research in Games, Graphs, Counting, and Complexity, Volume 2*. Princeton University Press, 09 2017.
5. J.B. Elder. *The Art of Arranging and Orchestration*. Independently Published, 2018.
6. Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama. Input-output HMM applied to automatic arrangement for guitars. *Journal of Information Processing*, 2013.
7. Gen Hori, Yuma Yoshinaga, Satoru Fukayama, Hirokazu Kameoka, and Shigeki Sagayama. Automatic arrangement for guitars using hidden markov model. *Proceedings of 9th Sound and Music Computing Conference (SMC2012)*, pages 450–455, 7 2012.
8. Jiun-Long Huang, Shih-Chuan Chiu, and Man-Kwan Shan. Towards an automatic music arrangement framework using score reduction. *ACM Trans. Multimedia Comput. Commun. Appl.*, 8(1), feb 2012.
9. Hiroshi Maekawa, Norio Emura, Masanobu Miura, and Masuzo Yanagida. On machine arrangement for smaller wind-orchestras based on scores for standard wind-orchestras. In *International Conference on Music Perception and Cognition, ICMPC 2006*, pages 268–273, 2006.
10. Eita Nakamura and Kazuyoshi Yoshii. Statistical piano reduction controlling performance difficulty. *APSIPA Transactions on Signal and Information Processing*, 2018.
11. Sho Onuma and Masatoshi Hamanaka. Piano arrangement system based on composers' arrangement processes. In *International Computer Music Conference, ICMC 2010*, 2010.
12. T.H. Stefan Kostka, T.H. Dorothy Payne, and B. Almén. *Tonal Harmony*. McGraw-Hill Education, 2017.
13. Hirofumi Takamori, Haruki Sato, Takayuki Nakatsuka, and Shigeo Morishima. Automatic arranging musical score for piano using important musical elements. In *Proceedings of the 14th Sound and Music Computing Conference 2017, SMC 2017*, 2019.
14. Moyu Terao, Yuki Hiramatsu, Ryoto Ishizuka, Yiming Wu, and Kazuyoshi Yoshii. Difficulty-aware neural band-to-piano score arrangement based on note- and statistic-level criteria. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200, 2022.
15. D.R. Tuohy and W.D. Potter. Ga-based music arranging for guitar. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1065–1070, 2006.
16. G.C. White. *Instrumental Arranging*. McGraw-Hill, 1992.

# Automatic Orchestration of Piano Scores for Wind Bands with User-Specified Instrumentation

Takuto Nabeoka<sup>1</sup>, Eita Nakamura<sup>1</sup> and Kazuyoshi Yoshii<sup>1\*</sup>

Kyoto University

eita.nakamura@i.kyoto-u.ac.jp

**Abstract.** We present a deep learning method for generating wind band scores with user-specified instrumentation from piano scores. The difficulty in curating large-scale pair data with accurately aligned wind band and piano scores poses two major challenges: (i) efficient preparation of training data and (ii) effective learning of orchestration rules, particularly for infrequently used instruments. We propose using an automatic piano arrangement method to generate pair data from existing wind band scores. Our method utilizes U-Net to assign notes in an input piano score to individual instrument parts, and we propose refined network architectures for efficient learning of characteristics of instrument parts in the wind band scores. We show that the method can generate partially playable scores that capture voicing rules and mutual relationships among instrument parts.

**Keywords:** symbolic music processing; automatic arrangement; orchestration for wind band; deep learning; U-Net.

## 1 Introduction

Wind band is a popular form of musical performance for amateur musicians; numerous schools and communities own wind bands. These bands often have only limited kinds of musical instruments, and the instrumentation may vary from year to year depending on the members' circumstances. Consequently, the repertoire for amateur wind bands is limited because wind band scores in the market tend to be expensive and may be difficult to perform due to discrepancies in the instrumentation of a particular band. This study aims to expand the available repertoire for wind bands by studying automatic orchestration of piano scores, which are relatively easy to obtain, for wind bands with user-specified instrumentation.

Orchestration is a challenging task even for human experts. It requires a high degree of expertise because it must take into account the simultaneous and temporal relationships among dozens of instrument parts, in addition to their pitch ranges and characteristics [1,2]. A previous study developed a method for converting a large wind band score

\* We thank Moyu Terao for cooperation and Hitomi Kaneko for useful discussions. This work was supported by JST PRESTO No. JPMJPR20CB, JSPS KAKENHI Nos. 19H04137, 21H03572, 21K02846, 21K12187, 22H03661, and JST FOREST Program No. JPMJPR226X.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

to a smaller one, by implementing manually constructed criteria for phrase segmentation, instrument group extraction, and instrument assignment [3]. This approach cannot be easily generalized for orchestration of piano scores and it is difficult to manually set up constraints to incorporate all of the aforementioned aspects of expert knowledge. A viable approach is then to use machine learning and infer such constraints from data. Another study explored a spectral-based approach for orchestration [4]. However, it is inappropriate for orchestrating piano scores since spectral features cannot directly capture relevant musical structures such as melody and bass lines.

Recent studies have explored the potential of deep neural networks (DNNs) and statistical models for automatic music generation and arrangement (e.g. [5, 6]). A study attempted automatic orchestration of piano scores using a restricted Boltzmann machine [7]. It was shown that curation of pair data with accurately aligned orchestra and piano scores requires high cost [8]. A more recent study used a Transformer to generate symphonic music using a larger dataset [9]. In these studies, how to control the instrumentation and assure the playability of the output was not focused on. The problem in data curation is even more severe when we allow arbitrary instrumentations because some instruments are much less frequently used in wind band scores than others. Therefore, to train DNNs for converting piano scores into wind band scores, we need to solve two problems: (i) efficient preparation of pair data and (ii) effective learning of orchestration rules, particularly for infrequently used instruments.

To address these problems, we attempt to create pair data by generating piano scores from existing wind band scores using an automatic piano arrangement method [10]. Then, using the U-Net [11], we estimate a mask that determines whether or not to assign each note of the piano score to an instrument part. To improve the quality of infrequently used instrument parts, we propose refined network architectures to effectively use instrumentation information during training and inference. The results are evaluated quantitatively and analyzed in terms of the ability to reproduce the co-occurrence and exclusion relations among instrument parts.

## 2 Method

### 2.1 Problem setup

The input of the proposed method is a piano score consisting of two parts for both hands, and the output is a wind band score with an instrumentation specified by the user. We assume that the user specifies the instrumentation by selecting any number of parts from the maximum instrumentation. Based on several sources of information (e.g. [2]), we define the maximum instrumentation to be consisting of  $N = 43$  parts for 28 commonly used instruments (e.g. clarinet in B $\flat$  has three parts), excluding percussion instruments with no pitch. Abbreviated labels for these 43 instrument parts will be listed in Fig. 4C. Thus, the user-specified instrumentation  $I = (I_n)_{n=1}^N$  is represented by an  $N$ -dimensional binary vector ( $I_n = 1$  indicates that instrument part  $n$  is used).

Each of the two hand parts,  $A_L = (A_L^o, A_L^a)$  and  $A_R = (A_R^o, A_R^a)$ , in the piano score and each instrument part  $B_n = (B_n^o, B_n^a)$  ( $n = 1, \dots, N$ ) in the wind band score are represented by a pair of binary matrices,  $M^o = [M^o(q, t)]$  and  $M^a = [M^a(q, t)]$ ,

representing the onset times and activations for individual pitches, respectively; the number of rows is  $Q = 128$ , same as the number of pitches in the MIDI format, and the number of columns is the length of the piece with  $1/3$  of a 16th note as the unit. For example,  $B_n^o(q, t) = 1$  indicates that instrument part  $n$  has an onset of pitch  $q$  at time  $t$ , and  $B_n^a(q, t) = 1$  indicates that part  $n$  is playing pitch  $q$  at time  $t$ . Thus, the activation matrix  $B_n^a$  represents the piano roll when graphically visualized, and correspondingly, the onset matrix  $B_n^o$  the onset positions. The latter is necessary to represent repeated notes of the same pitch without gaps. For the input and output of the U-Net described below, these matrices are segmented by a time length of  $T = 192$  corresponding to four measures in  $4/4$  time (zero padding is applied for fractional segments).

## 2.2 Preparation of pair data

First, we collected wind band scores in the MusicXML format from a public website (musescore.com). We extracted from the obtained scores only the 43 parts in the maximum instrumentation and used them for the following analysis.

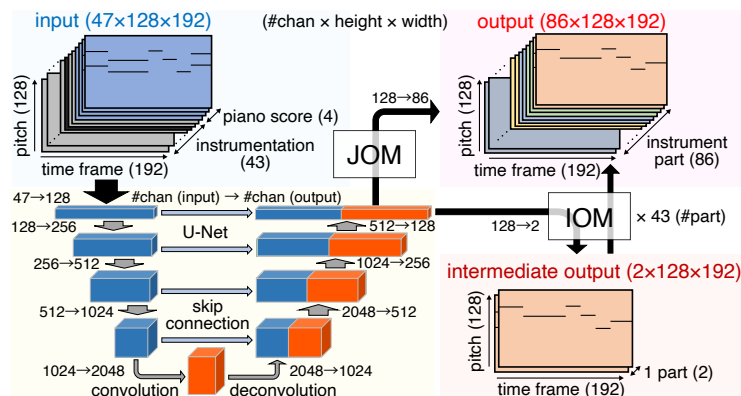
Next, an automatic piano arrangement method [10] was applied to convert these wind band scores to piano scores, thus obtaining pair data with accurately aligned wind band and piano scores that can be used as output and input data for training a DNN. Since this method generates a piano score by selecting some of the notes in an input ensemble score, the notes in the obtained piano score are included in the wind band score. This is a desired property for our method, which generates a wind band score by assigning the notes of the piano score to individual instrument parts.

## 2.3 Network architecture

We formulate the problem of converting a piano score  $A = (A_L, A_R)$  to a wind band score  $B = (B_n)_{n=1}^N$  as the estimation of a mask indicating whether or not to assign the notes of the piano score to each instrument part [7]. We use U-Net [11], which has been successfully applied to mask estimation problems such as singing voice separation [12] and piano reduction [13]. U-Net is an encoder-decoder model that performs feature extraction at multiple levels by a stack of convolution and deconvolution layers (Fig. 1). At each level, the features extracted in the encoder side are concatenated to the decoder side. This is expected to enable processing that captures properties at multiple resolutions in the pitch and time directions.

The output of the U-Net is a set of matrices,  $\tilde{B}_n^o$  and  $\tilde{B}_n^a$  ( $n = 1, \dots, N$ ), each of which corresponds to a binary matrix representing the wind band score. More specifically, for example, the element  $\tilde{B}_n^o(q, t)$  represents the probability that the corresponding element  $B_n^o(q, t)$  of the wind band score have a value 1. The following cross-entropy loss function is used for training:

$$\mathcal{L} = - \sum_{n=1}^N \sum_{q=1}^Q \sum_{t=1}^T \left\{ w B_n^o(q, t) \log \tilde{B}_n^o(q, t) + [1 - B_n^o(q, t)] \log [1 - \tilde{B}_n^o(q, t)] \right. \\ \left. + w B_n^a(q, t) \log \tilde{B}_n^a(q, t) + [1 - B_n^a(q, t)] \log [1 - \tilde{B}_n^a(q, t)] \right\}.$$



**Fig. 1.** Proposed network architecture. The output from the U-Net is differently processed in the joint output method (JOM) and individual output method (IOM).

Here, we introduced a weight  $w$  for positive samples since the onset and activation matrices are generally sparse in our data ( $w = 100$  in our analysis). As we explain below, in the inference step the matrices  $\hat{B}_n^o$  are subjected to thresholding and other post-processes to obtain a wind band score. We consider the following three network architectures with different formats of input and output for the U-Net.

First, in the simple method (SM), Only the piano score  $A$  (4 channels) is used as input, and the maximum instrumentation wind band score  $B_{\text{all}}$  (86 channels) is obtained as output. During training, the loss function is computed using all instrument parts including those not used in each piece. During inference, only the instrument parts used in the specified instrumentation  $I$  are extracted. This method cannot adaptively change the output depending on the specified instrumentation.

Second, in the joint output method (JOM), we add to the input 43 channels of matrices  $C_n = [C_n(q, t)]$  representing the instrumentation  $I$  (Fig. 1). All elements of matrix  $C_n$  are set to one, i.e.  $C_n(q, t) = 1$  for all  $q$  and  $t$ , if instrument part  $n$  is used and  $C_n(q, t) = 0$  if it is not used. During training, we set  $C_n(q, t) = 1$  at all time frames in a piece if instrument part  $n$  plays at least one note in the piece. In this way, the network is trained to learn note assignment including rest intervals. The output form and loss function are the same as those of the SM. This method is expected to be more robust to unbalanced frequencies of use of instrument parts in the training data and to learn the dependence on instrumentation, such as balance among instrument parts.

Third, in the individual output method (IOM), the output is the score  $B_n$  (2 channels) of each instrument part  $n$ , and a single U-Net is used to process all instrument parts. As in the JOM, 43 matrices  $C_n$  representing instrumentation  $I$  are added to the input, but here all the matrices except for the instrument part to be processed are filled with zeros. During training, a loss function is computed for each instrument part in each piece. During inference, the output  $B_n$  for each instrument part  $n$  used in the specified instrumentation is combined to generate a wind band score. With this method, it is difficult to adjust the balance among instrument parts according to the instrumentation  $I$ , but even more efficient learning of infrequently used instruments is expected.

Method	Octave augmentation	Precision	Recall	F-score
SM		29.2	30.1	28.8
SM	✓	32.3	21.1	25.1
JOM		22.0	2.3	3.8
JOM	✓	19.5	6.2	8.4
IOM		<b>33.9</b>	41.3	<b>36.8</b>
IOM	✓	31.9	<b>42.2</b>	35.9

**Table 1.** Average accuracies (%) for the simple method (SM), joint output method (JOM), and individual output method (IOM). The highest values are indicated in bold fonts.

In all of the above three methods, the following processes are applied in the inference step. After thresholding the probability estimates of the onset time matrix of each instrument part, the output score is obtained by selecting only the notes contained in the input piano score and imposing the instrument’s pitch range and monophonic constraint. We use the pitch ranges written in standard books on orchestration. To impose the monophonic constraint, if more than one onset remain as candidates at a time frame, we choose the one with the largest probability. The duration of each note obtained is determined by referring to the input piano score.

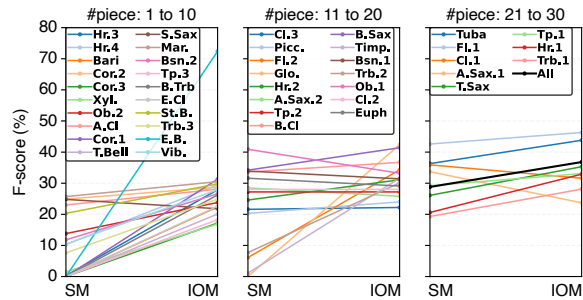
Finally, to generate wind band scores that take advantage of the wide pitch range, it is desired to extend the method and utilize octave-shifted notes from the input piano score. This can be realized in the same mask estimation framework by adding the octave-shifted piano scores,  $A_L^+$ ,  $A_R^+$ ,  $A_L^-$ , and  $A_R^-$ , to the input. For example,  $A_L^{\circ+}(q, t) = A_L^{\circ}(q - 12, t)$  and  $A_R^{\circ-}(q, t) = A_R^{\circ}(q + 12, t)$ . With this octave augmentation, the number of channels in the input increases by 8.

### 3 Result

From the pair data of 110 pieces obtained as in Sec. 2.2, we used randomly selected 80 pieces as training data and the remaining 30 pieces as test data. As evaluation metrics, we used the precisions, recalls, and F-scores for the output scores calculated individually for all instrumental parts with a criterion of exact match of pitch and onset time. The networks were trained by the AdamW optimizer with a learning rate of  $10^{-6}$  for the SM and JOM and  $10^{-7}$  for the IOM, batch size of 32, and dropout ( $p = 0.5$ ) applied to the first two layers of the decoder. A threshold value of 0.5 was used for inference.

The results in Table 1 show that the IOM outperformed the SM in F-scores, confirming the effectiveness of the method using instrumentation information as input<sup>1</sup>. A comparison of the F-scores for individual instrument parts for the SM and the IOM shows that the latter method significantly improved the F-scores, especially for instrument parts that are used infrequently (Fig. 2). On the other hand, the JOM, which was expected to be the most effective, showed significantly lower accuracies, suggesting that the complex network structure may have reduced the learning efficiency. Therefore, for

<sup>1</sup> See also our demo webpage [https://nabeshinabe.github.io/PianoToBrassBand\\_nabeoka/demo.html](https://nabeshinabe.github.io/PianoToBrassBand_nabeoka/demo.html)



**Fig. 2.** Partwise F-scores for the SM and IOM (without octave augmentation), shown in three groups according to the number of pieces in the test data in which each instrument part is used.

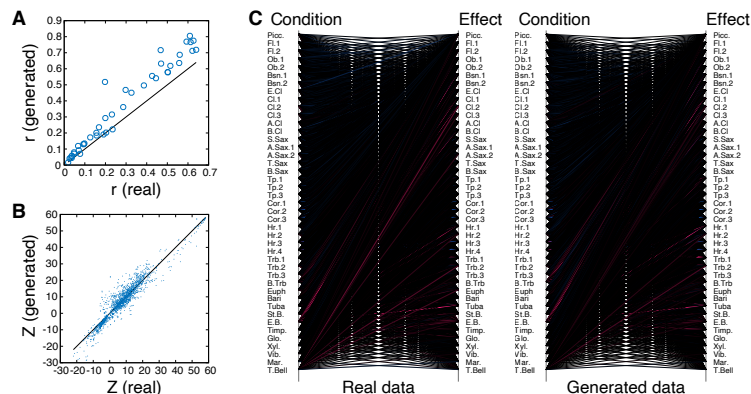


**Fig. 3.** Wind band score generated by the IOM (without octave augmentation) from Joplin's "The Entertainer." All instruments are notated in concert pitches.

the JOM, refinement of the learning method, for example, by improving the optimization method and increasing the amount of data, should further be investigated. As for the effect of octave augmentation, the metrics changed only slightly for the IOM, with an increased recall and decreased precision and F-score.

Fig. 3 shows an output score with seven instrument parts score obtained by the IOM with piece-level instrumentation information. The input was an existing piano score that was not included in our dataset. The three woodwind parts play the notes in the right hand part of the piano score, and the four brass parts mainly play the notes in the left hand part. The voicing of the chords follows the natural order of the parts within each instrument group. This suggests that the method enables orchestration that captures not only the pitch range of each instrument part, but also the characteristics of the instruments and the mutual relationships among the instrument parts. On the other hand, the IOM has limitations that it cannot adaptively change the roles of the instrument parts according to the specified instrumentation and it cannot assign notes from the piano score to each instrument part without omission. In addition, in the second measure of the 2nd Flute, only some notes of the melody are assigned, which is usually judged as inappropriate. Thus, a proper handling of sequential dependencies of notes, which is necessary for generating smoothly playable arrangements, needs to be improved.





**Fig. 4.** Correlations of the sounding rates (A) and of the conditional significances (B) between real data and data generated by the IOM (with octave augmentation). C: Conditional significances between all pairs of instrument parts, with positive and negative significances indicated by blue and red lines, respectively (high significances are indicated in dark colors).

To examine the potential of the IOM for learning the interdependence between instrument parts in a larger time scale, we analyzed the sounding rates of individual instrument parts and their correlations. Let  $h_{mn} \in \{0, 1\}$  represent whether part  $n$  plays at least one note in measure  $m$  ( $h_{mn} = 1$ ) or not ( $h_{mn} = 0$ ). We define the sounding rate  $r_n$  of part  $n$  as  $r_n = \sum_m h_{mn}/M$ , where  $M$  is the total number of measures analyzed. Similarly, we define the simultaneously sounding rate  $r_{nn'}$  of parts  $n$  and  $n'$  as  $r_{nn'} = \sum_m h_{mn}h_{mn'}/M$ . Then, their correlation can be calculated as  $\rho_{nn'} = r_{nn'} - r_n r_{n'}$ , which measures the deviation from the independence hypothesis. The statistical significance of this quantity can be measured by the conditional significance  $Z(n'|n) := (r_{n'n} - r_n r_{n'})\sqrt{M}/\sqrt{r_n r_{n'}(1 - r_{n'})}$ , where we assumed a binomial process for estimating the statistical error. A positive (negative) value of  $Z(n'|n)$  indicates a co-occurrence (exclusion) of part  $n'$  conditioned on the presence of part  $n$ .

Results in Fig. 4 show that both the sounding rates and simultaneous sounding rates were highly correlated between the real and generated data of wind band scores. This indicates that the U-Net trained by the the IOM learned the co-occurrence and exclusion relations between instrument parts. For example, Fig. 4C indicates a co-occurrence of Soprano Sax and Cornet parts, both of which are expected to be used in large bands but not in small bands, and an exclusion relationship between Bass Trombone and Tuba and between 2nd Bassoon and Electric Bass, which are likely to be a result of substitutability of these instrument parts. These properties of wind band scores were reproduced in the data generated by the IOM. We also conducted the same analysis for the SM but did not observed such clear correlations in the data generated by this method, showing the nontriviality of learning these statistical properties.

## 4 Discussion

In this paper, we showed the possibility of training DNNs for automatic orchestration of piano scores for wind bands, by generating pair data only from existing wind

band scores using a method for piano arrangement. The experimental results indicated the ability of the proposed U-Net-based method to learn voicing rules and co-occurrence/exclusion relations among instrument parts, and demonstrated the potential for generating partially playable wind band scores in user-specified instrumentations.

A number of challenges remain for the generation of wind band scores suitable for actual performance. Increasing training data and further refinements of network architectures should be attempted to successfully train the JOM or similar networks that can adaptively change the roles of instrument parts according to the specified instrumentation. To suppress note sequences with unnatural leap motions, rhythms, etc. in the outputs that are difficult to play, use of autoregressive networks, such as a long short-term memory (LSTM) network and Transformer, is expected to be effective. More thorough evaluations by arrangement experts and through actual performance tests of the output results should be conducted in the future.

## References

1. Berlioz, H., Strauss, R.: *Treatise on Instrumentation* (transl. by T. Front). Dover Publications, New York (1991)
2. Newton, B.: *Band Orchestration: Volume 1: Introduction and Orchestration*. CreateSpace Independent Publishing Platform (2016)
3. Maekawa, H., et al.: On machine arrangement for smaller wind-orchestras based on scores for standard wind-orchestras. In: *Proc. Int. Conf. on Music Perception and Cognition*, pp. 278–283. Bononia University Press, Bononia (2006)
4. Cella, C.E.: Orchidea: a comprehensive framework for target-based computer-assisted dynamic orchestration. *Journal of New Music Research* 51(1), 40–68 (2022)
5. Roberts, A., et al.: A hierarchical latent vector model for learning long-term structure in music. In: *Proc. Int. Conf. on Machine Learning*, pp. 4364–4373. ICML, Stockholm (2018)
6. Huang, Y.S., Yang, Y.H.: Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In: *Proc. ACM Int. Conf. on Multimedia*, pp. 1180–1188. ACM, Seattle (2020)
7. Crestel, L., Esling, P.: Live Orchestral Piano, a system for real-time orchestral music generation. In: *Proc. Int. Sound and Music Computing Conf.*, pp. 434–442. Aalto University, Espoo (2017)
8. Crestel, L., et al.: A database linking piano and orchestral MIDI scores with application to automatic projective orchestration. In: *Proc. Int. Society for Music Information Retrieval Conf.*, pp. 592–598. ISMIR, Suzhou (2017)
9. Liu, J., et al.: Symphony generation with permutation invariant language model. In: *Proc. Int. Society for Music Information Retrieval Conf.*, pp. 551–558. ISMIR, Bengaluru (2022)
10. Nakamura, E., Yoshii, K.: Statistical piano reduction controlling performance difficulty. *AP-SIPA Transactions on Signal and Information Processing* 7(e13), 1–12 (2018)
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: *Proc. Int. Conf. on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, Switzerland (2015)
12. Jansson, A., et al.: Singing voice separation with deep U-Net convolutional networks. In: *Proc. Int. Society for Music Information Retrieval Conf.*, pp. 745–751. ISMIR, Suzhou (2017)
13. Terao, M., et al.: Difficulty-Aware Neural Band-to-Piano Score Arrangement based on Note- and Statistic-Level Criteria. In: *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 196–200. IEEE, Singapore (2022)

# **A quantitative evaluation of a musical performance support system utilizing a musical sophistication test battery**

Yasumasa Yamaguchi<sup>1</sup>, Taku Kawada<sup>2</sup>, Toru Nagahama<sup>3</sup> and Tatsuya Horita<sup>3</sup>

<sup>1</sup> Sendai University

<sup>2</sup> Sendai Shirayuri Gakuen Elementary School

<sup>3</sup> Graduate School of Information Sciences, Tohoku University  
ys-yamaguchi@sendai-u.ac.jp

**Abstract.** This article discusses the effects of a pitch feedback system integrated into Google Glass, called the MVP (Musical pitch Visualization Perception) support system, on the musical performance of wind instrumentalists. The study adopted the Goldsmith Musical Sophistication Index (Gold-MSI) to discuss the contribution of the MVP support system to the improvement of musical performance. The Gold-MSI is a popular tool in the field of music research that measures musical sophistication based on observable behaviors. The study reports the effects of the MVP support system from a quantitative standpoint, and the results show that the system had a positive impact on the participants' pitch accuracy.

**Keywords:** ICT, Performance support, Pitch Feedback, Performance evaluation, Quantitative analysis

## **1 Introduction**

The intonation of instrumentalists has been extensively discussed in the literature [1], with correct intonation being viewed as particularly important for novice instrumentalists when performing in an ensemble. Effective methods for improving intonation have been widely studied in music education research [2]. In recent years, many researchers have shown an interest in real-time pitch feedback systems, owing to the development of Information and Communication Technologies (ICT). For instance, Wang et al. [3] investigated the potential of real-time feedback for violinists. This study, however, will focus on wind instrumentalists who perform in a concert band or orchestra.

Instrumentalists must pay attention not only to their intonation but also to other crucial aspects of performance, such as sheet music, rhythm, dynamics, fingering, tempo, expression, ensemble, and conductor cues[4]. Information provided by the conductor is



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

as essential as auditory information. However, during performances, some novice and student instrumentalists tend to rely on their tuner, which is typically placed on their music stand. This habit may cause fixation of gaze, narrowing of visual field, and poor posture. Therefore, Yamaguchi et al. [5,6] proposed a pitch feedback system integrated into Google Glass as a Musical pitch Visualization Perception (MVP) support system to solve the problems mentioned above.

Yamaguchi et al. [5,6] argue that the MVP support system has advantages over the conventional tuner in that it provides flexibility to the instrumentalist's physical and visual angle and reduces cognitive load during performance, according to usability tests and interview surveys. However, these qualitative methods have not escaped criticism, such as subjectivity and generalizability. Furthermore, these kinds of investigations do not fully represent the contribution of the musical performance system. The difficulty of quantitative analysis is an overall problem regarding musical performance research, however, evaluation of musical skill and ability have been conveniently defined by years of experience of musical activity. Yet these ideas should be dealt with as a multifaceted and complex concept. Thus, some of the aforementioned, convenient definitions are unworthy of trust. Controlling for musical sophistication, including skill and ability, is a major problem which still exists in the literature.

In this research, we report the effects of the MVP support system from a quantitative standpoint. To discuss the contribution of the MVP support system to the improvement of musical performance, we adopted the Goldsmith Musical Sophistication Index (Gold-MSI) [7, 15]. It is promising that the level of musical sophistication, such as skill and experience, will greatly affect the effectiveness of musical performance support systems.

## **2 MVP support system**

The MVP support system has been developed for wind instrumentalists. Although, the main concept was a pitch feedback system for them, it would also become a performance support system by utilizing Google Glass. The system can be defined as a glass-type tuner for instrumentalists and shows promise regarding the reduction of physical burden and the improvement of gaze flexibility and cognitive load during musical performances when compared to using a conventional tuner on a music stand [5,6].

The feedback system utilized a three-tier scale rating system of "correct", "higher", or "lower" compared to the correct pitch, with the participant receiving real-time feedback through color indicators on the display. The correct pitch range was defined as the target, expressed in Hertz,  $\pm 1\%$  [5,6]. The Glass Enterprise edition 2 by Google was used in the study. The reliability of the Google Glass system as a musical tuner was verified by a professional musician and the first author, who has experience conducting and training concert bands (for a detailed explanation, see [5]). The system was developed with four key standpoints: timeliness, ease of understanding, recordability, and stability [5]. To achieve this goal, the system utilized the ml5.js library [8], run in TensorFlow, which includes the PitchDetection package that implements the deep-learning-based CREPE algorithm [9,10]. In this study, we also used the Glass Enterprise

edition 2 by Google to send tonal pitch feedback to participants in real time. We also recorded the performance data and pitch estimate data in a CSV file, with the evaluation of the performance pitch stored in a separate column [5]. The schematic drawing of the MVP support system is shown in Figure 1.

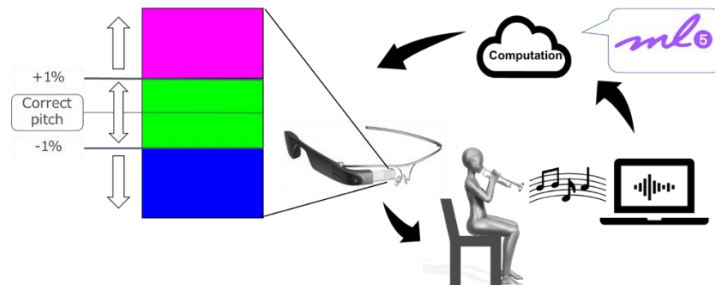


Fig. 1. The schematic of the MVP support system

### 3 Goldsmith Musical Sophistication Index

The Gold-MSI is a popular tool in the field of music research [7]. The term "musical sophistication" was introduced by Müllensiefen and colleagues as a more inclusive and neutral alternative to terms such as "talent" and "aptitude," and is based on observable behaviors. The Gold-MSI is composed of 38 self-report questions divided into five subscales: Active Engagement, Perceptual Abilities, Musical Training, Singing Abilities, and Emotions. These questions were carefully selected from a pool of 153 statements extracted from previous studies. The Gold-MSI has been validated by a large number of primarily English-speaking participants and has demonstrated good internal reliability for each subscale as well as the overall sophistication index, high test-retest reliability, and reliable correlation with a variety of objective listening ability tests.

The Gold-MSI has been widely used since its inception in 2014 and has been translated into various languages including Traditional Chinese, Simplified Chinese, Portuguese, German, and French [11-14]. These translations have shown high internal consistency and test-retest reliability, and the validation data collected from these studies indicate a good fit with the bifactor model structure proposed by the original Gold-MSI. These findings suggest that the structure and set of questions used to measure musical sophistication by the Gold-MSI are applicable to other cultures and languages.

The development of batteries for assessing musical abilities in Japan has resulted in the creation of several tests, including the Onken Musical Aptitude Test for Young Children [16], and the New Musical Aptitude Test [17]. Most of these tests are designed to assess the musical abilities of children. While there are few non-Japanese standardized tests that have been translated into Japanese, some, such as the Bentley Measure of Musical Abilities [18], have been translated and are available for use. However, there is currently no validated Japanese version of the Gold-MSI, which is widely used to assess musical sophistication.

Therefore, Sadakata et al. [15] translated the Gold-MSI into Japanese (Gold-MSI-J) and, after validating the translation with 689 Japanese speakers, it was found that the internal consistency and test-retest reliability were excellent. Furthermore, the confirmatory factor analysis showed that the bifactor model structure proposed by the original study of Gold-MSI is reasonably maintained in the data.

## **4 Method**

### **4.1 Participants**

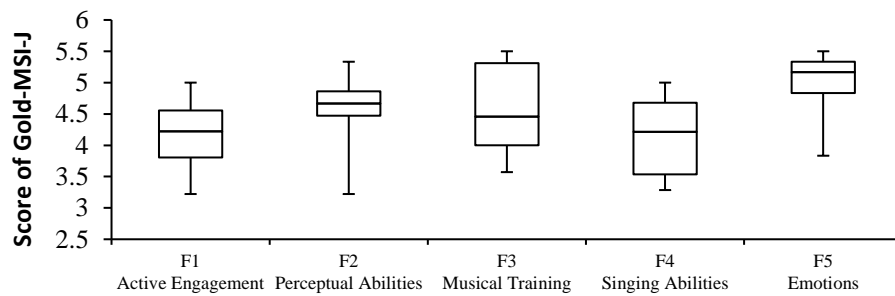
We conducted the experiment which consisted of 22 student participants with an average age of 22.14 years ( $SD=1.39$ ). Students were selected from national universities' concert band and orchestra clubs. Their musical ability as wind instrumentalists was confirmed to be at an intermediate or advanced level by the Japanese version of Goldsmith Musical Sophistication Index (Gold-MSI-J) [15]. Their musical training factor score in Gold-MSI-J was 3 or more. The participants were 4 clarinet players, 4 trumpet players, 4 flute players, 3 French horn players, 2 saxophone players, 2 trombone players, 1 oboe player, and 1 bassoon player.

### **4.2 Procedure**

The study comprised two distinct sections, namely a performance task section and a questionnaire and interview section. In the performance task section, the participants were instructed to perform the B-flat equal temperament major scale in long-tone while synchronizing with a metronome presented on the screen in front of them, set to a beat per minute (BPM) of 60. This long-tone scale task is the fundamental performance for wind instrumentalists in Japanese concert band societies. To gauge the efficacy of the Google Glass tuner system, a commercially available conventional tuner (YAMAHA TDM-70) which was placed on a music stand, was employed as a baseline. The order effects were counterbalanced for the performance task section. In both the baseline and Google Glass settings, the participants were advised to focus on their tempo and intonation while performing the task. There was no repetition and the duration of the experiment was approximately 30 minutes including explanation and warming up. After the performance section, we conducted a questionnaire survey regarding the musical sophistication using Gold-MSI-J. Finally, we ensured that all measurements were made using the same microphone, computer, browser, network environment and Google Glass system.

## **5 Results**

Figure 2 shows the boxplots regarding the scores of the Gold-MSI-J subscales. Compared to the results of Sadakata et al. [15], these scores clearly show that musical sophistication of the participants is much higher than laypeople.



**Fig. 2.** The boxplots regarding the results of Gold-MSI-J

**Table 1.** List of participants and pitch accuracy data

ID	Instrument	Conventional tuner	Google Glass
1	Clarinet	78.52%	81.87%
2	French Horn	62.26%	91.93%
3	Euphonium	73.61%	86.70%
4	Trombone	75.65%	76.91%
5	Oboe	94.52%	96.58%
6	Flute	98.95%	97.75%
7	Trumpet	91.79%	95.16%
8	French Horn	84.67%	82.78%
9	Clarinet	54.88%	62.58%
10	French Horn	82.36%	92.86%
11	Flute	87.49%	96.18%
12	Trombone	54.66%	72.18%
13	Trumpet	85.56%	89.90%
14	Trumpet	93.68%	87.45%
15	Saxophone	71.54%	82.78%
16	Flute	95.32%	91.34%
17	Flute	97.63%	95.61%
18	Saxophone	87.59%	88.24%
19	Clarinet	94.70%	94.22%
20	Bassoon	95.46%	96.07%
21	Clarinet	94.37%	97.65%
22	Trumpet	76.15%	85.28%

The median score of each factor in Gold-MSI-J was used as the threshold to divide participants into low and high groups for each factor, which were then used as between-

participant factors. The tuner use condition and the Google Glass use condition were defined as within-participant factors. Two-way repeated measures analysis of variance, for the effects of musical sophistication and the tuner system condition, was conducted on the pitch accuracy during the performance. Pitch accuracy was estimated by the recorded data of the MVP support system and transformed to angle data before analysis because it was ratio data by arcsine transformation. Table 1 shows the instrument of each participant and the pitch accuracy under both conditions. The accuracy rate was calculated by dividing the length of time the performed pitch was within the correct pitch range by the total length of the performance.

The results show that there was no significant main effect between participant groups for the Active Engagement factor ( $F(1,20)=0.31$ , *ns.*,  $\eta_p^2=0.02$ ), and only the main effect of tuner system factors was significant ( $F(1,20)=9.68$ ,  $p<.01$ ,  $\eta_p^2=0.33$ ). There was no significant interaction between Active Engagement factor and tuner system factors ( $F(1,20)=2.68$ , *ns.*,  $\eta_p^2=0.19$ ). For the Perceptual Abilities factor, there was no significant main effect between participant groups ( $F(1,20)=1.90$ , *ns.*,  $\eta_p^2=0.09$ ), and only the main effect of tuner system factors was significant ( $F(1,20)=5.27$ ,  $p<.05$ ,  $\eta_p^2=0.21$ ). There was no significant interaction between Perceptual Abilities factor and tuner system factors ( $F(1,20)=3.27$ , *ns.*,  $\eta_p^2=0.14$ ). For the Musical Training factor, both the main effect between participant groups ( $F(1,20)=6.04$ ,  $p<.05$ ,  $\eta_p^2=0.23$ ) and the main effect of tuner system factors ( $F(1,20)=9.96$ ,  $p<.01$ ,  $\eta_p^2=0.31$ ) were significant. There was no significant interaction between Musical Training factor and tuner system factors ( $F(1,20)=2.44$ , *ns.*,  $\eta_p^2=0.11$ ). However, this factor can be hypothesized that it has an effect of the performance, we conducted exploratory testing of simple main effects. The result showed that pitch accuracy increased significantly only when the Google Glass tuner was used in the low group of between-participants factor ( $F(1,20)=8.92$ ,  $p<.01$ ,  $\eta_p^2=0.47$ ). For the Singing Abilities factor, there was no significant main effect between participant groups ( $F(1,20)=1.26$ , *ns.*,  $\eta_p^2=0.06$ ), and only the main effect of tuner system factors was significant ( $F(1,20)=7.43$ ,  $p<.05$ ,  $\eta_p^2=0.27$ ). There was no significant interaction between Singing Abilities factor and tuner system factors ( $F(1,20)=2.68$ , *ns.*,  $\eta_p^2=0.03$ ). For the Emotions factor, there was no significant main effect between participant groups ( $F(1,20)=0.32$ , *ns.*,  $\eta_p^2=0.02$ ), and only the main effect of tuner system factors was significant ( $F(1,20)=6.32$ ,  $p<.05$ ,  $\eta_p^2=0.24$ ). There was no significant interaction between Emotions factor and tuner system factors ( $F(1,20)=2.26$ , *ns.*,  $\eta_p^2=0.10$ ).

For all analyses, the main effect of tuner system factors was significantly higher for Google Glass tuner conditions than for tuner conditions in terms of pitch accuracy. The main effect between participant groups for the Musical Training factor showed that the high group had significantly higher pitch accuracy than the low group. For this factor, the result of the test of simple main effects showed that the Google Glass conditions significantly improved pitch accuracy only in the low group.



## 6 Discussion

When using the Google Glass tuner, the accuracy of participant's pitch showed good values. Regarding the relationship with Gold-MSI-J, participants were divided into high and low groups based on their factor scores, and for all factors of Gold-MSI-J, the main effect of the tuner system was significant. On the other hand, the main effect between participant groups was significant only for the Musical Training factor. Furthermore, as a significant difference in pitch accuracy was observed only in the low group in the simple main effect test, it is suggested that the Google Glass tuners may have a strong effect as a performance support system for individuals with relatively low levels of musical training among instrument players. In the high musical training group, pitch accuracy was sufficiently high even under tuner use conditions, and there was no significant difference between the two conditions. However, the sample size is relatively small, so the statistical analysis was conducted with the aim of obtaining an overview of the data. Thus, we will have to conduct large-scale experiments for valid discussions, for example, the influence different instruments have on the results and the usability of the system. It will be also necessary to rely on qualitative analysis or subjective evaluations of participants, as described below, to understand how this system contributes to such individuals.

The results of this study show the effect of the MVP support system from each factor of musical sophistication by Gold-MSI-J. Compared with existing qualitative surveys, this method allowed a valid investigation regarding the individual performance and musical sophistication from multifaced standpoints. It should be noted that the present study also emphasized the effectiveness of the MVP support system. Further work in this area is underway to develop an effective pitch feedback system that is suitable to each instrumentalist. It is hoped that the outcome of this study will be of use for future empirical research regarding musical performance study.

## References

1. Madsen, C. K. , & Geringer, J. M. Preferences for trumpet tone quality versus intonation. *Bulletin of the Council for Research in Music Education*, 46, 13–22 (1976).
2. Elliott, C. A. Effect of Vocalization on the Sense of Pitch of Beginning Band Class Students. *Journal of Research in Music Education*, 22(2), 120–128. (1974).
3. Wang, J. H., Wang, S. A., Chen, W. C., Chang, K. N., & Chen, H. Y. *Real-time pitch training system for violin learners*. In 2012 IEEE International Conference on Multimedia and Expo Workshops. 163-168) IEEE. (2012, July).
4. Drake, C., & Palmer, C. Skill acquisition in music performance: relations between planning and temporal control. *Cognition*, 74(1), 1-32. (2000).
5. Yamaguchi, Y., Kawada, T., Nagahama, T., & Horita, T. A Pilot Study of the MVP Support System using Google Glass. In EdMedia+ Innovate Learning. 17-28. Association for the Advancement of Computing in Education (AACE). (2022, June).
6. Yamaguchi, Y., Kawada, T., Nagahama, T., & Horita, T. Development and Evaluation of a Musical Instrument Performance Support System Using Smart Glasses: from the Subjective

- Evaluation of Form, Gaze, and Performance. *Japan journal of educational technology*, 46, 185-188. (2023).
7. Müllensiefen, D., Gingras, B., Musil, J., & Stewart, L. The musicality of non-musicians: An index for assessing musical sophistication in the general population. In J. Snyder (Ed.), *PLoS ONE* (Vol. 9, Issue 2, p. e89642). <https://doi.org/10.1371/journal.pone.0089642>. (2014).
  8. NYU. ITP “*ml5js-Friendly Machine Learning for the Web*.” ml5js website. Accessed March 16, 2022. <https://ml5js.org/>, last accessed 2023/5/4.
  9. Kim, J. W., Salamon, J., Li, P., & Bello, J. P. *Crepe: A convolutional representation for pitch estimation*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 161-165). IEEE. (2018).
  10. Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. *Yaafe, an easy to use and efficient audio feature extraction software*. In Ismir. (2010).
  11. Degraeve P., Dedonder J. A French translation of the Goldsmiths Musical Sophistication Index, an instrument to assess self-reported musical skills, abilities and behaviours. *Journal of New Music Research*, 48(2), 138–144. <https://doi.org/10.1080/09298215.2018.1499779> (2019).
  12. Lima C. F., Correia A. I., Müllensiefen D., Castro S. L. Goldsmiths Musical Sophistication Index (Gold-MSI): Portuguese version and associations with socio-demographic factors, personality and music preferences. *Psychology of Music*, 48(3), 376–388. (2020).
  13. Lin H. R., Kopiez R., Müllensiefen D., Wolf A. The Chinese version of the Gold-MSI: Adaptation and validation of an inventory for the measurement of musical sophistication in a Taiwanese sample. *Musicae Scientiae*, 25(2), 226–251. (2021).
  14. Schaal N., Bauer A.-K., Müllensiefen D. Der Gold-MSI: Replikation und Validierung eines Fragebogeninstrumentes zur Messung Musikalischer Erfahrungheit anhand einer deutschen Stichprobe. [The Gold-MSI: Replication and validation of a survey instrument for measuring musical sophistication with a German sample.] *Musicae Scientiae*, 18(4), 423–447. [10.1177/1029864914541851](https://doi.org/10.1177/1029864914541851). (2014).
  15. Sadakata, M., Yamaguchi, Y., Ohsawa, C., Matsubara, M., Terasawa, H., von Schnehen, A., Müllensiefen, D., & Sekiyama, K. The Japanese translation of the Gold-MSI: Adaptation and validation of the self-report questionnaire of musical sophistication. *Musicae Scientiae*, 0(0). <https://doi.org/10.1177/10298649221110089>. (2022).
  16. Ongaku Shinri Kenkyu Sho. *Onken-shiki Youji no ongaku tokusei tesuto [Onken’s Musical Aptitudes Test for Young Children]*. Nihon Bunka Kagakusha Co. Ltd. (1969).
  17. Ogawa Y., Murao T., Mang E. H. S. Developing a music aptitude test for school children in Asia [Paper presentation], 10th International Conference for Music Perception and Cognition, Hokkaido, Japan. (2008).
  18. Furuichi H., Umemoto T. Bentley ongaku nouryoku tesuto no hyoujun ka [The translation and standardization of the Bentley Measure of Musical Abilities]. *Journal of the Musicological Society of Japan*, 21, 65–77. (1975).
  19. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
  20. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
  21. LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2016/11/21.

# SBERT-based Chord Progression Estimation from Lyrics Trained with Imbalanced Data

Mastuti Puspitasari<sup>1</sup>, Takuya Takahashi<sup>1</sup>, Gen Hori<sup>2</sup>,  
Shigeki Sagayama<sup>1</sup>, Toru Nakashika<sup>1</sup>, \*

<sup>1</sup> Department of Computer and Network Engineering  
The University of Electro-Communications, Tokyo 182-8585, Japan

<sup>2</sup> Department of Data Science, Faculty of Business Administration  
Asia University, Tokyo 180-8629, Japan  
m2131179@gl.cc.uec.ac.jp

**Abstract.** In this research, we developed a model that can estimate appropriate chord progression based on lyrics input. It outputs a sequence of chord that can be used to compose the corresponding lyrics input. By training the model with different datasets, it is also possible to estimate other musical components that are correlated with lyrics, for example rhythm pattern, instrument, tempo, and drum pattern. Using this set of musical components as a setup recommendation for composition can potentially automate the configuration process on AI-based composition tools. We sourced our training data from “Orpheus”, a web-based automatic composition system, resulting in more than 6,000 paired data of lyrics and musical components chosen by users who published their songs in the platform. Lyrics are pre-processed into semantics embedding using Sentence-BERT before being fed as training data into the multi-layer perceptron model as a classifier to estimate chord progression. Evaluation of this model is done objectively with ROC and F1 score, and subjectively through a survey.

**Keywords:** chord progression estimation, lyrics pre-processing, musical components, automatic composition, Orpheus, semantics embeddings, Sentence-BERT, multi-layer perceptron

## 1 Introduction

Following the recent trend in AI research, there have been tools (eg: soundraw.io, Orpheus [1]) developed to automate music composition. They depend on user input to generate music, some by asking users to select genre or mood, while others expect more detailed input such as lyrics and chord progression. The simpler a tool is, the more attractive it is to new users, but unfortunately, the output will never be as personal as the input is limited. On the other hand, while a more complex tool can result in more personalized music, it can be overwhelming for new users.

\* This research was funded by Grant-in-Aid for Scientific Research (B) No. 21H03462 from Japan Society for the Promotion of Science (JSPS) and a scholarship from The Ministry of Education, Culture, Sports, Science and Technology (MEXT) Japan with the cooperation of fellow members of Nakashika Laboratory, The University of Electro-Communications (UEC).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Ideally, such tedious process should be presented with an offer of automated assistance. The easiest approach to this would be by recommending randomly selected setup during configuration. However, this can possibly result in music that does not match the lyrics. The system may end up recommending an upbeat musical configurations when a user inputs sad lyrics, for example. As mood and nuance can be inferred from text, we argue that it should be possible to estimate appropriate musical compositions based on lyrics input. To achieve this, we decided to experiment with a number of classifier models and train them on relevant training data. We use semantics embeddings of lyrics as input and estimate the appropriate chord progression and other musical components based on what they learned from the training data.

Fortunately, such data can be extracted from existing compositions, as long as the necessary musical components data are also accessible. Even with appropriate training data, however, estimating appropriate musical components is not that straightforward. Since music is not strictly derivable from lyrics, there will never be one true exact match of a composition setup for a specific lyrics input. In fact, we cannot say that any setup is wrong at all, considering that one lyrics input can potentially result in various compositions that can equally be considered as good matches. For this reason, subjective approach is also necessary to evaluate the model performance.

By automating the selection process based on lyrics input, we offer a solution that can leverage a tedious process to be more user-friendly, and thus, encourage existing or potential users to use the system to compose more music. The data of future compositions can also be used to further train the system and improve its performance, allowing the system to evolve over time.

## **2 Related Works**

Our work was initially inspired by [2] in which Turkish lyrics are used to estimate the meta-data of the song, which includes: genre, authors, and year of publication. Similar studies had also been done on genre classification for lyrics in different languages. In [3] for example, an approach similar to [2] is applied on Nordic lyrics. These works were done with conventional approach using feature-based text pre-processing.

In [4], word2vec [5] is used to pre-process the lyrics. Their goal was to estimate chord progression based on lyrics using the data extracted from Orpheus, which then made it the base of this research. It is unfortunate that their model was of a low accuracy, but we argue that it is expected as they included all chord progressions available on Orpheus regardless the number of samples. It is not ideal to train a model to classify a class with insufficient number of samples as it will result in overfitting. To ensure that each class has enough samples for training, we decided to focus our research on the top 10 chord progression available on Orpheus.

Another problem with this approach is that using word2vec to pre-process lyrics means the semantics of the sentence is not considered, as it is meant to be used for word pre-processing. Different lyrics that consist of the same words will result in the same embedding despite the order, for example "king likes queen" shares the same embedding as "queen likes king". To consider the semantics of the lyrics, we decided to take a more state-of-the-art approach for the lyrics pre-processing by utilizing a language model that is able to directly derive embedding from sentences.

Fortunately, many language models have been developed in recent years. An example of this would be BERT [6], which is designed to pre-train deep bidirectional representations from unlabeled text. Several task specific modifications have also been done on BERT, including Sentence-BERT [7], which can be used to quickly measure similarity between two or more sentences, which would originally take hours for BERT to compute. By using SBERT, we convert our lyrics data into their semantics embeddings, which can then be paired with chord progression or other relevant musical components data and used to train our multi-layer perceptron models. Considering that it has been proven possible to infer genre from lyrics by [2], [3], and other studies, we argue that it should also be possible to infer specific musical components based on lyrics input.

### 3 Dataset

To train our models, we extracted composition data of the published songs in Orpheus [1], a Japanese automatic composition system with over 700,000 pieces composition generated by their users. As shown in table 1, this data consists of lyrics, musical components, and several statistics in regard to the composition. According to [8], chord progression can be used to infer music emotion, which has been proven by [10] to be derivable from lyrics. In Orpheus, there are over 1500 variations of chord progression to choose from, available for view on page<sup>1</sup>.

**Table 1:** Raw Data Sample of a Published Composition in Orpheus

Lyrics	Chord	Rhythm	Instr.	Tempo	Drum	#Likes	#Bms
からまつの林を過ぎて、 からまつをしみじみと見き。 からまつはさびしかりけり。 たびゆくはさびしかりけり。	Pachelbel- Kanon	sync- auf-3- 8sf	48	100	perc- hirata- rocknroll2	114	3

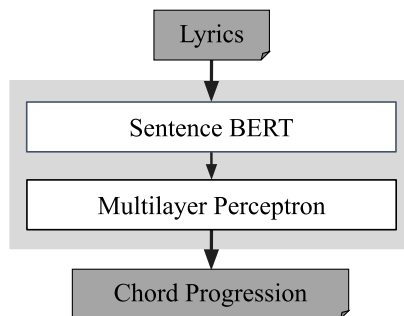
We extracted paired data of lyrics and chord progression for our main experiment and truncated our dataset by only taking samples of the top 10 chord progression to avoid overfitting. Experiment results of the other musical components will be included as ablation studies to consider potential future research.

### 4 Proposed Model

Conceptually, our system takes lyrics as input and outputs a recommendation of chord progression. To achieve this, we convert lyrics into semantics embedding with SBERT models pre-trained with Japanese corpus before feeding them into a multi-layer perceptron model as training data. The conversion from lyrics into numerical embedding is necessary because computers do not understand the meaning of words. This conversion allows computers to assign values to lyrics and understand which lyrics are similar or different based on their numerical representations.

For comparison purpose, we also rebuilt the word2vec model as proposed in [4] with the Japanese corpus used by the SBERT model. The pre-processing results of these models differ in terms of dimension, with a size of 768 for the SBERT model and 50 for the word2vec model which affects the input layer size of the multi-layer perceptron model used for chord progression estimation as shown in Fig. 1:

<sup>1</sup> <https://www.orpheus-music.org/Orpheus-lib-harmony.php>

**Fig. 1:** Estimation Model Architecture

In [4], there was no mention of using a specific loss function on the training phase. For this reason, we used categorical cross-entropy to rebuild the word2vec mode, which is defined as follows, where  $p_i$  is the softmax probability of the  $i^{th}$  class:

$$L_{CE} = - \sum_{i=1}^n \log(p_i) \quad (1)$$

Unfortunately, applying this loss function to train an estimation model with imbalanced training data will likely result in overfitting. To mitigate this issue, we attempted a different approach that is based on [9], which claimed that applying focal factor  $(1 - p_t)^\gamma$  can help to balance the weight of easy and hard samples and thus minimize the overfitting problem. Focal loss is calculated as follows:

$$L_{FCE} = - \sum_{i=1}^n (1 - p_i)^\gamma \log(p_i) \quad (2)$$

## 5 Experiments

### 5.1 Training with the Top 10 Chord Progression Dataset

In this section, we will discuss the result of our experiments on top 10 chord progression in terms of having the highest the number of samples. This was extracted from published Orpheus data with number of samples as shown in Table 2.

**Table 2:** Top 10 chord progression in published Orpheus Data

Label	#Samples
pattern O	1121
pattern FF	947
pattern Q	606
pattern P	570
pattern H	567
pattern E	539
pattern W	508
pattern R	402
Pachelbel Kanon Ending	394
User Harmony zkrxx7	388

Looking at the table, it is clear that there is a big difference in number of samples between the labels, showing an imbalance in data. Note that these labels represent different sequence of chords and not the chord progression itself. Refer to the link provided in section 3 for the full list of chord progression available on Orpheus.

## 5.2 Lyrics Pre-Processing and Loss Function

We experimented with the pre-processing using Japanese SBERT model and compare it with the word2vec model. The multi-layer perceptron models are also trained with two different cross-entropy (CE) loss functions, resulting in a total of four model variations: Word2Vec Categorical CE (WC), Word2Vec Focal CE (WF), Japanese SBERT Categorical CE (JC), and Japanese SBERT Categorical Focal CE (JF). They were trained with the top 10 chord progression dataset in 1000 epochs, with the ratio of 8:1:1, for training, validation, and test data respectively.

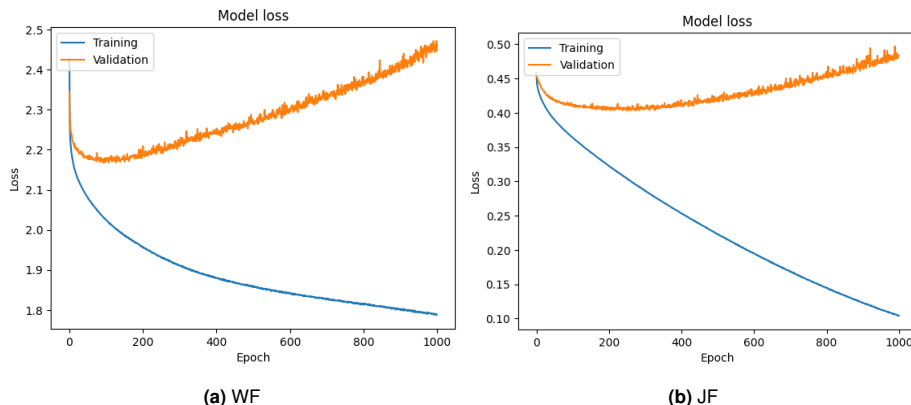
## 5.3 Final Accuracy and Overfitting of the Models on Chord Progression

We have compiled the final accuracy on both training (T) and validation (V) of each model in Table 3. We can see that using SBERT model pre-trained with the Japanese corpus results in higher accuracy (JC and JF) compared to those of word2vec (WC and WF). Note that due to the dataset unavailability, the word2vec model was pre-trained with a newer version of the Japanese corpus, and despite having this advantage, it was not able to achieve comparable accuracy values.

**Table 3:** Final training (T) and validation (V) accuracy of the 4 models

Model	T. Acc.(%) $\uparrow$	V. Acc.(%) $\uparrow$
WC	37.3	21.4
JC	<b>96.0</b>	<b>31.2</b>
WF	32.7	23.4
JF	<b>80.7</b>	<b>31.1</b>

In Fig. 2, we can see that the models with categorical CE (WC and JC) are overfitting, and this can be minimized by applying focal CE during training (WF and JF) as shown in Fig. 3. Note that while JF seems to overfit badly based on the graph, the loss value is still below 0.5, which is still not to far off of WF.



**Fig. 2:** Loss over time of the 2 models trained with CE

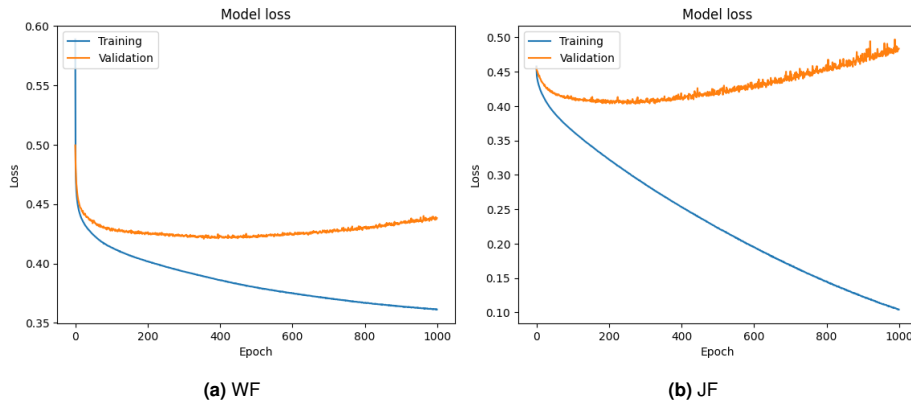


Fig. 3: Loss over time of the 2 models trained with focal CE

Judging from both the accuracy and overfitting, it is safe to say that JF can replicate human preference in chord progression based on lyrics better than the other models, according to the data taken from composition published in Orpheus.

#### 5.4 Objective and Subjective Evaluation

We evaluate the proposed model JF objectively by comparing its ROC AUC and F1 score with the word2vec approach WC and subjectively through survey to evaluate the quality of the generated music, in which a random composition published in Orpheus is recomposed with the chord progression recommended by JF and WC. We asked 10 respondents to score them based on how well the music matches the lyrics on a Likert scale from 1 (bad) to 5 (good). Note that the original composition is used as the ground truth in objective evaluations, and thus the lack of scores in Table 4.

Table 4: ROC AUC, F1, and Likert scores on chord progression

Model	ROC AUC (%)	F1 (%)	Likert (ave.±dev.)
Original	-	-	<b>3.5</b> (±1.08)
WC	64.2	23.4	2.7(±1.16)
JF	<b>69.6</b>	<b>32.1</b>	2.8(±1.03)

JF managed to get higher ROC AUC and F1 scores compared to WC. The Likert score of JF is also higher than WC with the lowest deviation. It can be concluded that using semantics instead of word embedding and changing the loss function to minimize overfitting result in better performance of the models in terms of recreating human preference and selecting the proper chord progression based on lyrics input.

#### 5.5 Ablation Studies

To see the potential of applying this approach on other musical components, we considered four other subjects: rhythm pattern, instrument, tempo, and drum pattern. In [11], rhythm patterns were used to generate lyrics, which led us to believe that the opposite can also be done. Instruments are generally chosen by a composer according to the genre of music they are trying to produce. There is a typical tendency in tempo according to the genre of music as mentioned in [12]. Lastly, drum patterns in Orpheus were created with regards to musical genre with some variations.



As they are correlated with genre which is derivable from lyrics, it may be possible to derive them straight from lyrics. We experimented on top 10 dataset of these subjects with WC and JF and have compiled the evaluation result in Table 5. More details on these experiments, including the number of samples of each class in the top 10 datasets are available on their respective sheet in this spreadsheet <sup>2</sup>.

**Table 5:** Models evaluation on the other 4 musical components

Musical Component	Model	ROC AUC (%)	F1 (%)	Likert (ave.±dev.)
Rhythm Pattern	Original	-	-	2.8(±1.14)
	WC	59.8	35.8	3.0(±1.41)
	JF	<b>67.4</b>	<b>38.2</b>	<b>3.0</b> (±1.15)
Instrument	Original	-	-	2.3(±1.25)
	WC	60.2	25.1	2.0(±0.94)
	JF	<b>65.5</b>	<b>28.8</b>	<b>3.0</b> (±1.56)
Tempo	Original	-	-	<b>3.6</b> (±1.17)
	WC	58.7	23.3	3.2(±1.03)
	JF	<b>66.2</b>	<b>28.9</b>	3.4(±1.35)
Drum Pattern	Original	-	-	<b>2.9</b> (±0.88)
	WC	56.0	21.8	2.7(±0.82)
	JF	<b>61.5</b>	<b>24.5</b>	2.7(±1.06)

Table 5 shows that JF is consistently superior than WC in terms of ROC AUC and F1 score. In the survey, it is also generally better in terms of performance compared to WC, with the exception on drum pattern. However, the drum pattern survey data shows that respondents were unsure of the sample difference and not confident in their answers. It can be concluded that JF performs better than WC when there is clear differences between the samples and respondents are confident. Another interesting point that is worth mentioning here is that on rhythm pattern, both WC and JF scored higher than the original composition, which shows the potential of these models in recommending appropriate musical components based on lyrics input.

## 6 Discussion

The proposed model JF managed to achieve higher ROC AUC, F1, and Likert score on chord progression estimation in comparison to the model WC as proposed in [4], and it is interesting that with similar approach, similar results are also reflected on other musical components, although with some degrees of deviation. However, the ROC AUC and F1 scores of the proposed model JF are still considerably low and mixed results can be seen on the survey. As the training is done by labelling to represent each class, similarities between each class are not considered.

By considering the feature similarities that are unique to each musical component, we argue that it is possible to achieve higher ROC AUC, F1, and Likert score of the classifier model. Chord sequence, for example, may be processed better with seq2seq approach instead of considering each sequence of chord as an entirely different class, rhythm pattern can be labeled with their individual notes, instrument can be grouped according to their similarities in terms of timbre, and so on.

<sup>2</sup> <https://docs.google.com/spreadsheets/d/16-MMdycFS2SN44hR5kFLermyNNquK3hHwhs4Lou3kY8/edit?usp=sharing>

## 7 Conclusion and Future Works

In this paper, we proposed a an approach to estimate chord progression, and potentially other specific musical components based on lyrics input by using SBERT model for lyrics pre-processing instead of word2vec as proposed in [4]. We also consider the imbalance in data and limit our scope by using top 10 dataset as training data. During training, focal cross-entropy is applied instead of cross-entropy loss function to mitigate the overfitting caused by the difference in number of samples between the classes.

The proposed model achieved higher ROC AUC and F1 score in comparison to the model proposed in [4]. Through a survey that compares audio samples configured with the two models and the original composition, it can be concluded that the proposed model generally performs better than the previous model, and can potentially generate music better than the original work in terms of how well they match the lyrics input. The proposed model can also be potentially improved by considering similarities between each class and features that are unique to each musical component.

## References

1. Sagayama, S.: Orpheus : An Automatic Music Composition System. In: The journal of the Institute of Electronics, Information and Communication Engineers, pp.214–220. The Institute of Electronics, Information and Communication Engineers (2019)
2. Oğul, H., Kırmacı, B.: Lyrics Mining for Music Meta-Data Estimation. In: 12th IFIP International Conference on Artificial Intelligence Applications and Innovations, pp.528–539. HAL open science, Greece (2016)
3. de Lima, A., Nunes, Rodrigo M., Ribeiro, Rafael P., Silla, Carlos N.: Nordic Music Genre Classification Using Song Lyrics. In: Natural Language Processing and Information Systems, pp.89–100. Springer International Publishing, Cham (2014)
4. Shinohara, K.: Automatic Chord Progression Setting Considering the Meaning of Japanese Lyrics in Automatic Composition. Meiji University Graduation Thesis, Meiji University (2018)
5. Mikolov, T., Chen K., Corrado, G., Dean, J.: “Efficient Estimation of Word Representations in Vector Space”, arXiv (2013).
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Computing Research Repository (CoRR), arXiv (2018)
7. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Computing Research Repository (CoRR), arXiv (2019)
8. Cho, Y.H., Lim, H., Kim D.W., Lee, I.K.: Music emotion recognition using chord progressions. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.2588–2593 IEEE (2016)
9. Lin, T.Y., Goyal, P., Girshick ,R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: Facebook AI Research (FAIR), arXiv (2018)
10. Edmonds, D., Sedoc J.: Multi-Emotion Classification for Song Lyrics. In: Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp.221–235. Association for Computational Linguistics (2021)
11. Oliveira, H.R.G., Cardoso, F.A., Pereira, F.C.: Tra-la-Lyrics: An approach to generate text based on rhythm. In: Computational Creativity 2007, pp.47–54. University of London, Goldsmiths (2005)
12. Wolf, T.: Genre Classification of Electronic Dance Music Using Spotify’s Audio Analysis. Towards Data Science (2020)

# PolyDDSP: A Lightweight and Polyphonic Differentiable Digital Signal Processing Library

Tom Baker, Ricardo Climent, and Ke Chen

University of Manchester

{tom.baker, ricardo.climent, ke.chen}@manchester.ac.uk

**Abstract.** This paper presents a work-in-progress DSP architecture<sup>1</sup> building from the basis of the Differentiable Digital Signal Processing (DDSP) library by Engel et al. (2020). The architecture is designed to process polyphonic musical audio in real-time, making use of classical DSP methods for greater interpretability. Utilising recent advancements in lightweight polyphonic pitch detection models, multiple input audio streams can be processed simultaneously, and with a novel stochastic latent dimension, the model can generate novel audio timbres outside of the training dataset. Due to its lightweight nature, the proposed architecture is designed to be used for live audio transformations with minimal input latency. The paper also discusses the limitations of the existing state-of-the-art model, which is deterministic and restricted to monophonic processing. Throughout, the paper explores potential applications of the proposed model. These include not only versatile timbre transfer between distinct instruments but interpolation between timbres, resulting in the creation of new sounds that can expand the aural pallet of musicians, sound designers, and experimental composers using live electronics. Furthermore, the model extends the library’s toolkit, such as natural pitch shifting and room acoustic reverb modelling to previously unusable polyphonic inputs.

**Keywords:** Digital Signal Processing, Machine Learning, Real-time, Polyphony, Timbre Transfer

## 1 Introduction

Digital signal processing (DSP) refers to the utilisation of algorithms and methodologies to process and analyse signals, including but not limited to audio and video. The use of DSP is a cornerstone in creative expression for the digital artist, using technology to explore sounds otherwise not possible acoustically. These techniques are often developed based on a solid foundation of knowledge and theory, enabling the creation of processes that can effectively extract desirable features or reduce unwanted noise,

---

<sup>1</sup> Code and audio examples at <https://github.com/TeeJayBaker/PolyDDSP>



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

among other applications. This theoretical basis enables the creation of efficient models that can generalise well over the relevant domain. This approach is also reflected in the use of structural priors within neural networks, such as convolution and recurrence, which are designed to take advantage of underlying patterns and relationships in the data being processed.

For Musical audio, we can exploit many structures straight from music theory and spectral analysis. Given its harmonic nature, we can approximate music as a combination of two components: harmonic - a series of sinusoids with integer multiples of a fundamental frequency, and noise - any remaining elements not so clearly defined in the frequency domain. We can leverage this decomposition to construct a decoder using classical synthesis techniques, resulting in lightweight, robust, and expressive audio generation. While the DDSP [1] library's current state-of-the-art (SOTA) work offers an excellent implementation of these techniques, it has a significant limitation: it cannot handle polyphonic audio.

In music, polyphony denotes the act of playing or singing multiple distinct notes at the same time. While some instruments, such as most woodwind instruments, are monophonic and can generally only play one note at a time<sup>2</sup>, many others are polyphonic and rely on playing multiple notes at once for creative expression. The CREPE [2] pitch encoder is a critical component for gathering pitch information within the DDSP library, and while CREPE is both lightweight and state-of-the-art for pitch accuracy, it is limited by its monophonic nature. Consequently, the DDSP library is only capable of reproducing monophonic audio signals, which restricts its applicability to polyphonic musical audio.

In this paper, we introduce the PolyDDSP model, which combines the modular and classical techniques from the DDSP architecture with state-of-the-art polyphonic pitch detection models. This new architecture is designed to handle polyphonic audio while maintaining lightweight performance and modular interpretability through the incorporation of multiple audio channels throughout the model. In addition, incorporating a stochastic latent dimension that closely resembles that of a traditional VAE will enable a more organic variation in the generated sounds, including the ability to interpolate between various timbres that have been learned. This allows the creation of new novel hybrid instrument sounds, broadening the possibilities of musical expression in the digital studio. The lightweight design of the proposed model also creates the opportunity for the development of a real-time audio plugin, similar to DDSP-VST, which can be used for live audio transformations within a digital audio workstation powered by machine learning.

The main feature of the DDSP toolkit is its timbre transfer capability, creating a unique tool for the digital studio to surpass the limitations of acoustic instruments by facilitating novel routes for real-time timbral hybridisation. However, its capabilities extend far beyond this. Through the complete reconstruction of audio from fundamental elements, the toolkit can accomplish tasks such as transposing audio while maintaining accurate instrument timbre, modifying performance dynamics, and even manipulating reverb characteristics, including complete dereverberation. The generalisation work

---

<sup>2</sup> They are capable of polyphonic expression using contemporary extended techniques such as multiphonics.

presented in this paper extends the applicability of this toolkit to polyphonic audio, significantly expanding the potential usage by accommodating audio inputs with multiple simultaneous notes.

## 2 Related Work

**Transcription:** Automatic Music Transcription (AMT) has been a long-standing problem in music, with Klapuri et al.'s probabilistic model for polyphonic pitch estimation setting the baseline in 2006 [4]. Recently, the CREPE model [2] achieved state-of-the-art accuracy for monophonic pitch estimation and was utilised in the original DDSP paper [1]. However, there has been no model that has matched CREPE's monophonic accuracy in the polyphonic domain. Bittner et al. proposed a deep learning-based approach for polyphonic pitch estimation in their paper on deep salience representations [5], laying the groundwork for further development. In their latest work, Basic-Pitch [6], Bittner et al. split the pitch detection pipeline into three tasks and created a lightweight, instrument-agnostic model that accurately detects pitch deviations and relates them to score-level note continuity. The accuracy of frame-level pitch detection is only slightly less than that of more computationally intensive, instrument-specific models.

**Style/Timbre Transfer** In the relatively young field of timbre transfer, earlier approaches such as the Universal Music Translation Network [7] relied on multiple separately trained decoders for domain transfer. This led to the timbre reconstruction falling entirely on the decoder, causing costly training. In contrast, Engel et al. [1] split the encoding between a fundamental pitch encoder, a residual encoder for timbre, and a raw extracted loudness envelope. Their model has a strong pre-baked music theory foundation, which allows it to require less training time and data to specialise to a specific domain and generate high-quality audio. However, the model's restrictive monophonic pitch detector and lack of latent interpolation leave significant room for improvement.

**Audio Generation** Audio generative modelling encompasses various disciplines, such as music, speech, and sound design. Various methods have been developed to address the complexity and controllability of generating high-quality audio. WaveNet [8] is a pioneering autoregressive generative model that produces realistic audio. However, it comes at a high computational cost, particularly for long output sequences. Recently, models based on techniques such as Diffusion [9] and Language Modelling [10] have been developed that produce excellent audio quality from minimal input. However, these models are also computationally costly and lack fine user control.

In contrast, Spectral Modelling Synthesis (SMS) [11] is a lightweight and modular approach that splits audio into harmonic and noise components [12]. These components can be generated separately using simpler techniques like additive and subtractive synthesis, driven by simple parameters. SMS provides greater control over the synthesis process, avoiding issues such as phasing alignment and spectral leakage and offering fully parametric flexibility. Thus, it is capable of producing quality audio from minimal training.

### 3 Methodology

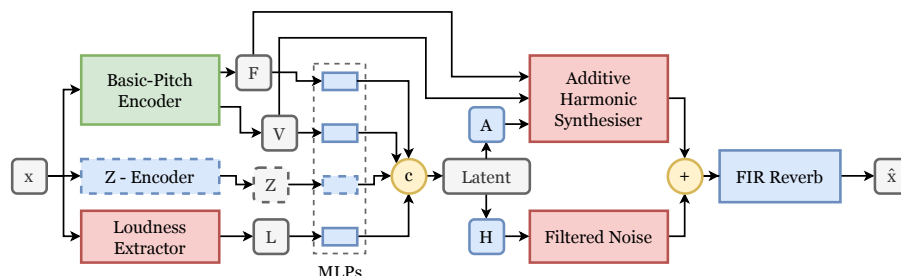


Fig. 1: Pathway through the model, with deterministic elements in red, trainable elements in blue, and the pre-trained elements in green. Tensor operations are labelled in yellow (with  $c$  being concatenation and  $+$  being addition) and dashed lines indicate optional components.

In this section we will discuss the methodology currently implemented within the proposed audio generative approach as well as the DDSP components that have been modified for multi-channel operation. Within this paper, each audio channel within the model will be referred to as a voice, in line with more traditional synthesis vocabulary.

#### 3.1 Encoders

**Pitch encoder:** The model utilises a pre-trained basic pitch encoder developed by Bitner et al. [6]. Unlike most AMT models, this fully-convolutional model generates three distinct posteriorgrams ( $Y_o, Y_p, Y_n$ ), each representing a different aspect of musical transcription.  $Y_o$  captures note onsets,  $Y_p$  tracks fine pitch, and  $Y_n$  records note events. This approach enables the model to achieve precise frequency quantised note-tracking while retaining detailed pitch information necessary for expressive performance, such as bends and vibrato.

To ensure continuous reproduction for each note during the synthesis step, we use the note-tracking  $Y_n$  to allocate each full note instance to a single voice in the pitch encoding  $F$ . As new notes appear, we assign them to the next inactive voice to ensure multiple non-overlapping voices. Finally, we apply fine pitch changes from  $Y_p$  to more closely match the input pitch and create a matching array within  $V$  with relevant note velocity values for each note in  $F$ .

**Z-Encoder** In certain musical instruments like the violin and piano, each performed note is typically played with a consistent timbre<sup>3</sup>. However, with instruments like the

<sup>3</sup> This is in the context of the general performer. Virtuoso performers will often use many techniques to explore the timbral aspects of their instrument for expressive performance.

guitar, a single pitch can be and is often produced with a diverse range of timbres, influenced by factors like the neck position and use of extended techniques. To accurately reproduce the audio in these cases, it is important to extract additional timbre information from the input and therefore we employ the optional Z-Encoder.

The Z-Encoder extracts this extra timbre information from the input audio in the form of Mel Frequency Cepstrum Coefficients (MFCCs). These coefficients are extracted by analysing the spectral envelope of the audio through a log mel-scaled spectrogram and they represent the distribution of spectral energy across the frequency scale. To utilise these coefficients, they are passed through a scalable normalisation layer, followed by a 512 unit Gated Recurrent Unit (GRU), then finally each time-step is fed through a linear layer to obtain  $Z$ , a frame-wise timbre representation for the input audio.

To improve the accuracy of timbre construction and reduce model complexity in reconstruction steps further into the model, we are developing a novel convolution based, pitch-informed source separation step. Utilising the temporally aligned transcription of the audio input provided by our pitch encoder, we can more easily separate individual voices by fundamental frequency from input spectrograms using simple lightweight convolution steps. This will allow for individual voices to have unique MFCCs and loudness envelopes to more closely reconstruct each voices timbre at later stages in the model, more closely to the much simpler monophonic case.

**Loudness Extraction:** To extract a loudness envelope, we also utilise the same steps as Hantrakul et al. [13] based on a simple psychometric model of perceived loudness. An A-Weighting of the power spectrum of input audio is log-scaled and centred according to the mean and standard deviation of the whole dataset. This specific weighting places higher value on higher frequencies to more closely match human perception.

### 3.2 Decoders

**Latent Spaces and Envelopes:** The majority of the control parameters driving the output synthesisers are contained in filter envelopes,  $A$  and  $H$  as shown in Figure 1. The tensor  $A$  contains a concatenation of both the global output amplitude envelope  $A_G$ , functionally controlling the ADSR envelope of each note generated, and each voice's harmonic spectra amplitude envelope  $A_{v,i}$ , responsible for creating the correct harmonic balance for each instrument. The envelope  $H$  controls the individual frequency bands in our filtered noise.

The multi-voice operation of the model requires the use of more complex latent space structure. Some components of the model perform better with limited input information, while other aspects require global information to function. Extracting each envelope from the latent space involves a two-step process: the specific features are fed through a GRU layer, followed by a dense linear layer. However, there is a distinction in the choice of features. In the case of  $A_{v,i}$ , the GRU layer for each voice only receives input related to the pitch, velocity, loudness, and timbre encoding of that specific voice. On the other hand, for the global amplitude envelope  $A_G$  and noise filter envelopes  $H$ , their respective GRU layers are fed with inputs consisting of encoding for the pitch and velocity of every voice, as well as the global amplitude and timbre.

**Additive Harmonic Synthesiser:** An Additive Harmonic Synthesiser is a type of synthesizer that generates complex waveforms by adding multiple simpler waveforms, typically sinusoidal waves. All acoustic instruments produce sound by using a resonating body, which is often a string or an air chamber. Due to the physics of standing-wave oscillations in resonant bodies, the timbre's generated by these instruments are characterised by a spectrum of harmonics. These harmonics start with a fundamental frequency denoted by  $f_0$  and are followed by an infinite series of integer multiples of that frequency  $i * f_0$ . The key to recreating an instrument's timbre lies in accurately recreating the correct balance of these harmonics via our harmonic amplitude envelope  $A_{v,i}$ .

In this model, the sinusoidal oscillator is constructed as a bank of  $V * H$  oscillators, where  $V$  is number of voices and  $H$  is our set harmonic cutoff, that outputs signal  $x(n)$  of discrete time steps  $n$ :

$$x(n) = A_G(n) \sum_{v=1}^V \sum_{i=1}^H A_{v,i}(n) \sin(\phi_{v,i}(n))$$

Where  $A_G(n)$  is our global amplitude envelope,  $A_{v,i}(n)$  is our specific harmonic amplitude envelope,  $\phi_{v,i}(n)$  is the instantaneous phase at timestep  $n$ , obtained from the frequency embedding  $F_v$  as follows:  $\phi_{v,i}(n) = 2\pi \sum_{m=0}^n i * F_v(m)$

In summary, the whole harmonic oscillator is parameterised by the three time dependent parameter sets:  $F_v(n)$  the fundamental frequencies,  $A_G(n)$  the global amplitude envelope and  $A_{v,i}(n)$  the harmonic distribution for each voice.

**Filtered Noise:** Subtractive synthesis works in the opposite way to additive synthesis. Rather than compounding simple waveforms to create more complex sounds, it starts with a colourful audio signal such as white noise and filters it until it reaches the desired sound. In this work, we implement a filtered noise technique similar to that of Engel et al. [1] by applying a Linear Time-Variant Finite Impulse Response (LTV-FIR) filter to a stream of uniform noise. To process this efficiently, we use frame-wise convolution through multiplication in the Fourier domain. Our extracted envelope tensor, denoted as  $H$ , represents our filter convolution function for each frequency band. We then apply this filter to the Inverse Discrete Fourier Transform (IDFT) of uniform noise,  $N$ , to obtain  $Y$ . We convert back to the audio domain by taking the IDFT of  $Y$ , resulting in the framed audio output,  $y$ , from which we construct the full audiorate signal using overlap-add.

**Reverb:** In most neural synthesis models, the room reverb is baked into generative process, as it is an essential component of producing realistic sounding audio. In contrast, this model applies room reverb after synthesis using a convolution step. This approach offers several benefits: it allows for greater transparency by enabling the extraction of dry audio from the model, and it offers more control over the room acoustics in the generated audio. However, standard convolution via matrix multiplication is computationally intensive and can hinder training and performance. To address this, we utilise the same techniques as those used in the filtered noise model - explicit convolution via multiplication in the frequency domain, which has been found to produce sufficiently accurate reproduction.



### 3.3 Other Methodology

**Upsampling:** The information contained within audio data is very dense, and at pure audiorate, it has a resolution that is too high to work with in real-time, even at the reduced 16kbps used in this model. To solve this problem, the model employs audio frames with a 64-bit length, and the encoder only extracts information at this frame rate level before upsampling it back up to audiorate much later in the model for resynthesis. Each frame lasts for 4ms, which is fine enough to fully track changes in important attributes such as  $F_0$  and loudness envelopes while reducing the temporal dimension of a second of audio from 16,000 to 250.

Bilinear interpolation is sufficient to upsample discrete variables, such as  $F_0$ , for parameterising the additive synthesizer. However, when it comes to smoothing the up-sampling of various continuous envelopes and preventing artifacting, we use overlapping Hamming windows centered at each frame.

**Spectral reconstruction loss:** For our training objective we utilise spectral reconstruction loss. This will allow for comparisons without considering audio phase differences between input and output, as these will not affect how the reconstruction sounds and therefore are not important to consider during the training process.

For input spectrogram  $\mathcal{S}$  and reconstruction  $\hat{\mathcal{S}}$ , the L1 loss for a given spectrogram is as given:

$$\mathcal{L} = \|\mathcal{S} - \hat{\mathcal{S}}\|_1 - \alpha \|\log \mathcal{S} - \log \hat{\mathcal{S}}\|_1,$$

where  $\alpha$  is log weighting term. This is summed over multiple FFT window sizes  $i$  to get a multi-scale loss  $\mathcal{L}_{multi-scale} = \sum_i \mathcal{L}_i$ . Calculating the sum of different windows sizes produces a better match over multiple resolutions, some fine detail matching without loss of the overall picture.

## 4 Experiments

The proposed polyphonic model’s effectiveness hinges on two key contributions: extending the DDSP model for polyphonic use ensuring no loss in performance compared to the monophonic case, and the effectiveness of a stochastic latent space for learning various timbres within one model. To evaluate, tests mirroring the original DDSP paper will be conducted, assessing timbre and loudness accuracy through MFCC and Loudness  $L_1$  deviations on GuitarSet and the Solo Violin dataset. GuitarSet allows for complex timbre test on a polyphonic dataset, while Solo Violin allows a direct comparison to the older model. The models performance on both sets combined will showcase the abilities of the stochastic latent space.

### 4.1 Datasets

**GuitarSet** [14] is a dataset of 360 audio recordings of guitars, each lasting approximately 30 seconds, featuring six different players playing 30 musical leadsheets across various genres and tempos in both comping and soloing styles. The recordings were played on the same guitar with consistent room acoustics to ensure uniform timbre.

**Solo Violin** comprises 13 minutes of monophonic solo violin performances by John Garner from the MusOpen royalty-free music library. All performed on a single violin in a consistent room environment.

## 4.2 Evaluation Metrics

**MFCC  $L_1$  Distance:** An important measure of the models reconstruction accuracy is it's ability to match input and output timbre. Mel Frequency Cepstrum Coefficients are used as part of the timbral encoder (z-encoder) step as they are an accurate representation of timbral quality and so two identical timbres at the same pitch should produce identical MFCCs. The  $L_1$  distance between input and output MFCC vectors should provide a representative measure of the models ability to match timbre.

**Loudness  $L_1$  Distance:** Similarly to MFCCs, the reconstructed track should produce an identical loudness envelope if it is reproduced accurately. Again, this is computed by computing the  $L_1$  distance between the ground truth audio and the synthesised audio's loudness encoding  $L$ . Please note that neither of these metrics are used by the model to evaluate during training, so there should be no inherent training bias.

## References

1. J. Engel, L. Hantrakul, C. Gu and A. Roberts. DDSP: Differentiable Digital Signal Processing. ICLR (2020)
2. J. W. Kim, J. Salamon, P. Li and J. P. Bello. CREPE: A Convolutional Representation for Pitch Estimation. arXiv (2018)
3. DDSP-VST. <https://magenta.tensorflow.org/ddsp-vst> (2022)
4. A. Klapuri. Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes. ISMIR (2006)
5. R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, Deep Saliency Representations for f0 Estimation in Polyphonic Music. ISMIR (2017)
6. R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal and S. Ewert. A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation. arXiv (2022)
7. N. Mor, L. Wolf, A. Polyak and Y. Taigman. A Universal Music Translation Network. ICLR (2018)
8. A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. arXiv (2016)
9. S. Forsgren and H. Martiros. Riffusion - Stable diffusion for real-time music generation. <https://riffusion.com/about> (2022)
10. 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 (2022)
11. X. Serra and J. Smith. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. Computer Music Journal (1990)
12. J. W. Beauchamp. Analysis, synthesis, and perception of musical sounds. Springer (2007)
13. L. Hantrakul, J. Engel, A. Roberts and C. Gu Fast and Flexible Audio Synthesis ISMIR (2019)
14. Q. Xi, R. Bittner, J. Pauwels, X. Ye, and J. P. Bello. Guitarset: A Dataset for Guitar Transcription. ISMIR (2018)

# The Unfinder: Finding and reminding in electronic music

Rikard Lindell<sup>1</sup> and Henrik Frisk<sup>2</sup> \*

<sup>1</sup> Mälardalen University, School of Innovation, Design and Engineering and  
Dalarna University, Dalarna Audiovisual Academy

<sup>2</sup> Royal College of Music, Institution of Composition, Conducting and Music Theory  
rikard.lindell@mdu.se, henrik.frisk@kmh.se

**Abstract.** In this article we examine how we as composers of electronic music organize our material, files, samples, settings, and compositions, and how existing technologies fails to meet our expectations. This text is based on a pseudo-autobiographical pilot study, where we and one other composer wrote journal notes of a preparation for an improvisation based on previous works or other material. The notes were coded and analyzed using thematic analysis that resulted in six themes: *Storage media*; *Date, time, and remembering*; *Matured material*; *Structure, metadata, and collection of material*; *Associations*; and *Tool*. Despite the enormous amounts of storage capacity available, the practice we use today we bear similarities to Barreau and Nardi's [1] nearly 30-year-old article *Finding and Reminding*. However, current operating systems were originally designed primarily to handle text files, the file system user interface has shortcomings in allowing for the kind of diversity and plethora of methods for storing and finding audio files in current music practices. Our study indicates that in order to support the way electronic music composers work, we need a usable, dynamic, plain, and transparent storage and material retrieval system.

**Keywords:** Personal Information Management, Information Retrieval, Artistic Sensibility, Electronic Music Composition, Thematic Analysis

## 1 Introduction

Although we believe that it is feasible that the study of artistic practices may generate results that are of general value also outside of the field of the arts, any results that may be drawn from this particular study are only valid in relation to the artistic practices of the three participants. In their often cited and important paper in personal information management Barreau and Nardi [1] describe how users organize and retrieve files relying on the hierarchical file system. The similarities between the results of this almost 30 year old article, written at a time when storage devices were measured in mega-bytes

---

\* KKS



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

at best, and the practice of digital artists in the present day with enormous amounts of digital material at their hands, still has relevance. In this paper we present the results of a pilot study of electronic musicians' personal information management of digital material: How do musicians organize their material (samplings, settings, naming files, etc.) on the various tools that they use before, during, and after a performance, and what are the needs that are not satisfied by existing technologies?

This question has a particular meaning in the genre of electronic music since there is often a lack of visual traces of what goes on in the sound producing engine. Whereas drummers, for example, moves their hands and hits cymbals that move. A laptop performer lacks a similar sense of immersion in performance: moving a finger, if even that, may result in a range of sonic effects, all of which lacks visual aspects.

Our study is part of a larger research project where we explore new designs to handle artistic material in electronic instruments, before, during, and after a performance. One of the central threads in the project is allowing for a widened view on information retrieval as a method. The hypothesis is that improved access to material creates an opportunity for a sense of immersion in electronic music. In order to address this we need to better understand what constitutes *relevant* material in a musical performance in electronic music, and how this material is assembled.

This paper is based on a pre-study that we have performed partly to develop a reasonable method with which we can gather results about how a musician working with electronic instruments handle the material that is generated through their practice, both live and in the studio. Both of the authors and one student at the Electroacoustic Music composition program at the Royal College of Music in Stockholm participated as subjects in the study.

## **2 Background**

The background to this study is the need to better understand how material that is generated in the process of composing music on a computer or with electronic instruments is handled by the user. How are files stored? How are they named? How are they retrieved? A session can generate large amounts of material, this material, though often invisible to the listener, and even the musician, can contain structural information about the piece. Furthermore, it can be of interest to the musician to re-use the material in forthcoming projects. To approach this complex field we have chosen to study how we, in our musical practices, handle the situation and compare it to results in previous studies in personal information management.

As mentioned in the Introduction the paper Finding and Reminding: File Organization from the Desktop by Barreau and Nardi[1] is an important reference. It "suggests that the way information is used is a primary determinant of how it will be organized, stored, and retrieved." They write that the users practice has a bigger impact on the strategies than the design of the system. Another finding they present is that the value and quality of the information decrease with time and that users give up on elaborate filing systems because in the end they do not yield enough value.

Ravasio, Schär, and Krueger [3] investigated how office personnel use their computers. In their study they found two overarching problems. First, the computer desktop

interface itself and the users' dealings with the technology; and second, the way the hierarchical file system navigation tools failed to support the information management. Of specific interest to our study are their findings concerning users' problems that often the information was distributed into different parts of the system, such as files, e-mails, and bookmarks, when these disparate pieces of information formed parts of a larger whole. These complicated searching and backup procedures forced users to redundantly move material across different storage media.

With the aforementioned paper being twenty years old, and the changes that cloud storage backup and ubiquitous WiFi and mobile internet connection has led to, one might conclude that this paper is no longer relevant. However, Wilken and Kennedy [4] recently found that people today still use and rely on portable storage devices, both for file sharing and for backup. File navigation activity still plays a crucial part in moving files across these different storage media.

Bergman et al. [5] suggests that sophistication of the organizational strategy makes a difference to the time it takes to retrieve files and that visual cues also speed up finding files. They also indicate that despite the advent of sophisticated tools for system wide text based search, such as Google Desktop, Catfish or Apple's Spotlight, users still rely on file navigation for their information retrieval and personal information management. Another study by Horst and Sinanan [6] finds that some of the participants feel a strong sense of nostalgia, evoked from file navigation, towards old data which affects the way users deal with their material: the emotional relation to the material impacts on choices made concerning storage. This contradicts the suggestion by Barreau and Nardi [1] that old files lose value over time.

Dupont et. al [7] found that there was a lack of efficient systems for retrieving data: "the tools available today for browsing through large musical libraries hinders the creative process", and "with the growing availability of multimedia content, there is a still larger demand for more flexible and efficient tools to access content and search for data". This is consistent with several of the findings in the more general studies of personal information management and retrieval above.

Recent contributions suggest that machine learning and results from the field of music information retrieval can support artists and supply artistic materials in performances [8, 9]. Here, Knees and Schedl [10] showed that context-based music information retrieval methods in general outperform content-based methods, whereas, content-based methods capture qualities closer to the material.

## **2.1 Theory**

Contemporary music since the twentieth century, including popular music, is full of examples of how change is a quality in itself: unexpected turns, erratic behavior and unpredictability are virtues that have been revered and supported creating stylistic changes at an ever increasing rate, where in popular music there is an abundance of genres and sub-genres. This is most likely connected to the fact that artistic work in general is engaged in multiple methods and is governed by change and difference to a large degree.

Artistic practice in music, and in particular experimental electronic music, which is the style we are focusing on in this paper, encapsulates all the things musicians do when they are engaged in making music. There is no real distinction between composer

and performer in this genre and the practice includes everything from thinking about making music and thinking back on past activities involving music to preparing for a performance or talking to a sound engineer. The artistic practice is guided by artistic sensibility which operates in a logic of non-conceptual free play where associations can shift rapidly. Ingman [11] defines artistic sensibility “as the sensitivity and capacity to appreciate and act upon concerns of or pertaining to art and its production”. Thompson [12] includes an intersubjective perspective into the definition of artistic sensibility and claims it “embodies the awareness of the self as an artist through the integration of artistic and aesthetic attributes toward self and other.” This awareness is of importance in this project as it extends not only to the other musicians that may be involved in the performance, but also to the material that the artist is handling. It contributes to “a certain freedom of responsiveness as the client’s artistic sensibility pervades and informs affective and cognitive reactions to his or her internal process and the wider environment.” [12]. Even if the genre, context, economy, ethics, and social circumstances set the confines for what is possible, there may not be a set goal against which the results can be measured. The critical judgment needs to happen continuously. Even if activities are geared towards a general end objective, such as a concert, distinctive for artistic practices is that they may always change direction at any time.

Artistic practices may have much in common with, and be very similar to, other practices such as programming, engineering, publicizing and many other tasks, however artistic practices are distinct from these because they are governed by an artistic sensibility towards all materials in the project. Schön [13] delineates this process as “a conversation with the material of a situation.” In our study we have considered the material to be the materials of a composition such as score, source code, sound patches, text, images, and recordings that can be stored in files. In conversation with the material, artistic practices pivot around methods of finding rather than creating, of uncovering rather than control, change rather than uniformity, and they are generally experiential rather than propositional.

### **3 Method**

As part of a larger study where we engage with electroacoustic under graduate composition students, this pilot study includes one male student and the two authors as previously mentioned. One of the purposes here is to develop the method for a larger study in which gender balance, background, and genre are of importance. In the study all participants worked on a computer and a set of various software and hardware. We kept a journal of the preparation for an improvisation based on material from previous works or other sources. In a project journal the participants reflect on how they organize their material, and in a final reflection to identify and describe advantages and disadvantages of the structure, or lack thereof, they were using. Furthermore, we asked him to briefly describe an utopian optimal solution for how to organize material in a practice such as his own that would support the composition process. Because we as authors are also part of the electronic music community our perspective in this study is emic. According to Pike [14], this means analyzing the unique meanings and symbols used within a culture or group that we as researchers are part of. He argues that the un-

derstanding of a culture requires examining the language and communication methods from within. Our insight in the experiences and perspectives of the practices of electronic music composers allowed us to also understand the underlying assumptions and values. On the other hand there is a risk that we from our privileged perspective have assumptions that may mask events in the practice that we then fail to expose because they are, to us, self-evident. We managed this risk by staying open to the data we analyzed and by relying in the thematic analysis method [16, 18]. In the written reflections in the study related to the information retrieval process and how material is organized the method is inspired by autoethnography [15]. Thus, the main data for this pilot study consists of the written reflections of three participants. Instead of proof by numbers we have cross-examined the qualitative data, and trusted in our reflections and in the thematic analysis method.

Thematic analysis [16, 18] is a pragmatic method to guide and organize the interpretation of data performed in six steps: (1) become familiar with the data, (2) generate codes, (3) search for themes, (4) review themes, (5) define themes, and (6) write-up. We made the initial three steps individually, where we both analyzed the student's journal and each other's journal. In the second step we relied on open coding without preset codes. To document initial ideas and interpretation of the codes and themes we wrote memos inspired by the Strauss and Glaser's [19] grounded theory method. We made the remaining three steps collaboratively. This allowed us to calibrate ourselves in the analysis of the student's reflections and double check our codes and themes because we both looked at each other's texts where the interpretation, codes and themes, of an extract from the reflection could be discussed with its author. Initially we had seventeen, mostly semantic, themes [16], based on the explicit meaning of utterances. In the fourth and fifth step we used the initial memos, revised, and redefined the themes into latent themes [16]) where the themes express underlying ideas and processes.

#### **4 Research ethics statement**

The study did not handle any sensitive personal data. To use the data produced in the study, we have obtained the explicit consent from the participant. The participant has certified voluntary participation, providing us the right to publish pseudonymized quotes from the journal and sample recordings that illustrate findings, and the participant has certified awareness of the study's procedure. Quotations from the project journal have been presented pseudonymously. The data from the study is stored pseudonymously.

#### **5 Findings**

From the data and codes we initially found twelve themes where each theme was built from a few codes. These themes were further developed and focused into six themes: *Storage media*; *Date, time, and remembering*; *Matured material*; *Structure, metadata, and collection of material*; *Associations*; and *Tool*. Figure 1 presents a model of these themes and their relative inter-connections, and the themes are described in the following.

### **5.1 Storage media**

This theme refers to the activity of dealing with storage media. Media types and media storage location are both aspects that are included, as explained by P2: “the resulting piece, when I believe it is worth it, is stored via SoundCloud or Vimeo”. The development of storage media impacts on the workflow of artists and P2 further comments upon the ways in which this development allowed for new practices. When audio files “were too large for the floppy discs” the relationships changed and “eventually I did not store anything except the recording to the fixed media” (P2). P3 comments that they “usually copy the whole directory [...] which is problematic and takes up disk space”. Economic awareness of disk space usage may impact on artistic processes. Storage media is at the very heart of the activities discussed here and is of central importance. It can stretch from discontinued media such as physical tapes and disks, hard drives and SSD disks, and online storage formats. P1 explains: “Hard drives, cassette tapes, USB sticks and memory cards are tucked away everywhere in the studio and the workflow takes shape there as well.” These types of storage media have a huge impact on the material and a sound file stored on one media may change to a significant degree if transferred to another, both in terms of its properties and the ways in which it can be interacted with. This theme is connected to the themes *Structure, metadata, and collection of material*; *Date, time and remembering*; and, *Matured material*, see Figure 1, because all these themes are consequences of the design of, and interface for the hierarchical file system media storage.

### **5.2 Date, time, and remembering**

This theme relates to the navigation, structuring, sorting, and information management of content material based on time and date. For instance, P2 describes how files are organized in date order: “However, the bundle files are sorted in the date order. For more elaborate pieces that constitute a plethora of different files, sketches, samples files, images, source code for SuperCollider.” It also accounts for associations to when pieces were made, or locating files from date associations. *Date, time and remembering* is important within the storage container for a work (see *Structure and metadata, collection of material*) but also describes the organization of the works themselves. Furthermore, forgetting about content material is also a part of this theme explained by P1 as “what is lost or forgotten is not used”. P1 continues by describing how sorting based on date can help the rediscovery of such forgotten content: “Sometimes I use sorting principles for lists of folders like “last opened/modified” to see what’s hiding among the storage devices.” Memory is obviously central to this kind of practice since all musical practices rely on memory and mnemonic associations, also on an overarching level. P3 describes how remembrance is a part of a performance: “The advantage with my method is that once I start to play, I know what kind of material I have to work with.” P2 comments that “I can more easily remember when I made a piece of music rather than what I called it” which hints at the organizational importance of date and time. Materials that have been forgotten or lost that are perceived in a new way related to, acousmatic listening [17], are directly connected to the theme of *Matured material*. Figure 1, shows that this theme has a relationship to *Matured material* where time alters the perception



of the material. It is related to the theme *Structure, metadata, and collection of material* where date and time is explicitly used to organise the material. This theme is also related to *Associations* due to the date supported in current systems, and to media storage because of the current implementation support for creation, last edited, and last opened dates.

### **5.3 Matured material**

This theme is about the phenomenon of materials and the way they change over time, such as files maturing as a function of change in the user's perception. This is illustrated by P1 in the following quote: "In my view, it is like having a wine cellar without an inventory list and that the work in the studio can be equated with collecting the bottles that can be thought of as working well together or in some other way picked out to be uncorked." This may be both due to the altered perspective, and because the material has actually deteriorated expressed by P1 as in "files may be processed and/or they may retain their original appearance". Furthermore, files can be reused and repurposed based on their matured artistic qualities. There is a natural connection between this theme and the *Date, time and remembering* theme because the main operative structure here is the passing of time, see Figure 1. *Matured material* is also related to Schaeffer's reduced listening [17], where time makes us forget the origin of, or the circumstances of the creation of the sound. This is poetically described by P1 as a Darwinian archive in which only the strong, or used, material survives. Matured material that resides in folders or on external drives, relates this theme to the *Storage media* theme and to the *Structure, metadata, and collection of material* theme, see also Figure 1.

### **5.4 Structure, metadata, and collection of material**

This theme describes the organization of content and material in files. A common exemplification of this is where pieces and works are structured inside a container, or a folder in a file system. Within this container, however, the file order is often unstructured, and the structure of the files may instead rely on metadata, for instance, date, color tags, file type, location — the cartesian position within a folder. This can be achieved loosely via the file manager tools of the operating system, as illustrated in the quote: "Tags and color coding of files as well as combinations of these to be able to more easily navigate what is linked by association due to content or area of use. [...] I use a hard drive where files are arranged in different formations purely graphically in a folder." (P1). These associations between material files makes this theme related to the *Associations* theme, files located on different hard drives relates to the media storage theme, and the use of date metadata relates this theme to *Date, time, and remembering*, see Figure 1. Another, more rigid, method to organize content in the container formally is through a structured description, such as a makefile. As reflected on by P3, choosing a strict method like this impacts on the voice of software. (See more under the theme of *Tool*). Maintaining structure may be difficult due to inconsistencies in file names and metadata. P3, however, comments that "the disadvantage is that it is difficult to use/reuse material in other projects" if all the project files are assembled in one container. To reuse parts of a project, or the parameter structure and relations within the project container, while

preserving the integrity of the material; related to the theme *Associations*, see Figure 1; it may be necessary to make copies of the entire container rather than copying individual files. This kind of organization is akin to the common paradigm of files in a DAW where, as commented by P2, “each song was a file or bundle that included everything for the piece”. P2 continues that “the disadvantage [with this method] is the lack of order inside each piece, in particular for the more elaborate and experimental works that include many different file formats.” Dynamic organization of content, where the work rather than the storage constitutes the semantic structure, with querying for metadata across all projects would support the reuse of content.

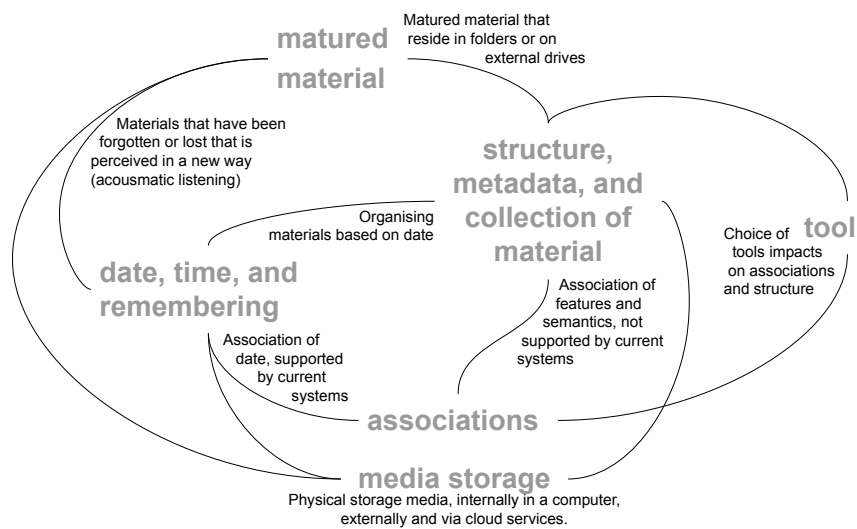
### **5.5 Associations**

This theme is about various associations between media, files and other aspects of artistic production, or systems that may operate with such associations. In the following quote, the whole network of association within a piece, between its content, parameters, and external connections is reused: “A composition is often based on presets from a previous project, but channels and connections go to new inputs. Control data with step patterns or start and stop points, pitch changes, or automated effect parameters and pans can be transferred from one thing to another, and in an improvisational/intuitive way create a whole reminiscent of the previous project because junctions between them are built into the technical infrastructure of the work.” (P1)

In commonly used personal computer file navigation systems associations based on date are important, which is related to the *Date, time, and remembering* theme, see Figure 1, whereas current information retrieval tools appear to not sufficiently support complex associations. P1 comments: “I sometimes use principles for sorting material that are related to ‘most recently opened/changed’ whereby I get information about what is hidden behind the structures of storage, and the result is often creative.” This may be understood as a method of choosing files that is not necessarily based on a musical association to the present material, but one that may generate surprising effects. The association in this case is temporal, and meta-structural (as in material often used are getting more exposure than others) rather than relational.

### **5.6 Tool**

This theme concerns the tools, or lack thereof, in the artistic work for the management of content, which is either tool dependent, where the tool “dictates” how to handle content, or the lack of tool independent means for organizing and/or finding materials. Tools can also be aspects of a larger system for production such as a DAW that contains synths with presets, the use of which may also be considered a tool. Here the theme is related to *Associations* theme, see Figure 1, because tools may impose or capture the associations of materials. A distinction between “tool” and “content” may be difficult to draw but is of interest in this context and is explored in the following quote: “It is a lengthy process if I want to change the sounds loaded, it’s simply not possible to change loaded sounds in real time; they are hard coded.” (P3) Here, the tool makes it difficult to handle content. The actual practice of using an instrument may dictate the kinds of tools that are useful. In the case of a modular synthesizer, for example,



**Fig. 1.** This figure depicts the relationships between the themes. Each line represents an explicit relationship from the findings, where most of the relationships has a caption. For instance, the *Date, time, and remembering* and *Associations* themes have a relationship based on the file date metadata supported in current operating systems file navigation interfaces.

a mobile phone camera may be useful to recreate a piece, whereas in programming a photo is less useful. A modular synthesizer, for example, depicts the possible means for documentation: “When I believe the patch of the modular synthesizer is worth saving, I use a notation or a patch description language that allows me to recreate a patch with some degree of precision.” (P2) Recording and storing wave files in the file structure of the operating system is different from recording using a DAW that also provides the user with a management system, hence this theme is also related to the *Structure, metadata, and collection of material* theme, see Figure 1.

## 6 Conclusions

Wilken and Kennedy’s [4] notes on the nostalgia of data and its age may determine its value has links to the analysis under *Matured material, Storage media* (Section 5.1) and *Date, time and remembering* (Section 5.2) where in particular the discussion on “old wine” (see also Figure 1) suggests an objectification of the material. This aspect of nostalgia give rise to a further abstraction where the actual storage results in a representation of a memory and becomes more important than the data it holds: “archives are felt to be significant, even if the data is no longer accessible” [4] - the media is truly the message.

Our findings, in particular for the theme describing the *Structure, metadata, and collection of material* (Section 5.4) indicate that electronic music composers are filing information according to systems of keywords, tags, and carefully architected logical schemes. This contradicts one of the key points of Barreau and Nardi [1]. Although our study shows that there are systematic and logical schemes for storing files by the users, these strategies were constructed based on the needs of the current project rather than on a general and reusable format. In other words, organization of files is structured according to the composition and production work, which is loosely in line with the conclusion by Wilken and Kennedy [4]. Barreau and Nardi are also stressing that “finding and reminding are intimately linked in users’ practice and should be considered together” [1]. Storage arrangements today commonly range over a large number of different kinds of systems, such as cloud based, disks and USB-sticks, each with different levels of tangibility that offer different possibilities. These do indeed support individuality (see Section 5.1) but are commonly tied to the logic of the file system at hand. File access in current operating systems were originally constructed primarily for handling text files. According to aspects discussed in the themes *Date and time, and remembering* (Section 5.2), and *Associations* (Section 5.5) relating to the organization of audio file and music information, our study indicates that the file system user interfaces has deficiencies in allowing for the kind of multiplicity of methods for storing and finding audio files that the participants in this study deploy. However, this study is mainly valid in relation to the three participants. Hence, our findings indicate the need for a larger study, with possibly more general results. Such a study could furthermore provide insight into other fields of creative practices but most importantly: We believe that there need to rethink the design of a usable, dynamic, plain, and transparent storage and material retrieval system to support how electronic music composers and performers work.

## References

1. Barreau, D. & Nardi, B. Finding and reminding: file organization from the desktop. *ACM SigChi Bulletin*. (27):39–43 (1995)
2. Schnell, N. & Battier, M. Introducing composed instruments, technical and musicological implications. *Proceedings Of The 2002 Conference On New Interfaces For Musical Expression*. 156–160 (2002)
3. Ravasio, P., Schär, S. & Krueger, H. In Pursuit of Desktop Evolution. *ACM Transactions On Computer-Human Interaction*. (11):156–180 (2004), <http://dx.doi.org/10.1145/1005361.1005363>
4. Wilken, R. & Kennedy, J. Everyday Data Cultures and Usb Portable Flash Drives. *International Journal Of Cultural Studies*. (25):192–209 (2021), <http://dx.doi.org/10.1177/13678779211047917>
5. Bergman, O., Whittaker, S., Sanderson, M., Nachmias, R. & Ramamoorthy, A. How do we find personal files?. *Proceedings Of The SIGCHI Conference On Human Factors In Computing Systems*. 2977–2980 (2012,5)
6. Horst, H. & Sinanan, J. Digital Housekeeping: Living With Data. *New Media & Society*. (23):834–852 (2021), <http://dx.doi.org/10.1177/1461444820953535>
7. Dupont, S., Dubuisson, T., Urbain, J., Sebbe, R., D'Alessandro, N. & Frisson, C. AudioCycle: Browsing Musical Loop Libraries. *2009 Seventh International Workshop On Content-Based Multimedia Indexing, Content-Based Multimedia Indexing, 2009. CBMI '09. Seventh International Workshop On*. 73–80 (2009)
8. Ordiales, H. & Bruno, M. Sound recycling from public databases Another BigData approach to sound collections. *ACM International Conference Proceeding Series*. (2017)
9. Xambó, A., Roma, G., Roig, S. & Solaz, E. Live Coding with the Cloud and a Virtual Agent. *NIME 2021*. (2021)
10. Knees, P. & Schedl, M. Contextual Music Similarity, Indexing, and Retrieval. *Music Similarity And Retrieval*. 133–158 (2016)
11. Ingman, B. Artistic Sensibility is Inherent to Research. *International Journal Of Qualitative Methods*. (21) (2022,1)
12. Thompson, G. Artistic sensibility in the studio and gallery model: Revisiting process and product. *Art Therapy*. (26)159–166 (2009)
13. Schon, D. *The Reflective Practitioner: How Professionals Think In Action*. 78–79 (Basic Books, 1983)
14. Pike, K. L. *Language in Relation to a Unified Theory of the Structure of Human Behavior*. (De Gruyter Mouton, 1967)
15. Ellis, C. *The ethnographic I: A methodological novel about autoethnography*. (Rowman Altamira, 2004)
16. Maguire, M. & Delahunt, B. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *All Ireland Journal Of Higher Education*. (9) (2017)
17. Kane, B. Pierre Schaeffer, the Sound Object, and the Acousmatic Reduction. *Sound Unseen: Acousmatic Sound in Theory and Practice*. (Oxford University Press, 2014).
18. Braun, V. & Clarke, V. Using thematic analysis in psychology. *Qualitative Research In Psychology*. (3):77–101 (2006)
19. Glaser, B. & Strauss, A. *Discovery of grounded theory: Strategies for qualitative research*. (Routledge, 2017)

# Towards Potential Applications of Machine Learning in Computer-Assisted Vocal Training

Antonia Stadler<sup>1</sup>, Emilia Parada-Cabaleiro<sup>1,2,3</sup> and Markus Schedl<sup>1,2</sup>

<sup>1</sup> Institute of Computational Perception, Johannes Kepler University Linz, Austria

<sup>2</sup> Human-centered AI Group, Linz Institute of Technology (LIT), Austria

<sup>3</sup> Department of Music Pedagogy, Nuremberg University of Music, Germany  
emiliaparada.cabaleiro@hfm-nuernberg.de

**Abstract.** The usefulness of computer-based tools in supporting singing pedagogy has been demonstrated. With the increasing use of artificial intelligence (AI) in education, machine learning (ML) has been applied in music-pedagogy related tasks too, e. g., singing technique recognition. Research has also shown that comparing ML performance with human perception can elucidate the usability of AI in real-life scenarios. Nevertheless, this assessment is still missing for singing technique recognition. Thus, we comparatively evaluate classification and perceptual results from the identification of singing techniques. Since computer-assisted singing often relays on visual feedback, both an auditory task (recognition from *a capella* singing), and a visual one (recognition from spectrograms) were performed. Responses by 60 humans were compared with ML outcomes. By guaranteeing comparable setups, our results indicate that ML can capture differences in human auditory and visual perception. This opens new horizons in the application of AI-supported learning.

**Keywords:** AI-supported Education, Singing Techniques, Perception

## 1 Introduction

Singing techniques, as well as the strategies to teach them, have evolved over the history, in correspondence with chronological and geographical factors influencing music development [1]. Nevertheless, singing pedagogy has been mostly based in oral tradition, which is the reason why the description of how to perform such techniques is, in some cases, vague and imprecise [2]. Due to this, while experienced singers and teachers can naturally evaluate the quality of singing by simply following their intuition [3], this task might be particularly challenging for beginners.

The advantages of using computer-based applications to support teaching and learning have been shown [4]. Within music pedagogy, the use of computer-assisted singing tools, able to enhance singers' awareness, have become of common use in combination with traditional pedagogy [5]. Indeed, some of these tools have shown to be particularly



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

assessment of singing quality [7]. Similarly, research on the automatic recognition of specific singing techniques has recently gained popularity [8, 9].

Nevertheless, the development of ML tools to support singing training is still on its infancy, which comes along with not yet well-defined use-cases and prevents a real connection between music pedagogy and the AI field. In this work, we present a preliminary study aimed to pave the way for future research on the use of AI in singing pedagogy. Since it has been shown that assessing how well a ML algorithm performs in comparison to humans can bring light about the utility of AI in real life [10–13], we assess, for the first time, the performance of ML methods in singing technique classification with respect to humans. By evaluating the perceptual ratings of two participant groups (with and without musical expertise) in comparison to ML we aim to: (i) assess how different feature representations perform in comparison to different learners level; and (ii) try to define potential applications of ML in singing education scenarios.

## 2 Related Work

The use of technology as an auxiliary educational tool has shown to successfully enhance singing pedagogy [14]. This is achieved by integrating acoustic voice analysis in the learning context as well as by using it as a biofeedback for singers' training [15]. Indeed, analysing audio recordings and computer-based feedback are two important elements of up-to-date singing pedagogy [16]. In particular, it has been shown that using visual representations of vocal properties effectively supports learners [5]. For instance, the understanding of phrasing can be enhanced by illustrating vocal pressure [17]. ALBERT [18] and VOXed [19], aiming to promote a more effective singing learning, are tools developed for real-time educational visual feedback. Finally, the use of computer-based tools complementing traditional pedagogy has shown to effectively promote curiosity and motivation [20], two essential aspects for a successful learning.

Within AI, the automatic classification of singing techniques has gained relevance, which lead to the development of dataset such as VocalSet [8] or J-POP [21]. Research on VocalSet showed that features learned from multi-resolution-spectrograms can outperform the original baseline, based on a Convolutional Neural Network (CNN), with a much less sophisticated architecture, i. e., Random Forest [9]. Similarly, a recent work on automatic recognition of paralinguistic singing attributes, e. g., vocal register and vibrato, has confirmed that feeding traditional ML models, such as Support Vector Machine (SVM), with spectrograms is a suitable approach for singing-related tasks [22].

## 3 Methodology

### 3.1 Dataset, Preprocessing, and Evaluation Metrics

In this work, we use VocalSet [8], a dataset consisting of 3 560 audio instances (10.1 hours of recordings) produced by 11 male and 9 female singers performing 17 different singing techniques. As in the original baseline, the experiments were performed by considering only 10 singing techniques (1 736 audio instances), i. e., the most relevant in practice: *Belt*, *Breathy*, *Inhaled*, *Lip Trill*, *Spoken*, *Straight*, *Trill*, *Trillo*, *Vibrato*, and *Vocal Fry*. In Table 1, the frequency distribution of the used audio instances across the singing techniques, as well as their duration in minutes, is indicated.

Singing technique	Number of instances	Duration
Belt	205	26.24
Breathy	200	28.00
Inhaled	100	9.95
Lip Trill	202	24.40
Spoken	20	4.06
Straight	361	71.65
Trill	95	18.45
Trillo	100	14.54
Vibrato	255	57.79
Vocal Fry	198	34.10

Table 1: Overview of the samples from VocalSet used in the experiments. For each singing technique, the total number of instance and overall duration in minutes is given.

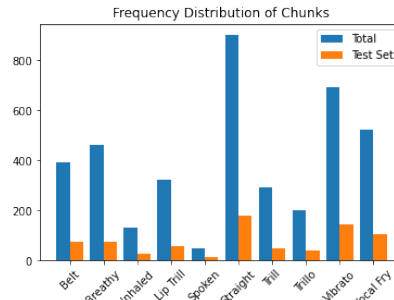


Fig. 1: Distribution of chunks across singing techniques. Besides the total, those used in the user study and as test set in the machine learning experiments, are displayed.

Following the pre-processing guidelines used in the baseline of VocalSet [8], the silence at the beginning, middle, and end of the audio files were removed and the instances were split into chunks of approx. 3 seconds length. The distribution of the resulting 3 934 audio chunks across the corresponding singing techniques is displayed in Figure 1 (cf. Total). For the user study and as a test set for the ML experiments, the chunks extracted from the audio instances produced by singers F2, F6, M3, and M11 (i. e., 777), were considered (cf. Test Set in Figure 1). These singers were selected as they produced samples for all the considered techniques.

The experimental results, for both the user-based and the ML experiments, will be evaluated in terms of Unweighted Average Recall (UAR), precision, and recall. UAR, also known as Balanced Accuracy, is the recommended metric for datasets with an imbalanced distribution of samples across classes [23]. Besides precision and recall, confusion matrices will be used to interpret confusion patterns amongst classes.

### 3.2 Singing Techniques

To enable a better interpretation of the results, a brief description of each singing technique (illustrated by a spectrogram generated with Praat, cf. Figure 2), is presented. Since not all the techniques are produced through the same vocalisations in VocalSet, the spectrograms display a variety of them, i. e., arpeggios, long tones, and scales.

The sound produced by the technique *Straight* is natural, without any pressure or ornamentation. This is what we typically refer to as ‘normal’ singing, with the complete elimination of vibrato [24], which is shown by the horizontal lines in the spectrogram representing the pitch (cf. Figure 2a). In contrast, when singing *Vibrato*, the fundamental frequency and amplitude are intentionally altered by the singer [25], oscillations clearly visible in the spectrogram generated from the same instance (cf. Figure 2b).

*Vibrato* is often confused with the technique *Trill*. However, *Vibrato* should sound like one single tone rather than two different ones, which is expected in *Trill* [24]. This is achieved by producing oscillations that do not exceed a semitone beyond the main tone [26]. On the contrary, *Trill* is perceived as a fluctuation between two clearly distinguished pitches [24]. This can be observed in the spectrogram (cf. Figure 2f), where the regular pitch oscillations are clearly defined contrasting with a dark background which indicates much less presence of upper and lower tones.



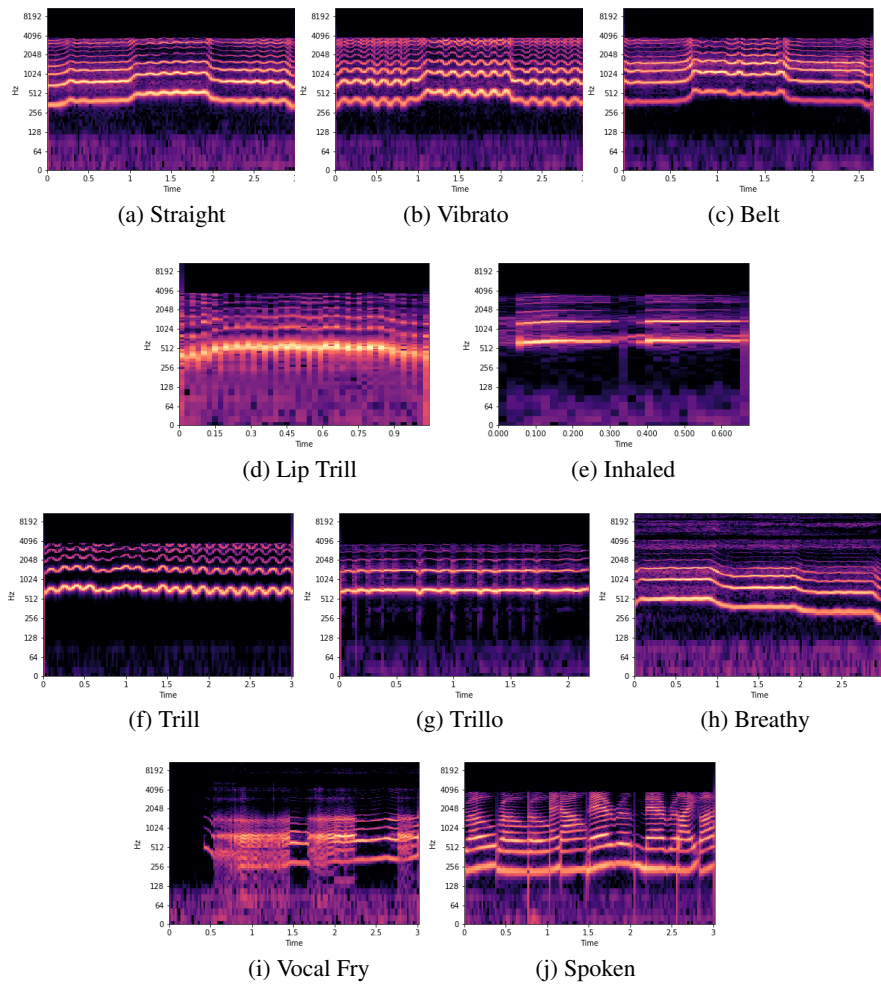


Fig. 2: Spectrograms displaying each of the evaluated singing techniques. All of them are generated from samples performed by the female singer F1 producing the vowel ‘a’ except *Spoken*, for which a text is read. The used vocalisations are: arpeggio (*Straight*, *Belt*, *Vibrato*, *Lip Trill*); long tone (*Inhaled*, *Trill*, *Trillo*); scale (*Breathy*, *Vocal Fry*).

*Trillo* is a singing technique described as a rapid *Trill* similar to the sound of a ‘bleating goat’ [24]. It sounds like a quick repetition of one single note and is produced by larynx movement. In the spectrogram (cf. Figure 2g) it can be observed that the pitch oscillations are much less pronounced than for *Trill*. Another distinguishable property are the pitch breaks visible in the spectrogram, which are due to breaks needed by the singer to catch air when performing this exhausting technique.

In comparison to ‘normal’ singing, *Belt* is produced through a higher subglottal pressure and by keeping more firm vocal cords adduction, which results in higher sound levels [25, 27]. This technique sounds ‘forced’, i. e., it is not perceived as relaxed singing

but rather uptight. *Belting* is referred to as raising the chest voice above the typical register and implies a higher level of physical effort [28]. This can be observed in the spectrogram by the rather straight and tense pitch lines (cf. Figure 2c).

The technique *Lip Trill*, often used as a warm up exercise, is done by continuously vibrating with the lips while simultaneously maintaining phonation [29]. This technique is the only one where the mouth and lips remain closed, something distinctive in the spectrogram, where there is barely any black background (cf. Figure 2d).

Another characteristic technique is *Inhaled*, as its main feature is that, unlike all the other techniques, the sound is produced using an inspiratory airflow instead of an expiratory one. Therefore, the sound is generated while the singer inhales [30], which can be observed in the spectrogram by less clearly defined pitch lines (cf. Figure 2e).

The technique *Inhaled* sounds, to some extent, similar to the techniques *Breathy* and *Vocal Fry*. In *Breathy*, a low subglottal pressure is combined with a less efficient adduction of the vocal cords [31]. This results in a sound characterised by audible airflow, which is shown in the spectrogram by broader and blurrier pitch lines (cf. Figure 2h). In *Vocal Fry*, characterised by lower subglottal air pressure and transglottal air flow, the vocal folds are shortened, even when frequency increases [32]. This is shown in the spectrogram by diffuse and irregular pitch lines (cf. Figure 2i).

Finally, *Spoken*, in contrast to singing, is the only technique that does not require the control of the pitch. The distinguishing feature visible in the spectrogram is a grid-like pattern (cf. Figure 2j) where the horizontal lines (relatively stable) represent the pitch and the vertical ones (unequally spaced out) correspond to the words' articulation.

### 3.3 User Study

The user study consists on two experiments performed by different groups: (i) musically trained individuals (task based on auditory perception); (ii) non-musically trained individuals (task based on visual perception). Both experiments were performed through a web-based interface and began with an example (either an audio or an spectrogram) of each singing technique. Then, an explanation of the task, presented as a multiple choice test, was given. For each sample, the participants could choose one singing technique out of the ten given possibilities. 60 volunteers (31 female, 29 male;  $\mu = 32.3$  years) participated in the study. Most of them were Austrian (43), the rest were German (14) and Australian (3).<sup>4</sup> They were recruited through the authors' social networks and consent, requested through the interface, was a requirement to take part in the experiment.<sup>5</sup>

In the auditory experiment, the participants were expected to identify the singing techniques by listening to the audio excerpts. Since a trained ear is necessary for this task, in the auditory task only participants with a musical education (9 female, 11 male) took part. Their formal training included choir conductor, singing, and vocal studies. In the visual experiment, the participants were expected to identify the techniques by looking at spectrograms generated from the audio excerpts. Spectrograms were chosen since typically used in singing lessons [16], specially to support beginners [33]. Since for the

<sup>4</sup> Due to the imbalanced distribution of participants, nationalities' role will not be evaluated.

<sup>5</sup> The procedures used in this study adhere to the tenets of the Declaration of Helsinki. Participants consented the use of their anonymous responses only for research.

auditory task a trained ear is needed, the visual task was considered a more suitable alternative for the participants without musical background (22 female, 18 male).

In order to avoid fatigue, the 777 excerpts were randomly distributed across the participants. For the auditory task, this was made in a way that each would annotate between 75 and 80 audio chunks. Since we expect the evaluation of spectrograms to require more time than assessing audio samples, in order to preserve the reliability of the experiment, for the visual task each participant would annotate between 37 and 41 images. In both experiments, in order to prevent individual biases, each sample was evaluated by two different participants, which lead to 1 554 annotations per task.

We are aware that assessing two user groups (experts and non-experts), makes the setups not comparable within the user study. However, the final goal of this study is to make a one-to-one comparison between perception (auditory as well as visual) and ML. In addition, in base of the principle that learning should be tailored to individuals capabilities [34] (which are not the same for musically trained users and non-trained ones) we believe that considering the same task for both user-groups would heavily penalise the non-trained group. Thus, to perform a fair comparison of trained and non-trained users with the ML algorithms, two different perceptual experiments were performed.

### **3.4 Machine Learning Setup**

Following previous works on singing classification [8, 22], both traditional models and neural-based were implemented. Due to space limitations, the results for the traditional models (outperformed by the neural ones) will not be reported. A Neural Network (NN) and a Convolutional Neural Network (CNN) were implemented in the tensorflow framework. The NN, presenting eight layers, Relu as activation function, and categorical crossentropy as loss function, was trained for 40 epochs. The CNN was implemented as in the VocalSet baseline [8], i. e., consisted of seven convolutional layers, seven max pooling layers, learning rate of 0.001, a momentum of 0.6, and categorical crossentropy as loss function. It was trained for 30 epochs.

Two type of features were considered: Mel-Frequency Cepstral Coefficients (MFCCs) and spectrograms. They were chosen as suitable representations according to state-of-the-art literature [35] and their corresponding outcomes will be compared with the auditory and visual perceptual results, respectively. The features were extracted from the audio files (sampling rate: 44100 Hz) with default parameters of the librosa package: fft-size of 2048; frame size of 93 ms; and frame step of 23 ms. For the MFCCs, the first 20 coefficients were extracted. As already mentioned, the 777 excerpts produced by the singers F2, F6, M3, and M11 were used as test set and the remaining 3 157 excerpts as training set. By this guaranteeing a comparable setup w. r. t. the user study, where only the 777 excerpts were assessed.

## **4 Results**

### **4.1 User Study**

As expected, the experimental outcomes show a higher performance from the musically trained participants: UAR = 76% for the auditory task w. r. t. to a UAR = 41% for the visual one. In Figure 3 the confusion matrices for both experiments are displayed. The

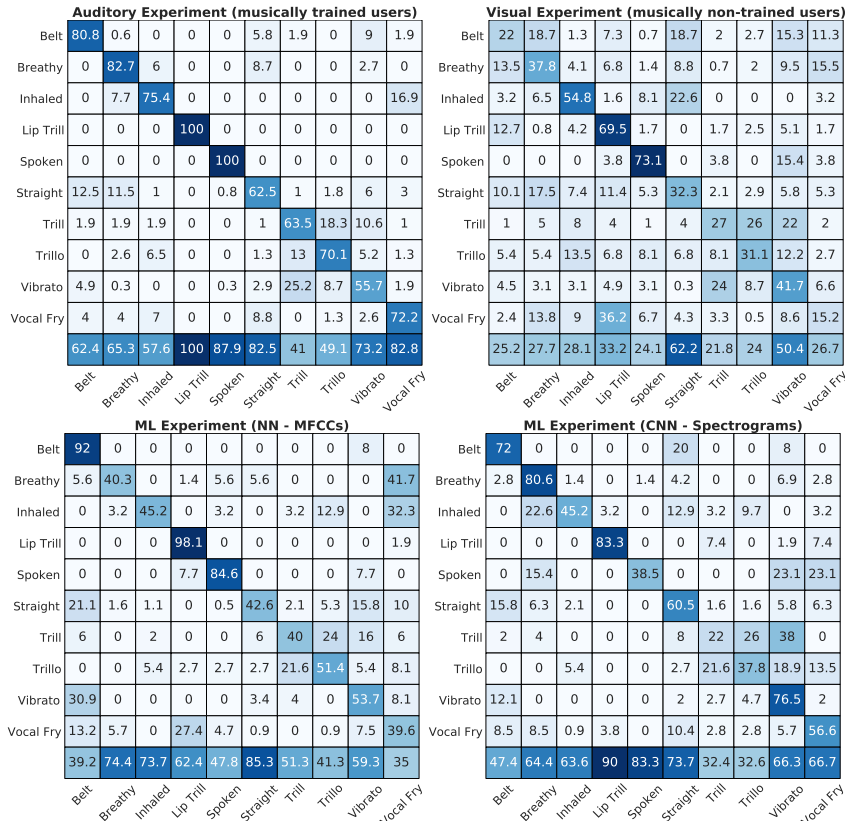


Fig. 3: Confusion matrices for: perception in the auditory task (UAR = 76%); perception in the visual task (UAR = 41%); classification from a Neural Network (NN) fed with MFCCs (UAR = 59%); and classification from a CNN fed with Spectrograms (UAR = 57%). Darker cells indicate higher values (%); rows encode real labels. Recalls are given in the diagonal; precisions are shown in the last row of each matrix. Note that the UAR is an overall measure computed from the whole confusion matrix.

higher recall and precision achieved by musically trained users is shown for all the techniques, which is displayed by a well defined diagonal and a darker precision row for the auditory results. The confusion between singing techniques experienced by users without musical training is shown by the spread of responses across the matrix as well as by the lower precision (cf. light colour of the last row) for the visual results.

Remarkable results are shown for the techniques *Lip Trill* and *Spoken*, recognised with the highest recall in both experiments: in the auditory, both techniques achieved 100% recall; in the visual experiment, they achieved 69.5% and 73.1%, respectively. Indeed, these two techniques are particularly distinctive w. r. t. the others, which make them more easily recognisable. As mentioned in Section 3.2, from an auditory point of view, *Lip Trill* is the only technique produced with a closed mouth and *Spoken* is the only one for which the pitch is not controlled. Although these aspects are visible in the spectrograms, it is important to note the low precision achieved for both techniques

in the visual task: 33.2% and 24.1%, respectively; which indicates that despite their characteristics, these techniques are often wrongly chosen by the non-experts group.

Beyond the expected performance differences between listeners' groups, a prominent confusion pattern is common in both experiments, i. e., samples from *Vibrato* are wrongly identified as *Trill*. In both tasks, the amount of misclassifications is nearly half of the correctly identified samples. For the auditory experiment, 25.5% misclassifications vs. 55.7% correct hits; for the visual one, 24% misclassifications vs. 41.7% correct hits. The confusion pattern is also shown in the opposite direction, i. e., *Trill* instances are wrongly identified as *Vibrato*, a result consistent with previous research showing that *Trill* might be similar to *Vibrato* performed with an 'exaggerated extent' [36]. The described confusion pattern involves *Trillo* as well, i. e., *Trill* and *Trillo* are misclassified not only as *Vibrato*, but also amongst themselves. Indeed, the three techniques are similar, since produced by modulating the fundamental frequency (cf. Section 3.2).

Finally, a prominent confusion is displayed for the visual experiment, i. e., *Vocal Fry* is wrongly identified as *Lip Trill*. The percentage of misclassifications exceeds by far the amount of correctly identified instances: 36.2% vs 15.2%. The pattern is not shown for the auditory experiment, which suggest that this type of confusion relates to similarities in the spectrograms difficultly disentangled without audio information.

#### 4.2 Machine Learning

Amongst the evaluated algorithms and feature sets, the best performing model was the NN fed with MFCCs (UAR = 59%) followed by the CNN fed with spectrograms (UAR = 57%). Confirming the results shown in both perceptual experiments, *Lip Trill*, and to some extent *Spoken*, are also the two techniques best recognised by the model fed with MFCCs: 98.1% and 84.6% of recall, respectively; cf. diagonal in Figure 3 (NN - MFCCs). This was also shown for the model fed with spectrograms concerning *Lip Trill*, achieving the highest recall (83.8%), but not for *Spoken*, reaching only 38.5% recall; cf. Figure 3 (CNN - Spectrograms). It is important to note, that despite the low recall for *Spoken*, the precision for this technique is lower for the NN than for the CNN, which indicates that the promising recall is only due to the high confusion attracted by the class; the same is displayed for the visual experiment but not for the auditory one.

The results from the model trained with MFCCs show that except for *Belt* (recall = 92%), all other techniques achieved a considerably lower recall:  $39.6\% \leq \text{recall} \leq 53.7\%$ . *Belt* was also well recognised in the auditory experiment but not in the visual one, which suggests that acoustic properties characteristic of this technique, recognisable by ear, can be better captured by specific acoustic features such as MFCCs than by spectrograms. In fact, this is to some extent confirmed by the lower recall for *Belt* achieved by the CNN trained with spectrograms, i. e., 72%.

As shown in the user study, the most prominent confusion pattern displayed by the ML results is between *Trill*, *Trillo*, and *Vibrato*. This is clearly shown by the misclassification of *Trill* instances as *Trillo*: 24% and 26% for the model trained with MFCCs and spectrograms, respectively; as well as those misclassified as *Vibrato*: 16% and 38%, respectively. However, unlike in the user study, this confusion is not displayed in the opposite direction for the ML task, i. e., almost no instances of *Vibrato* are wrongly classified as neither *Trill* nor *Trillo*, misclassifications  $\leq 4.7\%$  for both models.

Interestingly, *Vibrato* is particularly well classified by the CNN, i. e., the model trained with the spectrograms (76.5%). This is also shown, to some extent, by the non-trained user participating in the visual task, for whom this technique is identified as the fourth best (41.7%). Differently, in the auditory study, *Vibrato* was the technique worse recognised (55.7%), and also for the NN (model trained with MFCCs), *Vibrato* was by far worse classified than for the CNN (53.7% vs 76.5%). This suggests that spectrograms are more suitable than acoustic features for characterising *Vibrato*'s properties, something observable both perceptually and from a computational point of view.

Finally, another prominent confusion pattern shown by the model trained with MFCCs is given by the high percentage of *Breathy* and *Inhaled* samples wrongly classified as *Vocal Fry*: 41.7% and 32.3%, respectively. This is partially mirrored by the results from the user study. A major confusion of *Inhaled* towards *Vocal Fry* is shown in the auditory task (16.9%); while a major confusion of *Breathy* towards *Vocal Fry* is shown in the visual experiments (15.5%). However, this confusion pattern is not shown for the model trained with spectrograms, for which the misclassification is shown between *Breathy* and *Inhaled* themselves: 22.6% of *Inhaled* samples are wrongly classified as *Breathy*. This suggests that training a ML model with acoustic features such as MFCCs might enable to artificially mirror, and even amplify, perceptual patterns shown by humans assessing different modalities. Something not possible when using spectrograms.

## 5 AI in Singing Education: Future Directions

Within the e-learning context, the most obvious use-case for a system able to recognise singing techniques is to provide feedback during students' training. For instance, since the singing technique *Breathy*, sometimes also referred to as *Rough*, is not desired in most genres [22], the ML-based application would first detect *Breathy* singing and subsequently suggest exercises to prevent it. Our comparative results confirm previous works on human vs. machine speech identification [13], indicating that the most predominant perception patterns shown by humans can be mirrored by ML. Nevertheless, while our models outperform non trained users, they are still less accurate than musically trained individuals. This indicates that standard ML architectures (as those used in this study) could be useful in providing feedback to beginners; however, more sophisticated models should be developed to meaningfully support advanced learners.

Our experimental outcomes also show that ML can capture confusion patterns coming from different perceptual modalities. This type of parallelism might be particularly informative when integrated in a XAI system, i. e., a ML systems which besides giving a prediction, is also able to provide a human-understandable reasoning justifying it. Thus, an XAI assistant could propose specific warm-up exercises depending on the singers' voice [37], subsequently assess whether the performed technique match the target, and finally illustrate (either visually or acoustically, depending on which feature representation is more informative), the predicted class (performed by the student) with respect to the target one (performed by a professional singer of the system's database).

Similarly, in base of our results, an XAI assistant could also highlight the most prominent confusion patterns shown for both perception and classification, i. e., the confusion between *Trill*, *Trillo*, and *Vibrato*. By displaying not only a visual (qualitative) representation but also precision (quantitative) measures achieved by the model,

learners might gain a more objective understanding of the similarities between techniques, something that beyond being perceived, can also be measured. At the same time, this would also illustrate real challenges in distinguishing amongst some techniques, which would encourage a more constructive learning experience. We believe that the use of intelligent systems as the one just described, specially when including an XAI component, would promote in first place exploration, motivated by the curiosity of interacting with the XAI assistant. Furthermore, another important expected outcome is to encourage the students to carefully evaluate their own performance, both visually and acoustically, which would lead to the development of self-reflective and critical skills.

Needless to say that such a system, in particular considering that the current results are way below human proficiency, would be expected to be used as a complementary tool to traditional teaching, i. e., supporting the student (specially during individual learning), but used under the close supervision of the teacher. Indeed, a full development of the system, including an user interface as well as a usability assessment in a real pedagogical scenario, is still to be done and constitutes one of our future priorities. In this process, a continuous monitoring from singing educators, critically assessing the potential of the system in complementing their own practice, is essential.

Finally, beyond supporting vocal training, the recognition of specific singing techniques in a song might also enable the classification of a given piece into a musical style or genre. For instance, the use of the *Belting* technique, particularly for women, is typically used in pop genre [38] while *Vibrato* is a strong indicator of operatic singing style [39]. The application of this technology in the context of automatic genre classification is clearly relevant for music recommendation systems [40]. Similarly, an efficient singing detection system could also be utilised for an e-learning application aimed to support students' understanding of musical genres in relationship to singing styles.

## 6 Conclusions

We presented a comparative assessment of humans' and ML performance in singing technique recognition. Our study shows that some confusion patterns typical of perception are mirrored by ML, which highlights the potential of supporting education with AI to illustrate (and further understand) perceptual processes. Our results also indicate that ML can capture patterns displayed by different perceptual cues: auditory and visual. This suggests that AI could be of interest to enhance learning through different perceptual modalities. The presented results seem to encourage further research on the application of XAI in singing pedagogy, which could promote students' reflective and critical skills, by this enhancing the outcomes of a student-centered learning process.

## References

1. White, B.D.: *Singing Techniques and Vocal Pedagogy*. University of Surrey Press, Surrey, UK (1985)
2. Stark, J.: *Bel canto: A history of vocal pedagogy*. University of Toronto Press, Toronto, Canada (1999)

3. Nakano, T., Goto, M., Hiraga, Y.: Subjective evaluation of common singing skills using the rank ordering method. In: Proceedings of the International Conference on Music Perception and Cognition, Bologna, Italy (2006) 1507–1512
4. Haßler, B., Major, L., Hennessy, S.: Tablet use in schools: A critical review of the evidence for learning outcomes. *Journal of Computer Assisted Learning* **32**(2) (2016) 139–156
5. Lã, F.M., Fiuza, M.B.: Real-time visual feedback in singing pedagogy: Current trends and future directions. *Applied Sciences* **12**(21) (2022) 10781
6. Wilson, P.H., Thorpe, C.W., Callaghan, J.: Looking at singing: Does real-time visual feedback improve the way we learn to sing. In: Proceedings of the Asia-Pacific Society for the Cognitive Sciences of Music Conference, Seoul, South Korea (2005) 4–6
7. Gupta, C., Li, H., Wang, Y.: Automatic leaderboard: Evaluation of singing quality without a standard reference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28** (2019) 13–26
8. Wilkins, J., Seetharaman, P., Wahl, A., Pardo, B.: Vocalset: A singing voice dataset. In: Proceedings of the International Society for Music Information Retrieval Conference, Paris, France (2018) 468–474
9. Yamamoto, Y., Nam, J., Terasawa, H., Hiraga, Y.: Investigating time-frequency representations for audio feature extraction in singing technique classification. In: Proceedings of the IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Tokyo, Japan (2021) 890–896
10. Burkhardt, F., Brückl, M., Schuller, B.: Age classification: Comparison of human vs machine performance in prompted and spontaneous speech. In: Proceedings of Studentexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung, Magdeburg, Germany (2021) 35–42
11. Koemans, J.: Man vs Machine: Comparing cross-lingual Automatic and Human Emotion Recognition in Background Noise. Master Thesis, Radboud University (2020)
12. Parada-Cabaleiro, E., Schmitt, M., Batliner, A., Hantke, S., Costantini, G., Scherer, K., Schuller, B.: Identifying emotions in opera singing: Implications of adverse acoustic conditions. In: Proceedings of the International Society for Music Information Retrieval Conference, Paris, France (2018) 376–382
13. Parada-Cabaleiro, E., Batliner, A., Schmitt, M., Schedl, M., Costantini, G., Schuller, B.: Perception and classification of emotions in nonsense speech: Humans versus machines. *PLoS ONE* **18**(1) (2023) e0281079
14. McCoy, S.: Singing pedagogy in the twenty-first century: A look toward the future. In Harrison, S.D., O’Byrne, J., eds.: Teaching singing in the 21st century. Springer, New York, NY, USA (2014) 13–20
15. Miller, D.G.: Resonance in singing: Voice building through acoustic feedback. Inside view press, Gahanna, OH, USA (2008)
16. Lã, F.M.: Teaching singing and technology. In Basa, K.S., ed.: Aspects of singing II: Unity in understanding - Diversity in aesthetics. VoxHumana, Nürnberg, Germany (2012) 88–109
17. Friberg, A., Bresin, R., Sundberg, J.: Overview of the kth rule system for musical performance. *Advances in Cognitive Psychology* **2**(2) (2006) 145
18. Rossiter, D., Howard, D.M.: Albert: a real-time visual feedback computer tool for professional vocal development. *Journal of voice: official journal of the Voice Foundation* **10**(4) (1996) 321–336
19. Welch, G.F., Howard, D.M., Himonides, E., Brereton, J.: Real-time feedback in the singing studio: An innovatory action-research project using new voice technology. *Music Education Research* **7**(2) (2005) 225–249
20. Stavropoulou, S., Georgaki, A., Moschos, F.: The effectiveness of visual feedback singing vocal technology in greek elementary school. In: Proceedings of the International Computing Music Conference, Athens, Greece (2014) 1786–1792



21. Yamamoto, Y., Nam, J., Terasawa, H.: Analysis and detection of singing techniques in repertoires of j-pop solo singers. In: Proceedings of the International Society for Music Information Retrieval Conference, Bangaluru, India (2022) 384–391
22. Xu, Y., Wang, W., Cui, H., Xu, M., Li, M.: Paralinguistic singing attribute recognition using supervised machine learning for describing the classical tenor solo singing voice in vocal pedagogy. *EURASIP Journal on Audio, Speech, and Music Processing* (1) (2022) 1–16
23. Bekkar, M., Djemaa, H.K., Alitouche, T.: Evaluation measures for models assessment over imbalanced data sets. *Journal of Information Engineering and Applications* **3** (2013) 27–39
24. Isherwood, N.: Vocal vibrato: New directions. *Journal of Singing* **65**(3) (2009) 271
25. Kob, M.: Physical Modeling of the Singing Voice. PhD thesis, Bibliothek der RWTH Aachen (2002)
26. Sangiorgi, T., Manfredi, C., Brusciaglioni, P.: Objective analysis of the singing voice as a training aid. *Logopedics Phoniatrics Vocology* **30** (2005) 136–146
27. Sundberg, J., Thalén, M.: Respiratory and acoustical differences between belt and neutral style of singing. *Journal of Voice* **29**(4) (2015) 418–425
28. LeBorgne, W.D., Lee, L., Stemple, J.C., Bush, H.: Perceptual findings on the Broadway belt voice. *Journal of Voice* **24**(6) (2010) 678–689
29. Gaskill, C.S., Erickson, M.L.: The effect of a voiced lip trill on estimated glottal closed quotient. *Journal of Voice* **22**(6) (2008) 634–643
30. Vanhecke, F., Moerman, M., Desmet, F., Six, J., Daemers, K., Raes, G., Leman, M.: Acoustical properties in inhaling singing: A case-study. *Physics in Medicine* **3** (2017) 9–15
31. Proutskova, P., Rhodes, C., Crawford, T., Wiggins, G.: Breathily, resonant, pressed-automatic detection of phonation mode from audio recordings of singing. *Journal of New Music Research* **42**(2) (2013) 171–186
32. Appleman, R., Bunch, M.: Application of vocal fry to the training of singers. *Journal of Singing* **62**(1) (2005) 53–9
33. Hoppe, D., Sadakata, M., Desain, P.: Development of real-time visual feedback assistance in singing training: A review. *Journal of Computer Assisted Learning* **22**(4) (2006) 308–316
34. Schleicher, A.: Educating learners for their future, not our past. *ECNU Review of Education* **1**(1) (2018) 58–75
35. Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.Y., Sainath, T.: Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing* **13**(2) (2019) 206–219
36. Sundberg, J.: Acoustic and psychoacoustic aspects of vocal vibrato. *Vibrato* (1995) 35–62
37. Elliot, N., Sundberg, J., Gramming, P.: What happens during vocal warm-up? *Journal of Voice* **9**(1) (1995) 37–44
38. Spivey, N.: Music theater singing... let's talk. Part 2: Examining the debate on belting. *Journal of Singing* **64**(5) (2008) 607–614
39. Howes, P., Callaghan, J., Davis, P., Kenny, D., Thorpe, W.: The relationship between measured vibrato characteristics and perception in western operatic singing. *Journal of Voice* **18**(2) (2004) 216–230
40. Schedl, M., Knees, P., McFee, B., Bogdanov, D., Kaminskas, M.: Music recommender systems. *Recommender Systems Handbook* (2015) 453–492

# Effects of Convolutional Autoencoder Bottleneck Width on StarGAN-based Singing Technique Conversion

Tung-Cheng Su, Yung-Chuan Chang, and Yi-Wen Liu \*

Department of Electrical Engineering, National Tsing Hua University  
ywliu@ee.nthu.edu.tw

**Abstract.** Singing technique conversion (STC) refers to the task of converting from one voice technique to another while leaving the original singer identity, melody, and linguistic components intact. Previous STC studies, as well as singing voice conversion research in general, have utilized convolutional autoencoders (CAEs) for conversion, but how the bottleneck width of the CAE affects the synthesis quality has not been thoroughly evaluated. To this end, we constructed a GAN-based multi-domain STC system which took advantage of the WORLD vocoder representation and the CAE architecture. We varied the bottleneck width of the CAE, and evaluated the conversion results subjectively. The model was trained on a Mandarin dataset which features four singers and four singing techniques: the chest voice, the falsetto, the raspy voice, and the whistle voice. The results show that a wider bottleneck corresponds to better articulation clarity but does not necessarily lead to higher likeness to the target technique. Among the four techniques, we also found that the whistle voice is the easiest target for conversion, while the other three techniques as a source produce more convincing conversion results than the whistle.

**Keywords:** singing voice conversion, singing technique conversion, convolutional autoencoder, generative adversarial networks

## 1 Introduction

Singing voice conversion (SVC) is a task of converting prosodic features while retaining the linguistic content. The prosodic features to be converted can include singer identity, emotions, and singing techniques. Unlike speech conversion, the pitch contour of the singing voice is usually unchanged in SVC so that the melody of the original voice is preserved.

In recent years, many deep learning based methods of voice conversion (VC) have been shown to achieve state-of-the-art performance [1], and several methods have also

---

\* We thank the National Science and Technology Council of Taiwan for supporting this research under Grant No. 109-2221-E-007-094-MY3



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

been applied to SVC [2–8]. Compared to speech, singing is rich in terms of the voicing techniques that singers can apply to enhance their expressiveness, such as to switch between their chest voice, falsetto, whistle voice, and so on. Thus, one’s singing technique is an integral part of their singing performance [11], yet computer-based singing technique conversion (STC) is a less researched field compared to other SVC tasks. Previous works have applied the autoencoder (AE) for STC [5, 6] as well as other VC tasks [2–4]; however, how the architecture of the AE affects the synthesized voice quality has not been thoroughly studied.

Therefore, in this study we focus on the *bottleneck* of convolution autoencoders (CAEs) because it corresponds to the latent space representation of the features. Although the conversion process is operating within the latent space whose dimension is equal to the width of the bottleneck, the bottleneck architectures in existing SVC and STC models seem to be arbitrarily designed. To the best of our knowledge, few studies [9] focused on the effects of bottleneck architecture, and no study on bottlenecks was dedicated to STC or SVC in general, despite of AE’s prevalence in the field. To gain about to STC and explore bottleneck architectures, we presently built a STC system based on StarGAN [10], and experiments were conducted to compare the voice synthesis quality of STC with different bottleneck sizes.

The rest of this paper is organized as follows. Section 2 describes our voice conversion system and introduces StarGAN. Then, details of experiments and evaluation methods are described in Sec. 3. Results are reported and discussed in Sec. 4, and conclusions are given in Sec. 5.

## 2 Voice Conversion System Overview

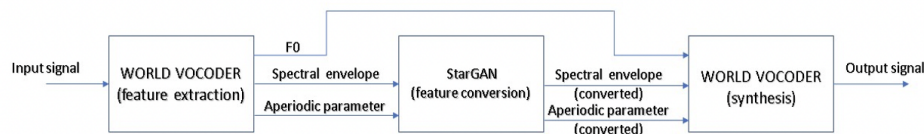


Fig. 1: The overall system block diagram for the present research

Figure 1 shows the overall system diagram of this research. We adopted the WORLD vocoder [14] to represent the signal by three sets of features, namely the fundamental frequency (F0), the aperiodic parameters (AP), and the spectral envelope (SP). The features were extracted every 5 ms. We concatenated SP and AP into 56-dimension Mel Cepstral Coefficients (MCC) for the SP plus 4 dimensions for the AP. Putting this 60 dimension feature into the StarGAN[10] model (illustrated in Fig. 2), we could then convert the signal in the vocoder domain. Figure 1 also shows that F0 was directly passed to the synthesizer in order to preserve the pitch of a singing voice.

The usage of the WORLD vocoder might limit the synthesized audio quality compared to what could be achieved by neurovocoders, such as HiFi-GAN[15]. Nevertheless, we chose to work with WORLD for it enables us to separately consider F0 and other acoustic features, which suits our purpose of transforming the singing technique while maintaining the original F0.

**StarGAN** StarGAN[10] is a GAN-based model consisting of a generator and a discriminator. The generator adopts a convolutional autoencoder architecture as shown in Fig. 2, which could be divided into two stages. The first stage can be viewed as an encoding stage that downsamples features to the latent space; the second stage is a decoding stage that upsamples the latent feature back to the original space. In this research, the original space is the 60-dimensional WORLD vocoder output mentioned above. In Fig. 2, the middle section between the last layer of the encoder and the first layer of the decoder is referred to as the *bottleneck*.

The first layer before the downsampling layers uses a kernel of size  $3 \times 9$  with a stride of 1. The output of each of the four downsampling layers are  $30 \times 200$ ,  $15 \times 100$ ,  $5 \times 100$ , and  $1 \times 100$  respectively; their kernel sizes are  $4 \times 8$ ,  $4 \times 8$ ,  $4 \times 7$ , and  $5 \times 7$  with corresponding stride of  $(2, 2)$ ,  $(2, 2)$ ,  $(3, 1)$ , and  $(1, 1)$  respectively. For experiments, the model with only the first two, three, or four down/upsampling layers are used. The bottleneck size is thus controlled by the last downsampling layer, which is the same as the encoder's output. Upsampling layers mirror the downsampling layers with the same number of layers and their kernel size and stride. The last layer after the upsampling layers uses a kernel of size  $7 \times 7$  with a stride of 1.

In Fig. 2, the attribute vector encodes the singing technique of an audio file. Here, we define *domain* as a set of audio files with the same attribute. Traditionally, a SVC model is only capable of performing conversion from one domain to another. In contrast, StarGAN achieves conversion between multiple domains with one single network. A key component of our proposed network is to represent the attribute by several channels with the same height and width as the bottleneck, in a similar fashion as one-hot vectors<sup>1</sup>. The target singing technique was thus informed to the decoder of CAE via the attribute channels.

The StarGAN is trained by minimizing the sum of three losses. The first is the adversarial loss, which makes the discriminator and generator work in an antagonistic fashion so that the generated features become more and more realistic. The second is the classification loss. The discriminator learns to classify WORLD vocoder features in the training set, while the generator aims to convert the features so that the classifier would put them into the target category. The third is the reconstruction loss. It forces the generator to reconstruct original features when given the original attribute vector. A lower reconstruction loss indicates less information loss in the bottleneck. In STC, lower reconstruction loss often indicates higher articulation clarity. Empirically, we observed that the width of the bottleneck had a significant impact on the reconstruction loss. This observation motivated us to conduct the experiments described next.

<sup>1</sup> We specify one out of four possible target domains by setting one of four channels to be all 1.

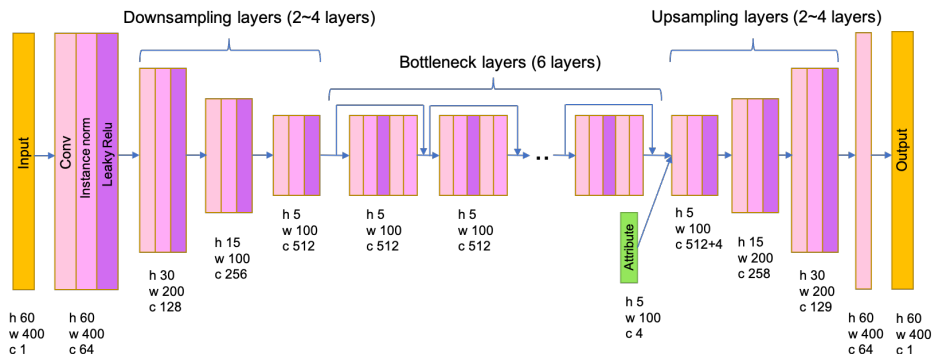


Fig. 2: The proposed convolutional autoencoder architecture for StarGAN. The figure shows a version of the model with three down/upsampling layers. The three downsampling layers have kernel sizes  $4 \times 8$ ,  $4 \times 8$ , and  $4 \times 7$ , with strides of  $(2, 2)$ ,  $(2, 2)$ , and  $(3, 1)$  respectively; the upsampling layers' settings mirror those of the downsampling layers.

### 3 Experiments

In this section, we describe the dataset, the network training strategies, the bottleneck architecture that was varied, and the evaluation metrics for this study.

#### 3.1 Dataset

While existing datasets, such as VocalSet [13], already featured diverse singing techniques and were used in previous STC studies [5, 6], we decided to collect a new dataset from scratch so as to focus on the singing techniques that are common in Chinese Mandarin pop singing. Our dataset contains non-parallel singing voice of four singing techniques, namely the chest voice, falsetto voice, whistle voice, and raspy voice. Two male and two female singers were recruited. Each singer sang Chinese Mandarin pop songs in their preferred techniques while they were instructed to maintain the same pitch range across different techniques, except for the whistle voice. Since the techniques were constrained by the singers' preferences, not all techniques were successfully recorded from all singers. In the end, the chest voice was sung by all four singers for a total of 53 minutes, the falsetto voice was sung also by all four singers for a total of 51 minutes, the whistle voice was sung by one female singer for 20 minutes, and the raspy voice was sung by one male singer for 7 minutes.

The audio was recorded with a large diaphragm condenser microphone<sup>2</sup> and sampled at 48 kHz in a vocal booth to approximate the recording environments of pop music vocals. The recordings were cut phrase by phrase afterwards, with each phrase being between 5 to 12 seconds. The audio data were then re-sampled to 16 kHz for STC experiments.

<sup>2</sup> Sontronics STC-2, without the built-in high-pass filtering or -10dB passive attenuation

### 3.2 Training configurations

For data augmentation, the model randomly selected 400 continuous frames (2 seconds) from the training set each time. We used the Adam optimizer [16] for training with  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$  for all the models. Each model was trained for 250,000 iterations with  $10^{-4}$  learning rate at the start and decays for the last 100,000 iterations. To optimize training quality, we updated the discriminator once for every three generator updates.

### 3.3 Bottleneck configurations

Our experimental design aims to investigate the influence of bottleneck width on singing voice technique conversion performance. Hence, we formulated three different bottleneck sizes,  $15 \times 256$ ,  $5 \times 512$ , and  $1 \times 1024$  (features x channels). To only change bottleneck sizes and not other CNN settings, downsampling/upsampling layers are added or eliminated for different sizes; as illustrated in Fig. 2, these three configurations correspond to encoders with two, three, and four downsampling layers, respectively.

### 3.4 Subjective Evaluation

The subjective evaluation test consisted of three listening tasks, and 27 participants were recruited.

**Bottleneck Comparison** We compared the conversion performance and articulation clarity of the synthesized voice produced by three different bottleneck widths across four distinct source techniques. Eight listening comparison tests were created. In each test, participants were provided with source and target audio samples beforehand for familiarization purposes. Then, they were asked to rank the audio conversion results of three different bottlenecks in terms of conversion performance and articulation clarity. The ranking was then given a score from 1-3, with the best receiving 3 points, 2 points for second-best, and 1 point for the worst.

**Likeness to the target** The performance of the multi-domain STC model was evaluated in terms of likeness to the target technique after a subject listened to the transformed audio. The model with 3 encoding layers was chosen for evaluation. Similar to the bottleneck comparison experiment, we provided the participants with source and target audio files for familiarization, but asked them to rate the transformed audio on a scale of 1-5, where 5 means most similar to the timbre of the target file, and 0 means most similar to that of the source file. For this part of the experiment, we performed pitch shifting when the whistle voice was involved in the conversion so the target pitch range sounded natural to the intended singing technique.

**Sound Quality** The final part of subjective evaluation aims to assess degradation in the sound quality after STC. To achieve this, we selected four audio files (C, F, W, R) and processed them by analysis-then-synthesis using the WORLD vocoder; the same audio files were also subjected to STC (C2F, F2W, W2R, R2C) so their sound quality could be evaluated.

## 4 Results and Discussion

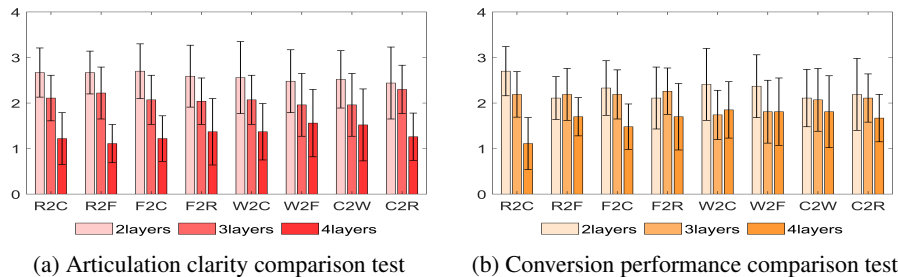


Fig. 3: For both comparison tests, each audio was scored on a scale of 1-3, with the best receiving 3 points, 2 points for second-best, and 1 point for the worst. C = chest voice, F = falsetto, W = whistle voice, and R = raspy voice. The error bar represents one standard deviation.

Figure 3 shows the mean and standard deviation of the 3-point scores for 8 different source-target combinations under three different widths of the CAE bottleneck. The results indicate that, in general, the widest bottleneck (which is 15 and corresponds to two encoding layers) produced the most clearly articulated speech. This makes sense, since a wide bottleneck encodes the input information into a higher-dimensional latent space and thus preserves more complete information about the voice content. A narrower bottleneck, in contrast, encodes information into a lower-dimensional latent space, making it more difficult to reconstruct the audio.

However, better articulation clarity was not always accompanied with a better conversion performance. Particularly, for conversion between raspy and falsetto voices (i.e., R2F and F2R), the results obtained with three downsampling layers in the encoder were slightly superior to those obtained with two downsampling layers. Also, while the system with four downsampling layers produced poor articulation clarity for the W2C transformation, it slightly outperformed the system with three encoding layers in terms of conversion performance. These findings suggest that the selection of an optimal bottleneck size is critical for singing voice techniques conversion, and best setting might depend on the intended source-target combination.

Figure 4 summarizes our evaluation of the system in terms of the timbral likeness to the target technique after conversion. In (a), the average results of four *source* techniques are shown, and (b), four *target* techniques; the average results of all source to target pair are shown in (c). The results indicate that the conversion of whistle voice to other techniques had limited success (with mean score < 3.0). However, the transformation from other techniques to whistle voice was effective, with a mean score of 3.63. This can be attributed to the unique timbre and the high pitch range of whistle voice, which were probably difficult to remove and easy for the listeners to recognize. The conversion of falsetto voice as a source has yielded satisfactory results, but achieved

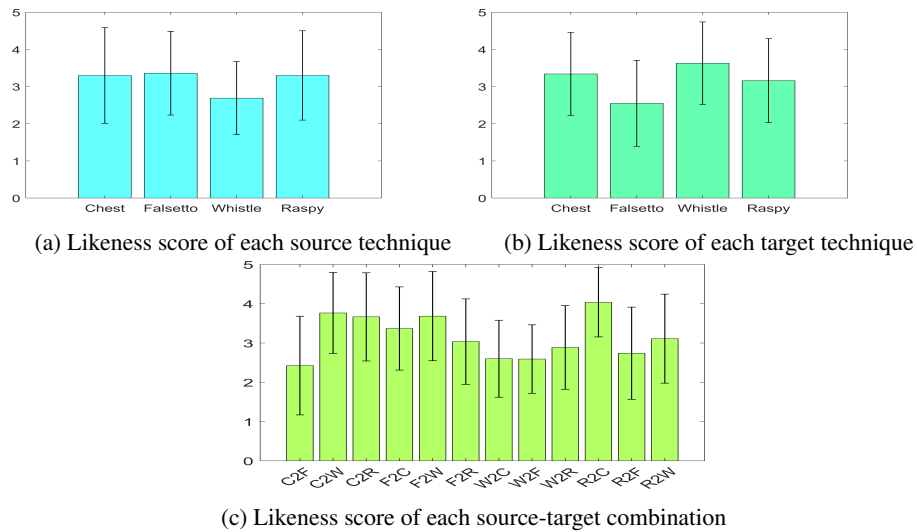


Fig. 4: For the likeness tests, 5 means most similar to the timbre of the target, and 0 means most similar to that of the source. C = chest voice, F = falsetto, W = whistle voice, and R = raspy voice. The error bar represents one standard deviation.

significantly poorer score as a target. Chest and raspy voices produced comparable results, regardless of them being utilized as source or target technique.

The mean opinion score (MOS) on a five-point scale was  $3.55 \pm 1.04$  for WORLD vocoder round-trip, and  $2.80 \pm 1.30$  after STC. Although the mean score decreased reasonably by 0.75 due to STC, several limitations of WORLD vocoder were noted in this research. First, we observed that WORLD encoding-decoding produced some cracking or breaking sound when we tested on the raspy voice. We suspect that the WORLD vocoder might have been optimized to handle monophonic sounds, whereas a raspy voice can have multiple concurrent fundamental frequencies, causing the vocoder to misinterpret the data. Additionally, we observed that the aspiration that was salient in the whistle voice could cause errors in voiced/unvoiced classification and thus pose a challenge for the vocoder-domain processing. To summarize, future fine-tuning of the vocoder should be warranted for improving the quality of STC.

## 5 Conclusion

In this research, we created a singing voice technique dataset that includes chest voice, falsetto, raspy voice, and whistle voice. The dataset was adopted to train a multi-domain singing technique conversion model. We found that the size of CAE’s bottlenecks affected the clarity of pronunciation and the likeness to the target technique after conversion, and the optimal size might depend on the intended source-target combination. Furthermore, we noted several audible defects when handling raspy or whistle voices with the WORLD vocoder, which ultimately limited the audio quality of STC. In the



future, we hope to continue improving the audio quality of STC and create different ways of vocal music production for amateurs and professionals to use.

## References

1. Sisman, B., Yamagishi J., King S. and Li H.: An overview of voice conversion and its challenges: From statistical modeling to deep learning. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 132-157 (2021)
2. Kameoka, H., Kaneko T., Tanaka K. and Hojo N.: StarGAN-VC: Non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, pp. 266-273 (2018)
3. Nachmani E. and Wolf L.: Unsupervised singing voice conversion. In: *Interspeech 2019*, pp. 2583-2587 (2019)
4. Deng, C., Yu C., Lu H., Weng C. and Yu D.: Pitchnet: Unsupervised singing voice conversion with pitch adversarial network. In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 7749-7753 (2020)
5. Luo, Y. -J., Hsu C. -C., Agres K. and Herremans D.: Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders. In: *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 3277-3281 (2020)
6. O'Connor, B., Dixon, S. and Fazekas, G.: Zero-shot singing technique conversion. In: *The 15th International Symposium on Computer Music Multidisciplinary Research*, pp. 235-244 (2021)
7. Liu, S., Cao, Y., Hu, N., Su D. and Meng H.: Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation. In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*, Shenzhen, China, pp. 1-6 (2021)
8. Liu, S., Cao, Y., Su D. and Meng H.: DiffSVC: A diffusion probabilistic model for singing voice conversion. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia, pp. 741-748 (2021)
9. Manakov, I., Rohm, M. and Tresp, V.: Walking the tightrope: An investigation of the convolutional autoencoder bottleneck. *arXiv preprint arXiv:1911.07460*. (2019)
10. Choi, Y., Choi, M., Kim, M., Ha, J. -W., Kim, S. and Choo, J.: StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 8789-8797 (2018)
11. Heidemann, K.: A system for describing vocal timbre in popular song. In: *Music Theory Online*, vol. 22, (2016)
12. Goodfellow, Ian, et al.: Generative adversarial networks. *Communications of the ACM* 63.11 (2020): 139-144.
13. Wilkins, J., Seetharaman, P., Wahl, A., Pardo, B.: VocalSet: A singing voice dataset. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)* Paris, France, September 23-27, 2018, pp. 468-474 (2018)
14. Morise, M., Yokomori, F. and Ozawa, K.: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems* 99.7, pp. 1877-1884 (2016)
15. Kong, J., Kim, J. and Bae, J.: HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 17022-17033 (2020)
16. Diederik P. K. and Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

# Historical Changes of Modes and their Substructure Modeled as Pitch Distributions in Plainchant from the 1100s to the 1500s

Eita Nakamura<sup>1</sup>, Tim Eipert<sup>2</sup>, and Fabian C. Moss<sup>2</sup> \*

<sup>1</sup> Kyoto University, Japan

<sup>2</sup> Julius-Maximilians-Universität Würzburg, Germany

eita.nakamura@i.kyoto-u.ac.jp

tim.eipert@uni-wuerzburg.de

fabian.moss@uni-wuerzburg.de

**Abstract.** Large-scale quantitative investigations into the cultural evolution of music have mostly focused on only a limited range of time periods and genres. Here, we analyze more than 40 000 pieces of plainchant to better understand the evolution of modes and pitch distributions in a period of five centuries that saw the development of the Western modal practice. Specifically, we employ a hierarchical Markov mixture model to analyze the eight medieval modes and their substructure represented as pitch distributions and observe their historical changes. We found that the individual modes exhibit internal clusters, that the relative frequencies of the eight modes remained remarkably stable over time, and that there were comparatively large changes in the pitch distributions of individual modes. We discuss our results on the background of musicological insights and point to the need for further interdisciplinary work.

**Keywords:** computational musicology; cultural evolution; plainchant; statistical modeling; mode classification.

## 1 Introduction

Quantitative analysis of music evolution has been gaining increasing attention in recent years. Previous studies have observed trends and regularities in musical styles in several cultural domains such as Western classical music [1–6] and popular music [7–10]. There are also several studies on evolution of folk and world music [11, 12]. Such studies inherently rely on the availability of large-scale music data that also include information about the time of composition to be used for evolutionary analysis. Most studies using quantitative methods in the Western classical context have focused on music from

\* This work was in part supported by JSPS KAKENHI Grant Numbers 21K12187, 21K02846, and 22H03661, and JST FOREST Program Grant Number JPMJPR226X.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Number	Name	Type	Final	Reciting tone	Range
1	Dorian	authentic	D	A4	D4–D5
2	Hypodorian	plagal	D	F4	A3–A4
3	Phrygian	authentic	E	C5	E4–E5
4	Hypophrygian	plagal	E	A4	B3–B4
5	Lydian	authentic	F	C5	F4–F5
6	Hypolydian	plagal	F	A4	C4–C5
7	Mixolydian	authentic	G	D5	G4–G5
8	Hypomixolydian	plagal	G	C5	D4–D5

Table 1: The eight medieval modes. The reciting tone and range are represented in the standard pitch notation for clarity, but the pitches have only relative meanings here.

the Renaissance, Baroque, Classical, or Romantic periods, and thus covered both modal and tonal practices [13]. However, this concentration on the period from approximately the 16th to the 19th centuries ignores several preceding centuries in which the Western modal practice developed. Here, we draw our attention to medieval monophony and its manifestation in chants [14], in order to shed light on our understanding of the development of pitch organization in Western music from its earliest beginnings.

Arguably, the most fundamental concept of medieval music theory for the categorization of chants is that of a *mode*. In the liturgical practice since the eighth century, we can find so-called *tonaries*: books that categorize chants into eight modes [15]. These are then identified in later manuscripts with one of the *finals* D, E, F, and G, each coming in two variants: authentic (the final is usually the lowest note) and plagal (the final is usually the central note in terms of pitch height). The eight medieval modes are conventionally labeled with a number or a Greek name (Table 1). It is, however, not entirely clear whether the concept of mode was merely used as a classification system to organize existing musical material into distinct categories based on some set of shared features (e.g. pitch-related or other), or whether mode was, in contrast, a concept existing prior to composition, that allowed music to be sung “in a certain mode.” It seems likely that these two conceptualizations were never strictly separated but that they are rather intricately entangled. Both the categorization and composition aspects probably played a role to varying degrees, as mode is a complex concept influenced by ancient music theory as well as medieval practice [16].

Commonly, the mode of a chant is determined on the basis of pitch-related features, e.g. which pitches are used in which frequency, which pitches are initial or final to a chant, etc. Consequently, modal characteristics should be reflected in pitch-distribution statistics, although it is not clear whether a mode can be modeled as a simple pitch distribution or a distribution with substructure reflecting other features such as the function of chant in liturgical use. These arguments indicate the importance of analyzing the evolution of modes and related pitch distributions and the need for addressing two major research questions: (i) How did the relative frequencies of modes change over time? (ii) How did the pitch distributions of individual modes change over time?

Our contribution shares a research interest with two prior studies on medieval mode. In their pioneering study on the pitch-class distributions of the eight modes, Huron and

Veltman [17] found a ‘supra-modal group’ consisting of modes 3, 5, and 8 (sharing reciting tone c), and another group consisting of modes 1, 4, and 6 (sharing reciting tone A), and suggested that this polarization facilitated the major-minor bifurcation in the 17th century. There has also been some musicological criticism on this work. Specifically, Wiering [18] noted that the abstraction of a pitch class ignores the different functions of octave-equivalent tones in medieval music. Moreover, the assumption of chromatic transpositions contradicts the medieval practice (and virtually all musical practice prior to the Romantic era). It was also noted that the melodic aspects were also ignored in the pitch-class profile approach, which does not account for pitch transitions and cannot represent subtle distinctions of modes based on melodic motions.

In a more recent study, Cornelissen et al. [19] examined mode classification in medieval plainchant melodies using a distributional model that improved some of the shortcomings of the earlier study by including both pitch information (as opposed to pitch classes) and n-gram models. Using the tf-idf vectors of chants, their model achieved a classification F-score of 93–95% and maintained F-scores of 81–83% even without absolute pitch information. The result suggested that plainchant contains ‘natural units’ that lie somewhere between the levels of individual notes and complete phrases.

To address our two research questions about mode in medieval plainchant, we analyze a large corpus of monophonic melodies that were almost exclusively written for liturgical use (see Sec. 2.1). We analyze the historical changes in pitch distributions in the chants whose source manuscripts date back to the range between the 12th and the 16th centuries. To examine the substructure of modes, we go beyond the approaches in previous research and apply an elaborated technique of machine learning to infer internal clusters of pitch distribution from data. We formulate a hierarchical Markov mixture model for this purpose and study the inferred parameters in terms of mode classification ability and the relationship with chant genres. We then analyze the historical changes in the relative frequencies of these clusters to draw conclusions for our research questions.

## 2 Method

### 2.1 Data

Our data source is *Cantus* [20], a database for Latin ecclesiastical chant that was created with the goal of digitizing and distributing indices of medieval chant manuscripts and early printed books [21] (see Fig. 1a for an example). Developed by Steiner in the 1980s, the *Cantus* database continues to provide an essential resource for scholars and researchers studying the history and evolution of Latin ecclesiastical chant. The central focus of the *Cantus* database is the so-called *liturgical office*, which is, besides the eucharist (mass), an essential element of the liturgy in almost all Christian denominations. It is a shared act of prayer, typically sung, that involves reciting the psalms and other supplementary texts throughout the various times of day and days of the year (referred to as the canonical hours) [22].

For the present study, we draw on a publicly available data resource that contains a total of 63 628 chants from the *Cantus* database, including a rich set of metadata [19]. Three types of information are used in particular: melody, mode, and source date. We

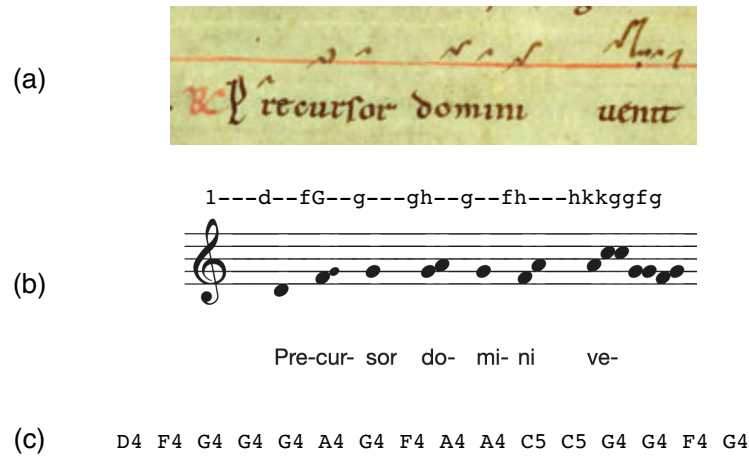


Fig. 1: Three different representations of a responsory chant incipit in mode 8 from the Cantus database (<https://cantus.uwaterloo.ca/chant/284929>). (a) Detail of the manuscript scan with only one staff line indicating the position of pitch F. (b) Volpiano encoding of the responsory assigning pitches to letters a to p, and modern staff notation. (c) Pitches in standard notation (note name + octave).

thus excluded in the following analyses chants with less than ten notes, without an annotated mode, or without source date information. Furthermore, we use the genre metadata with labels such as antiphon, responsory, and responsory verse<sup>1</sup>. Melodies are represented by a string in *Volpiano* encoding [23] (Fig. 1b). The alphabets represent pitches in the ascending order, and dashes indicate the hierarchical segmentation into words, syllables, and neumes. Conversion from this format to the standard pitch notation is straightforward (Fig. 1c).

A chant's metadata often contains a mode attribute extracted from the containing manuscript or assigned by experts. A majority of annotated modes are a single number from 1 to 8 corresponding to the eight modes explained in Sec. 1; only chants classified in these modes are used for analysis. Some of the other chants are transposed chants, indicated with a T, or verses that are sung with a special melody, indicated with an S. There are also chants whose mode is unknown or uncertain, indicated with a question mark. The date of a source manuscript, if it is given, is represented as a year range (e.g. 1201–1300). We use the middle values of these ranges as the time stamps of contained chants. After these data selection steps, we were left with 41 158 chants in total used for the following analysis.

<sup>1</sup> An antiphon is a short, mostly syllabic refrain that was commonly sung before and after a psalm in the liturgical chant. A responsory typically follows a scripture reading and comprises a verse sung by a soloist or small group, succeeded by a response from the choir or congregation. Its melodic structure is often more intricate and melismatic than an antiphon, and its content closely aligns with the theme of the reading it accompanies.

## 2.2 Markov mixture model

We use a Markov model to parameterize the pitch distribution of a certain set of chants, e.g. chants in one of the eight modes. To analyze the substructure of modes, we formulate a Markov mixture model to find internal clusters of chants according to their pitch distributions. This model can also be used to represent the different pitch distributions of the eight modes and to automatically estimate the mode of an unseen piece.

We represent a piece as a sequence of pitches  $\mathbf{x} = (x_\ell)_{\ell=1}^L$ . A Markov model describes the generative probability of  $x_\ell$  by the initial probability  $\psi^{\text{ini}}(q) = P(x_1 = q)$  and the transition probabilities  $\psi(q', q) = P(x_\ell = q | x_{\ell-1} = q')$  as

$$P(\mathbf{x}|\boldsymbol{\psi}) = \psi^{\text{ini}}(x_1) \prod_{\ell=2}^L \psi(x_{\ell-1}, x_\ell). \quad (1)$$

Given a set of pieces  $(\mathbf{x}_n)_{n=1}^N$ , the set of parameters  $\boldsymbol{\psi} = \{\psi^{\text{ini}}(q), \psi(q', q)\}$  can be optimized for maximizing the likelihood  $\prod_{n=1}^N P(\mathbf{x}_n|\boldsymbol{\psi})$ . The parameters learned in this way represent the pitch distribution in these pieces.

In a Markov mixture model, we consider a set of  $K$  Markov models parameterized by  $\boldsymbol{\psi}_k$  ( $k = 1, \dots, K$ ), each representing the pitch distribution of a class of data, and a mixture probability  $\pi_k$  representing the relative frequency of the  $k$ -th class. The probability of a pitch sequence is given as

$$P(\mathbf{x}) = \sum_{k=1}^K \pi_k P(\mathbf{x}|\boldsymbol{\psi}_k). \quad (2)$$

This model is not to be confused with a hidden Markov model, in which each latent variable is introduced for each pitch. In the Markov mixture model, the latent variable  $k$  is introduced for each sequence.

Supervised and unsupervised training methods can be derived for estimating the parameters  $\pi_k$  and  $\boldsymbol{\psi}_k$  from data. In the supervised setup, we consider that we have data divided into  $K$  classes. We can then estimate the mixture probability  $\pi_k$  from the relative frequencies of the individual classes and parameters  $\boldsymbol{\psi}_k$  from the subset of data in the  $k$ -th class. For example, using the mode label in the present data, we can train the Markov mixture model with eight classes corresponding to the eight modes. In the unsupervised setup, we can train a Markov mixture model from a dataset of pitch sequences without class labels. In this case, the number  $K$  of classes is an adjustable hyperparameter that defines a resolution of the analysis. The EM algorithm can be applied to estimate the parameters  $\pi_k$  and  $\boldsymbol{\psi}_k$ .

Given a Markov mixture model with trained parameters, we can estimate the posterior probability of the class of an unseen piece by the following equation:

$$P(k|\mathbf{x}) = \frac{P(\mathbf{x}, k)}{P(\mathbf{x})} \propto \pi_k P(\mathbf{x}|\boldsymbol{\psi}_k). \quad (3)$$

We can then take the class  $\hat{k}$  that maximizes the posterior probability as the estimated class for the piece.

In our analysis, we use the Markov mixture model in a hierarchical manner. We train a Markov mixture model with  $K_m$  classes from a subset of data in mode  $m$ , thus obtaining parameters  $\pi_k^{(m)}$  and  $\psi_k^{(m)}$ . We combine the eight Markov mixture models to obtain a hierarchical Markov mixture model represented as

$$P(\mathbf{x}) = \sum_{m=1}^8 \sum_{k=1}^{K_m} \pi_m \pi_k^{(m)} P(\mathbf{x}|\psi_k^{(m)}). \quad (4)$$

As in Eq. (3), we can use this model to estimate the posterior mode probability of a piece  $\mathbf{x}$  as

$$P(m|\mathbf{x}) \propto \sum_{k=1}^{K_m} \pi_m \pi_k^{(m)} P(\mathbf{x}|\psi_k^{(m)}). \quad (5)$$

The Markov mixture model and its hierarchical version can be used for addressing our two research questions. First, since the mode-level mixture probabilities  $\pi_m$  represent the relative frequencies of modes, the first research question can be examined by analyzing the temporal changes in their values over time. Next, the component Markov models of the hierarchical Markov mixture model represent internal clusters within individual modes and can thus be used for analyzing the modes' substructure. Specifically, the second research question can be examined by analyzing the temporal changes in the relative frequencies of the internal clusters.

The set of pitches, or the state space of Markov models, was constructed from the Cantus database. There were 22 pitches ranging from the lowest pitch F3 to the highest pitch D6. To account for the specific statistical features of the last note of a piece, we also introduce an additional state 'end' in the Markov models so that the statistics of the last note is incorporated in the transition probabilities to the 'end' state. Therefore, the number of states of the Markov models was 23.

### 3 Result

#### 3.1 Internal clusters of modes

We trained the hierarchical Markov mixture model with  $K_m = 3$  for all modes  $m$  (Fig. 2). The obtained internal clusters of a mode are ordered in the average time of appearance weighted by the relative frequencies, from the earliest to the latest (see Sec. 3.3). We can observe that the three internal clusters exhibit notable differences in pitch-class transition probabilities in mode 1, 3, and 5, whereas the substructures are less visible in the other modes.

The result of hierarchical clustering of mode-level pitch distributions is also shown in Fig. 2. The same tree structure was obtained when the symmetric Kullback–Leibler divergence and the squared distance were used as the distance measure. This result is similar to the result of [17], which used a single source and a subset of the data we used. Therefore, the structure with two supra-modal groups, one with mode 1, 4, and 6 and the other one with mode 3, 5, and 8 as core members, is shown to be a general characteristic over the time period from the 12th century to the 16th century.

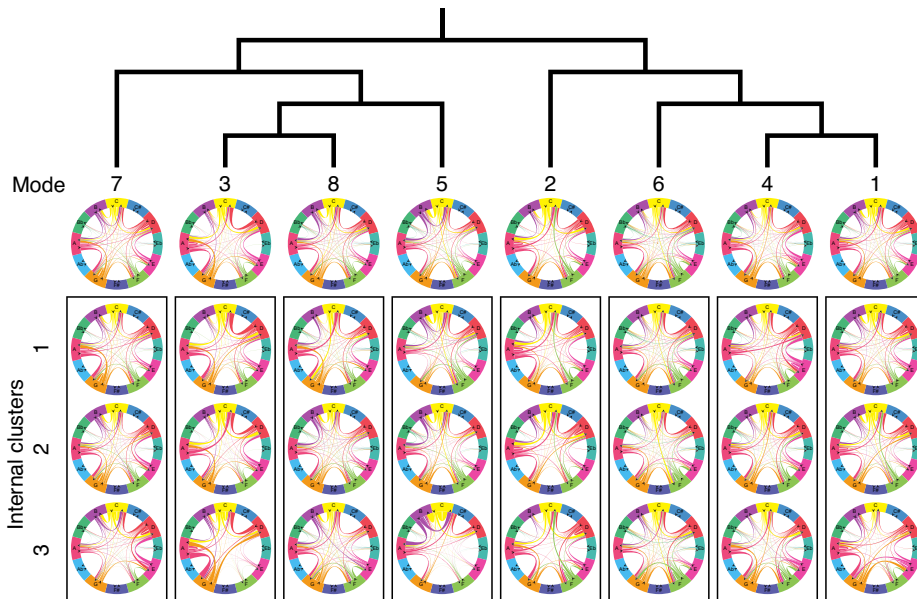


Fig. 2: Visualization of pitch distributions of the eight modes and their internal clusters. Here, the pitch bigram probabilities are reduced to pitch class bigram probabilities visualized as the band widths. The pitch classes are represented by different and arbitrary colors and the colors of the bands indicate the pitch class from which the corresponding pitch transitions occur. The dendrogram was obtained by complete-linkage hierarchical clustering.

To quantitatively measure the effect of the internal clusters for mode classification, we evaluated the accuracy of mode classification by the Markov mixture models with and without internal clusters. In this analysis, we randomly split the data into training (70%) and test (30%) data and used the former data for unsupervised training of the model and the latter data for evaluation. The accuracy was 84.0% without internal clusters and 85.3% with internal clusters, showing the positive effect of more precisely representing the distribution of pitch distributions using internal clusters. It was also confirmed that the accuracy further increased with  $K_m = 4$  and 5, indicating the existence of finer-grained internal clusters. As a reference, a previous study [19] reported an F-score of 88–90% by a classification method using the pitch profile (unigram distribution) and 91–92% using pitch bigram features, evaluated on a smaller subset of data with a larger average piece length. We expect that the present Markov model without internal clusters has an equivalent classification ability with these methods when compared in the same setup since the pitch transition probabilities are generally more informative than the pitch profile and essentially equivalent to the bigram probabilities.

Fig. 3 shows the confusion matrix of mode classification by the hierarchical Markov mixture model ( $K_m = 3$ ). The result shows that the classification errors generally occur within each of the two supra-modal groups, as expected. We can also find relatively high



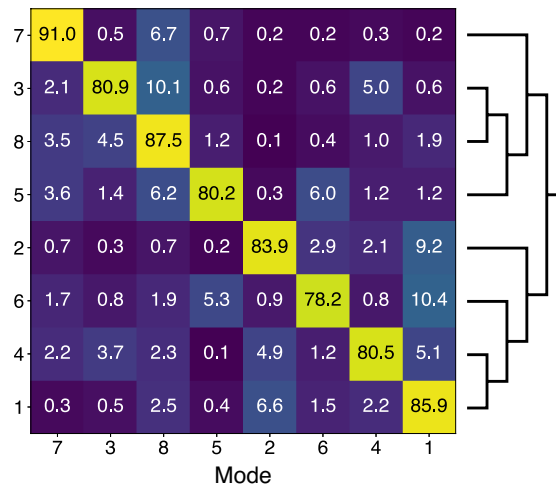


Fig. 3: Confusion matrix of mode classification rates (%) by the hierarchical Markov mixture model.

error rates across these groups between pairs of modes sharing the final, that is, between modes 3 and 4 (final E) and between modes 5 and 6 (final F).

### 3.2 Internal clusters and genres

To investigate the relationship between the internal clusters and the genres of chants, we analyzed their correlations. We focus on the three main genres, antiphon, responsory, and responsory verse, which cover 91% of the data, and analyzed the proportion of genres  $P(g|k)$  in each internal cluster  $k$ . More specifically, we used the posterior probability  $P(k|n)$  of internal clusters  $k$  for piece  $n$  and its annotated genre  $g_n$  to calculate the genre probability  $P(g) \propto \sum_n \delta(g_n, g)$ , the conditional internal cluster probability  $P(k|g) \propto \sum_n \delta(g_n, g)P(k|n)$ , and the proportion of genres  $P(g|k) \propto P(k|g)P(g)$  in internal cluster  $k$ .

The result in Fig. 4 shows that, for all modes, most of the pieces in genre ‘responsory verse’ belong to the third internal cluster. Since the labels for the internal clusters only indicate the order in the average time of appearance weighted by the relative frequencies, such a relationship indicates the heterogeneous time distributions of the genres as well as their distinctive features. The distinctive features of the genre ‘responsory verse’ are not surprising because its musical structure is fundamentally different from that of antiphon or responsory. For example, a responsory verse follows a psalm formula [24] and does not necessarily close with the mode’s final. Additionally, all the modes except mode 6 have an internal cluster dominated by the genre ‘antiphon’ and another one dominated the genre ‘responsory’. These results indicate that the three genres tend to have pitch distributions with different characteristics in most modes, and at the same time, that the internal clusters obtained by unsupervised learning do not perfectly match the genres.

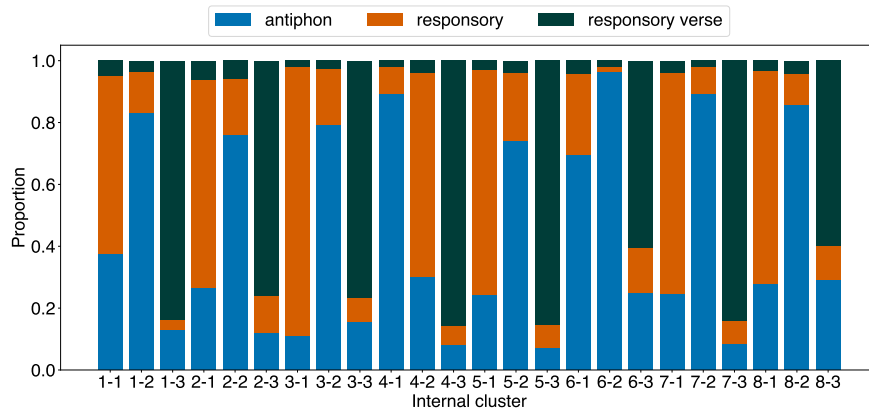


Fig. 4: Proportion of genres in each internal cluster. Internal cluster  $k$  of mode  $m$  is labeled as  $m-k$ .

### 3.3 Time evolution of mode frequency

The time evolution of the relative frequencies of modes and internal clusters is shown in Fig. 5. To obtain this result, we first calculated the posterior probability  $P(k|n)$  of internal clusters for each piece  $n$  using its mode annotation as a constraint, and used this probability and the time stamp  $t_n$  to calculate the relative frequency of internal clusters in each century.

Some interesting observations can be made from the result. First, we find that the changes in the mode frequencies are remarkably small throughout the analyzed time period. The largest relative changes can be observed between the 12th and 13th centuries and between the 13th and 14th centuries, but the relative changes of the mode frequencies are less than 100%. The observed stability of relative frequencies of clusters of musical styles is in stark contrast with the transitions of musical styles in Western classical music since the 16th century [1, 2] and popular music [7, 9].

Second, compared to the overall mode frequencies, the internal cluster frequencies have larger changes over time. For example, the frequency of cluster 3-3 increased by more than 100% from the 12th century to the 15th century, whereas the frequency of cluster 1-1 decreased by more than 50% in this period. With the result in Sec. 3.1, this means that there is some amount of internal changes in the average pitch distribution within each mode. We can also find some systematic tendencies across modes: the frequencies of the first internal clusters tend to decrease and those of the third internal clusters tend to increase. With the result in Sec. 3.2, we can relate these tendencies to the overall decrease of the proportion of genre ‘responsory’ and the overall increase of the proportion of genre ‘responsory verse’.

Finally, all internal clusters had a non-negligible frequency in the 11th century and emergences of new clusters were not observed. This means that there is no significant innovation in the pitch distribution in this data.

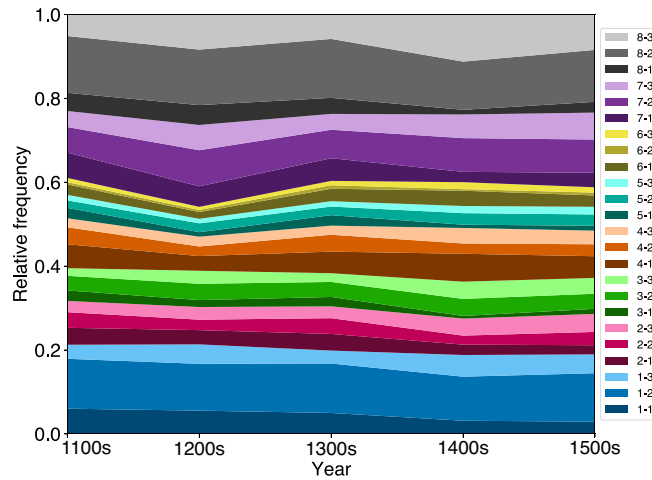


Fig. 5: Relative frequencies of modes and internal clusters over time.

#### 4 Discussion and conclusions

We here summarize and discuss our results. First, we found that the eight medieval modes exhibit internal clusters of pitch distributions, which correlate with the three major chant genres (antiphon, responsory, and responsory verse). Although we focused on the case of three internal clusters per mode, the experimental result suggested the existence of finer-grained internal clusters, which are also expectable from musicological considerations. For example, there are different types of responsories, such as *responsorium prolixum*, *responsorium breve*, and *responsorium graduale*, that were sung on different occasions or in distinct liturgical contexts and have different melodic features [25]. Incorporating deeper musicological insights into specific genre forms and their implications on mode is likely to shed further light on the substructure of the medieval modes.

Second, we found that the mode frequencies remained remarkably stable over five centuries. This can be explained by the fact that the responsories and antiphons were seen as divine texts that should by all means be authentically preserved. Incidentally, this was also a main driving force behind the development of music notation in the West and the emergence of other genres such as sequences and tropes that granted greater creative flexibility. It would be interesting to further study the Cantus database to reveal how such stability was attained when chants were transmitted across different geographical locations in relation to notational practice. It is also important to examine possible reasons for the observed small variations in mode frequencies, such as changing preferences to write chants in certain modes, changes in categorization practices, and artifacts/biases of sampling manuscripts in different time periods. Such a study will be facilitated when more manuscripts will be digitized and made available for computational analyses.

Third, we found considerable changes in the frequencies of internal clusters, which are related to changes in the proportion of the chant genres in the data. Additionally, the analysis conducted at the resolution of three internal clusters per mode revealed a lack of substantial innovation in the pitch distribution. As a caveat, we note that the Cantus database has a limited number of manuscript sources, and further study should also inspect possible biases by employing strategies such as downsampling and generating synthetic pieces to ensure a more balanced analysis.

Moreover, this study demonstrates how researchers can employ large datasets and computational modeling for investigating music-theoretical concepts and their cultural evolution. Finally, our work points to the necessity of increasing collaboration and exchange between researchers from the humanities and computer science. This pertains not only to the interpretation of quantitative results *post factum*. Rather, it is important to engage in interdisciplinary dialog early on in the research process, in particular when constructing and evaluating computational models. We believe that the vibrant field of cultural evolution provides an ideal forum for such exchanges to take place.

## References

1. Zivic, P.H.R., Shifres, F., Cecchi, G.A.: Perceptual Basis of Evolving Western Musical Styles. *Proceedings of the National Academy of Sciences of the USA* 110(24), 10034–10038 (2013)
2. Weiß, C., Mauch, M., Dixon, S., Müller, M.: Investigating Style Evolution of Western Classical Music: A Computational Approach. *Musicae Scientiae* 23(4), 486–507 (2019)
3. Nakamura, E., Kaneko, K.: Statistical Evolutionary Laws in Music Styles. *Scientific Reports* 9(1), 15993 (2019)
4. Anzuoni, E., Ayhan, S., Dutto, F., McLeod, A., Moss, F.C., Rohrmeier, M.: A Historical Analysis of Harmonic Progressions Using Chord Embeddings. In: *Proceedings of the 18th Sound and Music Computing Conference*, pp. 284–291. Axa sas/SMC Network (2021)
5. Harasim, D., Moss, F.C., Ramirez, M., Rohrmeier, M.: Exploring the Foundations of Tonality: Statistical Cognitive Modeling of Modes in the History of Western Classical Music. *Humanities and Social Sciences Communications* 8(5), 1–11 (2021)
6. Yust, J.: Stylistic Information in Pitch-Class Distributions. *Journal of New Music Research* 48(3), 217–231 (2019)
7. Mauch, M., MacCallum, R.M., Levy, M., Leroi, A.M.: The Evolution of Popular Music: USA 1960–2010. *Royal Society Open Science* 2(150081), 1–10 (2015)
8. Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., Komarova, N.L.: Musical Trends and Predictability of Success in Contemporary Songs in and out of the Top Charts. *Royal Society Open Science* 5(5), 171274 (2018)
9. Singh, R., Nakamura, E.: Dynamic Cluster Structure and Predictive Modelling of Music Creation Style Distributions. *Royal Society Open Science* 9(220516), 1–18 (2022)
10. Moss, F.C., Souza, W.F., Rohrmeier, M.: Harmony and Form in Brazilian Choro: A Corpus-Driven Approach to Musical Style Analysis. *Journal of New Music Research* 49(5), 416–437 (2020)
11. Savage, P.E., Passmore, S., Chiba, G., Currie, T.E., Suzuki, H., Atkinson, Q.D.: Sequence Alignment of Folk Song Melodies Reveals Cross-Cultural Regularities of Musical Evolution. *Current Biology* 32(6), 1395–1402 (2022)
12. McBride, J., Passmore, S., Tlustý, T.: Convergent Evolution in a Large Cross-Cultural Database of Musical Scales. *PsyArXiv* (2021). <https://doi.org/10.31234/osf.io/eh5b3>

13. Aldwell, E., Schachter, C., Cadwallader, A.: *Harmony and Voice Leading*. Cengage Learning, 4th ed. (2010)
14. Haug, A.: *Singing from the Book: The End of Sacrifice and the Rise of Chant in the Fourth Century*. *The Musical Quarterly* 104(3–4), 370–391 (2021)
15. Huglo, M.: *Tonary*. Grove Music Online (2001)
16. Atkinson, C.M.: *The Critical Nexus: Tone-System, Mode, and Notation in Early Medieval Music*. Oxford University Press (2008)
17. Huron, D., Veltman, J.: *A Cognitive Approach to Medieval Mode: Evidence for an Historical Antecedent to the Major/Minor System*. *Empirical Musicology Review* 1(1), 33–55 (2006)
18. Wiering, F.: *Comment on Huron and Veltman: Does a Cognitive Approach to Medieval Mode Make Sense?*. *Empirical Musicology Review* 1(1), 56–60 (2006)
19. Cornelissen, B., Zuidema, W., Burgoyne, J.A.: *Mode Classification and Natural Units in Plainchant*. In: *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pp. 869–875. Association for Computing Machinery, New York (2020)
20. Lacoste, D., Bailey, T., Steiner, R.: *Cantus: A Database for Latin Ecclesiastical Chant – Inventories of Chant Sources*. (2011)
21. Lacoste, D.: *The Cantus Database: Mining for Medieval Chant Traditions*. *Digital Medievalist* 7 (2012)
22. Dobszay, L.: *Art. Offizium*. MGG Online
23. Helsen, K., Lacoste, D.: *A Report on the Encoding of Melodic Incipits in the CANTUS Database with the Music Font ‘Volpiano’*. *Plainsong & Medieval Music* 20(1), 51–65 (2011)
24. Hucke, H., Hiley, D.: *Art. Responsorium, Die Melodien der Responsorialia*. MGG Online (2016)
25. Hiley, D.: *Western Plainchant: A Handbook*. Oxford University Press (1995)

# Computational Analysis of Selection and Mutation Probabilities in the Evolution of Chord Progressions

Eita Nakamura<sup>1\*</sup>

Kyoto University

eita.nakamura@i.kyoto-u.ac.jp

**Abstract.** We build a model of cultural evolution and study the properties of the process in which new chord progressions are repeatedly generated by referencing and modifying past chord progressions. As an extension of the models for biological molecular evolution, this model represents a stochastic process in which references are selected from an accumulating pool of chord segments and new chord segments are created by mutation including insertion, deletion, and substitution of chord symbols. We used a dataset of Japanese popular music and analyzed this evolutionary process by inferring the model parameters. A number of suggestive results regarding the evolution of the creative culture were obtained, including a strong recency bias, large mutation rates and large dynamic changes in mutation probabilities, and correlations between fluctuations and mutation probabilities and between the diffusedness of mutant chord segments and their mutation probabilities. Model-based predictions of new chord progressions were also made.

**Keywords:** cultural evolution; evolutionary model; symbolic music processing; chord progression; prediction of evolution; accumulating artifact pool

## 1 Introduction

Cultural development is a key aspect of human's intelligence, and musical culture provides a fruitful venue for studying its creative role. Quantitative studies on music evolution have revealed some interesting macroscopic phenomena. These include directional changes (trends) in average features continuing for decades [1–3] or centuries [4–6], punctuational short time periods with rapid changes [1, 4, 7], concurrent and transient cluster structure [1, 3, 4], and frequency-dependent selection bias [5, 8]. Since individual musical pieces are produced by creators who learn to create music from previous

---

\* The author thanks Hitomi Kaneko and Daichi Kamakura for useful discussions. This work was in part supported by JSPS KAKENHI Grant Numbers 21K12187, 21K02846, and 22H03661, and JST FOREST Program Grant Number JPMJPR226X.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

creators or musical pieces, revealing the microscopic processes of knowledge transmission and modification is essential for understanding the mechanisms underlying these phenomena [9].

Transmission processes of musical knowledge can be classified into two types, direct and indirect. In direct transmission, a song or other musical data serves as a reference and is replicated for producing a new one. For example, folk songs are typically transmitted in this way [10]. Some models of direct transmission of musical scale [11] or music sampling [8] have been proposed for testing musicological hypotheses. Direct transmission of music is also studied in laboratory experiments [12, 13]. In indirect transmission, on the other hand, knowledge for music creation is learned from a collection of past music or through teaching, and the acquired knowledge is used for creating (rather than replicating) new musical pieces. A dominant part of art music and popular music is considered to be created by indirectly transmitted knowledge, and there is some evidence from studies on automatic music composition showing the relevance of statistical learning [14]. Cultural evolution models incorporating indirect knowledge transmission have been studied to explain empirical laws found in music data [5, 15].

Here we focus on the evolution of chord progressions in popular music. Chord progressions outline how accompaniments are played and are of prime importance in the composition process of tonal music. In popular music, they are usually notated together with the melodies, forming a type of musical score called lead sheet. Unlike melodies, chord progressions are very often reused with possible modifications, and there are books [16] and websites [17] listing commonly used chord progressions, suggesting that direct transmission is at work. It is commonly known and has been shown by a corpus analysis [18] that patterns of chord progressions have changed significantly over the last decades while new chord progressions have continuously been invented. Therefore, chord progressions are scientifically and practically interesting objects to study how a creative culture evolves by knowledge transmission and modification.

To reveal the basic characteristics of the transmission and modification processes of chord progressions, we construct a stochastic model of evolution and analyze a dataset of chord progressions in popular music songs. We view a chord progression as a sequence of chord symbols and consider chord segments ( $L$ -grams) as the unit of knowledge transmission. The process of creating a new chord segment by (i) choosing a reference from the ‘artifact pool’ of previously created chord segments (i.e. selection) and (ii) possibly modifying it (i.e. mutation) is akin to that of biological molecular evolution, where nucleotides or amino acids correspond to chord symbols.

We thus build a model similar to the models of molecular evolution [19, 20], with extensions to incorporate essential factors of the cultural evolution. First, we formulate a model where created artifacts (chord segments) are accumulated in the artifact pool unlike individuals that are removed upon death from the population in biological models. Second, we incorporate in the selection process the recency and frequency-dependent biases, which are often relevant in cultural evolution [21, 22]. Third, we consider in the mutation process insertions and deletions of chord symbols, which are often ignored in molecular evolution models [20], as well as substitutions. These features also make our model different from the one previously used for chord progression data [23], enabling us to harvest a number of suggestive results. Another study analyzed folk songs

and estimated note-wise mutation probabilities using a dataset without time information [10]. With the use of mathematical model and data with proper time information, we here analyze more detailed properties of the evolutionary process such as dynamic changes of mutation probabilities, correlations between evolutionary parameters, and the characteristics of new chord segments that later become commonly used.

## 2 Method

### 2.1 Data representation

We consider a dataset of chord progressions represented in a standard popular music notation, transposed to the natural key (C major or A minor), and labelled with a year of creation. The set of distinct chords in the data is denoted by  $\Omega$  (e.g.  $\Omega = \{C, Am7, FM9, \dots\}$ ). From each progression, we extract  $L$ -grams (also called chord segments), which are segments of  $L$  consecutive chords, where we remove repetitions of chords. An  $L$ -gram so obtained is assigned a time stamp, which is the same as the year of creation of its source progression. The collection of all  $L$ -grams obtained from progressions created in year  $t$  is denoted by  $S_t^{(L)} = \{w_i | t_i = t\}$  and its index set by  $I_t^{(L)} = \{i | t_i = t\}$ , where  $i$  is used as an index for  $L$ -grams and  $w_i = (w_{i\ell})_{\ell=1}^L$  ( $w_{i\ell} \in \Omega$ ) denotes the corresponding  $L$ -gram. We also define  $S_{<t}^{(L)} = \bigcup_{s=1}^{t-1} S_s^{(L)}$ , where we take the starting time  $t = 1$  as the earliest year of creation in the data. For simplicity of notation, we define  $S_t = S_t^{(L)}$ ,  $S_{<t} = S_{<t}^{(L)}$ ,  $S_t^+ = S_t^{(L+1)}$ ,  $S_t^- = S_t^{(L-1)}$ , etc. For the result in Sec. 3, we consider the case  $L = 4$ .

### 2.2 Evolutionary model

We consider that each  $L$ -gram  $w \in S_t$  is stochastically generated by selecting a reference segment  $w'$  from past data and possibly mutating it. Three mutation modes are considered: substitution, deletion, and insertion. In the substitution mode, a reference  $w'$  is taken from the set  $S_{<t}$  of  $L$ -grams and mutated by changing one or more component chords, from  $w'_\ell$  to  $w_\ell$ , according to the symbol-wise substitution probability  $\pi_{\text{sub}}(w_\ell | w'_\ell)$ . The substitution probability from  $L$ -gram  $w'$  to  $w$  is defined as

$$P_{\text{rep/sub}}(w|w') = \prod_{\ell=1}^L \pi_{\text{sub}}(w_\ell | w'_\ell), \quad (1)$$

where we also include the pure replication case ( $w_\ell = w'_\ell$  for all  $\ell$ ) in this probability. In the deletion mode, a reference  $w'$  is taken from the set  $S_{<t}^+$  of  $(L+1)$ -grams and mutated by removing one of its components. Since a removal of the first or last chord in  $w' \in S_{<t}^+$  produces an  $L$ -gram in  $S_{<t}$ , we exclude such a case. Then, the deletion probability can be defined as

$$P_{\text{del}}(w|w') = \frac{1}{L-1} \sum_{\ell=2}^L \delta(w, w'_{1:(\ell-1)} w'_{(\ell+1):(L+1)}), \quad (2)$$



where  $w_{\ell:\ell'} = w_\ell w_{\ell+1} \cdots w_{\ell'}$ , and  $\delta(w_1, w_2) = 1$  if  $w_1 = w_2$  and 0 otherwise. In the insertion mode, a reference  $w'$  is taken from the set  $S_{<t}^-$  of  $(L-1)$ -grams and mutated by inserting a chord  $a$  after one of its chords  $w'_\ell$  according to the symbol-wise insertion probability  $\pi_{\text{ins}}(a|w'_\ell)$ . The insertion probability is defined as

$$P_{\text{ins}}(w|w') = \frac{1}{L-1} \sum_{\ell=1}^{L-1} \pi_{\text{ins}}(w_{\ell+1}|w'_\ell) \delta(w, w'_{1:\ell} w_{\ell+1} w'_{(\ell+1):(L-1)}). \quad (3)$$

Note that the mutation probabilities considered here are among the simplest choices and we can generalize them to more elaborated models. For example, while we assumed that the symbol-wise substitution probabilities are context free, that is, the probability is independent of the preceding or succeeding chord symbols, it is possible to include context dependence by extending the probability  $\pi_{\text{sub}}(w_\ell|w'_\ell)$  to such forms as  $\pi_{\text{sub}}(w_\ell|w'_{\ell-1}, w'_\ell)$  and  $\pi_{\text{sub}}(w_\ell|w'_{\ell-1}, w'_\ell, w'_{\ell+1})$ . Similarly, we can extend the insertion probability so that it also depends on the succeeding chord symbols. These refinements generally increase the complexity (the number of parameters) of the model and require a larger amount of data to reliably infer the parameters.

In the generative process, one of the mutation modes is first chosen according to the mutation mode probability  $P(b) = \lambda_b$  where  $b \in \{\text{rep/sub}, \text{del}, \text{ins}\}$ . We again note that the pure replication case is included in the mode  $b = \text{rep/sub}$ . Next, in this case, a reference segment  $w$  is chosen from  $S_{<t}$  according to the selection probability  $P_{\text{sel}}(w|S_{<t})$ . We incorporate two biases in the selection probability to represent potential tendencies of creators. The first is the recency bias [21], which represents the creators' tendency to more likely choose a reference that appears in a recently created song. This bias can be represented by a weighting factor  $e^{-(t-t_i)/\tau}$  for a segment  $i$ , where the time constant  $\tau$  represents the time scale for the bias. The second is the frequency-dependent bias [22], which represents the creators' tendency to more likely choose a reference that is more (or less) frequently used in  $S_{<t}$ . To formulate this bias, let  $F(w; I_s) = \#\{j \in I_s | w_j = w\} / \#I_s$  denote the relative frequency of  $w$  in  $S_s$ . The frequency bias can be incorporated in a factor  $[F(w; I_s)]^\alpha$  in the selection probability, where  $\alpha > 1$  ( $\alpha < 1$ ) represents a positive (negative) frequency-dependency bias.

The selection probability incorporating the two biases is then given as

$$P_{\text{sel}}(w; S_{<t}) \propto \sum_{s=1}^{t-1} e^{-(t-s)/\tau} [F(w; I_s)]^\alpha. \quad (4)$$

We note that this formulation removes a potential bias arising from the unbalanced numbers of chord segments created in individual years. Similarly, we define  $P_{\text{sel}}(w; S_{<t}^+)$  and  $P_{\text{sel}}(w; S_{<t}^-)$  for choosing a reference in the deletion and insertion modes, respectively, where the same  $\alpha$  and  $\tau$  are used.

We can summarize the generative probability of  $w \in S_t$  as follows:

$$P(w; S_t) = \sum_{b \in \{\text{rep/sub}, \text{del}, \text{ins}\}} \lambda_b P_b(w; S_t), \quad (5)$$

$$P_{\text{rep/sub}}(w; S_t) = \sum_{w' \in S_{<t}} P_{\text{rep/sub}}(w|w') P_{\text{sel}}(w'; S_{<t}), \quad (6)$$

$$P_{\text{del}}(w; S_t) = \sum_{w' \in S_{<t}^+} P_{\text{del}}(w|w')P_{\text{sel}}(w'; S_{<t}^+), \quad (7)$$

$$P_{\text{ins}}(w; S_t) = \sum_{w' \in S_{<t}^-} P_{\text{ins}}(w|w')P_{\text{sel}}(w'; S_{<t}^-). \quad (8)$$

We can also separately define the (pure) replication and substitution probabilities as

$$P_{\text{rep}}(w; S_t) = P_{\text{rep/sub}}(w|w)P_{\text{sel}}(w; S_{<t}), \quad (9)$$

$$\begin{aligned} P_{\text{sub}}(w; S_t) &= \sum_{w' \in S_{<t}, w' \neq w} P_{\text{rep/sub}}(w|w')P_{\text{sel}}(w'; S_{<t}) \\ &= P_{\text{rep/sub}}(w; S_t) - P_{\text{rep}}(w; S_t). \end{aligned} \quad (10)$$

### 2.3 Inference method

The parameters of the evolutionary model,  $\lambda_b$ ,  $\pi_{\text{sub}}(a|a')$ ,  $\pi_{\text{ins}}(a|a')$ ,  $\tau$ , and  $\alpha$ , can be estimated from the data by the maximum likelihood method. To estimate the first three sets of parameters, we apply the expectation-maximization (EM) algorithm by treating the mutation mode  $b$  and reference  $w'$  as latent variables for each observed  $w$ .

To estimate  $\tau$  and  $\alpha$ , we can apply a simple iterative grid search using the likelihood as the objective function. Since the optimal values of these parameters depend on the values of the other parameters and vice versa, we iterate the EM step and grid search step alternately until a convergence of the likelihood value. To evaluate their estimation variances, we can apply a Bayesian method based on Monte Carlo Markov Chain sampling. Specifically, we used the Metropolis method with a flat prior distribution and a log-normal distribution as the proposal distribution.

### 2.4 Posterior analysis

Given the set of model parameters estimated as in Sec. 2.3, we can apply the method of posterior analysis for analyzing possible dynamic changes in the evolutionary parameters. First, given an  $L$ -gram  $w$  at time  $t$ , its posterior probability of mutation modes  $\tilde{\lambda}_b(w, t) = P(b|w, t)$  ( $b \in \{\text{rep}, \text{sub}, \text{del}, \text{ins}\}$ ) can be obtained as

$$\tilde{\lambda}_{\text{rep}}(w, t) = \lambda_{\text{rep/sub}}P_{\text{rep/sub}}(w|w)P_{\text{sel}}(w; S_{<t})/P(w; S_t), \quad (11)$$

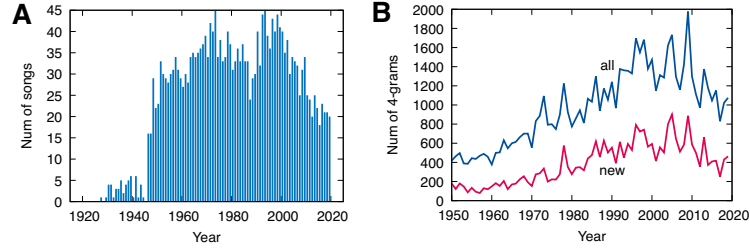
$$\tilde{\lambda}_{\text{del}}(w, t) = \lambda_{\text{del}}P_{\text{del}}(w; S_t)/P(w; S_t), \quad (12)$$

$$\tilde{\lambda}_{\text{ins}}(w, t) = \lambda_{\text{ins}}P_{\text{ins}}(w; S_t)/P(w; S_t), \quad (13)$$

$$\tilde{\lambda}_{\text{sub}}(w, t) = 1 - \tilde{\lambda}_{\text{rep}}(w, t) - \tilde{\lambda}_{\text{del}}(w, t) - \tilde{\lambda}_{\text{ins}}(w, t), \quad (14)$$

where the right-hand sides of these equations can be calculated using Eqs. (5) to (8) and the replication and substitution probabilities are separately defined here. Then, the mutation mode probabilities  $\tilde{\lambda}_b(t)$  at time  $t$  can be estimated, for example, as

$$\tilde{\lambda}_{\text{sub}}(t) = \frac{1}{\#S_t} \sum_{w \in S_t} \tilde{\lambda}_{\text{sub}}(w, t).$$



**Fig. 1.** A: The distribution of composition years in the dataset used. B: The yearly numbers of 4-grams (including duplications) and that of newly appeared 4-grams.

We can also calculate the posterior probability  $P(w'|w, t, b)$  of reference segments  $w' \in S_{<t} \cup S_{<t}^+ \cup S_{<t}^-$  in a similar way. For example, in the insertion mode,

$$P(w'|w, t, b = \text{ins}) \propto P_{\text{ins}}(w|w')P_{\text{sel}}(w'; S_{<t}^-).$$

The posterior probabilities obtained in this way can be used to estimate the mutation probabilities  $\tilde{\pi}_{\text{sub}}(a|a'; t)$  and  $\tilde{\pi}_{\text{ins}}(a|a'; t)$  at time  $t$ .

### 3 Result

#### 3.1 Dataset

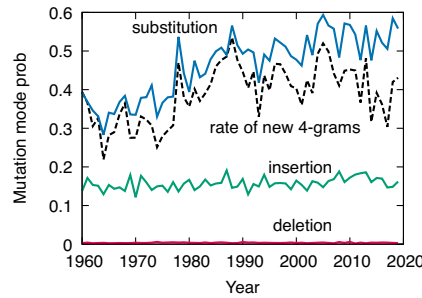
We used a dataset of Japanese popular music songs. The dataset was constructed by the author and comprised of 2419 songs. The songs were taken from top ranked songs in the Oricon yearly charts and from a compiled collection of historical popular songs [24]. The composition years spanned the range [1927, 2019] and we applied the evolutionary model for analysis in a range of years  $t \geq 1960$  (Fig. 1A).

Before extracting  $L$ -grams of chords from a song, we transposed the song into the natural key, converted a consecutive repetition of the same chord into a single chord, and converted a slash chord into a normal chord by removing the bass note. The target of the analysis was 4-grams ( $L = 4$ ). The number of distinct chord symbols was 232, from which approximately  $2.9 \times 10^9$  distinct 4-grams can be created in principle. The numbers of distinct 3-grams, 4-grams, and 5-grams appearing in the dataset were 12 258, 27 237, and 44 204, respectively. Fig. 1B shows the yearly numbers of 4-grams and those of newly appeared 4-grams; the average rate of new 4-grams was 39%.

#### 3.2 Selection biases

The inferred value of the time constant was  $\tau = 2.61 \pm 0.53$ . This means that the probability of a chord segment being chosen as a reference reduces by a factor of 10 in every 6.1 years, when other conditions are equal. The inferred value of the frequency-dependence parameter was  $\alpha = 1.16 \pm 0.16$ . The mean value indicates a slightly positive frequency-dependent bias, i.e., more common chord segments tend to be more frequently chosen as a reference than its frequency expected for random selection. However, the deviation of  $\alpha$  from unity is small and the result is consistent with the frequency-independent bias within the range of statistical error.

$\lambda_{\text{rep}}$	$\lambda_{\text{sub}}$	$\lambda_{\text{del}}$	$\lambda_{\text{ins}}$
0.38	0.46	0.00	0.16

**Table 1.** Inferred values of mutation mode probabilities.**Fig. 2.** Dynamic changes of mutation mode probabilities.

### 3.3 Mutation modes

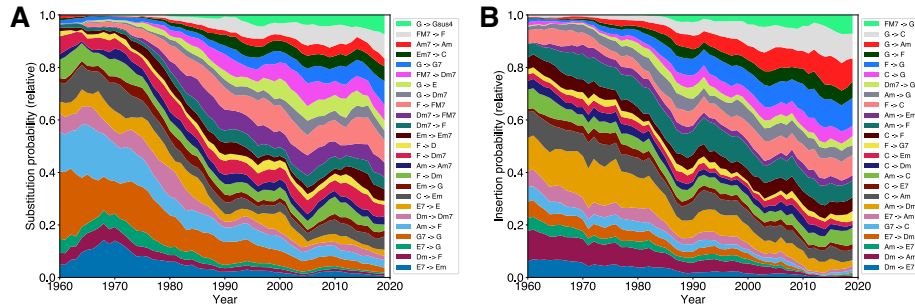
The inferred values of mutation mode probabilities are listed in Table 1. We found that the probability of choosing the deletion mode tended to converge to zero. To understand this result, we note that the probability of each mutation mode is of the same order,  $O(K^{-L})$  for  $K = \#\Omega$ , if we suppose a uniform distribution over all chord symbols. The result can then be explained by the fact that there are no tunable parameters for the deletion operation whereas the substitution and insertion probabilities,  $\pi_{\text{sub}}(a|a')$  and  $\pi_{\text{ins}}(a|a')$ , are trained so that the likelihoods of these modes will increase in the course of statistical inference. Consequently, in our model, the main mutation modes are substitution and insertion.

We see that the sum of mutation probabilities, which is equal to  $1 - \lambda_{\text{rep}}$ , is 62% and larger than the average rate (39%) of new segments. This value is substantially larger than the mutation probabilities in typical biological evolution and leads to distinct characteristics. For example, a significant proportion (38%) of reappeared segments are estimated to be created through a mutation process according to the present model.

A posterior analysis over time showed that the substitution mode probability had some variations across years and a general trend of increase from the 1960s to the 1990s (Fig. 2). Its temporal changes highly correlated with the rate of new segments ( $\rho = 0.86$ ,  $p < 10^{-10}$ ). On the other hand, the insertion mode probability had small fluctuation and no notable trend was observed.

### 3.4 Symbol-wise substitution and insertion probabilities

The most frequent modes of symbol-wise substitution and insertion are shown in Fig. 3 with their yearly relative frequencies obtained by the posterior analysis. For substitution probabilities (Fig. 3A), we see that there are significant changes over years and the diversity of applied substitution modes considerably increased from the early period to the late period. There is also a tendency that substitutions involving less frequent



**Fig. 3.** Dynamic changes of symbol-wise substitution (A) and insertion (B) probabilities. In each panel, the relative frequencies of the 25 most frequent modes are shown, and a smoothing with a window of 5-year width is applied.

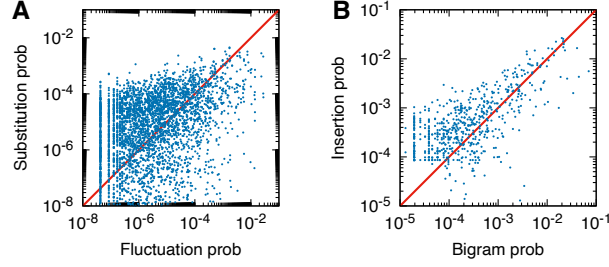
chords such as FM7 and Gsus4 became more frequent in later periods. From the list of substitution modes, we can find that most substitutions occur between chords with the same harmonic function. These chords often share the root tone (e.g. E7 → Em, Dm → Dm7, F → FM7, and G → Gsus4) or share constituent pitches (e.g. Dm → F, C → Em, and Em7 → C). A similar relation between the substitutability of chords and the harmonic function was also found in an analysis using hidden Markov model [25].

Significant changes over years were also found in the insertion probabilities (Fig. 3B). The list of insertion modes mostly consisted of common chord transitions (see also Sec. 3.5). We see a tendency that chord transitions used in the minor key (e.g. Dm → E7, E7 → Am, and Am → Dm) appear more frequently in the early period and those used in the major key (e.g. F → G, G → C, and FM7 → G) in the late period.

### 3.5 Correlation between fluctuation and mutation probabilities

We analyzed the correlations between evolutionary parameters to examine several expectations from the evolutionary theory. On the one hand, if new chord segments are stochastically generated by substitutions, we expect that the joint probability of different chord symbols  $P_{\text{var}}(a', a)$  in variants of segments related by a single substitution correlates with the joint probability of substitution  $\pi_{\text{sub}}(a' \rightarrow a) = P(a')\pi_{\text{sub}}(a|a')$ , where  $P(a')$  is the prior probability of chord symbols. On the other hand, if we consider the implicit effect of social selection in the data, among potential creators who generate new segments with different substitution probabilities, successful creators would be those with substitution probabilities that are similar to the fluctuation probabilities of chord symbols in past data. This also suggests that the probability  $\pi_{\text{sub}}(a' \rightarrow a)$  correlates with the fluctuation represented by the probability  $P_{\text{var}}(a', a)$  in past data.

To examine this expectation, we analyzed the correlation between the joint substitution probability  $\pi_{\text{sub}}(a' \rightarrow a)$  in a time range [2010 : 2019] and the fluctuation  $P_{\text{var}}(a', a)$  observed in a time range [1927 : 2009]. The result in Fig. 4A supports the expectation and shows a positive correlation ( $\rho = 0.18, p < 10^{-10}$ ). We also see a significant amount of deviation: in particular, a high fluctuation probability does not always indicate a high substitution probability.



**Fig. 4.** Correlations between fluctuation and mutation probabilities. A: The fluctuation probability  $P_{\text{var}}(a', a)$  and the joint substitution probability  $\pi_{\text{sub}}(a' \rightarrow a)$ . B: The bigram probability  $P_{\text{bi}}(a', a)$  and the joint insertion probability  $\pi_{\text{ins}}(a' \rightarrow a)$ .

Similarly, we expect that the insertion probabilities are related to the corresponding fluctuation probabilities. More specifically, we expect that the joint probability of insertion  $\pi_{\text{ins}}(a' \rightarrow a) = P(a')\pi_{\text{ins}}(a|a')$  correlates with the bigram probability  $P_{\text{bi}}(a', a)$  of chord symbols in past data. The result in Fig. 4B supports the expectation and shows a high correlation ( $\rho = 0.64, p < 10^{-10}$ ).

### 3.6 Diffusion of mutants

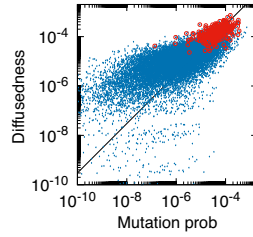
Characterizing the conditions of new mutant chord segments that will diffuse and become commonly used is important for understanding the macroscopic evolution of chord progressions. For biological evolution, where mutations are rare, a similar problem of fixation has been studied, and the fitness of the mutant and the random sampling in a finite population are studied as two major factors [26]. Our case of cultural evolution has two features that lead to an evolutionary process distinct from the typical case of biological evolution. First, as we discussed in Sec. 3.3, the mutation rate is much larger than in biological evolution so that the chance that mutants of the same form are independently generated is not negligible. Second, the evolutionary process is an accumulated process so that mutant segments will not be removed from the artifact pool.

Based on this consideration, we expect that the accumulation of independent mutants is relevant for the diffusion of a new chord segment. As a measure of diffusion of a segment  $w$ , we can use its probability of replication. More specifically, we define the diffusedness  $D^{10\text{yr}}(w, t_w^*)$  of a mutant segment  $w$  first appeared in year  $t_w^*$  as

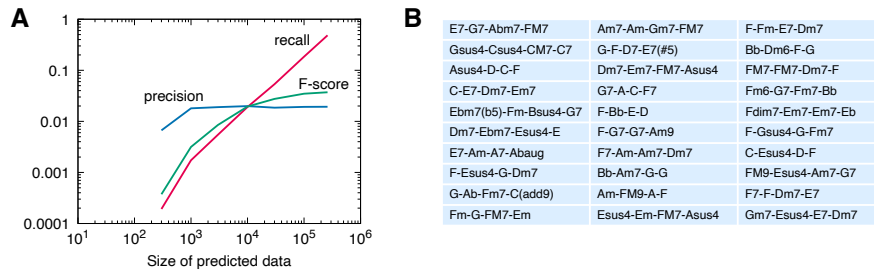
$$D^{10\text{yr}}(w, t_w^*) = \frac{1}{10} \sum_{t=t_w^*+1}^{t=t_w^*+10} \lambda_{\text{rep/sub}} P_{\text{rep}}(w, t), \quad (15)$$

where the replication probability  $P_{\text{rep}}(w, t)$  is given in Eq. (9). Our hypothesis is that a mutant with a larger mutation probability  $P(w; S_{t_w^*})$  has a higher chance of repeatedly introduced to the artifact pool and consequently has a higher diffusedness on average.

The relationship between the diffusedness and mutation probability analyzed for all mutant segments that first appeared in years between 1960 and 2009 is shown in Fig. 5. The observed high correlation ( $\rho = 0.57, p < 10^{-10}$ ) supports our hypothesis, and



**Fig. 5.** Diffusedness  $D^{10\text{yr}}(w, t_w^*)$  and mutation probability  $P(w; S_{t_w^*})$  for mutant segments  $w$  with the year of first appearance  $t_w^* \in [1960 : 2009]$ . Segments that appeared four or more times within ten years after the first appearance are indicated by red circles. A linear function  $y \propto x$  (black line) is shown as a guide for the eyes.



**Fig. 6.** Predictive ability of the evolutionary model. A: Prediction accuracies for varying sizes of predicted data. B: Most probable predictions by the evolutionary model.

the linear relation is particularly clear for mutant segments with highest diffusedness (marked in red circles in Fig. 5). The deviation from the linear relation in the small mutation probability regime can be explained by the finite size effect. We can also see that there is a variation of  $O(10^1-10^2)$  in the diffusedness for those samples with a mutation probability  $\sim 10^{-6}$ , which indicates that the mutation probability cannot be the only factor that determines the diffusedness.

### 3.7 Predictions

The present evolutionary model can be used for predicting new segments to appear in the future. To examine its potential, we trained the model with a subset of the data of segments created in 1999 or before and evaluated its predictive ability using as test data the remaining data of segments created in years 2000–2019. We randomly generated  $10^6$  4-grams by the model and obtained approximately  $2.6 \times 10^5$  segments after removing duplications and the samples already appearing in the training data. The generated 4-grams were sorted by the mutation probability in the decreasing order. We compared the predicted data with the test data by measuring the precision, recall, and F-score.

The recall achieved 48% on the whole predicted data (Fig. 6A). The precision was approximately 2%, which is much larger than the expectation value of 0.0004% by random sampling. Examples of predicted segments with highest mutation probabilities that were not included in the analyzed dataset are shown in Fig. 6B.

## 4 Conclusion and discussion

In this paper, we have studied the evolution of chord progressions based on a stochastic model of cultural evolution incorporating the selection and mutation processes. We summarize the main findings and discuss implications. First, the inferred selection biases showed a strong recency effect with a time constant of 2.61 yrs. This indicates that while chord segments are accumulated in the artifact pool, they will be effectively removed from the pool after some decades in the sense that their chance of being chosen as a reference will decrease significantly in that time interval. On the other hand, no significant sign of frequency dependence was observed.

Second, the analysis revealed large mutation rates and large dynamic changes in the substitution and insertion probabilities. The first feature reminds us of an interesting phenomenon known as the survival-of-the-flattest effect [27], which suggests the possibility that a chord segment with a high probability of replication can be outcompeted by segments that have lower probabilities of selection but are robust in usability against mutations. The second feature also suggests a selective advantage of chord segments that are robust in usability against mutations toward multiple directions. While the significance of this effect depends on the mutation rate and other model configurations, this observation may provide a new perspective on understanding why certain chord segments are more popular than others.

Third, the correlations found between fluctuation and mutation probabilities and between diffusedness and mutation probabilities support expectations from the evolutionary theory and may be useful for predicting the features of the evolutionary process. It is also important to seek for possible explanations for the observed deviations of  $O(10^1-10^2)$  in the mutation probabilities for similar values of fluctuation probabilities.

We remark that although the present evolutionary model was built upon empirical knowledge on the process of music creation, the results of this study do not verify that the assumed process is correct. We can think of other processes of creating chord progressions, for example, a process involving data generation through statistical learning. To formulate a more realistic model, we should incorporate the multilevel structure of music, consisting of chord segments, musical piece, composer, etc.; reference and selection can occur at each of these levels. Japanese popular music is not a closed system and some chord progressions should have been imported from Western musical cultures; such migrations were treated as mutations in this study. Future work should experimentally test the model with other possibilities and address the aforementioned theoretical issues. The present evolutionary model can also be applied to analyzing the origins and relationships of chord segments in a similar way as the stochastic models of molecular evolution are applied to phylogenetic analysis.

## References

1. Mauch, M., et al.: The evolution of popular music: USA 1960–2010. *Royal Society Open Science* 2(150081), 1–10 (2015)
2. Interiano, M., et al.: Musical trends and predictability of success in contemporary songs in and out of the top charts. *Royal Society Open Science* 5(171274), 1–16 (2018)



3. Singh, R., Nakamura, E.: Dynamic cluster structure and predictive modelling of music creation style distributions. *Royal Society Open Science* 9(220516), 1–18 (2022)
4. Weiß, C., et al.: Investigating style evolution of Western classical music: A computational approach. *Musicae Scientiae* 23(4), 486–507 (2019)
5. Nakamura, E., Kaneko, K.: Statistical evolutionary laws in music styles. *Scientific Reports* 9(15993), 1–11 (2019)
6. Moss, F.C., Neuwirth, M., Rohrmeier, M.: The line of fifths and the co-evolution of tonal pitch-classes. *Journal of Mathematics and Music* (2022)
7. Zivic, P.H.R., Shifres, F., Cecchi, G.A.: Perceptual basis of evolving Western musical styles. *Proceedings of the National Academy of Sciences of the USA* 110(24), 10034–10038 (2013)
8. Youngblood, M.: Conformity bias in the cultural transmission of music sampling traditions. *Royal Society Open Science* 6(191149), 1–8 (2019)
9. Cavalli-Sforza, L.L., Feldman, M.W.: *Cultural Transmission and Evolution*. Princeton University Press, Princeton (1981)
10. Savage, P.E., et al.: Sequence alignment of folk song melodies reveals cross-cultural regularities of musical evolution. *Current Biology* 32(6), 1395–1402.e8 (2022)
11. McBride, J.M., Tlusty, T.: Cross-cultural data shows musical scales evolved to maximise imperfect fifths. Preprint arXiv:1906.06171 (2019)
12. Ravignani, A., Delgado, T., Kirby, S.: Musical evolution in the lab exhibits rhythmic universals. *Nature Human Behaviour* 1(0007), 1–7 (2016)
13. Anglada-Tort, M., et al.: Large-scale iterated singing experiments reveal oral transmission mechanisms underlying music evolution. *Current Biology* 33(8), 1472–1486.e12 (2023)
14. Fernández, J.D., Vico, F.: AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research* 48, 513–582 (2013)
15. Nakamura, E.: Conjugate distribution laws in cultural evolution via statistical learning. *Physical Review E* 104(034309), 1–13 (2021)
16. Scott, R.: *Chord Progressions for Songwriters*. iUniverse, Bloomington (2003)
17. Hooktheory: Popular Chord Progressions, <https://www.hooktheory.com/theorytab/common-chord-progressions>
18. de Clercq, T., Temperley, D.: A corpus analysis of rock harmony. *Popular Music* 30(1), 47–70 (2011)
19. Felsenstein, J.: Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution* 17, 368–376 (1981)
20. Arenas, M.: Trends in substitution models of molecular evolution. *Frontier in Genetics* 6(319), 1–9 (2015)
21. Wang, D., Song, C., Barabási, A.L.: Quantifying long-term scientific impact. *Science* 342(6154), 127–132 (2013)
22. Boyd, R., Richerson, P.J.: *Culture and the Evolutionary Process*. The University of Chicago Press, Chicago (1985)
23. Warrell, J., Salichos, L., Gerstein, M.: Latent evolutionary signatures: A general framework for analyzing music and cultural evolution. Preprint bioRxiv doi:10.1101/2020.10.23.352930 (2020)
24. Shiiba, K., Kubo (ed.), S.: *Japanese Songs Vols. 3–9* (in Japanese). Nobarasha, Tokyo (1998/1999/2001/2004/2014)
25. Uehara, Y., Nakamura, E., Tojo, S.: Chord function identification with modulation detection based on HMM. In: *Proc. CMMR*, pp. 59–70. The Laboratory PRISM, Marseille (2019)
26. Kimura, M.: On the probability of fixation of mutant genes in a population. *Genetics* 47(6), 713–719 (1962)
27. Wilke, C.O., et al.: Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature* 412, 331–333 (2001)

## **A network approach to harmonic evolution and complexity in western classical music**

Marco Buongiorno Nardelli<sup>1,2</sup>

<sup>1</sup> Department of Physics and Division of Composition Studies,  
University of North Texas, Denton, TX 76203, USA

<sup>2</sup> Santa Fe Institute, Santa Fe, NM 87501, USA  
mbn@unt.edu

**Abstract.** I recently introduced the concept of dynamical score network to represent the harmonic progressions in any composition. Through a process of chord slicing, I obtain a representation of the score as a complex network, where every chord is a node and each progression (voice leading) links successive chords. In this paper, I use this representation to extract quantitative information about harmonic complexity from the analysis of the topology of these networks using state of the art statistical mechanics techniques. Since complex networks support the communication of information by encoding the structure of allowed messages, we can quantify the information associated with locating specific addresses through the measure of the entropy of such network. In doing so I can then introduce a measure of complexity that can be used to quantify harmonic evolution when applied to an extensive corpus of scores spanning 500 years of western classical music.

**Keywords:** music complexity; computational music theory; music analysis; music composition; music information retrieval; music evolution; music innovation

### **1 Introduction**

In the article *Topology of Networks in Generalized Musical Spaces*, published on the Leonardo Music Journal, [1] I have introduced the concept of harmony as a network representation of the musical structures built out of all possible combinatorial pitch class sets in any arbitrary temperament. Inspired by a long tradition of network representations of musical structures such as the circle of fifths [2], the Tonnetz [3], and recent works on the spiral array model of pitch space, [4] the geometry of musical chords [5] and generalized voice-leading spaces [6] [7], I interpret the harmonic structure of a composition as a large-scale complex network whose topological properties uncover its underlying organizational principles and demonstrates how classifications



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

or rule-based frameworks (such as common-practice harmony, for instance) can be interpreted as emerging phenomena in a complex network system. Since the conclusions of that study serves as foundations for the present paper, let me review some of its principal results.

Network analysis methods exploit the use of graphs or networks as convenient tools for modelling relations in large data sets. If the elements of a data set are thought of as “nodes”, then the emergence of pairwise relations between them, “edges”, yields a network representation of the underlying set. Similarly to social networks, biological networks and other well-known real-world complex networks, entire dataset of musical structures can be treated as networks, where each individual musical entity (pitch class set (pcs), chord, rhythmic progression, etc.) is represented by a node, and a pair of nodes is connected by a link if the respective two objects exhibit a certain level of similarity according to a specified quantitative metric. Pairwise similarity relations between nodes are thus defined through the introduction of a measure of “distance” in the network: a “metric” [8]. As in more well-known social or biological networks, individual nodes are connected if they share a certain property or characteristic (i.e., in a social network people are connected according to their acquaintances, collaborations, common interests, etc.) Clearly, different properties of interest can determine whether a pair of nodes is connected; therefore, different networks connecting the same set of nodes can be generated.

In this paper I construct networks where nodes are the individual chords that can be extracted from a score, and edges are built between successive chords in the progression: nodes are connected if they appear as neighbours in the sequence. Naturally, nodes are visited numerous times, and the score evolution implies a directionality of the links. The networks are thus “directed”, and each edge will have a weight (strength) proportional to the times the link is visited.

Given a network, we can perform many statistical operations that shed light on the internal structure of the data. In this work I will consider only two of such measures, degree centrality and modularity class. [9] The degree of a node is measured by the number of edges that depart from it. It is a local measure of the relative “importance” of a node in the network. Modularity is a measure of the strength of division of a network into communities: high modularity (above 0.6 in a scale from 0 to 1) corresponds to networks that have a clearly visible community structure. [10]. Isolating communities through modularity measures provides a way to operate within regions of higher similarity.

In a more recent work [11] I have proposed that this score network can be viewed both as a static graph that represents all the existing chord changes in a composition, or as a dynamical system, a time series of a non-stationary signal, and as such, it can be partitioned, as for community structures, using time series analysis and change point detection. This dual representation (static and dynamical) offers novel ways to quantify the harmonic complexity of a single score or a full corpus without relying on comparisons with pre-determined reference sets.

## **2 Methods**

### **2.1 Network models**

I will make use of two principal software libraries for computational music analysis, both written in the Python language: MUSICNTWRK (at [www.musicntwrk.com](http://www.musicntwrk.com)) and music21 (at <https://web.mit.edu/music21>). MUSICNTWRK is an open source python library for pitch class set and rhythmic sequences classification and manipulation, the generation of networks in generalized music and sound spaces, deep learning algorithms for timbre recognition, and the sonification of arbitrary data [11]. music21, developed at MIT [12], is an object-oriented toolkit for analysing, searching, and transforming music in symbolic (score-based) forms of great versatility, whose modularity allows a seamless integration with MUSICNTWRK and other applications.

Scores are read in musicxml format by the readSCORE function of MUSICNTWRK, where their harmonic content (and other relevant information, like in which bar the chord is found) is extracted using the music21 parser and converter (using the “chordify” method). With this we obtain easily the full sequence of pcs, chord by chord, where each change to a new pitch results in a new chord. Upon “chordification”, each pcs is reduced to its normal form. While such “quantization” of pcs is quite adequate for the analysis of pieces with a harmonic movement where each vertical pcs plays some functional values (for instance in the corpus of J.S. Bach’s chorales), for compositions where there is more contrapuntal development, the number of individual pcs in the sequence can become very large, without providing necessarily more detailed information, since many such chords are only modifications via passing notes or vagrant harmonies. To circumvent this problem and make the analysis more manageable without losing any functional value, we have devised a “filtering” algorithm based on the cumulative measure of how many times an individual pcs appears in the sequence. All pcs with a frequency lower than a threshold are eliminated.

Starting from digitalized scores (in musicxml or MIDI format) I then construct networks where nodes are the individual chords, and edges are built between successive chords in the progression: nodes are connected if they appear as neighbors in the sequence. Naturally, nodes are visited numerous times, and the score evolution implies a directionality of the links. The networks are thus directed, and each edge will have a weight (strength) proportional to the times the link is visited. In Figure 1 I show the network of one score from the L. van Beethoven’s corpus.

I have analyzed an extensive corpus of scores by composers spanning five centuries of western classical music: Josquin des Prez (1450-1521), G.P. da Palestrina (1525-1594), Claudio Monteverdi (1567-1643), J.S. Bach (1685-1750), J. Haydn (1732-1809), W.A. Mozart (1756-1791), L. van Beethoven (1770-1827), J. Brahms (1833-1897) and G. Mahler (1860-1911).

### **2.2 Conditional Degree Matrix**

The local metrics usually used in networks theory fall short of capturing the richness of the vast majority of natural network topologies. At the same time, one of the most commonly used (local) characteristics is a node’s degree. Based on this attribute, I propose

to use what has been named "conditional degree matrix"  $D$  [13] to characterize the topology of the harmonic networks.

This matrix captures the classical node distribution and shows the existing architecture between the network nodes taking into account their different degrees. Each element of the matrix,  $D_{i,j}$  is defined as the number of nodes of degree  $i$  connected to nodes of degree  $j$ ,  $N_{i,j}$ , divided (normalized) by the number of total nodes,  $N_t$ , that is:

$$D_{i,j} = \frac{N_{i,j}}{N_t}.$$

This definition produces a symmetric matrix and ensures that  $D$  is properly normalized. More generally, directed and weighted networks would result in non-symmetric matrices.

The structure of the  $D$  matrix allows to estimate the complexity of a given network and provides more information than the classical degree distribution:  $D$  effectively acts as a probability matrix and can be the input for the evaluation of other metrics such as entropy, divergence, and complexity among others.

One of the essential properties of this matrix is that it allows to explore the characteristics of the degree of connections of each node with its environment (its close neighborhood) in a direct way. Its importance can be understood in terms of information diffusion: the rows  $i$  of this matrix show the probability that nodes of degree  $i$  are connected with nodes of another degree  $j$ . Their frequency will finally be reflected in each of its elements  $D_{i,j}$ .

### 2.3 Kullback-Leibler divergence

To extract quantitative information from the network topology I use the the Kullback-Leibler (KL) divergence as metric. In both information theory and probability theory, the Kullback-Leibler divergence is used as a measure indicating the difference between two probability functions. In general terms, KL measures the expected number of extra bits or excess surprise from using  $Q$  as a model when the data distribution is  $P$ .

The Kullback-Leibler divergence for the conditional degree matrix is defined as:

$$KL = \sum_{i,j} D_{i,j} \log(D_{i,j}/Q_{i,j})$$

where  $D$  is our CDM, and the reference matrix  $Q$  is defined as the mean of all the  $D$  for the whole corpus:

$$Q_{i,j} = \frac{1}{N} \sum_n D_{i,j}^n,$$

and  $N$  is the total number of score networks.

Since the KL divergence quantifies how much the topology of any individual network "diverges" from the average of the reference corpus, it provides a way to quantify

the difference in the distribution of observed degrees and in particular, the way in which the occurrence and distribution of hubs (as chords that are more important in the harmonic progression of a piece) characterizes the harmonic structure of the composition.

## 2.4 Diffusion Entropy Analysis

Diffusion Entropy Analysis (DEA) is a time-series analysis method for detecting temporal complexity in a dataset; such as heartbeat rhythm [14] [15] [16] a seismograph [17], or financial markets [18]. DEA uses a moving window method to convert the time-series into a diffusion trajectory, then uses the deviation of this diffusion from that of ordinary brownian motion as a measure of the temporal complexity in the data. It is thus appropriate to analyze the score time series and derive quantitative estimates of complexity.

Diffusion Entropy Analysis was first introduced by Scafetta and Grigolini [19] as a method of statistical analysis of time-series based on the Shannon entropy of the diffusion process to determine the scaling exponent of a complex dynamic system. It was later refined with the introduction of "stripes" (MDEA) by Culbreth *et al.* [20] in the context of detecting crucial events. While the reader should refer to the publications above for a full treatment of DEA, here I use the realization that the scaling of the diffusion coefficient  $\delta$  obtained in DEA provides a measure of complexity of the time-series, measured through the statistics of occurrences of crucial events. Here,  $\delta$  ranges between 0.5 and 1.0: for a completely non-complex process, such as a random walk, MDEA yields  $\delta = 0.5$ . For a process at criticality, MDEA yields  $\delta = 1$ . Therefore,  $\delta$  represents a measure of the "strength" of the complexity present in the process: the closer  $\delta$  is to 1 the closer the process is to criticality.

In Fig. 2, we show a MDEA analysis of the first movement of Beethoven's string quartet Op. 127 n. 12, that was extensively discussed in [21]:  $\delta \approx 0.7$  indicates a "medium" level of complexity as observed in other composition of the same period as it will be discussed extensively below.

MDEA analysis has been carried out using the module DEA implemented in the MUSICNTWRK library [11].

## 3 Results and discussion

I have applied the above metrics to our selected corpus of composition and the results are summarized in Fig. 3 and 4.

In Fig. 3 I show the KL divergence calculated for the full corpus of compositions. Here the values are referenced to the average of the corpus, that is, I am capturing how much the topology of a given piece deviates from the cumulative average. The results point to a clear distinction between earlier polyphony (des Prez, Bach) and later chromaticism (Mahler). There is a marked transition starting in the XIX century and culminating with the works of Brahms and Mahler. It is important to note that this metric provides a somewhat indirect measure of complexity as a relative difference between compositions. Of course, more work is needed to understand this metric further: although large, the corpora I have analyzed are still small in statistical terms, and more

analysis should be done by extending the corpora to a larger repertoire and/or using simulated data as toy models of the musical practice of selected composers.

For a more direct evaluation of complexity, we turn now to the MDEA results. By applying the procedure outlined in Fig. 2 to the full corpus, I have extracted the values of  $\delta$  for each piece and collected the results in Fig. 4.

Although the data points show a wide distribution around the averages for each composer, the results point to an increase of harmonic complexity over time, a result that agrees broadly with other analysis based on different metrics. These results allow us to discriminate further among composers and different time periods. We can, in principle divide this graph in three regions. The first region corresponds to the Renaissance and Early Baroque composers, where  $\delta$  is consistently lower. Since this musical period is characterized by a modal approach to harmony, we can easily infer that modal harmony is characterized by a lower complexity, as observed in the scarcity of functional chords (functional chords are hubs in tonal harmony networks [21], a more homogeneous distribution of node degrees, and a lack of multiple tonal centers.<sup>1</sup>

The second section corresponds to the Common Practice period, that shows an average complexity measure of  $\approx 0.7$ . This is when tonal harmony has matured into an established musical language. Once again in the third section, corresponding to XIX and early XX century, harmonic complexity increases to an average of 0.8. Once more, enhanced criticality and complexity correspond to a fragmentation of the tonal harmony language towards increase chromaticism, as it was observed in the CDM data above.

Finally, I have superimposed measures of complexity for pieces of the pop/rock repertoire as a suggestion for further analysis and discussion beyond the realm of classical music genres.

## 4 Conclusions

In conclusion, with this study I have built on the concept of network representation of musical spaces, introduced the idea of a composition as a dynamical score network, and discussed two complementary measures of music complexity. Although the results point unequivocally to an increase of harmonic complexity in western classical music, the present study is just an initial exploration of this fascinating topic. One of the challenges is the availability of large score corpora that would make the analysis of a single composer's production more coherent. I am currently working to expand the availability of corpora and I hope to extend this work in the future. Notwithstanding its limitations, this study demonstrates that combining the abstraction of a score as network with established mathematical and statistical techniques is a powerful tool for a quantitative analysis of music production that is independent of prior musicological or music

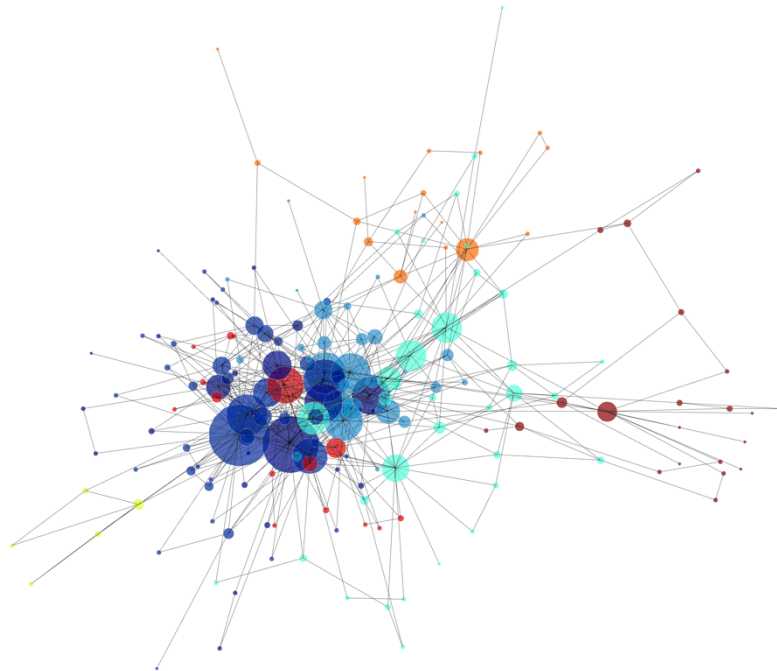
---

<sup>1</sup> It is important to note that here *complexity* must be interpreted as a statistical measure on the network topology associated to a dynamical system behavior, and not as a measure of how sophisticated a piece is or is perceived by a listener: a Palestrina's Mass can be more challenging and sophisticated than a Bach's chorale, although their measured complexities might suggest otherwise.

theoretic information and opens the way to a novel interpretation of music as a dynamical process.

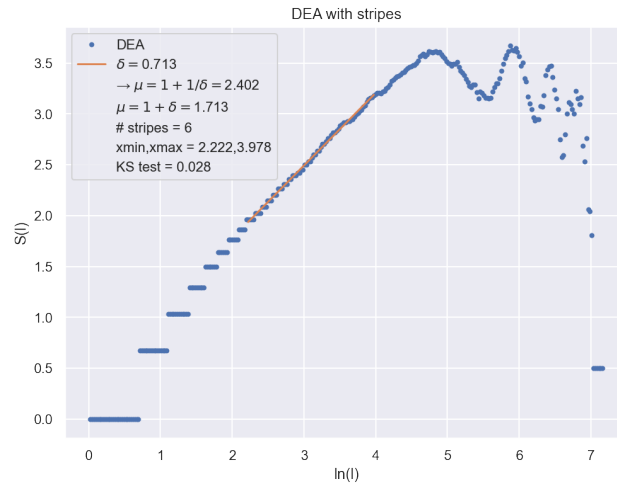
## 5 Acknowledgements

This paper is based on an earlier study published as: M. Buongiorno Nardelli, G. Culbreth and M. Fuentes, “Evolution of harmonic complexity in western classical music”, *Advances in Complex Systems*, vol. 25, No. 05n06, 2022.

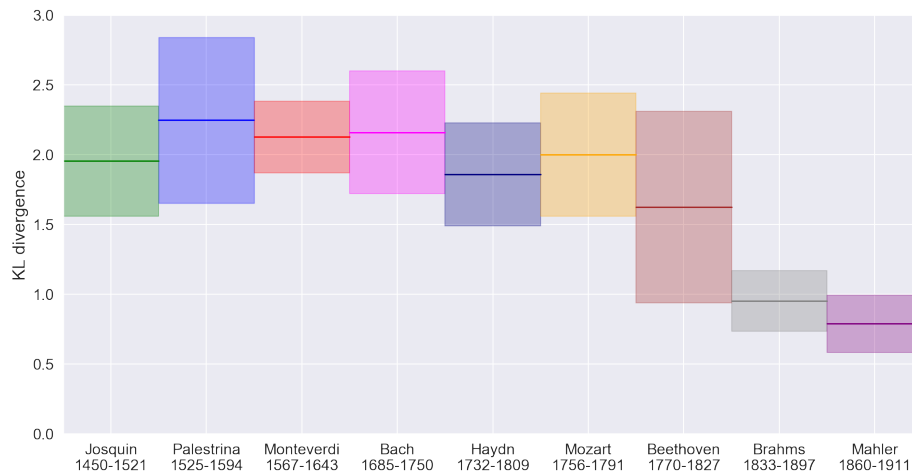


**Fig. 1.** Network structure of the third movement of Ludwig van Beethoven’s string quartet Op. 127 n. 12. Node size is proportional to degree, colors indicate the network community structure (the tonal region central to that particular section) and links correspond to voice leading (the transition from one chord to the next).

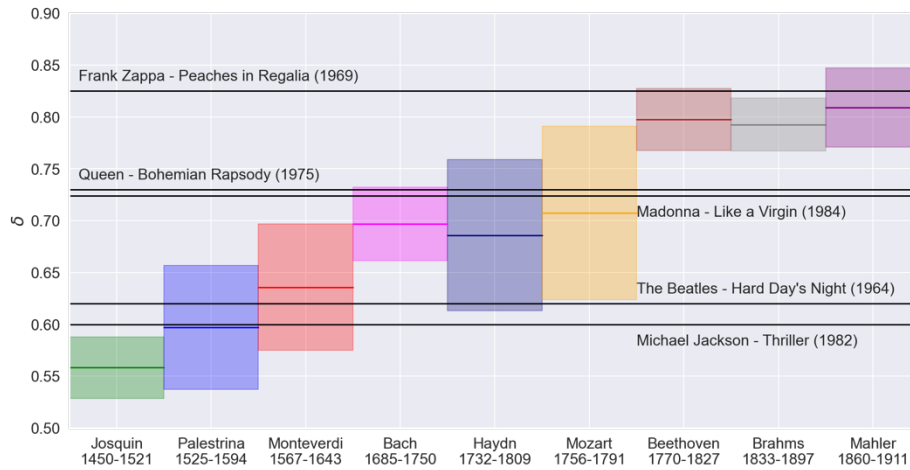




**Fig. 2.** Diffusion Entropy Analysis applied to the third movement of Beethoven's string quartet Op. 127 n. 12, whose network is shown in Fig. 1.



**Fig. 3.** Kullback-Leibler divergence. Horizontal lines are the average values of KL across the corpus of each composer; shaded areas indicate standard deviations.



**Fig. 4.** Complexity for different composers as measured from the diffusion entropy analysis. Horizontal lines are the average values of  $\delta$  across the corpus of each composer; shaded areas indicate standard deviations.

## References

- [1] M. Buongiorno Nardelli, "Topology of Networks in Generalized Musical Spaces," *Leonardo Music Journal*, vol. 30, pp. 38-43, 2020.
- [2] J. D. Heinichen, "Der General-Bass in der Composition," 1969.
- [3] L. Euler, *Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae*, Saint Petersburg Academy, 1739, p. 147.
- [4] E. Chew, *Mathematical and Computational Modeling of Tonality*, Springer US, 2014.
- [5] D. Tymoczko, "The Geometry of Musical Chords," *Science*, vol. 313, pp. 72-75, 2006.
- [6] C. Callender, I. Quinn and D. Tymoczko, "Generalized Voice-Leading Spaces," *Science*, vol. 320, p. 346, 2008.
- [7] D. Tymoczko, "The Generalized Tonnetz," *Journal of Music Theory*, vol. 56, no. 1, pp. 1-52, 2012.
- [8] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. 1, p. 47-97, 2002.

- [9] A.-L. Barabasi and M. Posfai, *Network Science*, Cambridge: Cambridge University Press, 2016.
- [10] D. Zinoviev, *Complex Network Analysis in Python: Recognize - Construct - Visualize - Analyze – Interpret*, Pragmatic Bookshelf, 2018.
- [11] M. Buongiorno Nardelli, "MUSICNTWRK: data tools for music theory, analysis and composition,," *Springer Lecture Notes in Computer Science*, vol. 12631, p. 190, 2021.
- [12] M. S. a. A. C. Cuthbert, " "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," in *International Society for Music Information Retrieval*, 2010.
- [13] J. C. Cardenas, G. Olivares, G. Vidal, C. Urbina and M. Fuentes, "The structure of online information behind social crises," *Frontiers in Physics*, vol. 9, p. 116, 2021.
- [14] G. Bohara, D. Lambert, B. J. West and P. Grigolini, "Crucial events, randomness, and multifractality in heartbeats," *Physical Review E*, vol. 96, no. 6, p. 062216, 2017.
- [15] R. Tuladhar, G. Bohara, P. Grigolini and B. J. West, "Meditation-induced coherence and crucial events," *Frontiers in physiology*, vol. 9, p. 626, 2018.
- [16] H. F. Jelinek, R. Tuladhar, G. Culbreth, G. Bohara, D. Cornforth, B. J. West and P. Grigolini, "Diffusion entropy vs. multiscale and r'enyi entropy to detect progression of autonomic neuropathy," *Frontiers in Physiology*, vol. 11, 2020.
- [17] M. S. Mega, P. Allegrini, P. Grigolini, V. Latora, L. Palatella, A. Rapisarda and S. Vinciguerra, "Power-law time distribution of large earthquakes,," *Physical Review Letters*, vol. 90, no. 18, p. 188501, 2003.
- [18] S.-M. Cai, P.-L. Zhuo, H.-J. Yang, C.-X. Yang, B.-H. Wang and T. Zhou, "Diffusion entropy analysis on the scaling behavior of financial markets,," *Physica A: Statistical Mechanics and its Applications*, vol. 367, pp. 337-344, 2006.
- [19] N. Scafetta and P. Grigolini, "Scaling detection in time series: Diffusion entropy analysis," *Phys. Rev. E*, vol. 66, p. 036130, 2002.
- [20] G. Culbreth, B. J. West and P. Grigolini, "Entropic approach to the detection of crucial events," *Entropy*, vol. 21, no. 2, 2019.
- [21] M. Buongiorno Nardelli, "Tonal harmony and the topology of dynamical score networks," *Journal of Mathematics and Music*, vol. <https://doi.org/10.1080/17459737.2021.1969599>.

# On the Analysis of Voicing Novelty in Classical Piano Music

Halla Kim<sup>1</sup> and Juyong Park<sup>1</sup>

Graduate School of Culture Technology, KAIST  
kimhalla@kaist.ac.kr juyongp@kaist.ac.kr

**Abstract.** Musical composition can be viewed as an act of conditional problem solving, the realization of musical ideas by arranging notes spatially and temporally. The resulting creations may constitute the unique style of the composer. In this paper we focus on how chord voicing – the expression of chords by choosing and stacking musical notes – has evolved in western classical piano music using large-scale music data sets. Our results shows that the level and variety of voicing novelty have increased throughout history. We also find that some composers exhibit a high level of voicing novelty due to the utilization of innovative pitch class sets, while others actually have pushed the boundaries of voicing with traditional pitch class sets. This study helps us to probe the emergence of expression of musical style on note level and to understand the evolutionary pattern of note arrangements.

**Keywords:** Musical style, Voicing, Evolution, Novelty

## 1 Introduction

Musical composition can be viewed as a process of conditional problem solving: Composers' creations reflect their musical ideas by way of the selection and arrangements of such musical elements as melody, rhythm, harmony, and structure, which results the creations manifesting their particular musical styles [1]. In this paper we investigate in particular **chord voicing** – how musical notes are vertically arranged to express a given harmonic scheme – which is the core element of harmonic progression often called the fundamental task in Western musical composition [2]. For example, a fundamental voice-leading rules in classical music of dominant to tonic chord is realized in different ways: A Pitch Class set (hereafter PC-set) movement from  $\{G, B, D\}$  to  $\{C, E\}$  can be written either as  $(G3, D4, B4)$  followed by  $(C3, E4, C5)$  or as  $(B2, G3, D5)$  followed by  $(C3, E3, C5)$  chosen by the composer.

By uncovering the historical compositional patterns of voicing, here we inspect how they developed over time. Previous works on musical novelty showed the stylistic evolution or the sweetspot in terms of success, including Park *et al.* [3] who showed that musical periods can be characterized by the novelty and influence of composers of



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

each era. Similarly, the predictability of subsequent chords is measured by exploring chord transition probability [4], whereas the time of musical revolutions can be traced where novelty of harmonic and timbre properties change drastically [5]. Other symbolic features such as melodic intervals [6] or triads [7] are also shown to be effective for identifying styles and distinguishing musical eras. Weiß *et al.* [8] observed that the frequency of tritones and tonal complexity have steadily increased over the history of Western classical music. Nakamura and Kaneko [9] developed a statistical evolutionary model that fits the frequency data of tritones and that of non-diatonic motions where the creators and the evaluators coevolve through a function of novelty and typicality in a process of social selection. Finally, O’Toole and Horvát [10] used audio features to evaluate novelty and asserted that optimal level of differentiation is needed to become the most popular song. Few, however, have explicitly examined chord voicing (the vertical placement of pc-sets), with Harrison and Pearce [2] being an exception who introduced a computational framework of voicing. They suggested a mathematical model to calculate the probability of choosing the next voicing given current voicing according to pre-defined perceptual rules of chord voicing. In this paper, we try to investigate the very fundamental aspect of voicing; We hypothesize that novel ways of placing notes have been developed that characterize the style of composer and musical era.

## 2 Methodology

### 2.1 Dataset

The data used in our analysis were collected mainly from three online sources<sup>1</sup>, and consist of 1 017 piano compositions by 40 historically prominent composers in MIDI format. We follow the common convention of dividing the history of classical music into the following five periods: Baroque, Classical, Romantic, Post-romantic, and Modern, and specifically use All Music Guide<sup>2</sup> to tag the year of compositions and the era to which the composers belong.

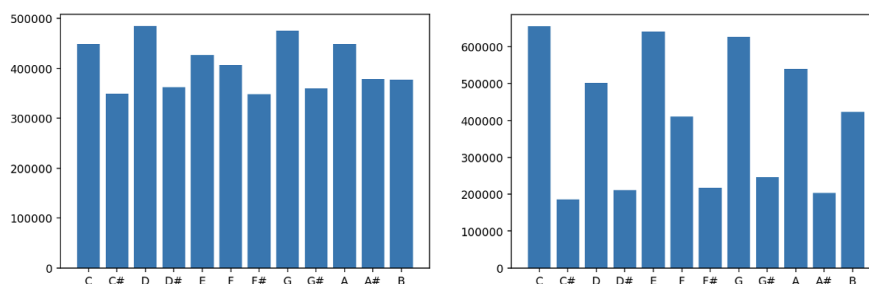
### 2.2 Key Normalization

The objective of key normalization is to treat the notes equally that serve identical harmonic functions, irrespective of their absolute pitch, to promote consistency. For example, the harmonic role of the chord  $C = \{C, E, G\}$  in C major is identical to that of the chord  $F = \{F, A, C\}$  in F major. The key of each composition in the dataset was estimated using a the Krumhansl-Schmuckler key detection algorithm [11]. This algorithm compares the pitch class distribution of a given piece of music with the key profiles obtained from music-cognitive experiments [12], and selects the key with the highest Pearson correlation among the 24 possible key. If the highest Pearson correlation coefficient value refers to an outlier among the values for the whole song, key normalization was not performed since the algorithm’s key estimation is unreliable. Any number that falls outside of the first quartile (Q1) or above the third quartile (Q3) by more than

<sup>1</sup> <http://www.piano-midi.de>, <https://www.classicalarchives.com>, <http://www.kunsterfuge.com>

<sup>2</sup> <http://www.allmusic.com>

1.5 times the interquartile range (IQR=Q3-Q1) was considered an outlier. After each composition's key was estimated, all non-outlier major compositions were transposed to C major, and minor compositions to A minor (Fig. 1). It is clear from Fig. 1 that the comparatively high non-diatonic frequency is a result of outlier songs that would have chromatic scales or key modulation.



**Fig. 1.** Pitch Class distribution of the compositions before (left) and after (right) key normalization.

### 2.3 Encoding of Voicing

Voicing refers to the simultaneous vertical placement of notes in relation to each other [13] or assigning pitch heights to pitch classes [2]<sup>3</sup>. To conduct voicing analysis, we first encode each musical composition as a series of group of notes played simultaneously (which we call codewords) [14]. Next, we focus on the voicing of each codeword to calculate the novelty of voicing used by composers. While there is an issue of not being able to clearly distinguish between voicing and harmonic skeleton [2], here we model a composer as choosing the pc-set first then subsequently determining the voicing as an elaboration or an embellishment.

**Voicing Encoding Given a PC-set** Regarding the voicing as an implementation of pc-set, we can express the probability of choosing a voicing  $v_i$  given a pc-set  $s_i$  as

$$P(v_i|s_i) = \frac{z(s_i \rightarrow v_i) + \alpha(s_i \rightarrow v_i)}{\sum_{v \in V(s_i)} z(s_i \rightarrow v) + \alpha(s_i \rightarrow v)}, v_i \in V(s_i), \quad (1)$$

where  $V(s_i)$  is the set of all possible voicings for a pc-set  $s_i$ ,  $z(s_i \rightarrow v_i)$  the number of occurrences of voicings  $v_i$  that have a pc-set  $s_i$ , and  $\alpha$  is a constant representing an uninformed prior, a type of additive Laplace smoothing. Setting  $\alpha = 1$  means that every conceivable voicing element in  $V(s_i)$  has a finite chance of being chosen [3]. Let

<sup>3</sup> Another definition of voicing, a placement of notes among various instruments, does not fit to our analysis since we only address musical pieces for piano solo.

us take an example of a pc-set  $s_i = \{C, E, G\}$ . This pc-set's possible voicings  $V(s_i)$  includes  $v_i = (C4, E4, G4), (E4, G4, C5), (C3, E3, G3)$  etc. The total number of all possible voicings is equal to the number of piano key combinations available in a given pc-set. Note that the two voicings  $(C4, E4, G4)$  and  $(C3, E3, G3)$  differ by an octave. To discard octave position and only handle the relative spacing between notes, we use the voicing notation as the pitch interval between adjacent notes starting from bass note (Fig. 2).

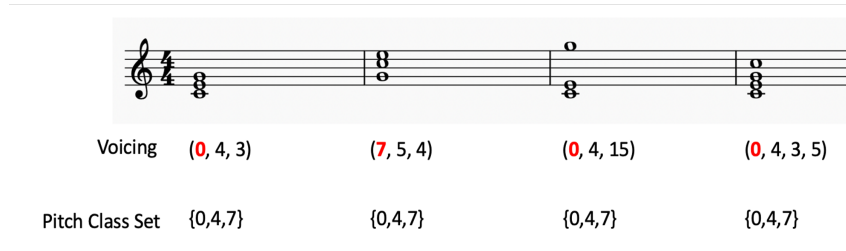


Fig. 2. Example of voicing and pc-set notation

## 2.4 Calculating Voicing Novelty

To see how the voicing style has evolved, we first measure the novelty of voicing. Representing a composition as a sequence of codewords (configuration of voicings)  $\zeta = \{v_1, v_2, \dots, v_m\}$ , we can write the generation probability of  $\zeta$  as first-order Markov chain

$$\Pi(\zeta) = P(v_1|s_1)P(v_2|s_2) \dots P(v_m|s_m). \quad (2)$$

Its log inverse is the magnitude of surprise in information theory. We can thus quantify the novelty in voicing as an average unexpectedness of all voicings in a composition normalized by the length of a composition  $m$ :

$$Novelty(\zeta) = \frac{1}{m} \log \frac{1}{\Pi(\zeta)} = \frac{1}{m} \left[ \sum_{k=1}^m \log \frac{1}{P(v_k|s_k)} \right] \quad (3)$$

## 2.5 Calculating PC-set Novelty

Here we discuss the pc-set and compare it with voicing novelty. pc-set novelty measures how novel the current codeword's pc-set is when the preceding codeword's pc-set is given. Representing a composition as a sequence of codewords' pc-set,  $\zeta = \{s_1, s_2, \dots, s_m\}$ , the probability of choosing a pc-set  $s_{i+1}$  after a pc-set  $s_i$  is written as,

$$P(s_{i+1}|s_i) = \frac{z(s_i \rightarrow s_{i+1}) + \alpha(s_i \rightarrow s_{i+1})}{\sum_{k \in S} z(s_i \rightarrow k) + \alpha(s_i \rightarrow k)}, s_i \in S, \quad (4)$$

where  $S$  is the set of all possible pc-sets,  $z(s_i \rightarrow s_{i+1})$  the number of occurrences of pc-set transition from  $s_i$  to  $s_{i+1}$ . Then the pc-set novelty can be acquired by plugging Eq. 5 into Eq. 6.

$$\Pi(\zeta) = P(s_1)P(s_2|s_1) \dots P(s_m|s_{m-1}) \quad (5)$$

$$Novelty(\zeta) = \frac{1}{m} \log \frac{1}{\Pi(\zeta)} = \frac{1}{m} \left[ \log \frac{1}{P(s_1)} + \sum_{k=1}^{m-1} \log \frac{1}{P(s_{k+1}|s_k)} \right] \quad (6)$$

### 3 Results

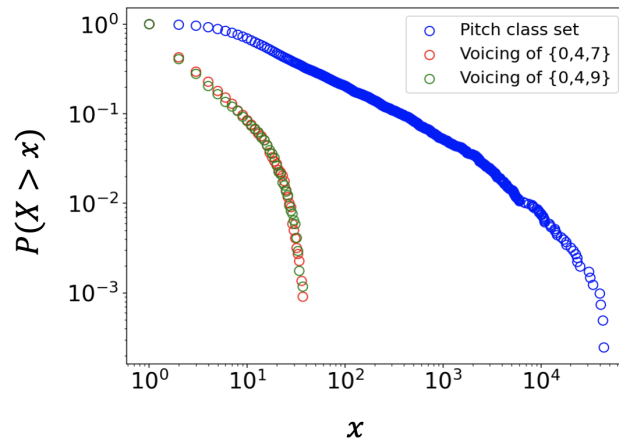
By examining the usage patterns of both the pc-set and the voicing of codewords, we investigate the composing style of Western classical composers. In the data set the total number of codewords is 1 230 441, and its unique number of pc-set and voicing set were 4 071 and 209 439, respectively. Note that we only include the codewords that consist of at least two notes. PC-set distribution is significantly skewed, which suggests that only a tiny portion of pc-set are employed for composition while the majority is rarely used (Fig. 3). The distribution of voicing is less skewed than that of the pc-set; the maximum frequency of voicing usage is less than  $10^2$  whereas the highest frequency of the pc-set is much more than  $10^4$ . We decide to display the voicings of both  $\{0,4,7\}$  and  $\{0,4,9\}$  since they are the tonic chords of the normalized key and are included in the top five most often used pc-sets (Table. 1). The second column shows the top five voicings of pc-set  $\{0,4,7\}$  (i.e.  $\{C,E,G\}$ ) and the third column shows the top five voicings of pc-set  $\{0,4,9\}$ , i.e.  $\{C,E,A\}$ .

**Table 1.** Top five frequently used pc-sets and voicings of two representative pc-sets.

Rank	PC-set	Voicing of $\{0,4,7\}$	Voicing of $\{0,4,9\}$
1	$\{0,4,7\}$	(4,3,5)	(0,4,5)
2	$\{0,4\}$	(0,4,3)	(9,3,4)
3	$\{0,4,9\}$	(7,5,4)	(4,5,3)
4	$\{0,7\}$	(0,7,9)	(9,3,4,5)
5	$\{4,7\}$	(0,4,3,5)	(9,7,8)

The heterogeneity in the use of pc-sets provides clues for interpreting the two novelties that we ultimately seek to analyze. Fig. 4 displays pc-set novelty and voicing novelty for composers, arranged chronologically based on the average years of birth and death. Composers like Couperin and Handel showed a high level of pc-set novelty





**Fig. 3.** The cumulative distribution on a log-log scale of the occurrences of pc-sets, voicing of  $\{0,4,7\}$ , and voicing of  $\{0,4,9\}$ . The frequency of pc-sets or voicing is represented on the horizontal axis, while the cumulative probability is represented on the vertical axis.

in their works due to the advantages of their time, and this remained consistent until later composers such as Elgar, Berg, Schoenberg, and Messiaen introduced new pc-sets, leading to greater variation. This figure agrees with Figure 1 (a) of Nakamura and Kaneko [9] which depicts the steadily increasing mean and standard deviation of tritone frequency, as tritone is one of the examples of historically important pc-sets. Similarly, increasing voicing novelty in later generations is a noticeable trend, with some prominent composers exemplifying it. When several new pc-sets are employed, the voicing novelty can rise only by virtue of the pc-set itself, or it can also increase if unique vertical arrangements are made using traditional pc-sets. These are the two scenarios where voicing novelty can be high. In order to identify the driving cause behind high voicing novelty, we computed the new pc-set ratio and new voicing ratio for each song. New pc-set ratio is the unique number of new pc-sets that were not used in the previous songs divided by the total unique number of pc-sets in a given song. It refers to the ratio of innovative pc-sets that the composer has chosen. New voicing ratio is the unique number of voicings for which the pc-set has already been used in previous songs but the present voicing is used for the first time in a given song, divided by the total unique number of voicings in a given song. The new voicing ratio serves as a metric to gauge the extent of reconfiguration undergone by existing pc-sets, as it specifically signifies the proportion of instances involving entirely novel voicings. According to Figure 4, while Handel exhibited the highest degree of pc-set novelty among all composers, benefiting from an early temporal advantage, Bach, despite sharing a similar advantage, displayed significantly lower pc-set novelty. This observation highlights Bach's propensity for predominantly composing using existing pc-sets. Composers such as Elgar, Berg, and Schoenberg, despite being situated in subsequent eras, stand out as instances where both the new pc-set ratio and new voicing ratio are elevated, resulting in pronounced

levels of pc-set novelty and voicing novelty. With the exception of a few later composers who introduced a significant number of new pc-sets, the majority enhanced the novelty of chordal expressions by creatively reconfiguring existing pc-set tones. Brahms compared to Beethoven serve as an example. Despite using a smaller percentage of new pc-set than Beethoven, Brahms had a greater new voicing ratio than Beethoven, which resulted in a higher level of voicing novelty (Figure 4). In this way, through the comparison of the ratios and novelty values of new pc-sets and voicings in each composition, it becomes possible to explore how the chordal expression in Western classical piano music has evolved uniquely for each composer.

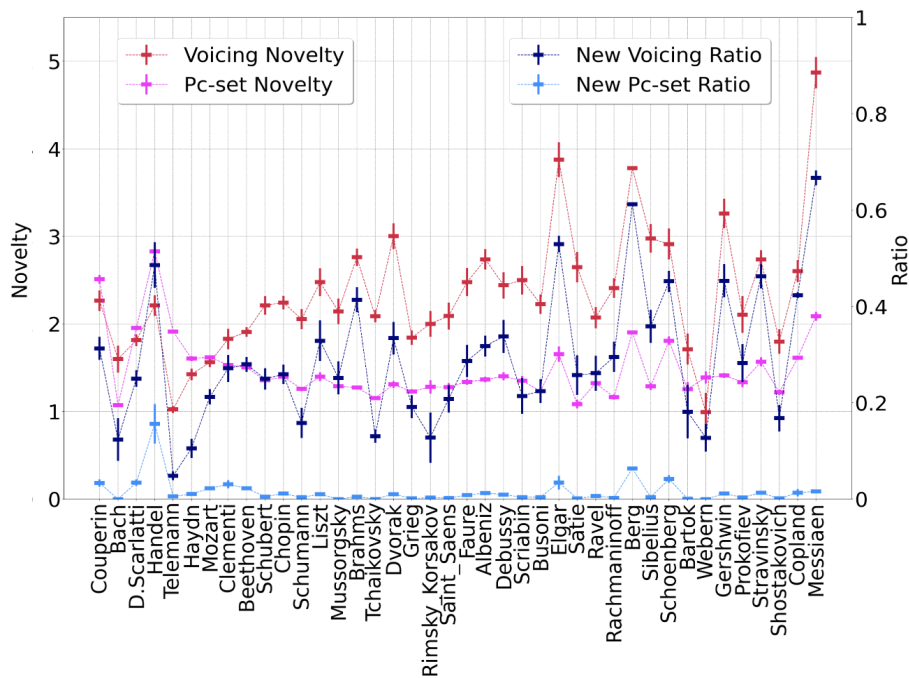


Fig. 4. Voicing and pc-set novelties in comparison to the new voicing ratio and new pc-set ratio.

#### 4 Conclusion

In this paper, we studied the compositional style of Western classical piano pieces, with a particular focus on chord voicing – the vertical arrangement of notes within a set of pitch classes. Despite using the same set of pitch classes, altering the octave’s location, the arrangement’s sequence, or the distance between notes might result in a significantly different sound. We first encoded the voicing for each codeword, computed the voicing novelty, and then looked at the historical evolution of voicings to understand how

composers chose and arranged notes. In Western classical piano music, voicing novelty exhibited a consistent upward trend over time, accompanied by an increasing divergence among composers. Early composers introduced a number of popular pc-sets that were widely used by later composers. Later composers mainly increased the novelty of the song by vertically arranging the pre-existing pc-set, with the exception of Elgar, Berg, and Schoenberg who used a sizable portion of novel pc-sets. Our examination of compositional trends among different composers reveals distinct patterns of novelty in their approach to chordal expression in Western classical piano music. As our study continues, we plan to explore the historical evolution of compositional style and how composers create their unique styles through influence scores of voicing, contributing to the understanding of musical styles at the note level using symbolic music data.

## Acknowledgement

This research was supported by KAIST Post-AI Research Grant and Korea Creative Content Agency funded by the Korean Government (RS-2023-00270043).

## References

1. LaRue, J. (1962). On style analysis. *Journal of Music Theory*, 6(1), 91–107.
2. Harrison, P. M., and Pearce, M. T. (2020). A computational cognitive model for the analysis and generation of voice leadings. *Music Perception*, 37(3), 208-224.
3. Park, D., Nam, J. and Park, J. Novelty and influence of creative works, and quantifying patterns of advances based on probabilistic references networks. *EPJ Data Sci.* 9, 2 (2020). <https://doi.org/10.1140/epjds/s13688-019-0214-8>
4. Moss, F. C., Neuwirth, M., Harasim, D., and Rohrmeier, M. (2019). Statistical characteristics of tonal harmony: A corpus study of Beethoven’s string quartets. *PLoS One*, 14(6), e0217242.
5. Mauch M, MacCallum RM, Levy M, Leroi AM (2015) The evolution of popular music: USA 1960-2010. *R Soc Open Sci* 2:150081
6. Rodriguez Zivic, P. H., Shifres, F., and Cecchi, G. A. (2013). Perceptual basis of evolving Western musical styles. *Proceedings of the National Academy of Sciences*, 110(24), 10034-10038.
7. Weiß, C., Mauch, M., and Dixon, S. (2014). Timbre-invariant audio features for style analysis of classical music.
8. Weiß, C., Mauch, M., Dixon, S., and Müller, M. (2019). Investigating style evolution of Western classical music: A computational approach. *Musicae Scientiae*, 23(4), 486-507.
9. Nakamura, E., and Kaneko, K. (2019). Statistical evolutionary laws in music styles. *Scientific reports*, 9(1), 15993.
10. O’Toole, K., and Horvát, E. Á. (2023). Novelty and cultural evolution in modern popular music. *EPJ Data Science*, 12(1), 3.
11. C. L. Krumhansl, *Cognitive foundations of musical pitch*, New York: Oxford University Press, 1990.
12. Krumhansl, C. L., and Kessler, E. J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological review*, 89(4), 334.
13. Corozine, V. (2015). *Arranging music for the real world*. Mel Bay Publications.
14. CONKLIN, D. (2002). Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand, A. Smaill (Eds.), *Music and artificial intelligence: Proceedings of ICMAI 2002* (pp. 32–42). Berlin, Germany: Springer-Verlag.

## **Bipartite network analysis of the stylistic evolution of sample-based music**

Dongju Park<sup>1</sup> and Juyong Park<sup>1\*</sup>

<sup>1</sup> Graduate School of Culture Technology, Korea Advanced Institute of Science & Technology,  
Daejeon, Republic of Korea  
pdj333@kaist.ac.kr / juyongp@kaist.ac.kr

**Abstract.** In this study we present a network analysis of the communities of artists based on sampling. We construct a bipartite network between the artists who perform the sampling and the samples, then detect communities of the artists and the samples. We find that sample-based music has a clear community structure where each community features artists (nodes) with high centralities, allowing us to determine its musical style. We also define and visualize the similarities between communities representing distinct generations to observe how sample-based musical styles have evolved or been “handed off” to the posterity. This study not only enhances our understanding of sampling-based music, but also presents a novel application of network community structure to a creative enterprise such as music.

**Keywords:** Sample-based music; Bipartite network analysis; Community detection; Music style evolution

### **1 Introduction**

Musical sampling is a technique used in popular music where one borrows some parts of existing recordings and incorporates them into new musical creations. Sampling can involve using any portion of a song, including the melody, drum parts, and vocals. While experimental music first began using sampling in the mid-20th century [1], it has since become extensively used in hip-hop, electronic, and pop music, particularly since the 1980s. The identity of sample-based music is profoundly related to the songs that were sampled. For instance, G-Funk, the dominant subgenre in West Coast hip-hop during the 90s, created its own rhythm by sampling George Clinton and other funk musicians [2]. Electronic music subgenres such as Jungle and Drum’n’Bass are built on the foundation of one of the most sampled songs in the world, “Amen, Brother.” [3] As such, the sampling practice of an artist reflects the characteristics of the subgenre or the music community to which the artist belongs. Therefore, analyzing sampling

---

\* corresponding author



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

relationships can help us comprehend the different musical styles in the in sample-based musical scene.

Music sampling data is a form of metadata about music that represents the sampling relationships between songs. It is relational data that can be analyzed using network analysis. Many studies have used network analysis to quantitatively analyze musical metadata. Notably, Bryan and Wang [4] created a musical influence network of songs based on sampling relationships and analyzed the network to identify the most influential songs in sample-based music. In another study that analyzed musical sampling using network methodology, Youngblood [5] utilized a network diffusion model to verify the hypothesis that the diffusion of drum breaks, which play a significant role as key samples in sample-based music, occurred through collaborative networks. Unlike this study that focused on drum breaks, our study considers the relationships of all samples with artists, analyzing the community structure of sample-based music that goes further beyond the influence between individual pairs of songs.

Community detection is one of the most standard methods of network analysis. It can also be applied to an influence network based on sampling relationships to discover groups within sample-based music. To do this, we take a cue from studies on citation networks. Musical sampling and academic citation are comparable in that they credit past works for the production of current works [6]. The concept of Author Bibliographic Coupling (ABC) exists in citation analysis [7], a measure of similarity between two authors who cite the same paper. When the author is replaced with an artist and the paper with a song, the similarity between two artists who sampled the same song can be defined in the same fashion.

In this study, we analyze the community structure of sample-based music by constructing an artist-sample bipartite network. The community detected in this network can be understood as reflecting a style in the sample-based musical scene. Furthermore, we define similarities between generations of communities to investigate the stylistic evolution of sample-based music, which we then visualize.

## **2 Materials and Methods**

### **2.1 Data and network construction**

Our data set consists of a total of 333 090 sampling cases between 1980 and 2019 procured from WhoSampled.com. Each case consists of the sampling relationship between song pairs and its metadata (artist, genre, year of release). The total number of songs included in the dataset is 296 456. Each artist's genre is set to be the most common one among the artist's songs in the data set. Since there can be many styles within a genre, it is impossible to specify an artist's musical style by the genre tag alone. To overcome this we collected the style tags shown on the artist's pages on Allmusic.com. Of the 42 969 sampling artists included in the sampling data, 14 124 style tags were collected. This low coverage is due to many of the artists who are relative obscure not having been tagged.

In this study, the 40-year period from 1980 to 2019 was divided into five-year intervals, yielding a total of eight generations. The songs in the data set were assigned

a generation by the year they were created. Then we constructed artist-sample bipartite networks inside each generation. The bipartite networks comprise two distinct node groups (artists and songs) with edges exclusively linking nodes between the opposing groups. Since an artist may sample a song multiple times, the network is weighted.

## 2.2 Community detection

When dealing with bipartite networks, community detection is often performed on the one-mode projection of the bipartite network into a unipartite network [8]. Alternatively, community detection can be performed without such projection, preventing the loss of data but is not as widely used. Various modifications of ‘modularity’ have been proposed for bipartite networks, where modularity maximization is a popular method for community detection in unipartite networks. Modularity is an index that quantifies how many more connections are inside the community compared to random expectation. Here we utilized Barber’s bimodularity [9], allowing both artist and sample nodes to be members of the same community, given as

$$Q = \frac{1}{w} \sum_i \sum_j \left( B_{ij} - \frac{d_{1,i}d_{2,j}}{w} \right) \delta_{c_{1,i},c_{2,j}}, \quad (1)$$

where  $B$  is the biadjacency matrix of the network, and  $w$  is the sum of the weights of all edges in the network.  $d_{1,i}$ ,  $d_{2,j}$  each denotes degree of node  $i$  of type 1, and  $j$  of type 2. And  $\delta_{c_{1,i},c_{2,j}}$  is 1 when node  $i$  of type 1 and  $j$  of type 2 are in the same community, and 0 otherwise.

To maximize Barber’s bimodularity, we used the Bilouvain algorithm [10], a bipartite variant of the Louvain algorithm. The Louvain algorithm is a heuristic algorithm applicable to weighted networks and is computationally efficient. In this study, the communities can contain both artists and samples in them.

## 2.3 Defining similarities between communities in different generations

The identity of the community can be determined from its samples; Artists resample previously sampled songs to acquire sounds similar to previous works or to demonstrate respect for senior artists. Consequently, if two communities from distinct generations share samples, it is likely that the two communities show a similar style. The similarity between two communities of distinct generations can be computed based on this idea.

A community is a bipartite network consisting of artist nodes, sample nodes, and the edges connecting them. A network centrality can be utilized to determine the importance of the samples. The degree centrality is the most fundamental centrality, but it only takes into account the local network information and has trouble differentiating nodes with the same degree due to being an integer value. The HITS (Hypertext Induced Topic Selection) score [?] is a centrality that incorporates nonlocal network information, and in this study, we employ the bipartite version of the HITS algorithm. HITS is a scoring algorithm for directed unipartite networks consisting of the ‘hub’ score and the ‘authority’ score. The hub score is the sum of the authority scores of nodes that the corresponding node points to, while the authority score is the sum of the

hub scores of nodes that point to the corresponding node. This can be extended to the bipartite networks [11] where the score for each node can be defined as the sum of the scores of the nodes it is connected to. This can be expressed using the formula below given as

$$p_j = \sum_{i=1}^{|U|} B_{ij}u_i; u_i = \sum_{j=1}^{|P|} B_{ij}p_j, \quad (2)$$

where  $B$  is the biadjacency matrix of the network, and  $U$  and  $P$  are separate node sets. The final scores are normalized to 1.

To determine the similarity between two communities, we first identify the shared samples. Then we merge the two communities into a single network and calculate their HITS scores. Then we compute similarity between the two communities as the sum of the HITS scores of the shared samples:

$$HITS\_Sim(C_1, C_2) = \sum_{s \in S_1 \cap S_2} HITS_{UC}(s), \quad (3)$$

where  $C_1 = \{A_1 \cup S_1, E_1\}$  and  $C_2 = \{A_2 \cup S_2, E_2\}$ .  $A_1$  and  $A_2$  are sets of artists and  $S_1$  and  $S_2$  are sets of samples.  $UC$  is a union of communities  $C_1$  and  $C_2$ .

### 3 Results

**Table 1.** Network information and community detection results for each generation.

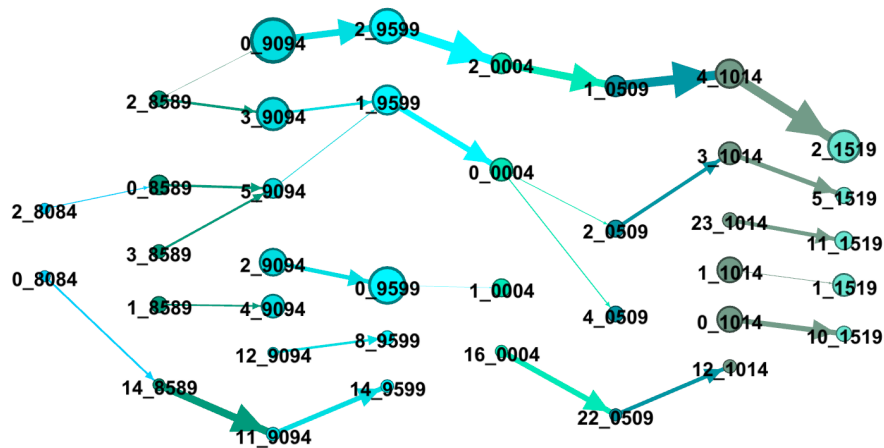
Generation	# of artists	# of samples	# of edges	# of communities	bimodularity
1980-1984	1060	2362	3460	465	0.884
1985-1989	3108	5614	19975	511	0.481
1990-1994	7871	14034	53916	966	0.491
1995-1999	9964	19196	46215	1761	0.667
2000-2004	8603	20874	36254	2521	0.784
2005-2009	9691	24147	41635	2785	0.777
2010-2014	14680	34469	63569	3805	0.757
2015-2019	15185	31002	53140	4369	0.827

Table 1 shows the information on the networks belong to the different five years-long generation. While the numbers of nodes in each group exhibit an upward trend over time, the number of edges exhibits a greater degree of variation. Detection results indicate that the number of identified communities increases over time. Bimodularity, quantifying the strength of the community structure, is comparatively low during the generations of 1985–1989 and 1990–1994 but increases subsequently.

The communities we derive from this study consist of artists and samples. Thus a community is not simply a group of artists but can be considered as representing

sample-based musical styles. We also identified artists and samples that played a significant role in the community (style) by calculating degree centrality, as the community evolved into another bipartite network. To better comprehend the musical styles of each community, we compiled the Allmusic style tags of each community’s artists and designated the five most common tags as the community’s main subgenres.

To investigate the evolution of sample-based music styles, we created a network of similarity between communities from successive generations. First, we see that the primary communities of each generation consist of more than 1% of the network’s total nodes. The similarity between the primary communities of successive generations was then computed using the similarity index defined earlier. Finally, we constructed a network consisting of the the primary communities of each generation as nodes and their similarity as edge weights. We set a threshold for the edge weights to visualize only connections above a certain level of similarity. Figure 1 depicts the network visualization resulting from a threshold value of 0.2. In terms of node labeling, for instance, ‘2\_9599’ represents the 2nd community of the 1995-1999 generation. The main subgenres of the primary communities visualized in Figure 1 are presented in Table 2.



**Fig. 1.** Similarity network of primary communities in each generation. Edges exceeding threshold 0.2 were excluded from visualization. The node size is proportional to the number of artists belonging to each community, and the thickness of the edge is proportional to the inter-community similarity. The color of nodes signifies the generation to which they belong. In terms of node labeling, for instance, ‘2\_9599’ represents the 2nd community of the 1995-1999 generation.

In Figure 1, a significant path is observed from 0\_9094 to 2\_1519 (top of the figure). These communities represent Jungle and Drum ‘n’ Bass, which are breakbeat-based subgenres of electronic music [3].(‘Jungle/Drum ‘n’ Bass’) Since these communities represent the largest electronic music samples of each generation, we can intuitively see that breakbeat-based music dominates sample-based electronic music. Breakbeat uses drum breaks included in funk, jazz, and R&B music, and the most famous drum break



**Table 2.** Main subgenres the visualized primary communities in Figure 1. The main subgenre of each community is determined by counting the style tags of artists belonging to the community.

comm.	main subgenre
0.8084	Dancehall, Roots Reggae, Ragga, Contemporary Reggae, Lovers Rock
2.8084	Golden Age, Old-School Rap, Electro, Alternative Pop/Rock, French
0.8589	Golden Age, Old-School Rap, Hardcore Rap, East Coast Rap, Club/Dance
1.8589	Club/Dance, House, Acid House, Dance-Pop, Techno
2.8589	Pop-Rap, Golden Age, East Coast Rap, Party Rap, Hardcore Rap
3.8589	Party Rap, Club/Dance, Bass Music, Quiet Storm, Modern Electric Blues
14.8589	Dancehall, Ragga, Roots Reggae, Contemporary Reggae, Lovers Rock
0.9094	Club/Dance, Jungle/Drum'n'Bass, Techno, House, Rave
2.9094	Gangsta Rap, Hardcore Rap, West Coast Rap, G-Funk, East Coast Rap
3.9094	Hardcore Rap, Pop-Rap, Golden Age, Contemporary R&B, East Coast Rap
4.9094	Club/Dance, House, Dance-Pop, Acid House, Euro-Dance
5.9094	Club/Dance, Party Rap, Bass Music, Southern Rap, Pop-Rap
11.9094	Dancehall, Ragga, Contemporary Reggae, Reggae-Pop, Club/Dance
12.9094	Hardcore Rap, Gangsta Rap, Southern Rap, Underground Rap, Dirty South
0.9599	Gangsta Rap, Hardcore Rap, West Coast Rap, G-Funk, Pop-Rap
1.9599	Club/Dance, Turntablism, Underground Rap, Hardcore Rap, East Coast Rap
2.9599	Club/Dance, Jungle/Drum'n'Bass, Techno, Hardcore Techno, Electronica
8.9599	Hardcore Rap, Gangsta Rap, Dirty South, Southern Rap, Adult Contemporary
14.9599	Dancehall, Contemporary Reggae, Ragga, Alternative Pop/Rock, Roots Reggae
0.0004	Alternative Rap, Hardcore Rap, Underground Rap, East Coast Rap, Turntablism
1.0004	Hardcore Rap, East Coast Rap, Gangsta Rap, Pop-Rap, West Coast Rap
2.0004	Jungle/Drum'n'Bass, Club/Dance, Techno, Electronica, IDM
16.0004	Contemporary R&B, Club/Dance, House, French House, Pop
1.0509	Jungle/Drum'n'Bass, Club/Dance, Garage, Breakcore, Dubstep
2.0509	Hardcore Rap, Alternative Rap, Alternative/Indie Rock, Underground Rap, French Rap
4.0509	Hardcore Rap, East Coast Rap, Alternative Rap, Trip-Hop, Club/Dance
22.0509	Pop, Dance-Pop, Adult Contemporary, Teen Pop, Contemporary R&B
0.1014	Southern Rap, Hardcore Rap, Pop-Rap, Gangsta Rap, East Coast Rap
1.1014	Pop, Alternative/Indie Rock, Club/Dance, Indie Electronic, EDM
3.1014	Hardcore Rap, East Coast Rap, Political Rap, Golden Age, Heavy Metal
4.1014	Club/Dance, Jungle/Drum'n'Bass, Dubstep, Garage, House
12.1014	Midwest Rap, Hardcore Rap, Left-Field Rap, Alternative Rap, French Rap
23.1014	Club/Dance, House, EDM, Dubstep, Pop-Rap
1.1519	Pop, Dance-Pop, Alternative Rap, Left-Field Rap, Acappella
2.1519	Club/Dance, Jungle/Drum'n'Bass, Dubstep, House, Garage
5.1519	Polish, Hardcore Rap, Central European Traditions, East Coast Rap, Political Rap
10.1519	West Coast Rap, Contemporary R&B, Left-Field Rap, Pop-Rap, Gangsta Rap
11.1519	Club/Dance, EDM, Pop, House, Downtempo

is “Amen, Brother” released by The Winstons. This song has been sampled the most across all communities on path. Other prominent drum breaks, such as those from Lyn Collins’ “Think (About It),” Bobby Byrd’s “Hot Pants,” and Incredible Bongo Band’s “Apache,” are commonly found on each community’s list of the top samples. Thus, the drum break, utilized primarily in breakbeat-based music, is fixed and can be seen to have been utilized throughout time.

Community 1\_9599 is notable as well. The predecessors of Community 1\_9599 refer to those that represent the old-school hip-hop style, including samples such as Beside’s “Change the Beat (Female Version)” and James Brown’s “Funky Drummer”. Specifically, “Change the Beat (Female Version)” is the most sampled song in the world and can be considered an iconic old-school hip-hop sample utilized in DJ scratch performances [13]. Community 1\_9599 represents Turntablism and underground hip-hop styles focused on famous hip-hop DJs (“Turntablism”, ‘Underground Rap’) and illustrates the success of Turntablism music in the late 1990s [12]. The successors of 1\_9599 can be considered to be the genres that retain the essence of classic hip-hop. Therefore, it is notable that the path following Community 2\_0509 is dominated by Polish hip-hop artists [14]. This suggests that in recent years, artists who inherit the old-school hip-hop style have emerged more frequently in European countries such as Poland than in the United States, the birthplace of hip-hop.

Also identified is a path connecting 0\_8084 → 14\_8589 → 11\_9094 → 14\_9599 (bottom left of the figure). These communities are synonymous with reggae music. Reggae is also a sampling-based music genre, like hip-hop and electronic music, primarily sampling prior reggae music [15]. Before 1980 when hip-hop was born, reggae was the major music type. Reggae songs such as “Funaany” by Admiral Bailey, “Full Up”, “Drum Song” by Jackie Mittoo, and “Real Rock” by Sound Dimension were frequently sampled in each community. The fact that this path was cut off in the 1995-1999 generation suggests that the sampling reggae became much less popular in the 00’s.

## 4 Conclusions

In this study we investigated the community structure of sample-based music post-1980 using bipartite network analysis. We constructed the sampling networks for each of the eight five year-long generations and then conducted community detection. The communities established in this manner were shown to be representing a style.

Our analysis focused on two significant ample-based music genres, electronic music and hip-hop. Jungle and Drum’n’Bass are the subgenres of electronic music that use a style that incorporates breakbeats. Since there are only a few types of breakbeats, we showed that the Jungle / Drum’n’Bass communities in each generation have strong ties. We also observed that the Old-school hip-hop style, which has persisted since the 1980s, is diverging into multiple branches and that non-European artists, such as those from Poland, continue to use this style into the 2000s.

In the future we intend to increase our understanding of sample-based music styles by conducting a more comprehensive analysis of the artists and genre information of the derived communities. We may also vary the time unit used to divide generations in order to conduct analyses on a different scale. Moreover, by varying the edge weight

threshold when visualizing the similarity network of consecutive primary communities, we may observe the evolution of sample-based music genres in more detail. Lastly, we could investigate generation-skipping transmission of musical styles by analyzing the similarity between two communities separated by more than one generation, which could show how the phenomenon of “revival” of musical styles occurs.

## Acknowledgements

This research was supported by KAIST Post-AI Research Grant and Korea Creative Content Agency funded by the Korean Government (RS-2023-00270043).

## References

1. Roads, C. *Early Electronic Music Instruments: Time Line 1899-1950*. Computer Music Journal 20, no. 3 (1996).
2. Williams, Justin A. ‘You Never Been on a Ride like This Befo’: Los Angeles, Automotive Listening, and Dr. Dre’s ‘G-Funk’. Popular Music History 4, no. 2 (2010).
3. Hockman, J., Davies, M. & Fujinaga, I. *One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass*. In Proceedings of the 13th International Society for Music Information Retrieval (ISMIR), Porto (2012).
4. Bryan, N. & Wang, G. *Musical Influence Network Analysis and Rank of Sample-Based Music*. Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011. pp. 329-334 (2011).
5. Youngblood, M. *Cultural transmission modes of music sampling traditions remain stable despite delocalization in the digital age*. PloS one, 14(2), e0211860 (2019).
6. Hess, M. *Was Foucault a Plagiarist? Hip-Hop Sampling and Academic Citation*. Computers and Composition 23, no. 3. pp. 280-295 (2006).
7. Zhao, D. & Strotmann, A. *Evolution of Research Activities and Intellectual Influences in Information Science 1996-2005: Introducing Author Bibliographic-Coupling Analysis*. Journal of the American Society for Information Science and Technology 59, no. 13. pp. 2070–86 (2008).
8. Zhou, T., Ren, J., Medo, M. & Zhang, Y. C. *Bipartite Network Projection and Personal Recommendation*. Physical Review E 76, no. 4 (2007).
9. Barber, M. J. *Modularity and Community Detection in Bipartite Networks*. Physical Review E 76, no. 6 (2007).
10. Pesantez-Cabrera, P. & Kalyanaraman, A. *Detecting Communities in Biological Bipartite Networks*. Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics. (2016).
11. He, X., Gao, M., Kan, M. Y., & Wang, D. *Birank: Towards ranking on bipartite graphs*. IEEE Transactions on Knowledge and Data Engineering, 29(1), 57-71 (2016).
12. Greasley, A. E., & Prior, H. M. *Mixtapes and turntablism: DJs’ perspectives on musical shape*. Empirical Musicology Review, 23-43 (2013).
13. Hansen, K. F. *The Basics of Scratching*. Journal of New Music Research 31, no. 4. pp. 357–65 (2002).
14. Aniskiewicz, A.G. *Cultural Remix: Polish Hip-Hop and the Sampling of Heritage* Ph.D. dissertation, University of Michigan (2019).
15. *Dub: Soundscapes and Shattered Songs in Jamaican Reggae*. Choice Reviews Online 45, no. 03 (2007).

# Algorithms for Roughness Control Using Frequency Shifting and Attenuation of Partial in Audio

Jeremy Hyrkas\*

University of California San Diego  
jhyrkas@ucsd.edu

**Abstract.** Though synthesis algorithms frequently use parameters to change the produced sound, it is not always the case that these parameters have a direct (or intuitive) correlation to the change made to the perceptual attributes—the more meaningful sound descriptors for the listener and/or musician. In this work we explore two strategies by which a perceptual descriptor, roughness, can be parameterized directly on a scale, much like how interactive sound allows for control of pitch and/or loudness. Here, roughness (often tied to dissonance) is controlled by changing either the frequency or amplitude of partials that lie within a critical band. Audio examples are provided to demonstrate use in audio mixing, sound (re)synthesis and audio effects, with two implementations made available: one for offline use and another for real-time interactive synthesis using Max/MSP.

**Keywords:** auditory roughness, additive synthesis, assistive audio production

## 1 Introduction

This work presents two algorithms for controlling the roughness of sound by adjusting the frequency or amplitude of individual partials or harmonics of the sound. Auditory roughness is a perceptual feature of sound that is often linked with an experience of dissonance, making it a particularly salient sonic parameter that a composer may want to manipulate. Roughness has historically been studied in relation to musical consonance with regards to interval choice and tuning. Consequently, previous approaches that manipulate the roughness of sound often utilize pitch shifting. The approaches described here avoid pitch shifting, favoring subtle changes to the spectrum to control roughness while maintaining as much of the original timbre as possible. A brief history of roughness and musical applications is presented in Section 2. Algorithms for roughness control are described generally in Section 3. Finally, implementations for audio files and additive synthesis are presented in Section 4. Implementations of the algorithms and audio examples are available online.

---

\* Special thanks to Miller Puckette and Tamara Smyth for advising portions of this project.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

## 2 Auditory roughness, musical consonance and tuning

Auditory roughness is a psychoacoustic and physical phenomena that occurs when sinusoidal components of sound fall within the critical band of the ear. Sinusoidal components that are very close in frequency (i.e. separated by less than 10 Hz), are heard as a single partial with slow beating, a special case of amplitude modulation where the modulating sinusoid (the sound's amplitude envelope) is sufficiently low in frequency that the sound is brought in and out of prominence on a perceptible time scale. While such sounds are generally considered to be *consonant*, increasing the modulation frequency or, equivalently, the frequency difference between the two components, increases the rate of beating, eventually leading to the listener being no longer able to track the beats; the tone takes on a more steady amplitude but with a distinct quality known as *roughness*, an attribute often associated with *dissonance*. Increasing the modulation frequency still, so that the frequency separation approaches the critical bandwidth, the listener begins to recognize the sound as separate tones, at which point discomfort decreases and eventually disappears. Below, we briefly present the history of the study of auditory roughness, its relation to musical consonance and its use in music systems.

### 2.1 Psychoacoustic models of roughness

Auditory roughness relates the experience of dissonance to the presence of sound partials that fall within a critical band of the human ear. Theories relating musical consonance and distance between sound partials date at least as far back as Helmholtz, who credited consonance to the lack of beating partials when harmonic instruments play intervals related by integer ratios [1].

Plomp and Levelt tested the perception of dissonance and proposed a general model of dissonance of sinusoidal tones based on critical bandwidth [2]. The resulting data informed a model by Sethares [6] who calculates the dissonance between two partials with frequencies  $f_1$  and  $f_2$  having amplitudes  $a_1$  and  $a_2$ , respectively, as

$$r(f_1, a_1, f_2, a_2) = a_1 a_2 \cdot [e^{-b_1 s x} - e^{-b_2 s x}], \quad (1)$$

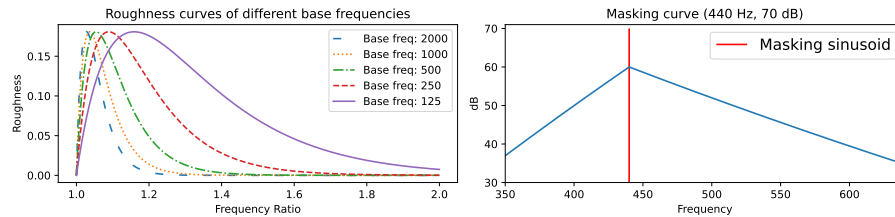
where  $b_1 = 3.5$  and  $b_2 = 5.75$  are chosen by fitting  $sx$  to Plomp and Levelt's model.

In (1),  $x = f_2 - f_1$  (where  $f_1 < f_2$ ) is the frequency difference between the two partials and  $s$  scales the frequency difference to fit Plomp and Levelt's standard curve, which was originally plotted with respect to critical bandwidth. The frequency difference is scaled by

$$s = \frac{d^*}{s_1 f_1 + s_2}, \quad (2)$$

where  $d^* = 0.24$  is the position of maximum dissonance and  $s_1 = 0.021$  and  $s_2 = 19$  are obtained using a least-squares fit. Equation 1 accounts for partials having different amplitudes, so softer components contribute less to dissonance. Figure 1 demonstrates how (1) varies across the frequency domain. We will later use (1) to calculate the roughness of sound partials and choose new frequencies to reduce roughness.

Kameoka and Kuriyagawa extended Plomp and Levelt's experiments and investigated the role of masking [3]. The perception of dissonance when two partials have different amplitudes was found to differ from the pattern found in masking curves. While



**Fig. 1. Left:** the roughness of two partials with equal amplitude by their frequency ratio (1). **Right:** a plot of (3), a simple model for estimating the masking curve of a tone given its frequency and sound pressure level in dB. Tones on or below the sloped lines are masked by the original.

a louder sinusoid will more easily mask a quieter sinusoid of higher frequency, pairs of sinusoids were found to be more dissonant when the tone with lower frequency was louder than the tone of higher frequency. Nevertheless, perceived dissonance dropped when one was completely masked by the other, suggesting that masking plays a role in perceived dissonance. A simple model for estimating the masking curve of a partial based on difference in Barks [8] can be computed as

$$mask(x|f, dB) = \begin{cases} dB - 10 - 27 [B_{Hz}(f) - B_{Hz}(x)], & \text{if } x < f \\ dB - 10 - 15 [B_{Hz}(x) - B_{Hz}(f)], & \text{if } x \geq f \end{cases}, \quad (3)$$

which we will later use to computationally control roughness by amplitude changes informed by masking.  $B_{Hz}$  converts from Hz to Bark using Trautmüller's model [4]. A plot of this masking curve is shown on the right side of Figure 1.

Concepts of consonance and dissonance have many definitions in the context of music. The sensation caused by beating partials in a critical band, which can also be described as the presence of amplitude modulation within a critical band, is now referred to as *roughness* in psychoacoustics to disambiguate it from compositional definitions. Based on the work of Plomp and Levelt, more sophisticated models of roughness have been developed. Sethares's model of Plomp and Levelt accounts for the amplitude of partials (see (1)). Vassilakis [14] additionally accounted for the role of amplitude modulation in an extension to (1)<sup>1</sup>. The roughness of a signal is usually defined as the sum of roughness between all pairs of partials.

## 2.2 Applications to tuning, mixing and synthesis

While roughness can be measured outside the context of musical applications, it has been supposed to be related to musical consonance [1] and has been used as a metric in tuning, mixing and composition. Sethares developed the Adaptive Tuning algorithm, which adjusts the fundamental frequencies of notes based on their spectra to minimize

<sup>1</sup> Vassilakis's model more thoroughly accounts for the role of loudness and amplitude modulation to roughness. However, this equation is more complex and was not found to improve synthesis in this study, so the simpler model from Sethares is used.

the expected roughness of the sound. Sethares's algorithm can be approximated in real-time on spectra that are known ahead of time [9] or computed exactly on mixtures of spectra in the full implementation [10]. Adaptive Tuning is based on minimizing (1) with respect to fundamental frequency (as opposed to individual partial frequency) and has been used in other musical contexts (for example, to control the pitch of a Theremin implementation in real-time using Pure Data [13]).

The roughness of an audio mix can be a useful metric to measure and control for in sound engineering. Vassilakis used an extension of (1) to analyze and annotate the roughness of sound files [14]. Real-time roughness estimators have been implemented in Pure Data [15][16], each of which use sinusoidal modeling to estimate partials of an incoming mix to determine the roughness. While the ability to analyze roughness in real-time can be helpful, these algorithms crucially will have difficulty detecting sinusoidal peaks that are nearby (i.e., less than 20 Hz difference) using solely FFT-based analyses. Vassilakis's software, on the other hand, uses frequency reassignment methods to obtain finer resolution of nearby partials.

The roughness of a sound can be used as a parameter to be increased or reduced in some synthesis or resynthesis methods. Molina et al. modeled audio signals of chords using sinusoid-plus-residual analysis [5] and reduced beating partials in resynthesis by forcing partial frequencies be an integer multiple of one of the fundamental frequencies of the chord [18]. Roughness was found to be the most impactful feature when pitch shifting one track to be consonant with another in the context of DJing [19][20]. When synthesizing impact sounds using additive synthesis, roughness has been used as a parameter to create convincing sound examples [17]. Finally, Park et al. created software to manipulate psychoacoustic features such as spectral slope or inharmonicity when composing using sound material [12]; auditory roughness control was not implemented in the software but may be another candidate for compositional control.

### **3 Roughness control by individual partial adjustment**

Previous approaches to roughness control have focused on changing fundamental frequencies (and therefore all partials) of notes [9], pitch shifting audio tracks [20], or quantizing partial frequencies to strict integer harmonics [18]. An unexplored approach is the selective adjustment of partials independent of tuning to control roughness while leaving the majority of the signal intact. Such an approach gives the composer options for reducing or increasing roughness in subtle ways that preserve the essence of the original material as much as possible. The methods developed here explore this approach in both a real-time and offline process. Real-time methods change either the frequency or amplitude of additive synthesis parameters in the Max/MSP computer music environment, while offline approaches selectively filter and resynthesize certain partials in audio files while preserving the remaining portions of audio, which may produce higher fidelity output than methods that use sinusoidal resynthesis [18]. Section 3.1 presents an algorithm for frequency adjustment and Section 3.2 presents an algorithm for amplitude adjustment. Offline and online implementations are described in Section 4.

### 3.1 Changing roughness by partial tuning

Here we define *frequency bashing*, a greedy algorithm for reducing or increasing roughness by changing the frequency of partials that lie within a critical band. Pairs of partials that overlap in time and cause roughness are selected and iteratively changed to adjust roughness. A partial may contribute to roughness in more than one pair of nearby partials, so as the algorithm iterates, partials that have already been adjusted are skipped.

In the algorithm, partials pairs are analyzed for their contribution to the overall roughness of the sound using (1), with the most rough partial pairs processed first. The quieter partial has its frequency changed to be the frequency that either minimizes or maximizes roughness with respect to the louder partial within a predefined distance. We restrict movement to a specified range in Barks due to the the shape of (1), which would otherwise cause partials to always move to an identical frequency when bashing for minimized roughness; we also aim to keep partials within their original critical band to avoid drastically altering the nature of the original sound. Based on the maximum dissonance of (1), distance in Barks is typically set to between 0.05 and 0.4 in our experiments, but the distance is exposed as a user parameter.

Frequency bashing for consonance is computed mathematically as

$$f_2^* = \arg \min_{f_{\min} \leq f^* \leq f_{\max}} r(f_1, a_1, f^*, a_2) \quad (4)$$

with constraints defined as

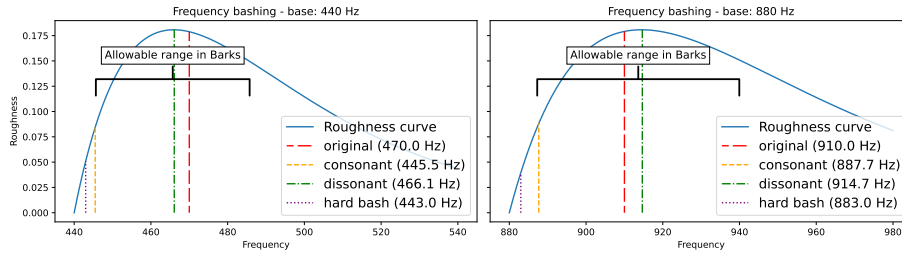
$$f_{\min}, f_{\max} = Hz_B(B_{Hz}(f_1) + B_L), Hz_B(B_{Hz}(f_1) + B_H) . \quad (5)$$

$Hz_B$  and  $B_{Hz}$  convert from Bark to Hz and vice versa [4], and  $B_L$  and  $B_H$  define the allowable distance in Barks. In (4) and (5),  $f_1$  is the louder partial whose frequency remains constant, while frequency  $f_2$  is the quieter partial whose frequency will be bashed to a new value. When increasing roughness instead of decreasing, the argmin operation in (4) is replaced by argmax. Equation 5 assumes  $f_1 < f_2$ ; when  $f_1 > f_2$  the Bark range is defined below  $f_1$  instead of above.

Figure 2 shows potential adjustments for sinusoid pairs. Note that for partials with identical differences in frequency, their position along the roughness curve and solutions for maximum consonance and dissonance will change depending on their location in the frequency domain. The partial pairs depicted (440 Hz and 470 Hz versus 880 Hz and 910 Hz) have unequal differences in their maximally consonant and dissonant solutions, even though they begin with the same difference of 30 Hz per pair.

Another option is *hard-bashing*, where the quieter partial is adjusted to be a specified difference in Hz from the unchanged partial. An advantage of hard-bashing is that when multiple partials are adjusted within a sound, they will have identical frequency difference from their neighboring partial, creating a slow-beating (tremolo) effect but only in certain frequency ranges of the signal. However, consonance and dissonance as defined by roughness models are disregarded. In Figure 2, hard-bashing partials to have a difference of 3 Hz maintains the equal difference between partial pairs after bashing and both new frequencies result in lower roughness. However, they now fall on different positions along the roughness curve due to their different critical bandwidths.





**Fig. 2.** Frequency bashing two pairs of partials with frequencies (440 Hz, 470 Hz) on the left and (880 Hz, 910 Hz) on the right. The partials with higher frequencies are adjusted. The original frequency of the higher partial is depicted in the long-dashed red line, the most consonant in the specified allowable range of movement in Barks is shown in the short-dashed yellow line and the most dissonant in the dash-dotted green line. Hard-bashing the partial to a difference of 3 Hz is shown in the dotted purple line. Hard-bashing ignores the roughness curve and allowable range.

Frequency bashing is implemented as described for offline processing of audio files, with further modifications described in Section 4.1. Simplifications made to this algorithm for real-time implementation are described in Section 4.2.

### 3.2 Changing roughness by amplitude adjustment

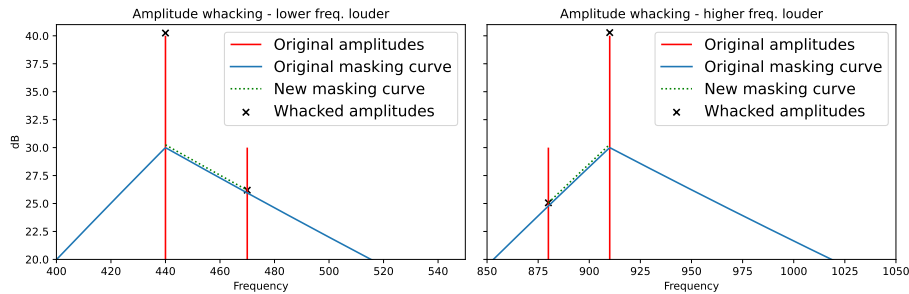
Another approach to roughness control is changing the amplitude of partials, as the absolute and relative loudness of partials contributes to roughness. A simple technique would be to lower the amplitude of the quieter partial in a pair that contributes roughness to the sound. However, lowering the amplitude of a partial will reduce the power of the signal. Instead, the quieter partial must have its amplitude decreased while the louder partial’s amplitude is increased to maintain the original signal power. This seesaw effect of amplitude adjustments may remind the reader of the children’s arcade game “whack-a-mole,” and is therefore named *amplitude whacking*.

Whacking can be performed on a scale between 0.0 (no change) and 1.0 (maximum amplitude change). The algorithm maximally adjusts amplitudes so that the quieter partial is fully masked by the louder partial, as masking plays a role in perceived roughness [3]. When adjusting the amplitudes of a pair of partials to reduce roughness, the resulting difference in dB ( $\Delta_{dB}$ ) should be the whacking percentage of the masking threshold specified by (3); additionally, the power of the signal should be retained. If  $a_1$  is the amplitude of the louder partial and  $a_2$  the amplitude of the quieter partial, these constraints are specified as

$$20 \log_{10}(a_1^*) - 20 \log_{10}(a_2^*) = \Delta_{dB}, (a_1)^2 + (a_2)^2 = (a_1^*)^2 + (a_2^*)^2 \quad (6)$$

respectively, where  $a_1^*$  and  $a_2^*$  are the new amplitudes of the partials. Solving the system of equations algebraically leads to the solutions:

$$a_2^* = \sqrt{\frac{(a_1)^2 + (a_2)^2}{1 + 10^{\frac{\Delta_{dB}}{10}}}}, a_1^* = \sqrt{(a_1)^2 + (a_2)^2 - (a_2^*)^2}. \quad (7)$$



**Fig. 3.** Amplitude whacking pairs of partials with frequencies (440 Hz, 470 Hz) on the left and (880 Hz, 910 Hz) on the right. Original amplitudes are shown as red solid lines, with whacked amplitudes depicted as black X's. Despite each pair being equidistant in Hz, the example on the right requires more attenuation of the quieter partial due to the asymmetry of the masking curve.

Amplitude whacking proceeds identically to frequency bashing, with pairs of partials that contribute the most roughness processed first and partials that have already been adjusted skipped in later iterations. Figure 3 shows potential adjustments for amplitudes of partial pairs to reduce the roughness of a sound. Offline and real-time implementations of the algorithm are further described in Sections 4.1 and 4.2, respectively.

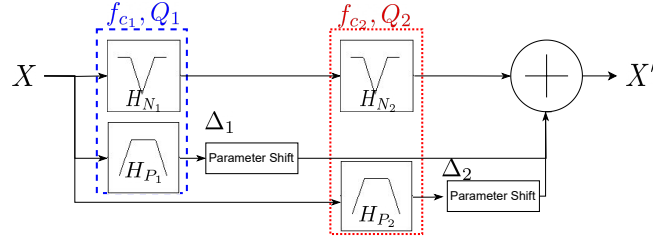
## 4 Algorithm Implementations

The algorithms for frequency bashing and amplitude whacking can be performed offline on mixes of audio files or in real-time in additive synthesis. Each approach requires certain tweaks and optimizations. The implementations are described below, followed by a discussion of potential applications with accompanying examples available online.

### 4.1 Offline implementations on audio files

Auditory roughness occurs most when partials fall very close to each other (i.e., on the order of 20-40 Hz difference). Finding partials with such resolution in a single audio file is difficult using only FFT-based methods [14], and isolating one partial in a digital filter without affecting the nearby partial also presents issues. Taking these limitations into account, and following previous application of roughness to combinations of sounds [9][18][19], we use the algorithms defined in Section 3 to control the roughness of audio files that are to be combined in a mix.

Given a collection of audio files, each file is analyzed using sinusoidal modeling [5] to find the top  $N$  partials of each frame of audio.  $N$  can be a small number (i.e., on the order of 10) because the sinusoidal tracks will not be used for resynthesis and are instead used to identify the most prominent partials at a given time. When a partial from signal  $x_i$  and a partial from signal  $x_j$  overlap in time and cause roughness, the partials are collected as candidates for frequency bashing or amplitude whacking with new frequencies or amplitudes determined using (4) and (7).



**Fig. 4.** Partials of a signal  $x$  are adjusted by removing them using notch filters, and in parallel, isolating them from  $x$  using amplitude-complimentary peaking bandpass filters. Complimentary filters share the same center frequency  $f_{c_k}$  and quality factor  $Q_k$ . The output of the peaking filters are processed to adjust either the frequency or the amplitude by some amount  $\Delta_k$ .

To adjust partials of a signal  $x$ , the offending partials are removed from  $x$  and in parallel isolated, altered, and added back in. The mean frequency and frequency range of a partial are used to set the center frequency  $f_c$  and quality factor  $Q$  of an amplitude-complimentary pair of  $H_N$ , a notch filter, and  $H_P$ , a peaking bandpass filter, such that

$$H_N(f_c, Q) \cdot X + H_P(f_c, Q) \cdot X = X . \quad (8)$$

If a signal has  $k$  partials to be adjusted, the output signal  $x'$  will be computed as

$$X' = X \cdot \prod_{i=1}^k H_{N_i}(f_{c_i}, Q_i) + \sum_{i=1}^k \Delta_i (H_{P_i}(f_{c_i}, Q_i) \cdot X) , \quad (9)$$

where  $\Delta_i$  changes either the amplitude or frequency of the partial isolated in the peaking filter.  $H_N$  and  $H_P$  are standard second-order IIR filters defined in [7] and implemented as the *iirnotch* and *iirpeak* functions in MATLAB and scipy.

Figure 4 depicts the audio processing of partials. Forward and backward filtering is used for zero-phase filtering. When frequency bashing, the output of a peaking filter is frequency shifted using single sideband modulation; when amplitude whacking, gain is applied to the output so that the mean amplitude matches the intended value. The processed signals are then added to the output of the series of notch filters. Additionally, cross-fades are applied so that filtered signals with adjusted partials are only heard during the lifetime of the partial. When partials that cause roughness are not present, the original sound files are used unaltered.

#### 4.2 Real-time implementations for additive synthesis parameters

Frequency bashing and amplitude whacking can be performed more straightforwardly on additive synthesis parameters, as no audio analysis is inherently required. In additive synthesis, frequency and amplitude pairs are used to control an oscillator bank, with each oscillator receiving one frequency and amplitude per synthesis step. Roughness control on additive synthesis parameters can be more easily performed in real-time unlike the previous methods for audio processing, although some simplifications to the algorithms are required to reduce computational load.

Two externals for the Max/MSP computer music environment were created to control roughness of additive synthesis parameters. The externals are control rate objects that take as input a list of frequency/amplitude pairs and output one list of frequencies and one list of amplitudes. In the *basher* object, frequencies are adjusted using (4) while amplitudes are passed through unchanged, while amplitudes are adjusted by the *whacker* object using (7) with the frequencies unchanged. The outputs can be connected to multichannel Max objects for additive synthesis (see the associated code for examples). The input list of sinusoidal parameters can come from an analysis-resynthesis system for sinusoidal modeling [5][11], but composers are free to use any method for generating sinusoidal parameters. Other algorithmic parameters include the Bark range of search and adjustment (see Equation 5), a toggle to change from decreasing roughness to increasing roughness, and the percent of movement from the original sinusoidal parameters to the adjusted parameters. These parameters can be fixed or adjusted by the user algorithmically or manually on-the-fly as a musical effect.

Equation 1 is expensive to compute in real-time for every potential pair-wise combination for every frame. When searching for potential candidates for parameter adjustment in the Max objects, partial pairs are sorted by frequency instead of by overall roughness to reduce computational load. Pairs are processed in ascending order of frequency, short-cutting whenever pairs start to fall outside of the defined Bark range. In many cases there will not be many partials lying within a fraction of a critical band of each other, but in general results may differ from the offline implementation.

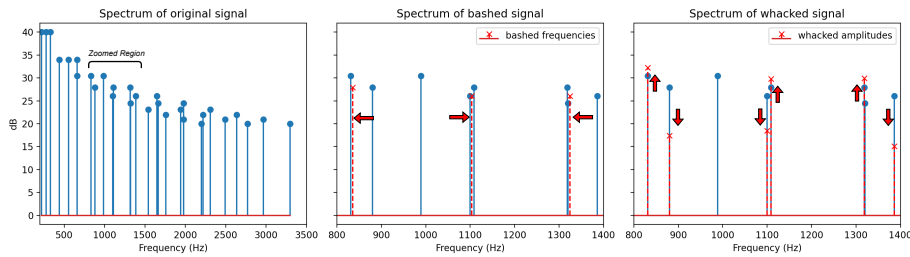
When changing amplitudes in Max, the user parameter for whacking amount maximally transfers all power from the quietest partial to the loudest partial to avoid computing (3) at every frame. As a result, the audible effect of the parameter will stop before reaching the maximum value once quieter partials are fully masked. This issue could be avoided by precomputing and quantizing masking curves. Finally, while offline methods change the parameters of partials across time, sinusoidal parameters here are adjusted at every frame in a memoryless fashion. This change reduces computation and computer memory necessary to track previous changes, but makes the objects more susceptible to rapid fluctuation in the case where two nearby partials are nearly the same amplitude and fluctuate between which one is louder.

### 4.3 Audio examples and observations

Examples of frequency bashing and amplitude whacking on audio files and additive synthesis parameters are available on the associated supplemental website.<sup>2</sup> The basic algorithms are demonstrated on the sinusoid examples in Figures 2 and 3 where the effect of each algorithm and parameter choice is most obvious. More complex examples showcase various use cases of the algorithms in more realistic musical contexts.

The effects of frequency bashing and amplitude whacking on synthesized notes is shown in Figure 5. Each algorithm is applied to an equal tempered major triad of sawtooth waves with 10 partials. The original spectrum is shown on the left side of the figure, alongside the adjusted partials in each algorithm in a zoomed region of the spectrum. Three pairs of partials are found to be nearby in a critical band and contribute

<sup>2</sup> <https://jeremyhyrkas.com/cmmr2023>

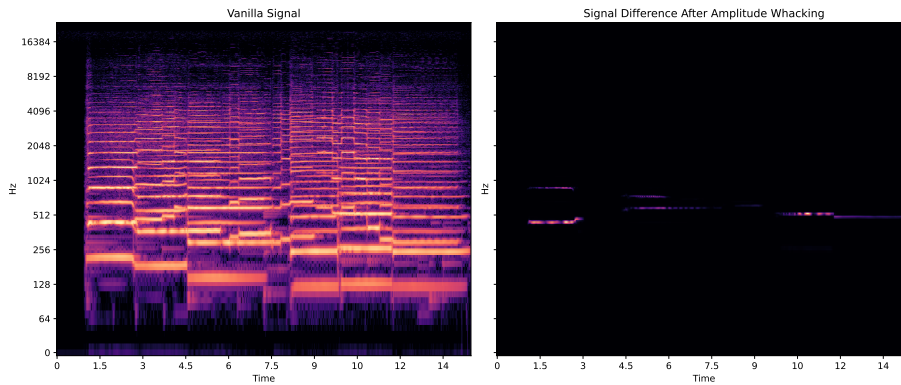


**Fig. 5.** **Left:** the spectrum of a major triad of sawtooth waveforms in equal temperament tuning. **Center:** zoomed region of the spectrum after frequency bashing. Three partials have been moved. **Right:** the same zoomed region of the spectrum after amplitude whacking. Three pairs of nearby partials (six partials total) have had their amplitudes adjusted.

to roughness. Although the algorithms only adjust a handful of partials, there is a noticeable difference in beating between the original and processed examples while the characteristic of the sound is largely intact. In contrast, an accompanying audio example demonstrates the chord in just intonation, which an algorithm such as Adaptive Tuning [9] may offer as a solution. The retuned example also contains less roughness than the original but sounds fundamentally different. This example demonstrates the difference in philosophy between our methods for roughness control versus tuning-based approaches, as the goal of our algorithms is to only change the perceived auditory roughness while maintaining as much of the original signal as possible.

More examples of roughness reduction include frequency bashing a slightly detuned major chord and amplitude whacking a horn line. The horn line is depicted in Figure 6, with a spectrogram of the original on the left and a spectrogram of the difference signal after amplitude whacking on the right. This example again demonstrates the very subtle changes made to the signal, as well as the preservation of the original signal during periods of time where no roughness is present. Another example demonstrates the effect of filtering a partial that contributes to roughness without resynthesizing it back in with a different frequency or amplitude. This approach will reduce the power of the signal and can cause the sound to feel hollow when too many partials are removed, but can be effective on audio examples where very few partials contribute to roughness. These examples demonstrate a potential use case of reducing roughness when mixing a track without the use of retuning or manually intensive EQing.

Two examples are presented where frequency bashing is used to introduce more roughness into a sound mix than was originally present. A sawtooth major seventh chord is used, as is a recording of a choir singing a chorale. This use case shows frequency bashing as an audio effect that may be useful for a composer who wishes to introduce dissonance without detuning or modulating the entire signal. Finally, an example is presented using hard-bashing where pairs that contribute roughness are moved to be exactly 3 Hz apart from one another. The effect is similar to a tremolo audio effect, but the modulation is only heard when roughness is present and only in some parts of the frequency domain. All examples described can be found on the accompanying website.



**Fig. 6. Left:** a spectrogram of a dynamic horn line featuring audio of three players performing. **Right:** the spectral difference of an amplitude whacked version of the horn line and the original. Amplitudes of partials are only adjusted in areas of roughness, after which point the original signals are faded back in. This example is time-varying and demonstrates subtle changes that achieve a reduction in roughness without retuning any notes.

The Max externals described in Section 4.2 are demonstrated in videos using harmonic and inharmonic drones, as well as a dynamic sinusoidal reconstruction of the horn line described previously. The reconstruction requires playback of an offline analysis using the SPEAR modeling software [11]. Preliminary versions of these Max externals that control the amplitude, frequency and spatial panning of partials to increase or reduce auditory roughness were used by the author to create a composition that musically investigates roughness, tuning and listening tests.<sup>3</sup>

## 5 Conclusion and Future Work

We present two algorithms for controlling auditory roughness by targeted sound partial adjustment. Frequency bashing modifies the frequencies of neighboring partials, while amplitude whacking modifies their amplitude. Considerations are made based on previous work modeling listener perception of roughness, auditory masking, and previous approaches to roughness reduction. Offline implementations of both algorithms are provided for audio files intended to be mixed in time<sup>4</sup>, and control-rate objects for additive synthesis are provided for the Max/MSP environment<sup>5</sup>. The accompanying audio examples demonstrate potential use cases in mixing and composition.

The algorithms reduce roughness as calculated by the roughness models by their definition, but listening tests would be beneficial to confirm the intended effect on listeners, as the perception of consonance and dissonance is affected by context in ways not accounted for in these models. Additionally, a VST implementation of the audio

<sup>3</sup> The piece described here was submitted for consideration to CMMR 2023.

<sup>4</sup> <https://github.com/jhyrkas/sms-tools-audio-bashing>

<sup>5</sup> [https://github.com/jhyrkas/basher\\_max](https://github.com/jhyrkas/basher_max)

processing algorithms would assist sound engineers in incorporating them into their workflow. While these algorithms do not currently work on incoming audio streams, a plug-in implementation may be beneficial for use on processed tracks before final mixing and mastering. Finally, simplifying certain portions of the audio implementations may make them viable for use in real-time applications.

## References

1. Helmholtz, H.: On the Sensations of Tone as a Physiological Basis for the Theory of Music. Longmans, Green and Co. (1877)
2. Plomp, R., Levelt, W.: Tonal Consonance and Critical Bandwidth. In: The Journal of the Acoustical Society of America, 38 #4 (1965), 548–560
3. Kameoka, A., Kuriyagawa, M.: Consonance Theory Part I: Consonance of Dyads. In: The Journal of the Acoustical Society of America, 45 #6 (1969), 1451–1459
4. Trautmüller, H.: Analytical Expressions for the Tonotopic Sensory Scale. In: The Journal of the Acoustical Society of America, 88 #1 (1990), 97–100
5. Serra, X.: A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition. Dissertation, Stanford University (1990)
6. Sethares, W.: Local Consonance and the Relationship Between Timbre and Scale. In: The Journal of the Acoustical Society of America, 94 #3 (1993), 1218–1228
7. Orfanidis, S. J.: Introduction to Signal Processing. Prentice Hall Upper Saddle River, 1996
8. Lagrange, M., Marchand, S.: Real-time Additive Synthesis of Sound by Taking Advantage of Psychoacoustics. In: Proc. of the International Conference on Digital Audio Effects (DAFx) (2001)
9. Sethares, W.: Real-time Adaptive Tunings Using Max. In: The Journal of New Music Research, 31 #4 (2002), 347–355
10. Sethares, W.: Tuning, Timbre, Spectrum, Scale. Springer Science & Business Media, 2005
11. Klingbeil, M.: Software for spectral analysis, editing, and synthesis. In: Proc. of the International Computer Music Conference (ICMC) (2005)
12. Park, T. H., Biguenet, J., Li, Z., Richardson, C., Scharr, T.: Feature Modulation Synthesis (FMS). In: Proc. of the International Computer Music Conference (ICMC) (2007)
13. Porres, A., Manzolli, J.: A Roughness Model in Pd for an Adaptive Tuning Patch Controlled by Antennas. In: Proc. of the Pure Data Convention (2007)
14. Vassilakis, P., Fitz, K.: SRA: A Web-based Research Tool for Spectral and Roughness Analysis of Sound Signals. In: Proc. of the Sound and Music Computing Conference (SMC) (2007)
15. MacCallum, J. and Einbond, A.: Real-time Analysis of Sensory Dissonance. In: Computer Music Modeling and Retrieval. Sense of Sounds. 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 2007, Revised Papers (2007)
16. Villegas, J., Cohen, M.: “Roughometer”: Realtime Roughness Calculation and Profiling. In: Proc. of the Audio Engineering Society Convention 125 (AES) (2008)
17. Aramaki, M., Gondre, C., Kronland-Martinet, R., Voinier, T., Ystad, S.: Thinking the Sounds: An Intuitive Control of an Impact Sound Synthesizer. In: Proc. of the 15th International Conference on Auditory Display (ICAD) (2009)
18. Molina, E., Barbancho, A., Tardón, L., Barbancho, I.: Dissonance Reduction In Polyphonic Audio Using Harmonic Reorganization. In: The IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22 #2 (2013), 325–334
19. Gebhardt, R., Davies, M., Seeber, B.: Harmonic Mixing Based on Roughness and Pitch Commonality. In: Proc. of the International Conference on Digital Audio Effects (DAFx) (2015)
20. Gebhardt, R., Davies, M., Seeber, B.: Psychoacoustic Approaches for Harmonic Music Mixing. In: The Journal of Applied Sciences, 6 #5 (2016), 123–143

# Bridging the Rhythmic Gap: A User-Centric Approach to Beat Tracking in Challenging Music Signals

António Sá Pinto<sup>1,2</sup> and Gilberto Bernardes<sup>1,2</sup>

<sup>1</sup> Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

<sup>2</sup> INESC TEC, Porto, Portugal  
asapinto@fe.up.pt

**Abstract.** Deep-learning beat-tracking algorithms have made significant advancements in recent years. However, despite these advancements, challenges persist when processing complex musical examples, which are often under-represented in training corpora. Expanding on our prior work, this paper delves into our user-centric beat tracking approach by subjecting it to highly challenging musical pieces. We probe the adaptability and resilience of our methodology, illustrating its ease of integration with state-of-the-art techniques through minimal user annotations.

The chosen samples, namely, Uruguayan *Candombe*, Colombian *Bambuco*, and Steve Reich's *Piano Phase*, not only demonstrate our method's efficacy but also exemplify challenging rhythmic dissonance effects such as *polyrhythms*, *polymetres*, and *polytempi*. Thereby, we demonstrate the applicability of our human-in-the-loop strategy in the domain of Computational Ethnomusicology, confronting the prevalent Music Information Retrieval (MIR) constraints found in non-Western musical scenarios. Our approach enables notable improvements in terms of the F-measure, ranging from 2 to 5 times the current state-of-the-art performance. In terms of the annotation workflow, these results translate into a minimum reduction of 50% in the number of manual operations required to correct the beat-tracking estimates.

Beyond beat tracking and computational rhythm analysis, this user-driven adaptation suggests wider implications for various MIR technologies, particularly when music signal ambiguity and human subjectivity challenge conventional algorithms.

**Keywords:** User-Centred, Transfer Learning, Beat Tracking, Computational Ethnomusicology

## 1 Introduction

Rhythm is a fundamental aspect of music, making computational rhythm analysis a critical topic within Music Information Retrieval (MIR). This area involves tasks such as tempo determination, rhythmic pattern recognition, and metre determination [9]. Algorithmic beat tracking, the automatic detection of a musical signal's pulse, plays an



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



essential role in various MIR applications that require the parsing of musical *time*, i.e., the beat. In the past decade, beat tracking has seen significant progress, with the current state-of-the-art achieving accuracy levels over 90% on benchmark datasets [5, 2]. However, even these advanced methods can face challenges with complex rhythms, especially if they differ from the features of their training data. These challenges are amplified in specialised areas like Computational Ethnomusicology (CE) [20]. In this domain, the availability of annotated datasets is limited, and the need for specialised cultural knowledge to annotate unique rhythmic examples is crucial. Due to these limitations, many musical traditions remain under-represented in MIR research. This gap highlights a known issue in MIR systems: a primary focus on Western (or *Eurogenetic*) music at the expense of diverse global genres and expressions [4, 8].

To overcome these obstacles, adaptive methods have been proposed for tasks like beat tracking [7] and metre determination [19]. While genre-aware knowledge models might provide solutions, they lack scalability. Fiocchi et al. [6] explored how beat tracking knowledge transfers from mainstream Western to Greek music, but their approach, besides being computationally intensive, did not perform as well as training on the same dataset from scratch and yielded less than satisfactory results on the established *SMC* dataset [10], designed with a focus on challenging musical audio examples.

In light of these shortcomings, we shifted towards a more streamlined solution. Our approach harnesses minimal user annotations to optimise a state-of-the-art beat tracker. In earlier works [16, 15], we introduced this user-centric method, aiming for very high accuracy on specific music pieces. Designed for computational efficiency and compatibility with personal computing devices, our methodology has outperformed established methods across various datasets, most notably on the demanding *SMC* dataset [14].

In this study, we expand the scope of our approach beyond Western music. We evaluate our beat-tracking method using challenging datasets such as the Uruguayan *Can-dombe* and the Colombian *Bambuco*, both distinguished by their respective *polyrhythm* and *polymetre* features. These musical traditions, with their intricate rhythmic structures, serve as a rigorous test bed to assess the adaptability and robustness of our method. Moreover, we apply our technique to Steve Reich's *Piano Phase*, a composition renowned for its innovative use of concurrent *tempi*. The choice to analyse this piece subjects our method to a formidable challenge: to our knowledge, it is the first reported attempt at beat tracking a *polytempo* composition. Our findings indicate that our method effectively manages diverse rhythmic intricacies, allowing for the streamlined adaptation of a leading beat-tracker across a spectrum of musical styles and genres.

## 2 Rhythmic Dissonance Challenges

Rhythm serves as a foundational scaffold for many musical traditions. Particularly, within African heritage cultures, there is a notable use of complex rhythmic techniques such as *polyrhythms*, *polymetres*, and to a lesser extent, *polytempi* [1]. While these rhythmic intricacies contribute to the distinctiveness of these traditions, they introduce unique challenges in Music Information Retrieval (MIR). In this section, we briefly address the

concept of rhythmic dissonance, emphasizing its manifestations in the datasets selected for our study.

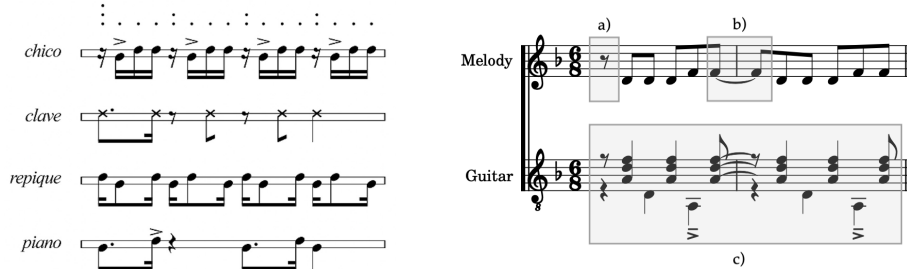


Fig. 1: *Left*: Interaction of the main *Candombe* patterns and the resulting metric structure levels (adapted from [13]). *Right*: Colombian *Bambuco* pattern showing a) downbeat in a rest; b) caudal syncopation; and c) guitar pattern suggesting 6/8 at the top voice and 3/4 at the bass voice (adapted from [3]).

**Polyrhythm in Uruguayan *Candombe*:** *Candombe* is an African-origin rhythm prominent in Uruguay and, to a lesser extent, in other South American countries [18]. Musically, as illustrated in Fig. 1, it is characterised by the interplay of three percussion instruments: the *chico*, the *repique*, and the *piano*, with an additional time-line pattern called *clave*, shared by the three drums [11]. This combination produces a typical rhythmic structure consisting of a four-beat measure evenly divided into 16 tatums, typically played at a tempo of about 110–150 bpm. *Candombe* distinguishes itself from other rhythms through two features that connect it to Afro-Atlantic music traditions [13]: a) the pulse pattern emphasises the second tatum rather than the one on the beat, and b) the *clave* divides the 16-tatum cycle irregularly (3+3+4+2+4), with only two of its five strokes synchronised with the beat. This interplay creates an overall polyrhythmic texture. Moreover, in actual performances, the primary pattern of *repique* leans towards a triplet feeling, and although the *chico* drum establishes the metrical foundation, its pattern exhibits a contraction of inter-onset intervals (IOIs). These unique characteristics of *Candombe* present challenges for both untrained listeners and standard beat-tracking algorithms, making it a challenging test case for evaluating our user-driven approach.

**Polymetre in Colombian *Bambuco*:** *Bambuco* is a Colombian traditional music genre known for its rhythmic complexity, characterised by heavy syncopation, odd accents, and a certain degree of rhythmic freedom, including tempo variations and micro-timing [3]. Its most distinctive aspect is the polymetric nature, resulting from the superposition or alternation of musical elements in two metres: a simple metre (3/4) and a compound one (6/8), as illustrated on the right part of Fig. 1. This phenomenon, commonly known as “hemiola” or the equivalent Latin term “sesquialtera”, is relatively common in other South American musical genres [18] but poses a challenge for computational metre and beat-tracking analysis of *Bambuco*. As illustrated by the guitar voice, depending on the simple or compound metre interpretation, the beats’ locations do not align, except for the downbeat. This indicates a close relationship between the tasks of metre analysis and beat tracking. Essentially, it implies that we can deduce the metric

interpretation from the placement of the beats. These properties make *Bambuco* an ideal test case. More specifically, while our approach primarily targets beat tracking, it also informs metre analysis due to the interconnected nature of these rhythmical facets.

**Polytempo in Steve Reich *Piano Phase*:** Steve Reich’s *Piano Phase* stands out as an interesting example of *polytempo*, a phenomenon mostly absent from mainstream music genres and unrepresented in datasets used to train deep-learning beat-tracking models. This rhythmic dissonance effect presents a significant challenge for general-purpose beat-trackers, as it involves concurrent and isochronous pulses within the same music piece. This compositional technique is primarily found in avant-garde Western music, with Charles Ives’s *Symphony no. 4* being considered the earliest formalised work featuring *polytempo*. Later, composers such as Conlon Nancarrow or György Ligeti explored this approach. Steve Reich’s *phasing* is a unique manifestation of *polytempo*, where identical phrases are played simultaneously at slightly different *tempi*, creating a gradual phase shift. *Piano Phase* brings Reich’s technique to live performance (a rendition of the original score is shown in Fig. 2), complete with a detailed set of instructions for performance, which we briefly summarise:

1. One performer starts, the other fades in unison (bars 1–2), and both continue playing the pattern over and over again;
2. The first performer keeps a constant tempo. The other performer gradually increases his tempo, until he is one note ahead of the first performer (bar 3);
3. After playing in synchronisation for a while, the second performer again begins increasing his tempo, and the phase shifting process starts again (bars 3-4);
4. In the first part of the piece, this procedure is repeated twelve times.

♩ = ca. 72

Repeat each bar approximately number of times written. / Jeder Takt soll approximativ wiederholt werden entsprechend der angegebenen Anzahl. / Répétez chaque mesure à peu près le nombre de fois indiqué.

The musical score consists of two staves. The top staff has six measures, each with a circled number and a repetition count: 1 (x 4-8), 2 (x 12-18), 3 (x 4-16), 4 (x 16-24), 5 (x 4-16), and 6 (x 16-24). The bottom staff has six measures, each with a circled number and a repetition count: 1 (x 16-24), 2 (x 4-16), 3 (x 16-24), 4 (x 4-16), 5 (x 16-24), and 6 (x 4-16). Performance instructions include 'hold tempo1', 'accel very slightly', and 'a.v.s.' (ad libitum).

Fig. 2: *Piano Phase*: Partial Reproduction of the Original Score.

### 3 Methodology

Building on our earlier contributions [14, 15], our approach integrates user knowledge with a state-of-the-art beat tracker [2], enabling direct, content-specific adaptation. Through minimal manual annotation, we tailor this system to the unique characteristics of a musical piece and the user’s own subjective musical judgement.

**Retraining and Inference:** To ensure this paper stands as a self-contained resource, we provide a concise overview of our fine-tuning parameterisation process. For an in-depth understanding and further details on the fine-tuning process, readers are directed to consult [15].

Fine-tuning is allowed for all layers of the baseline network. Given the present task is beat-tracking, the losses for tempo and downbeat tasks on the underlying multitask network [2] are masked. Common practice in transfer learning is followed, thus reducing the learning rate to one fifth of the rate used in the base training. To control network adaptation, we divide the fine-tuning segment into two adjacent, disjoint regions for (re)training and validation, setting a maximum of 50 epochs, and employing learning rate reduction and early stopping strategies. To account for the limited data in the fine-tuning region, target widening and data augmentation are employed. The user-annotated region also serves to parameterise the post-processing Dynamic Bayesian Network (DBN), which extracts beat positions from the Temporal Convolutional Network’s likelihood output. For DBN parameterisation, we employ two strategies: 1) adjusting the transition- $\lambda$  parameter for the adaptive processor type (pt), and 2) setting a tempo tolerance window using user annotations as the tempo guide (tg). While fine-tuning (ft) and data augmentation (da) are general user-driven techniques, strategies like the adaptive processor type and tempo guide are specific to networks employing DBN.

Lastly, the length and characteristics of the annotated region, determined by end-users in real-world situations, play a pivotal role in affecting the final performance. In the current experiment, we opted for a relative length, specifically a quarter of the total file length, to standardise the influence of the fine-tuning region length on the evaluation results.

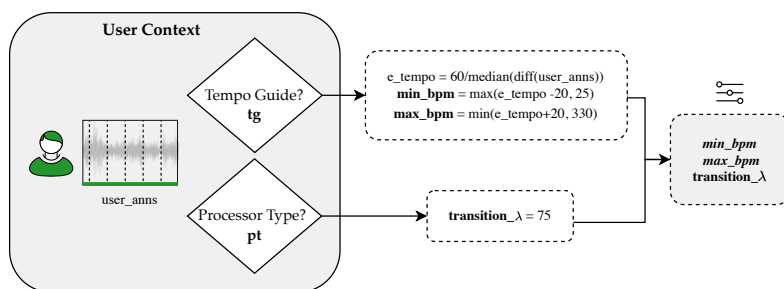


Fig. 3: DBN parameterisation (defaults to min\_bpm:50, max\_bpm:215, transition\_lambda:100).

**Scope of Evaluation:** In this study, we report results with (fullRes) and without (testRes) the fine-tuned part of the input signal for evaluation purposes, and consider the main combinations of user-driven techniques: fine-tuning (ft), data augmentation

(da), and DBN customisations (tg and pt). To minimise variability, we adapt the data augmentation procedure from [14] to a deterministic sampling approach based on a linear distribution with a  $\pm 30\%$  deviation from the local tempo, calculated using the median inter-beat interval across the annotated region. Results are averaged over three iterations, except for the *Piano Phase* analysis, which results include a single run. While there are 11 combinations of user-driven beat-tracking configurations, this report centres on the primary configurations: ft+da, ft+da+pt, ft+da+tg, and ft+da+tg+pt. These are compared with the state-of-the-art, denoted as baseline (bs1). Occasionally, we reference results from configurations that highlight the standalone application of specific techniques, namely ft, pt, and tg.

**Evaluation Metrics:** In the present study, we employ both the standard F-measure and a previously proposed *annotation efficiency* (Ae) metric [17] for beat tracking evaluation. The Ae conceptualises beat tracking evaluation from a user workflow perspective, framing it in terms of the effort necessary to modify a series of detected beats to align with the ground-truth annotations. It provides a more intuitive understanding of the evaluation process and aligns better with practical annotation workflows. This is quantified by counting the number of edit operations, specifically insertions and deletions, but also - contrarily to the F-measure -, including the *shifting* of poorly localised individual beats, a very common operation in annotation workflows. This dual evaluation framework, combining both traditional and user-centric metrics, offers a more comprehensive insight into beat tracking performance.

**Datasets:** We utilise two external datasets and a custom-developed dataset with a simplified version of *Piano Phase*. The *Candombe* dataset has 35 full-length songs with variable durations that accumulate to almost 2.5 hours [11]. The *Bambuco* dataset features two sets of ground-truth annotations corresponding to the predominant meters [12]: 3/4 and 6/8, referred to as *Bambuco (simple)* and *Bambuco (compound)* respectively.



Fig. 4: Musical score of the simplified version of *Piano Phase*.

To address the significant challenges *Piano Phase* presents for human annotators attempting to accurately annotate the beat “by ear”, we created a simplified version (as depicted in Fig. 4) of the composition using a PureData patch. This patch produced two streams of 12 MIDI notes played at slightly different *tempi*, and the audio was obtained using a piano synthesizer. Ground-truth beat annotations were generated for each stream, assuming a 6/8 time signature (thus adopting the dotted quarter note (♩.) as the beat, as inferred from the original score). The score of this simplified rendition is shown in Fig. 2. Our primary experimental objective is to assess the ability of our beat-tracker to synchronise with each of the *tempi* present in the music. To achieve this, the custom

dataset is composed of two files, pianophaseM.A and pianophaseM.B, representing the mixed audio (M:A+B) and ground-truth annotations for streams A and B.

## 4 Results

**Beat Tracking in Uruguayan *Candombe*:** Fig. 5 provides a summary of the overall results. A clear improvement in accuracy scores is observed across all fine-tuning configurations when compared to the baseline (bs1). Exceptions arise with configurations exclusively utilising DBN-parameterisation techniques (pt and tg), which yield scores similar to the baseline. Quantitatively, the best-performing configuration (ft) elevates the mean F-measure score from 0.280 to 0.952 when excluding the fine-tuned region (testRes), and from 0.334 to 0.956 when considering the entire file extent (fullRes).

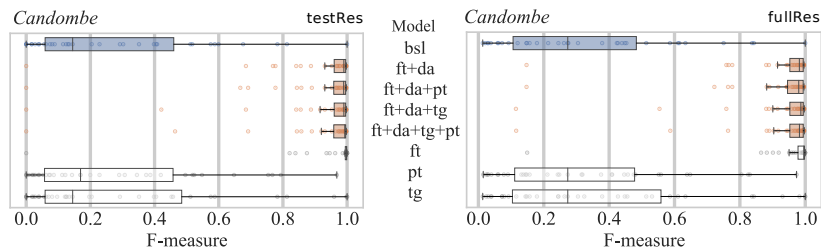


Fig. 5: Distribution of F-measure scores by configuration for the *Candombe* dataset.

When examining the annotation-correction workflow detailed in Table 1, it is observed that the Annotation Gain (Ag) improvements are marginally less than those of the F-measure. This indicates that the shift operation plays a minor role in this dataset’s annotation workflow. However, the results demonstrate that our method significantly enhances efficiency. The number of operations (#ops) required to correct beat detections drops from 12,912 in the baseline to just 1,904 with the ft configuration. Given that there are 19,136 total beat annotations in the *Candombe* dataset, this means that the ft configuration requires only about 10% of the total beats to be corrected, achieving a reduction of approximately 85% from the baseline. Even accounting for the required user annotations for fine-tuning (which amount to 4,757 in the current experimental scenario), the results demonstrate a compelling decrease in manual annotation effort.

Table 1: Mean of the Ae score and sum of the #det, #ins, #del, #shf, and #ops scores across the *Candombe* dataset for the main configurations. (fullRes)

Dataset	Model	Ae	#det	#ins	#del	#shf	#ops
<i>Candombe</i>	bsl	0.319	6,316	2,901	92	9,919	12,912
	ft+da	0.919	16,688	1,885	181	561	2,632
	ft+da+pt	0.915	16,575	1,997	178	563	2,739
	ft+da+tg	0.922	16,892	1,504	190	739	2,437
	ft+da+tg+pt	<b>0.923</b>	<b>16,903</b>	<b>1,504</b>	188	726	<b>2,421</b>

**Beat Tracking in Colombian *Bambuco*:** As summarised in Fig. 6, all primary fine-tuning configurations outperform the baseline (bsl). Results are consistent across both settings (testRes and fullRes), revealing notable F-measure improvements: around 25 percentage points (p.p.) for the simple metre and close to 30 p.p. for the compound metre datasets. The ft+da+tg configuration emerges as the standout performer in both scenarios. Although each of the techniques (ft, pt, and tg) yields different contributions individually, their combined implementation is what truly augments performance.

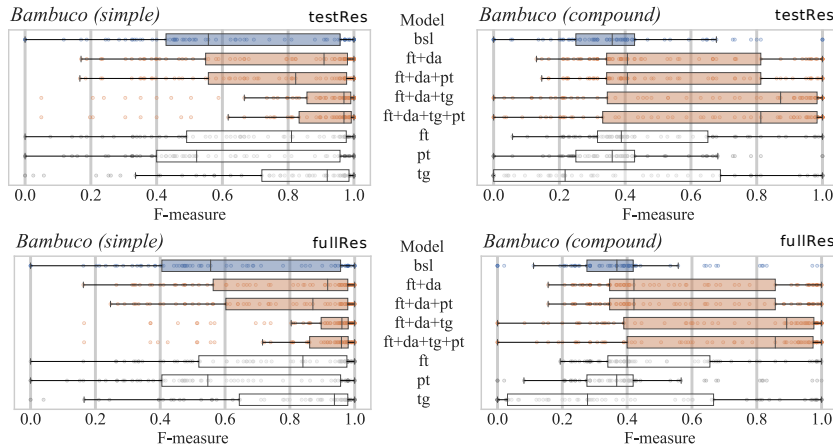


Fig. 6: Distribution of F-measure scores by configuration for the *Bambuco* datasets.

Table 2 shows Ae gains slightly outpacing F-measure, illustrating a greater relevance of the shift operation in this setting. Compared to the baseline, the ft+da+tg setup in simple metre trims beat estimate correction operations (#ops) by two-thirds (455 vs 1,610). For the compound subset, correct detections (#det) almost double in the optimal setting (from 899 to 1,665), underscoring our method’s enhancement over the state of the art.

Table 2: Mean of the Ae score and sum of the #det, #ins, #del, #shf, and #ops scores across the *Bambuco* datasets for the main configurations. (fullRes)

Dataset	Model	Ae	#det	#ins	#del	#shf	#ops
<i>Bambuco (simple)</i>	bsl	0.556	1,756	1,110	60	440	1,610
	ft+da	0.726	2,439	602	91	265	957
	ft+da+pt	0.718	2,428	588	94	291	972
	ft+da+tg	<b>0.869</b>	<b>2,990</b>	12	138	306	455
	ft+da+tg+pt	0.866	2,978	12	137	316	465
	<i>Bambuco (compound)</i>	bsl	0.338	899	424	410	947
	ft+da	0.509	1,319	285	340	665	1,292
	ft+da+pt	0.513	1,322	285	332	663	1,282
	ft+da+tg	<b>0.685</b>	<b>1,665</b>	63	62	541	667
	ft+da+tg+pt	0.671	1,640	73	61	557	691

**Beat Tracking in Steve Reich *Piano Phase*:** The primary experimental objective is to evaluate the capability of our beat-tracking method in synchronising with distinct *tempi* present in this musical piece. When referencing stream *A* or *B*, we are essentially assessing the beat tracker’s ability to tune into each stream’s tempo. This task, which is already immensely challenging for most humans, i.e., allowing themselves to align with one tempo while ignoring conflicting ones, presents an even more formidable test for an automatic beat tracker. Given this complexity, any advancement in performance, even if slight, can be considered significant. With this perspective, we now delve into the results obtained from our experiments.

From Fig. 7, results indicate improvements across all fine-tuning configurations when compared to the baseline for both streams (*A* and *B*). The F-measure score rises from approximately 0.2 to 0.7 across the main configurations. The role of fine-tuning (*ft*) is prominent, emerging as a key factor in performance enhancement. However, a more constrained adaptation to stream *B* is also apparent, an aspect we currently lack comprehensive data to fully elucidate. Another aspect worth further investigation is the observed efficiency of the adaptive processor type (*pt*) over the tempo guide (*tg*). This observation is somewhat counterintuitive, given that the primary goal of this beat tracking method aims to synchronise with conflicting, yet stable, *tempi*.

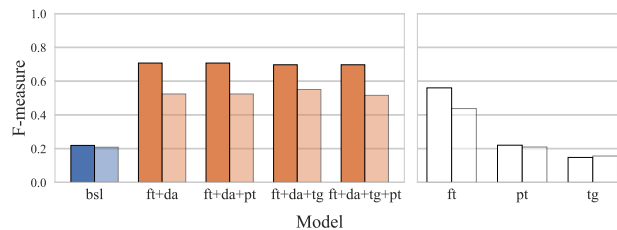


Fig. 7: F-measure vs model for *Piano Phase* (left:pianophaseM.A; right:pianophaseM.B).

As represented in Fig. 8, a closer examination of specific musical segments for stream *A* is provided. This figure offers a comparative perspective between the baseline approach and our best-performing configuration (*ft+da*). The superiority of the fine-tuned configuration over the baseline is evident across most parts of the musical segment. Notably, the beat estimates are accurate up to nearly bar 6 (or up to 68 seconds to be precise). However, around bar 6, signs of desynchronisation emerge, with the imprecise predictions persisting until bar 8. In this specific range, the baseline method manages to hold a slight edge over our approach by correctly identifying certain beats. In terms of the annotation-correction workflow, we see that the state-of-the-art correctly estimates 40 beats, while our fine-tuned configuration improves this count significantly, estimating 105 correctly. Even considering the required 19 user annotations for the fine-tuning segment, this is a notable improvement with such challenging material.

However, it is important to place the results obtained in the appropriate context. When comparing our method with non-adaptive beat trackers, including the current state-of-the-art, we recognize that this is not an even comparison. Most traditional beat trackers are designed for music that adheres to a single tempo, and data-driven methods



have not been exposed to similar training examples, as *polytempo* is absent from standard datasets. Despite these differences, it remains logical to use a baseline for performance assessment. Our focus is in demonstrating that with minimal user input, our approach can leverage the model’s general knowledge and adapt to music with rhythmic dissonance. This showcases the versatility of our approach and its applicability in diverse musical scenarios.

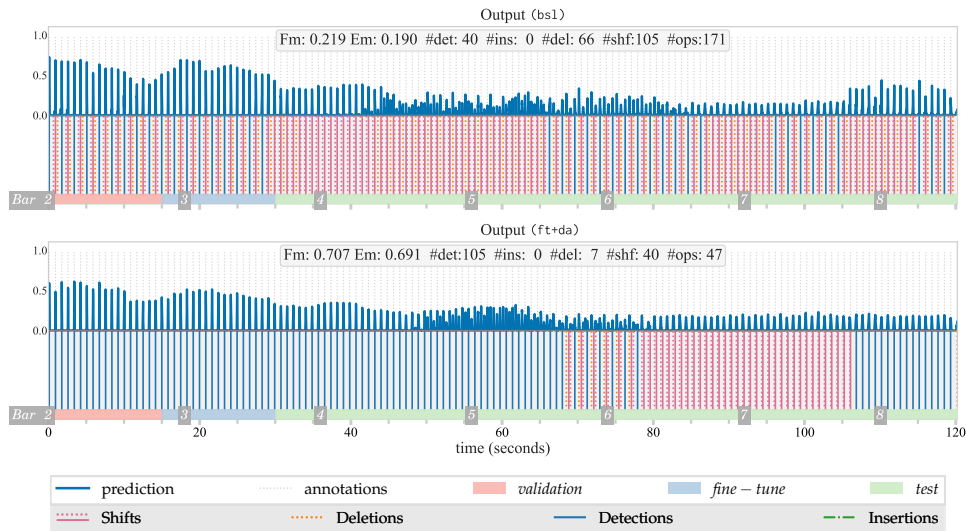


Fig. 8: Detailed analysis for *pianophaseM\_A* (Mixed audio and annotations for stream A tempo).

## 5 Conclusions

In this study, we presented a user-centric approach to beat tracking designed specifically for challenging music signals. By leveraging concise user-annotated regions, our method significantly enhanced the performance of current state-of-the-art beat tracking, especially in environments dominated by complex rhythms. The rhythmic intricacies of Colombian *Bambuco*, Uruguayan *Candombe*, and Steve Reich’s *Piano Phase* were put under scrutiny. These music forms represent, in order, the phenomena of *polyrhythm*, *polymetre*, and *polytempo*.

Among the notable results, for *Candombe*, our approach achieved an excess of 3-fold improvement over existing techniques. In the case of *Bambuco*, the performance was enhanced by approximately 25 p.p. for the simple metre and neared 30 p.p. for the compound metre datasets. With Reich’s *Piano Phase*, even though the F-measure score escalated 50 p.p., our primary objective was to underscore our method’s capability in handling the extreme challenges posed by *polytempo*. To the best of our knowledge,

this study is the first to attempt beat tracking of a musical composition with such compositional technique.

While these results are promising, it is essential to interpret accuracy variations carefully and circumscribe them to the scope of our investigation. Looking forward to future research directions, the exploration of extended musical segments, enriched with a diverse set of fine-tuning parameters, could provide more profound insights into *polytempo* adaptability. Though this study's scope was restricted, it introduces promising methodologies for situations where traditional techniques might not be as effective.

In summary, our research demonstrated the potential of transfer learning and user-driven adaptation for beat tracking in rhythmically complex musical contexts. Using minimal user feedback, we enhanced the state-of-the-art model, enabling its adaptability to challenging musical scenarios and underscoring its utility for specific applications, notably musicological analysis. Our research reach extends past beat tracking, touching upon rhythm-focused tasks such as metre determination and downbeat tracking. Yet, our user-centred approach suggests even wider application across various MIR tasks, beyond computational rhythm analysis. Given the inherent ambiguity in music signals, integrating a user-centric viewpoint is pivotal in integrating subjectivity and accurate analysis.

While our findings represent an encouraging step forward, there remains much to explore in this domain. We hope this study serves as a starting point for future endeavours, aiming to refine adaptive strategies and the human-in-the-loop paradigm. Ultimately, our goal is to promote the development of MIR tools capable of effectively handling a wider range of musical traditions, fostering inclusivity and a deeper appreciation of the world's rich musical heritage.

## References

1. K. Agawu and V. K. Agawu. *African Rhythm: A Northern Ewe Perspective*. Cambridge University Press, 1995.
2. S. Böck and M. E. P. Davies. Deconstruct, Analyse, Reconstruct: How To Improve Tempo, Beat, and Downbeat Estimation. In *Proc. of the ISMIR*, pages 574–582, 2020.
3. E. Cano, F. Mora-Ángel, G. A. L. Gil, J. R. Zapata, A. Escamilla, J. F. Alzate, and M. Betancur. Sesquialtera in the Colombian Bambuco: Perception and Estimation of Beat and Meter – Extended version. *Trans. of the ISMIR*, 4(1):248–262, 2021.
4. O. Cornelis, M. Lesaffre, D. Moelants, and M. Leman. Access to ethnic music: Advances and perspectives in content-based music information retrieval. *Signal Processing*, 90(4):1008–1031, 2010.
5. M. E. P. Davies and S. Böck. Temporal convolutional networks for musical audio beat tracking. In *Proc. of the 27th European Signal Processing Conference*, 2019.
6. D. Fiochi, M. Buccoli, M. Zanoni, F. Antonacci, and A. Sarti. Beat Tracking using Recurrent Neural Network: A Transfer Learning Approach. In *26th European Signal Processing Conf.(EUSIPCO)*, pages 1915–1919. IEEE, 2018.
7. 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 *Proc. of the ISMIR*, pages 251–258, 2019.
8. E. Gómez, P. Herrera, and F. Gómez-Martin. Computational Ethnomusicology: perspectives and challenges. *JNMR*, 42(2):111–112, 2013.

9. F. Gouyon and S. Dixon. A Review of Automatic Rhythm Description Systems. *Computer Music Journal*, 29(1):34–54, 2005.
10. A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Trans. on Audio, Speech and Language Processing*, 20(9):2539–2548, 2012.
11. L. Jure and M. Rocamora. Microtiming in the rhythmic structure of Candombe drumming patterns. In *Fourth Int. Conf. on Analytical Approaches to World Music (AAWM 2016)*, pages 1–5, 2016.
12. F. Mora-Ángel, G. L. Gil, E. Cano, and S. Grollmisch. ACMUS-MIR: A new annotated data set of Andean Colombian music. In *7th Int. Conf. on Digital Libraries for Musicology (DLfM)*, 2019.
13. L. Nunes, M. Rocamora, L. Jure, and L. W. Biscainho. Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan candombe drumming. In *Proc. of the ISMIR*, pages 264–270, 2015.
14. A. S. Pinto, S. Böck, J. S. Cardoso, and M. E. P. Davies. User-Driven Fine-Tuning for Beat Tracking. *Electronics*, 10(13):1518, 2021.
15. A. S. Pinto and M. E. Davies. Tapping Along to the Difficult Ones: Leveraging User-Input for Beat Tracking in Highly Expressive Musical Content. In R. Kronland-Martinet, S. Ystad, and M. Aramaki, editors, *Perception, Representations, Image, Sound, Music - 14th Int. Symposium, CMMR 2019, Marseille, France, Oct. 14-18, 2019, Revised Selected Papers*, volume 12631 of LNCS, pages 75–90. Springer, 2021.
16. A. S. Pinto and M. E. P. Davies. Towards user-informed beat tracking of musical audio. In *14th Int. Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 577–588, 2019.
17. A. S. Pinto, I. Domingues, and M. E. P. Davies. Shift If You Can: Counting and Visualising Correction Operations for Beat Tracking Evaluation. In *Extended Abstracts for the Late-Breaking Demo Session of the Int. ISMIR Conf.*, 2020.
18. J. M. Schechter, D. E. Sheehy, and R. R. Smith. Latin America. *Ethnomusicology*, 29(2):317, 1985.
19. A. Srinivasamurthy, A. Holzapfel, and X. Serra. Informed automatic meter analysis of music recordings. In *Proc. of the ISMIR*, pages 679–685, 2017.
20. G. Tzanetakis, A. Kapur, W. Schloss, and M. Wright. Computational Ethnomusicology. *J. of Interdisciplinary Music Studies*, 1(2):1–24, 2007.

# Creating a New Lullaby Using an Automatic Music Composition System in Collaboration with a Musician

So Hirawata<sup>1</sup>, Noriko Otani<sup>1</sup>, Daisuke Okabe<sup>1</sup>, and Masayuki Numao<sup>2</sup> \*

<sup>1</sup> Faculty of Informatics, Tokyo City University

<sup>2</sup> Institute of Scientific and Industrial Research, Osaka University  
otani@tcu.ac.jp

**Abstract.** Many parents often have problems with getting their children to sleep. A publishing company planned to produce a promotional video consisting of pictures from their published book and a new lullaby with a sleep-inducing effect. They requested that the new lullaby would be created through a collaboration of an automatic composition system and a musician. In our previous work, a melody generation method has been proposed to support the creative activities of a musician. However, this method requires too much intervention by a musician to meet the publisher's requirements. In this paper, we propose an automatic composition system that generates a new piece with a chord progression only by specifying some existing pieces. A case study is presented in which a professional musician completed a lullaby based on the piece generated by the proposed system.

**Keywords:** Lullaby, Music Composition, Symbiotic Evolution

## 1 Introduction

In a survey of 550 mothers of under-three-year-olds conducted by Interspace Co., Ltd. and Hakuhodo Inc., 66.4 % of the respondents reported having problems with their children's sleep. In addition, 71.3 % answered that they were stressed about getting their children to sleep, and 64.3 % answered that they were troubled about it. The top 3 methods the mothers used to get their children to sleep were pretending to sleep next to them, lying down with and watching over them, and sleeping with them. Although singing to children not only soothes them, but also activates maternal love, increases parental motivation, and improves the quality of parental behavior [3], only 21.5 % and 20.4 % said they read picture books and sang songs, respectively, a behavior that is often criticized in today's parents. However, according to Takamatsu's [12] survey of 337 parents of 18-month-olds, 87.0 % of the respondents reported having sung songs to put their children to sleep.

\* We would like to express our deepest gratitude to all the children and their parents for their cooperation in confirming the sleep-inducing effects, Tokyo City University Futako Kindergarten, Toho Co., Ltd., and Crimzon Technology, Inc.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

A sleep-inducing book written in 2015 by Carl-Johan Ehrlin, a psychologist, titled *The Rabbit Who Wants To Fall Asleep*, has become a global hot topic. It has been reported that when parents read this book aloud to their children at bedtime, the children fall asleep more easily. In Japan, a picture book written in 2020 by NOBU titled *Dream Rescue*, which features tapirs who help children with bad dreams, was published. Prior to the book's release, the publishing company planned to produce a promotional video consisting of pictures from the book and a new lullaby with a sleep-inducing effect in an effort to assist children with sleep and reduce parents' burden. According to the publisher's request, the new lullaby would be created through an AI-human collaboration; the music would be generated using an AI-based automatic composition system while humans would be responsible for writing lyrics, playing, and singing.

In our previous work[7], a melody generation method has been proposed to support the creative activities of musicians while satisfying the clients' requirements using a music composition system. The method is based on a constructive adaptive user interface (CAUI) [6, 4], whose goal is to compose music that arouses a particular sensibility in the listener. In order to reflect a musician's creativity and intention in the overall atmosphere of the music, the musician selects some existing pieces and uses them as training data to induce sensibility models. In addition, the musician specifies a chord progression, the pitch extent, and the length of a new piece, and then a melody is generated based on the specified contents and the sensibility models. The steps of generating a short melody are repeated until the musician is satisfied. The musician selects suitable chord progressions and melodies, arranges them, writes the lyrics, and finally obtains a complete piece. The effectiveness of this method has been evaluated by subjective experiments and two case studies involving collaborative work with professional musicians. However, this approach requires too much intervention by the musician to meet the publisher's requirements described in the previous paragraph.

In this paper, we propose an automatic music composition system that generates a new piece with a chord progression only by specifying some existing pieces, that follows basically our previous work [7]. Using the proposed system, we aim to create a new sleep-inducing lullaby that meets the publisher's requirements and assists parents put their children to sleep. We also present a case study in which a professional musician completed a lullaby based on the piece generated by the proposed system.

## 2 Music Composition Flow

The music composition flow of the proposed system is illustrated in Fig. 1. Some existing pieces are needed as the training dataset. The pieces included in the training dataset and the pieces generated by the system consists of a chord progression and a melody with a 4/4 time signature. The basic duration of a note or rest in a melody is defined as the duration of a sixteenth note. The basic duration of a chord in a chord progression is defined as the duration of a quarter note. A motif, which is the minimum unit of a piece, is set to two bars, and a piece is represented as a sequence of multiple motifs.

First, existing pieces are specified as the training dataset according to the user's sensibilities, aims, and/or the purpose of the intended composition. Beats per minute (BPM) is set to a value randomly selected from a  $\pm 0.5\sigma$  range of all the pieces' BPM

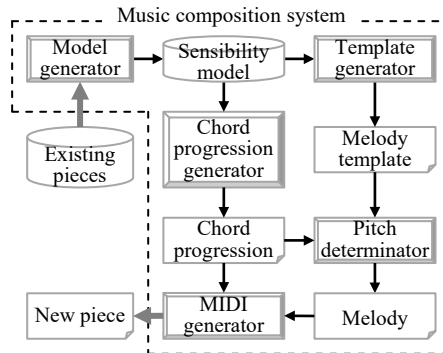


Fig. 1. Music composition flow

distributions in the training dataset. Sensibility models for the chord progression and the melody are obtained based on the training dataset. The next step is to generate a chord progression and a melody template that adapts to the sensibility models and the basic music theory. A melody template indicates the time at which each sound in the melody is played, the length of time each sound is played in succession, and the up-and-down stream of the melody line. In other words, a melody template is a melody without the pitch of each note. Subsequently, the pitch of each note in a melody is determined using the melody template and chord progression. Finally, the chord progression and the melody are combined and output in the form of a MIDI file.

Bainbridge [1] has shown that adult listeners accurately identify unfamiliar lullabies as infant-directed based on their musical features alone, and that infants relax more to unfamiliar foreign lullabies than to non-lullaby foreign songs. They suggested that infants might be predisposed to respond to common features of lullabies. Therefore, to generate a new lullaby using the proposed system, the characteristics of existing lullabies are regarded and used as sensibility models and some traditional lullabies are used as the training data. The generated output piece is then handed over to the musician, who modifies the melody, writes the lyrics, and completes the lullaby.

### 3 Sensibility Models

A sensibility model comprises a partial music structure that affects the user's particular sensibility or reflects their intentions and is represented by a set of patterns that are common to the pieces in the training dataset. In the proposed system, four sensibility models are induced: one each for the rhythm of the chord progression, chord name of the chord progression, rhythm of the melody, and up-and-down stream of the melody.

#### 3.1 Training Dataset

To induce frequent patterns in existing pieces, the rhythms and chord names of the chord progressions, and the rhythms and up-and-down streams of the melodies in the training dataset, are represented as element sequences or chunk sequences.

The rhythm of the chord progression is represented as a sequence of two types of elements, “beat” and “-,” for each motif. Each element represents the state of a note for one beat. The “beat” and “-” elements respectively indicate that a chord is played and the duration of the previous note is extended. The chord name of the chord progression is represented as a chunk sequence. Each chunk indicates one chord and consists of three elements: a degree-notated root note and type pair, tension, and degree-notated on-chord. If there is no tension and no on-chord, the element is “+.”

The rhythm of the melody is represented as a sequence of chunks for each motif. Each chunk indicates one beat rhythm, and consists of four elements. Each element represents the state of a note for 1/4 beat, and can be one of “beat,” “-,” and “null.” The elements “beat,” “-,” and “null” respectively indicate that a sound is played, the duration of the previous note or rest is extended, and no sound is played. The up-and-down stream of the melody is represented as a sequence of elements in which each note other than the first is replaced by an “up,” “down,” or “flat” profile for each motif. Each element indicates the change in pitch from the previous note: “up” means higher, “down” means lower, and “flat” means the same pitch.

An example of element sequences and chunk sequences is shown in Fig. 2. Chunks are enclosed in parentheses. For the rhythm of the chord progression and the up-and-down stream of the melody, two element sequences are generated from the score, respectively. One chunk sequence for the chord name of the chord progression and two chunk sequences for the rhythm of the melody are generated from the score.

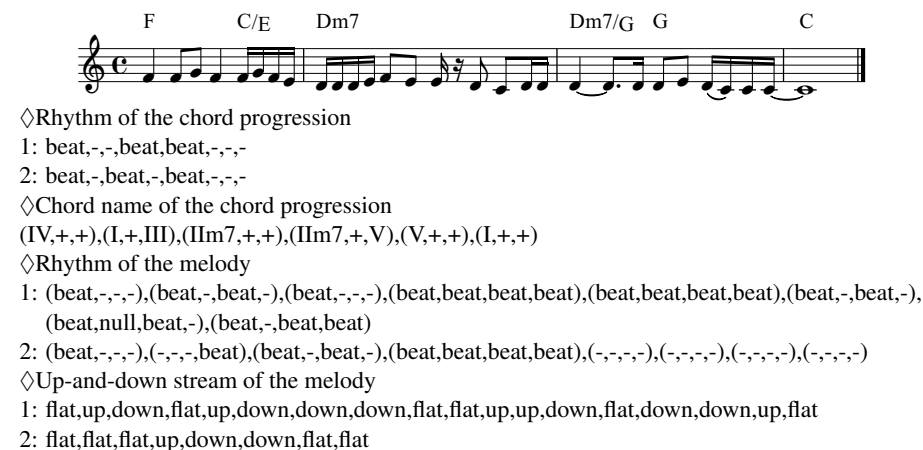


Fig. 2. Examples of element sequences and chunk sequences

### 3.2 Induction of Sensibility Models

A “don’t-care” element “\*” that is considered equal to any element is introduced in obtaining a sensibility model for the rhythm of the chord progression. For any element

sequence  $s_i$  with  $i$  elements in an element sequence in the training dataset replaced by “\*” ( $0 \leq i \leq D_{max}$ ), the set of element sequences in the training dataset contains more than  $F_{min}$  element sequences in which all elements are equal except for “\*”  $F_{min}$  or more, then  $s_i$  is defined as a frequent pattern. Frequent patterns are extracted for all element sequences and used as sensibility models for the rhythm of the chord progression. In a chunk sequence of the chord names of the chord progression, a sequence of any consecutive chunks is called a chunk subsequence. Of all the chunk subsequences in the training dataset, those with a length of  $L_{min}$  or more and occurrence of more than  $F_{min}$  times in the training dataset are extracted as the sensibility models for the chord names of the chord progression. In addition, to replace the on-code with “+” in all chunk sequences and replace tension and on-code with “+,” the chunk subsequences with a length of  $L_{min}$  or more and occurrence of more than  $F_{min}$  times are also extracted as the sensibility models for the chord name of the chord progression.

In a chunk sequence of the rhythm of the melody, a sequence of any consecutive chunks is called a chunk subsequence. Of all the chunk subsequences in the training dataset, those with a length of  $L_{min}$  or more and occurrence of more than  $F_{min}$  times in the training dataset are extracted as the sensibility models for the rhythm of the melody.

In an element sequence of the up-and-down stream of the melody, a sequence of any consecutive elements is called an element subsequence. Of all the element subsequences in the training dataset, those with a length of  $L_{min}$  or more and occurrence of more than  $F_{min}$  times in the training dataset are extracted as the sensibility models for the up-and-down stream of the melody.

To shorten the processing time, the PrefixSpan approach [10] is adopted to induce the sensibility models.

## 4 Chord Progression and Melody

Symbiotic evolution is applied to generate a chord progression and melody template. This section describes the characteristics of symbiotic evolution and how it is applied to generate a chord progression and melody template.

### 4.1 Symbiotic Evolution

Symbiotic evolution is an evolutionary computation algorithm that was proposed for forming neural networks [5]. This algorithm results in a fast, efficient search and prevents convergence to suboptimal solutions. It is characterized by maintaining two separate populations: a partial solution population, the individuals of which represent partial solutions, and a whole solution population, the individuals of which are combinations of individuals in the partial solution population and represent whole solutions. In the former population, partial solutions that may be components of the optimal whole solution are generated. In the latter population, combinations of the partial solutions that may be the optimal solution are generated.

A piece of music can be considered a combination of motifs; it is essential to find motifs that may be contained in a suitable piece of music as well as a suitable combination of motifs. As symbiotic evolution is appropriate for generating a piece of music



owing to its suitable characteristics, a chord progression and a melody template are generated based on symbiotic evolution in the proposed system. Each motif in a chord progression or melody template is expressed as an individual in the partial solution population, and a whole chord progression or a whole melody template is expressed as an individual in the whole solution population. When generating a piece of  $2N$  bars, a chromosome of the whole solution individual is expressed as a pointer sequence to the  $N$  partial solution individuals.

Individuals of both whole and partial solution populations in the next generation are generated using the GA operators: two-point crossover and mutation. The partial solution population is evolved with the strategy described in [5]. The minimal generation gap model [11], which is an effective evolution strategy for avoiding early convergence, is applied to the whole solution population.

After generating the partial and whole solution populations of the initial generation, alternation in all the partial and whole solution populations and evaluation of all the whole and partial solution individuals are repeated a specified number of times. Finally, the sequence of the genes of the partial solution individuals pointed out by the best whole solution individual is generated as the output.

## 4.2 Generation of Chord Progression

Chord progression is generated to adapt to the sensibility models for the rhythm and the chord name of the chord progression using chords contained in the training dataset.

When there are  $R$  types of degree-notated root notes and type pairs in the training dataset, each pair is called  $rt_1 - rt_R$ . The set of tensions of chords whose root note and type pair is  $rt_i$  is called  $T_i$ , and the set of on-chords is called  $O_i$ . A chromosome of a partial solution individual has 24 genes that include 8 root-type genes, 8 tension genes, and 8 on-chord genes. The first root-type gene is a natural number less than or equal to  $R$ . The second and subsequent root-type genes are natural numbers less than or equal to  $R$  or 0, with 0 representing “-” and a non-zero value  $i$  representing  $rt_i$ . The tension genes and on-chord genes are 0 or 1 to represent the presence or absence of tension and on-code, respectively. When the root-type gene is a non-zero value  $i$  and the tension gene or on-chord gene is 1, the tension and on-chord of the chord are selected from  $T_i$  and  $O_i$ , respectively, according to their frequency of occurrence in the training dataset.

After the tail of a piece represented by a whole solution individual is converted to the perfect cadence regardless of the gene value, the fitness value  $f_{cw}(W_c)$  of a whole solution individual  $W_c$  is calculated using (1).

$$f_{cw}(W_c) = \sum_{W_c \rightarrow P_c} \{f_{cr}(P_c) + f_{ct}(P_c)\} + f_{cn}(W_c) + f_{ct}(W_c) . \quad (1)$$

where  $W_c \rightarrow P_c$  means that a partial solution individual  $P_c$  is pointed out by a whole solution individual  $W_c$ . The function  $f_{cr}(P_c)$  indicates the degree of adaptability of a partial solution individual  $P_c$  to the music theory, and the function  $f_{ct}(W_c)$  indicates the degree of adaptability of a whole solution individual  $W_c$  to the music theory. The function  $f_{cr}(P_c)$  indicates the degree of adaptability of a partial solution individual  $P_c$  to the sensibility model for the rhythm of the chord progression, while  $f_{cn}(W_c)$  indicates

the degree of adaptability of a whole solution individual  $W_c$  to the sensibility model for the chord name of the chord progression. The two are calculated using (2) and (3), respectively.

$$f_{cr}(P_c) = \sum_{e \in S_{cr}(P_c)} \{bn(e, P_c) \cdot fq_e(e)\} . \quad (2)$$

$$f_{cn}(W_c) = \sum_{c \in S_{cn}(W_c)} \{ln_c(c) \cdot fq_c(c)\} . \quad (3)$$

where  $S_{cr}(P_c)$  is the set of element sequences that were extracted as the sensibility model for the rhythm of the chord progression and contained in  $P_c$ .  $S_{cn}(W_c)$  is the set of chunk sequences that were extracted as the sensibility model for the chord name of the chord progression and contained in  $W_c$ .  $bn(e, P_c)$  is the number of beats other than the “don’t-care” ones that an element sequence  $e$  covers in  $P_c$ ,  $fq_e(e)$  is the frequency with which  $e$  appears in the training dataset,  $ln_c(c)$  is the length of a chunk sequence  $c$ , and  $fq_c(c)$  is the frequency with which a chunk  $c$  appears in the training dataset.

A partial solution individual is evaluated using whole solution individuals that point to the partial solution individual. The fitness value  $f_{cp}(P_c)$  of  $P_c$  is the largest fitness value of these whole solution individuals, as given by (4). The partial solution individual receives a higher evaluation when it is pointed to by a better whole solution individual.

$$f_{cp}(P_c) = \frac{1}{N} \max_{W_c \rightarrow P_c} f_{cw}(W_c) + f_{cr}(P_c) + f_{ct}(P_c) . \quad (4)$$

### 4.3 Generation of Melody Template

A melody template is generated to adapt to the sensibility models for the rhythm and up-and-down stream of a melody. A chromosome of a partial solution individual has 32 genes. Each gene is  $-1, 0, 1, 2,$  or  $3$ , which mean “rest,” “extend,” “beat + down,” “beat + flat,” and “beat + up” respectively.

The fitness value  $f_{mw}(W_m)$  of a whole solution individual  $W_m$  is defined by (5).

$$f_{mw}(W_m) = \sum_{W_m \rightarrow P_m} f_{mm}(P_m) \times \alpha^{k(W_m)} . \quad (5)$$

where  $W_m \rightarrow P_m$  means that a partial solution individual  $P_m$  is pointed by a whole solution individual  $W_m$ .  $\alpha$  is a parameter greater than 1 that promotes a longer phonetic value of the last note in the melody. Let  $k'(W_m)$  be the number of genes 0 following the end of the whole solution individual  $W_m$ , then  $k(W_m)$  is calculated by (6).

$$k(W_m) = \begin{cases} k'(W_m) & (k'(W_m) \leq 3) \\ 4 & (\text{otherwise}) \end{cases} . \quad (6)$$

The function  $f_{mm}(P_m)$  indicates the degree of adaptability of a partial solution individual  $P_m$  to the sensibility models, and is calculated using (7).

$$f_{mm}(P_m) = f_{mr}(P_m) + \frac{1}{4} f_{mu}(P_m) . \quad (7)$$

The functions  $f_{mr}(P_m)$  and  $f_{mu}(P_m)$  indicate the degree of adaptability to the sensibility model for the rhythm and up-and-down stream of the melody, respectively, and are calculated using (8) and (9).

$$f_{mr}(P_m) = \sum_{c \in S_{mr}(P_m)} \left[ \{ln_c(c)\}^2 \cdot fq_c(c) \right] . \quad (8)$$

$$f_{mu}(P_m) = \sum_{e \in S_{mu}(P_m)} \{ln_e(e) \cdot bn(e, P_m) \cdot fq_e(e)\} . \quad (9)$$

where  $S_{mr}(P_m)$  is the set of chunk sequences that were extracted as the sensibility model for the rhythm of the melody and contained in  $P_m$ .  $S_{mu}(P_m)$  is the set of element sequences that were extracted as the sensibility model for the up-and-down stream of the melody and contained in  $P_m$ .  $ln_c(c)$  is the length of a chunk sequence  $c$ ,  $fq_c(c)$  is the frequency with which a chunk  $c$  appears in the training dataset, and  $bn(e, P_m)$  is the number of beats that an element sequence  $e$  covers in  $P_m$ .

The fitness value  $f_{mp}(P_m)$  of a partial solution individual  $P_m$  is the largest fitness value of these whole solution individuals, as given by (10).

$$f_{mp}(P_m) = \frac{1}{N} \max_{W_m \rightarrow P_m} f_{mw}(W_m) + f_{mm}(P_m) . \quad (10)$$

#### 4.4 Determination of Pitch of Notes in the Melody

In the score of the generated melody, notes are placed at the position where the melody template value is 1 to 3, that is, “beat + down,” “beat + flat,” and “beat + up.” The pitch of each note in the melody is determined based on the generated melody template. First, a pitch candidate set is prepared according to the scale of the tonality and pitch extent. In determining the pitch of a note that is played at the same time as a chord, discords of the chord are deleted from the pitch candidate set.

The pitch of each note is an element of the pitch candidate set. The pitch of the first note in a motif is chosen at random from the pitch candidate set. The pitch of a “beat + flat” note is set to be the same as that of the previous note. The pitch of a “beat + up” note is set to the lowest pitch among the pitches of the pitch candidate set that are higher than that of the previous note. The pitch of a “beat + down” note is set to the highest pitch among the pitches of the pitch candidate set that are lower than that of the previous note. If the target pitch is not contained in the pitch candidate set, the nearest pitch in the pitch candidate set is chosen. In addition, the pitch of the last note in the piece is set to the lowest key pitch among pitches that are higher than that of the previous note for “beat + up,” and the highest key pitch among pitches that are lower than that of the previous note for “beat + flat” or “beat + down.”

## 5 Case Study

### 5.1 Creation of a New Lullaby

A new lullaby was created according to the procedure described in Section 2 with the help of a professional musician who is a member of a Japanese pop duo . The 60 tradi-

tional lullabies collected for the training data were classified according to the four criteria listed below. There were 48 categories in total, 22 of which at least one traditional lullaby was classified to. Possible values for each criterion are given in parentheses. “Structure” refers to whether the lullaby is constructed as a one-part or two-part song divided by the rehearsal mark.

- Tonality (Major / minor)
- Structure (one-part / two-part)
- Tempo (slow / medium / fast)
- Rhythm pattern (bounce / four on the floor / eight beat / sixteen beat)

In total 22 pieces were created, with one piece in each category using the lullabies classified to that category. For one-part structure category, a piece with eight bars was generated. For two-part structure category, the first part of the piece was generated using the first part of the lullabies classified to that category. In the same way, the second part was generated using the second part of the lullabies classified to that category. Then the first and second parts were joined to form a piece of sixteen bars. The members of the publisher selected one of pieces: a two-part minor with a slow tempo, and bounce rhythm pattern, and commissioned it to the musician. The musician changed the piece as follows.

1. The first and second bars were moved to the seventh and eighth bars.
2. The third and fourth bars were moved to the first and second bars.
3. The seventh and eighth bars were moved to the third and fourth bars.
4. The pitch of the second and third notes in the first bar were raised by one tone.

The musician wrote the lyrics to this melody and sang the song with the partner of the duo. A promotional video with this song was created and published on the picture book’s website.<sup>3</sup> A warning “Do not play while driving” is attached to the video.

## **5.2 Effect on Falling Asleep**

Parents with children under the age of six were asked to play the new lullaby while putting their children to sleep and to report the children’s sleeping behavior and their observations each time they did this. The 58 participants’ usual average time to fall asleep by age, the need to put the child to sleep, and the method of putting the child to sleep, which were obtained in the preliminary survey, are shown in Fig. 3, 4, and 5, respectively. Multiple answers were allowed for the latter two measures.

According to the results, 67.2 % of children, regardless of age, took more than 16 minutes to fall asleep. In fact, of the 58, only two three-year-olds, one at four-year-old, seven five-year-olds, and three six-year-olds could fall asleep by themselves; assistance from their parents or other adults was not needed. These results indicate the importance of reassuring the children by snuggling with them. They also reveal that 17.2 % of participants use music to put their children to sleep.

<sup>3</sup> <https://yume-rescue.com/>

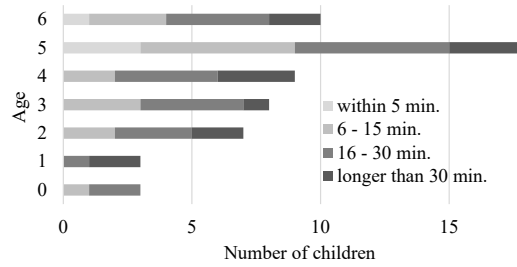


Fig. 3. Usual average time to fall asleep by age

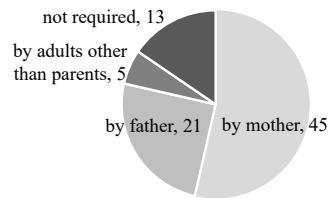


Fig. 4. The need to put the child to sleep

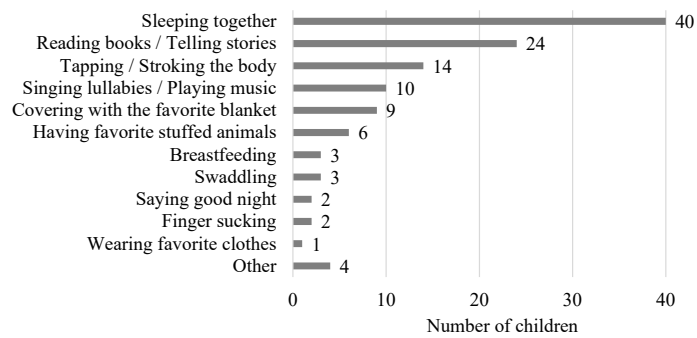


Fig. 5. Method of putting the child to sleep

Each participant used the lullaby 1-9 times to put their children to sleep, producing a total of 126 responses. The children's sleeping behaviors while using the new lullaby are shown in Fig. 6. Although the lullaby was not effective for all of the children, it was effective for the majority of them.

Out of the 126 responses collected, 94 included free descriptions regarding the children's sleeping behavior, such as "Although there was no change in bedtime, a change in being able to go to bed alone without needing to be rocked to sleep continued." These responses were then analyzed using SCAT (Steps for Coding and Theorization) [8, 9]. SCAT is a qualitative data analysis method that weaves themes and constituent concepts that emerge from four steps of coding into a story line and theory. In qualitative research, coding refers to the task of assigning "codes" to text data and codes are concepts that make up the text. The four steps of coding in SCAT are as follows.

1. Identify key phrases in the data
2. Substitute those phrases with phrases outside of the data
3. Provide explanations for those phrases using concepts outside of the text data
4. Identify and conceptualize the themes that have surfaced during the steps 1-3

As a result, it was found that this lullaby has a relaxing effect, but the support of a person the child trusts may be necessary to ensure this effect, and there may be differences in how the length of the song is felt depending on the time it takes to fall asleep. There may also be differences in the sleep-inducing effect depending on the growth stage of the child. If the lullaby could cause the children to fall asleep by conditioned reflex when it is played, then it can be expected that the relaxing effect has improved and that nightmares could possibly be avoided.

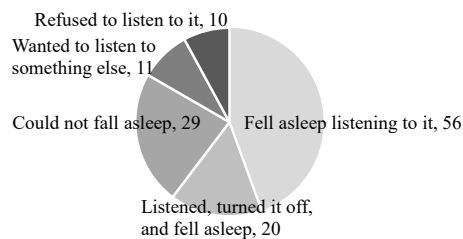


Fig. 6. Sleeping behavior while using the new lullaby

## 6 Conclusion

In this study, we proposed a new automatic music composition system. This system is not specific to any particular genre, but rather generates music that is based on a personal sensibility. As it can generate melodies and chord progressions simply by specifying some existing pieces of music as the training dataset, it is suited for providing musicians with a basis for their creative activities. In addition, by setting music pieces

of a specific genre or existing music pieces that have characteristics that one wants to incorporate in a new piece as the training dataset, it is possible to generate genre-specific or various purpose pieces. A sensibility model that shows characteristics common to training data is not a black box, but it is made explicit, which also helps to identify characteristics.

Here, the proposed system was used to generate a new lullaby. The musicians were able to complete the lullaby without significantly modifying the system-generated piece, and a trial study with under-six-year-old children showed that the lullaby was effective in putting children to sleep to some extent.

In the future works, the target pieces, both inputted into the system and generated by the system, will be expanded to include elements, such as time signatures and the basic durations of notes. In addition, it is important to confirm the effectiveness of the system when applied to various musical genres.

## References

1. Bainbridge, C.M., Bertolo, M., Youngers, J., Atwood, S., Yurdum, L., Simson, J., Lopez, K., Xing, F., Martin, A., Mehr, S.A.: Infants Relax in Response to Unfamiliar Foreign Lullabies. *Nature Human Behaviour*, 5(2), 256–264 (2021)
2. Ehrlin, C.F.: *The Rabbit Who Wants to Fall Asleep: A New Way of Getting Children to Sleep*. Crown Books for Young Readers, New York (2015)
3. Kobayashi, N., Matsui, I., Tanimura, M.: Parenting and Lullabies (in Japanese). *Perinatology*, 23(6), 885–889 (1993)
4. Legaspi, R., Hashimoto, Y., Moriyama, K., Kurihara, S., Numao, M.: Music Compositional Intelligence with an Affective Flavor. *Proceedings of the 12th International Conference on Intelligent User Interfaces*, 216–224 (2007)
5. Moriarty, D.E., Miikkulainen, R.: Efficient Reinforcement Learning through Symbiotic Evolution. *Machine Learning*, 22(1), 11–32 (1996)
6. Numao, M., Takagi, S., Nakamura, K.: Constructive Adaptive User Interfaces –Composing Music Based on Human Feelings. *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 193–198 (2002)
7. Otani, N., Okabe, D., Numao, M.: Generating a Melody Based on Symbiotic Evolution for Musicians’ Creative Activities. *Proceedings of the Genetic and Evolutionary Computation Conference 2018*, 197–204 (2018)
8. Otani, T.: A Qualitative Data Analysis Method SCAT by Four-Step Coding – Theorization Process That is Easy Startable and Applicable to Small Datasets –. *Bulletin of the Graduate School of Education and Human Development, Nagoya University*, 54, 27–44 (2008)
9. Otani, T.: SCAT: Steps for Coding and Theorization – a Qualitative Data Analysis Method That is Easy Startable with Explicit Procedures and Applicable to Small Datasets –. *Kansei Engineering*, 10(3), 155–160 (2011)
10. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Mining Sequential Patterns by Pattern-Growth: the Prefixspan Approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11), 1424–1440 (2004)
11. Satoh, H., Yamamura, M., Kobayashi, S.: Minimal Generation Gap Model for Gas Considering both Exploration and Exploitation. *Proceedings of the 4th International Conference on Soft Computing*, 494–497 (1996)
12. Takamatsu, A.: Lullabies Today (in Japanese). *Memoirs of the Faculty of Education and Regional Studies, Fukui University, Series VI, Fine arts & music, physical education*, 35, 1–19 (2002)

# Automatic Phrasing System for Expressive Performance Based on The Generative Theory of Tonal Music

Madoka Goto<sup>1</sup>, Masahiko Sakai<sup>2</sup>, and Satoshi Tojo<sup>3\*</sup>

<sup>1</sup> Hitachi, Ltd. madoka@trs.css.i.nagoya-u.ac.jp

<sup>2</sup> Nagoya University sakai@i.nagoya-u.ac.jp

<sup>3</sup> Asia University tojo.satoshi@asia-u.ac.jp

**Abstract.** Music phrase is an ambiguous notion since it often depends on the performer's subjective view. Thus far, we have employed Director Musices (DM) for automatic expressive performance, however, segmentation of phrases has only been given manually. In order to identify phrases from an objective viewpoint, we propose to obtain them from the trees acquired by the Generative Theory of Tonal Music (GTTM). We select the usable subtrees and regard the scope of the subtrees as phrases. We introduce a test tool to generate an expressive performance, given original music data to DM together with GTTM trees, to facilitate the phrasing steps.

**Keywords:** Automatic Expressive Performance, Generative Theory of Tonal Music, Director Musices

## 1 Introduction

Automatic expressive performance is an attractive challenge in music information processing, and competition such as RENCON [7] has been held for us to obtain more natural, smooth, and comfortable performance by computers [9]. The key issues in expressive performance concern *dynamique* (loudness of each tone) and *tempo* (speed).

Director Musices<sup>TM</sup> (DM), one of the distinguished generators of expressive performance, also gives variation in dynamique and tempo upon a phrase, with a specific rule called *phrase arch*. However, a phrase is not given in DM but needs to be given by human hands. Here, since a phrase is a subjective notion dependent on each performer, such a phrase arch also needs to be given by experienced human hands, and thus DM is not user-friendly, especially for those musically untrained users.

We consider giving phrase information automatically, independent of such subjective views, to DM. In this research, we propose to acquire phrases from the Generative Theory of Tonal Music (GTTM) [10]. From the theory, we can acquire syntactic tree

\* This research was supported by JSPS Kaken 20H04302 and 21H03572. We thank Gilles Baroin for discussions on effects of composed phrases.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



structures on a given music score, regarding each note as a linguistic morpheme, as is explained in the later section. Since such trees may include extraneous information for DM, we need to consider how we can retrieve usable phrases from them.

In this research, we have implemented a user-friendly interface to facilitate the automatic phrasing process, where we show how GTTM analyses are combined with music data, and report examples of expressive performance.

## 2 Phrase Arches in Director Musices

Director Musices (DM) [2] is a computer system that generates expressive performance based on given performance rules [1]. Input to DM is restricted either to MIDI or to its proper `mus` type file. DM, together with a rule palette of `pal` file in which performance rules are written, renders expressive articulation, and saves its MIDI.

Each performance rule accompanies a parameter called *k*-value, which specifies a grade of the intended effect of the rule upon music pieces. Among these rules, *phrase arch* that acts on a phrase, plays an important role, as it controls loudness (tone volume) and duration of each note. In the concrete, the beginning part of a phrase receives *accelerando* (gradually faster in tempo), and the ending part does *ritardando* (gradually slower); the loudness in *accelerando* grows larger while in *ritardando* smaller. This effect is illustrated in Fig. 1 though here the penult to the final note receives an *accelerando*.

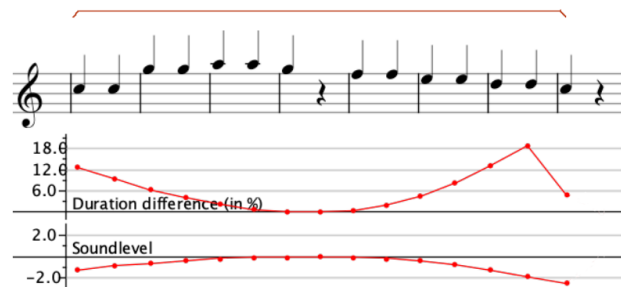


Fig. 1. DM phrasing application

For example, Fig. 2 is a screenshot of DM system. Here appear three layers of phrase arches above the score; since each of which can be given a different *k*-value, each layer is arranged in a different grade. While MIDI input is automatically converted to `mus` type, phrase arch itself must be edited manually; otherwise, the target music remains insipid and tasteless.



Fig. 2. Example of phrase arch: *Jupiter of The Planet* Op.32, Gustav Holst

### 3 Phrases obtained from GTTM

In order to avoid subjectivity in identifying phrases, we would like to rely on an external method to obtain them. In this section, we introduce a Generative Theory of Tonal Music (GTTM) and show how we can retrieve phrases from its analysis.

#### 3.1 Generative Theory of Tonal Music

At the end of the 19th century, Heinrich Schenker proposed the *reduction principle*; that is, we can reduce the number of notes appearing on the score surface (*Vorgrund*), disregarding decorative notes, and can reach the fundamental structure (*Hintergrund*) or the basic melody line (*Urlinie*), consisting only of intrinsic notes to form cadences.

In order to embody the process from *Vorgrund* to *Hintergrund* in music, GTTM [10] invented a method to build a hierarchical tree in a bottom-up way, at each node of which two adjacent notes are compared and the more structurally important note goes upward, absorbing the less important one. Therefore, each of its nodes becomes either left-branching or right-branching. We call such importance among notes *salience*, according to [10]. Hereafter, we call the salient branch the *prime* branch, and the other the *secondary* branch.

GTTM consists of well-formedness rules that constrain rigid syntax, and other preference rules. In the process of building a tree, multiple preference rules may be applicable, and thus, the process necessarily becomes ambiguous. Hamanaka et al. [6] then assigned weighted parameters to all those applicable rules of GTTM, gave an algorithm to choose the most adequate rule in generating a tree, and realized a semi-deterministic procedure as a computer process.

GTTM consists of four sub-theories of grouping analysis, metrical analysis, time-span analysis, and prolongational analysis. The first three theories contribute to the construction of the time-span tree, and the prolongational analysis, together with the time-span tree, results in the prolongational tree. We summarize these trees as follows.

**Time-span tree** The grouping analysis finds boundaries in a sequence of musical notes, based on strength, duration, register, accent, and so on. Then, the metrical analysis identifies those notes with strong/weak beats in meters.

Here, we consider the note of group boundary (the beginning or the end of a group) with a stronger beat to be more salient than the neighboring note. Since the grouping structure is hierarchical, that is, smaller groups are merged into a larger group recursively, the comparison of salience also becomes hierarchical. Therefore, notes compose a knockout tournament in regard to structural salience.

We illustrate a time-span tree of *Jupiter* of Holst which we have employed in Fig. 2, in the left figure of Fig. 3.

**Prolongational tree** The time-span tree does not reflect the harmonic structure of the music piece. In order to represent the dependency of chords, and to organize cadences, we rearrange the branchings of the time-span tree to construct the prolongational tree.

Actually, the fundamental structure that Schenker originally intended was such cadences that are I (tonic) – V (dominant) – I (tonic), I – IV (subdominant) – V – I, I – IV – I, and so on. As a result, the left-hand side of the binary tree represents a progression of chords to cause *tension* while the right-hand side represents *relaxation*.

We show a prolongational tree of *Jupiter* in the right figure of Fig. 3.

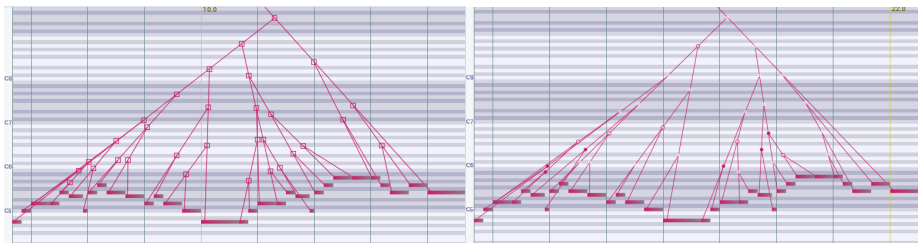


Fig. 3. Time-span (left) and Prolongational (right) trees of *Jupiter* [5]

### 3.2 Extraction of Phrases

From the two tree structures obtained from GTTM analyses, we can naturally consider that the scope of one subtree becomes a candidate of a phrase. In this research, we assign a phrase to each of hierarchical subtrees, as is shown in Fig. 4.

According to this, phrases become hierarchical; two adjacent phrases compose a larger phrase recursively in a higher hierarchy. Here, we can also naturally abandon smaller phrases, that are near to leaves (notes) in a tree, for the following two reasons.

- Even though we give expressive phrasing in a short phrase, the human auditory sense cannot catch it.
- Useless multiple layers of phrases blur each effect of expressiveness; overlapping of expressive effects may cancel each other, or may unnecessarily be augmented.

In order to avoid the above issues, we exclude those deep nodes counting from the top (root) node. Note that the top node represents the whole piece, and thus, the whole piece itself could be a phrase; however, in this research we do not regard the whole piece as a target to give expressiveness, since we pay attention to local phrasings.

Now, let  $root(t)$  be a root node of tree  $t$ . Since  $root(t)$  can possess two immediate branches, one of the two is more salient than the other; we name the prime (more salient) one  $prm(v)$  and the second one  $snd(v)$ . Also, we provide the following notions.

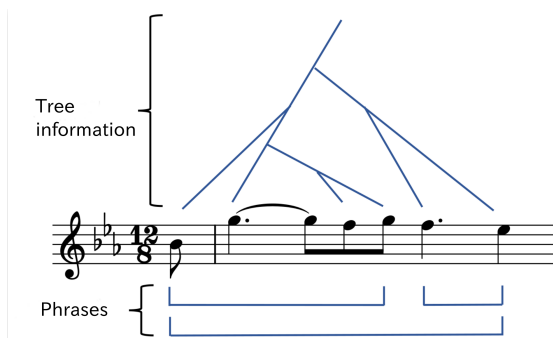


Fig. 4. Subtree as a phrase

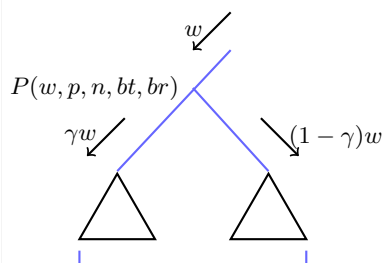


Fig. 5. Weight distribution

- $\#Nt(v)$ : the number of notes below  $v$  (except for slurred notes).
- $\#Bt(v)$ : the number of beats in terms of meters below  $v$ .
- $\#Br(v)$ : the number of bars below  $v$ .

We define a recursive function  $phGen(v, w, p)$  for tree node  $v$ , weight  $w$ , and phrase-level  $p$ ; when predicate

$$P(w, p, \#Nt(v), \#Bt(v), \#Br(v))$$

holds, a phrase is recognized and we assign the total weight of  $w$  which would be distributed to her sub-branches under  $v$ , with the ratio of  $\gamma: 1 - \gamma$  ( $0 \leq \gamma \leq 1$ ), between  $prm(v)$  and  $snd(v)$ , respectively. The validity of  $P$  is adjustable dependent on  $w$  so that we can restrict the number of layers of phrases.

The phrase detection algorithm is summarized as follows.

**Input** : a tree  $t$ , an initial weight  $w_0$ , a distribution ratio  $\gamma$ , and a predicate  $P$ .

**Output** : layer of phrases, produced by  $phGen(root(t), w_0, 1)$ .

**Procedure**  $phGen(v, w, p)$  :

1. Regard the scope of  $t$  as a phrase with level  $p$  if  $P(w, p, \#Nt(v), \#Bt(v), \#Br(v))$  holds.
2. Call both  $phGen(prm(v), \gamma w, p + 1)$  and  $phGen(snd(v), (1 - \gamma)w, p + 1)$  recursively and return.

The final step is shown in Fig. 5.

### 3.3 Adjustable Parameters

In the algorithm in the previous section, phrase arches are constructed depending on a tree  $t$ , an initial weight  $w_0$ , a distribution rate  $\gamma$ , and a predicate  $P$ . Therefore, for a given tree we can adjust these parameters to obtain plausible results.

To fine-tune these parameters through machine learning, we require the appropriate phrase information in DM format though, unfortunately, it is currently not available. As a result, we have opted for a less sophisticated approach as our initial step. We observe

	$w_0$	$\gamma$	$P(w, p, n, bt, br)$
Alg.0	$2^3$	1/2	$w > 1$ and $bt \geq 2$
Alg.1	$b_0$	2/3	$w \geq 4$ and $n \geq 2$
Alg.2	$(b_0)^2/n_0$	1/2	$w \geq 3.75$ and $n \geq 2$
Alg.3	$(b_0)^2/n_0$	3/5	$w \geq 5.9$ and $n \geq 2$
Alg.4	unused	unused	$bt/n \geq 0.6$ , $bt > 4$ , $n \geq 2$ , and $p < 4 \vee n \leq 4$
Alg.5	$b_0$	1/2	$w \geq 0.5$ , $n \geq 2$ , and $br \leq 10/p$

$b_0 : \#Bt(\text{root}(t))$ ,  $n_0 : \#Nt(\text{root}(t))$

**Table 1.** Proposed set of parameters

	Alg.0	Alg.1
<i>Le Cygne</i>	8	5
<i>Salut d'amour</i>	9	4
sum	17	9

**Table 2.** Preliminary Experiments

the behavior of the algorithm in multiple preliminary experiments with Alg.0 and Alg.1, and propose three different assignments of parameters Alg.2, Alg.3, Alg.4, and Alg.5 in Table 1.

Each preliminary experiment is based on the following consideration. First, Alg.0 simply restricts the number of layers to three; then, Alg.1 revises Alg.0 as follows: (i) a long piece needs more minute segmentation and needs to increase the number of layers, and (ii) the primary branch may need the larger number of layers than the secondary branch.

In order to compare the efficacy of Alg.1 with that of Alg.0, we have experimented on *Le Cygne* (The Swan) of Camille Saint-Saëns, and on *Salut d'amour* (Love's greetings) of Edward Elgar. Table 2 shows that Alg.1 is unpopular; it is said that its tempo shift sounds unnatural. This result seems to be caused by the distribution ratio of  $\gamma = 2/3$ , which is too unbalanced and may generate too different numbers of layers.

Revising Alg.0 and Alg.1, we propose Alg.2 and Alg.3, the policy of which is commonly the following.

- We revise the distribution ratio to be flatter, as  $1/2 < \gamma < 2/3$ .
- Those pieces with a smaller number of notes, as opposed to the length of the piece, require more expressiveness. We augment the number of layers if  $\#Nt/\#Bt$  is smaller.

In the process of weight passing from upper layers to lower ones, when the number of notes is unbalanced, the number of layers may not be even. To avoid unnatural expressive performance, we further propose Alg.4 based on the following two policies.

- We take  $\#Nt$  and  $\#Bt$  into account when we decide if a phrase is producible.
- When  $\#Nt/\#Bt$  is large, we should avoid minute expressive performance, and avoid also small phrases.

## 4 System Implementation

We have implemented the algorithm proposed in Section 3.2, and have publicized this system.

#### 4.1 Environmental Notes

We have developed an environment [11] that eases testing phrase-creation strategies, where we can compare performances generated from different phrases/palettes for DM on-the-fly. Prior to that, we needed to extend the file converter `kern2dm` in Humdrum Toolkit [8], to translate a `kern` music file into a `mus` DM-specific music file without phrase information. Thus, we revised `kern2dm` to accept a tree structure in `xml` as well as `kern` file [3]. In addition, we offer the following facilities.

**Data downloader** accesses GTTM database, and patches their musicXML scores on information such as tempo, title, and composer name if necessary. It generates pdf scores by using MuseScore™.

**Phrase identifier** generates scores with phrase information for DM by applying the extended `kern2dm`.

**Performance arranger** executes DM to create performances in midi formats, and transforms them into `wav/mp3` formats.

**Screen interface** prepares `html` files to present them on the screen, to compare the performances. Fig. 6 is a part of the index list of detailed pages like Fig. 7 for pieces GTTM music database.<sup>4</sup>

The biggest feature of this system is that it recalculates only the parts affected by the changed files by adopting the *make* system for program development. As a result, the waiting time required for recalculation after changing parameters can be greatly reduced.

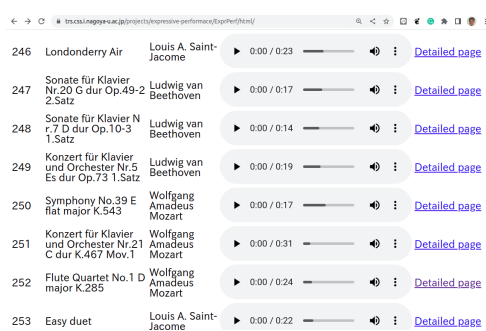


Fig. 6. A Screen Shot of Sample Pieces

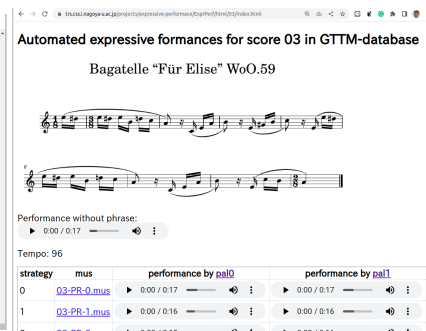


Fig. 7. A detailed page

#### 4.2 Examples

We have conducted experimental analyses. We have applied Alg.2 and Alg.3 to the time-span and the prolongational trees for four pieces in [5], which showed conspicuous

<sup>4</sup> See sample page:  
<https://www.trs.css.i.nagoya-u.ac.jp/projects/expressive-performance/ExprPerf/html/>

effects both in good and bad meanings; that are, Holst: *Jupiter of The Planet*, J. S. Bach: *Jesu, Joy of Man's Desiring*, Tchaikovsky: *Waltz from Swan Lake*, and Ravel: *Pavane pour une infante défunte*. We show all the phrases obtained by our method and rule palette employed in artificial expressive performance in Appendix A.

In comparison between the time-span tree and the prolongational tree, we found that there were no big differences. However, as to *Jesu, Joy of Man's Desiring* of J. S. Bach, this result could not be applied because the time-span tree of the piece is extremely deformed to be left-recursive branching. Since we cannot know if this is the adequate result of time-span analysis, we should doubt the reliability of the process of GTTM. In other words, if the original tree is not reliable, the resultant phrase structure also becomes unreliable.

Uncomfortable expressions are sometimes observed when a short phrase is located at the end of a long phrase. This situation is illustrated in Fig. 8, where each blue and green phrase affects the corresponding colored duration differences, and the red line denotes their additive effect. In this occasion, the conflict occurs between the deceleration due to the short phrase and the acceleration due to the long phrase in the first part of the short phrase.

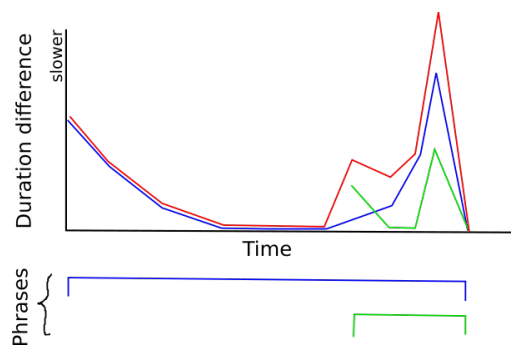


Fig. 8. An effect of composed phrases

In order to confirm the effects of these analyses, we have conducted a questionnaire of 17 examinees, including both of musically trained/ untrained listeners. In comparison between Alg.2 and Alg.3, Alg.2 had a good reputation in both trees (see Table 3); as for the time-span tree 19 vs 14 and for the prolongation tree 26 vs 9, and in sum 45 vs 23 that is 66% vs 34%. Even for each piece, Alg.2 is felt better than Alg.3. Thus, we can say the number of layers should be even.

Table 3. Questionnaire result (pr: prolongational tree and ts: time-span tree)

	Alg.2		pr		ts		pr+ts	
	pr	ts	Alg.2	Alg.3	Alg.2	Alg.3	Alg.2	Alg.3
<i>Jupiter of The Planet</i>	5	12	4	1	9	3	13	4
<i>Jesu, Joy of Man's Desiring</i>	15	2	10	5	1	1	11	6
<i>Waltz of Swan Lake</i>	6	11	6	0	5	6	11	6
<i>Pavane pour une infante défunte</i>	9	8	6	3	4	4	10	7
sum	35	33	26	9	19	14	45	23

## 5 Conclusion and future works

In this research, to avoid the arbitrary choice of phrases in expressive performance, we proposed to give phrases by subtrees obtained from GTTM analysis. We have expanded a file converter to include trees as input besides symbolic music data, implemented an environment for performance comparison, and have experimented to give expressiveness for selected pieces in GTTM database.

We have offered multiple parameters concerning the ratio between the number of notes and that of beats, the weight distribution between two branches at a tree node, and so on. As a result, effects upon time-span trees were found to be more natural than those upon prolongational trees, supposedly because of the balanced length of phrases.

In order to aim at better expressive performance, we need to consider the genre and age of target music when we adjust parameters. In general, baroque music is performed stably in tempo and only cadences should be played in *ritardand* as is provided in DM as FINAL RITARD [1], while in the romanticist age the tempo fluctuates rather freely dependent on performers. Thus, the phrase arch effect should be expressed more conspicuous in romanticist music.

As for the overlaid phrase arches, we need further improvement to avoid mutual cancellation/ augmentation of effects given by each phrase. In order to do this, we need to analyze the innate algorithms inside of DM and to revise them so as to include the interaction between phrase effects; this task remains a future work.

Machine learning is promising for accurately adjusting these parameters. For that purpose, creating a corpus consisting of musical scores with phrases extracted from actual performances by humans is necessary. Therefore, extracting phrases from actual performances is an essential issue for the future.

## References

1. A. Friberg, R. Bresin, and J. Sundberg. Overviews of the kth rule system for musical performances. *Advances in Cognitive Psychology*, 2(2–3):145–161, 2006.
2. A. Friberg, V. Colombo, L. Fryden, and J. Sundberg. Generating musical performances with director musices. *Computer Music Journal*, 24(3):23–29, 2000.
3. M. Goto and M. Sakai. Extended kern2dm. <https://git.trs.css.i.nagoya-u.ac.jp/transcription/humextra/-/tree/kern2dm>.
4. M. Goto, M. Sakai, and S. Tojo. <https://www.trs.css.i.nagoya-u.ac.jp/projects/expressive-performance/cmmr2023/>.
5. M. Hamanaka. GTTM database. <https://gttm.jp/gttm/ja/database/>.
6. M. Hamanaka, K. Hirata, and S. Tojo. Implimenting a generative theory of tonal music. *Journal of New Music Research*, 35(4):249–277, 2006.
7. M. Hashida, T. M. Nakra, H. Katayose, T. Murao, K. Hirata, K. Suzuki, and T. Kitahara. Rencon: Performance rendering contest for automated music systems. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC10)*, 2008.
8. D. Huron. Design principles in computer-based music representation. In *Computer Representations and Models in Music*, pages 5–59. Academic Press, 1992.
9. A. Kirke and E. R. Miranda. A survey of computer systems for expressive music performance. *ACM computer surveys*, 42(1):3:1–3:41, 2009.
10. F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, 1983.



11. M. Sakai. Environment for automatic generation of expressive performances. <https://git.trs.css.i.nagoya-u.ac.jp/transcription/dm-env>.

## A Example Analyses

We show phrases employed for phrase arches for four selected pieces, from Fig. 10 to Fig. 13, with a palette of Fig. 9. We also provide supplemental sound data, at [4].

```
(in-package "DM")
(set-dm-var 'all-rules '(
  (DURATION-CONTRAST 1.0
   :amp 1 :dur 1)
  (DOUBLE-DURATION 1.0 )
  (PHRASE-ARCH 1.4 :phlevel 1
   :turn 0.5 :last 0.2 :amp 2)
  (PHRASE-ARCH 1.4 :phlevel 2
   :turn 0.5 :last 0.2 :amp 5)
  (PHRASE-ARCH 1.4 :phlevel 3
   :turn 0.5 :last 0.2 :amp 3)
  (PHRASE-ARCH 1.4 :phlevel 4
   :turn 0.5 :last 0.2 :amp 2)
))
(set-dm-var 'sync-rule-list
 '( (NO-SYNC NIL)
   (MELODIC-SYNC T)))
```

prolongational tree, Alg.2

prolongational tree, Alg.3

time-span tree, Alg.2

time-span tree, Alg.3

**Fig. 9.** Rule Palette of DM

**Fig. 10.** *Jupiter* (GTTM DB No.49, #Bt = 24)

prolongational tree, Alg.2

prolongational tree, Alg.3

time-span tree, Alg.2

time-span tree, Alg.3

**Fig. 11.** *Jesu, Joy of Man's Desiring*, (GTTM DB No.70, #Bt = 36)

prolongational tree, Alg.2

prolongational tree, Alg.3

time-span tree, Alg.2

time-span tree, Alg.3

**Fig. 12.** Waltz of *Swan Lake*, (GTTM DB No.33, #Bt = 32)

prolongational tree, Alg.2

prolongational tree, Alg.3

time-span tree, Alg.2

time-span tree, Alg.3

**Fig. 13.** *Pavane pour une infante défunte*, (GTTM DB No.73, #Bt = 28)

# NUFluteDB: Flute Sound Dataset with Appropriate and Inappropriate Blowing Styles

Sai Oshita<sup>1</sup> and Tetsuro Kitahara<sup>1\*</sup>

<sup>1</sup>Graduate School of Integrated Basic Sciences, Nihon University  
Setagaya-ku, Tokyo, Japan  
{ohshita, kitahara}@kthrlab.jp

**Abstract.** This paper describes a dataset of flute sounds with appropriate and inappropriate blowing styles. The flute is known as a difficult instrument to learn. We, therefore, have been developing a support system that automatically identifies the appropriateness of blowing in flute performances. To develop such a system, we need a dataset that consists of various sounds with various blowing styles, including both appropriate and inappropriate ways, but there are no such datasets. In this paper, we present the dataset that we have been developing. This dataset consists of sounds played by various players with various blowing styles, and also it has annotations of each sound's subjective evaluation.

**Keywords:** Flute, Dataset, Subjective evaluation

## 1 Introduction

The flute is an instrument whose sound changes greatly when the breath's direction and the mouth's size are changed. Therefore, it is necessary to carefully control the size of the mouth and the direction and strength of the breath to play the flute with appropriate tone quality. However, although many books on the market instruct how to play the flute, only a few clearly describe these points. Therefore, even if one reads a detailed book, it is not easy for a beginner to listen to their sound and judge how to improve it by themselves.

To facilitate beginners' practice of the flute, we have been developing a system that analyzes users' flute sounds and feeds back on how inappropriate their sounds are and/or why they are inappropriate. To achieve such technologies, we need a lot of flute sounds played by various players with different skill levels.

Several studies have been conducted on flute performance support systems. Yoonchang[1] created a system to judge whether the player was playing appropriate sounds by evaluating the head-tube relationship, air pressure, and fingering from the sounds. Kuroda et al.[2] created a dataset that includes sounds played by a robot and human

---

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

players to control the blow's strength and direction strictly. After creating their dataset called *Good Sounds*, Romani et al.[3] created a system to analyze the acoustic characteristics of flute sounds.

Datasets of flute sounds have also been developed recently. Brum[4] created a dataset comprising performances of four pieces with different directions. Cantos[5] created a dataset of flute sounds to research automatic music transcription; it contains monophonic and polyphonic flute sounds, their MIDI transcriptions, and objective evaluations of the transcription accuracy. Goto et al.[6] created a dataset of sounds of various instrument by different performers with different intensities. In addition, multiple datasets of sounds other than flute performances have been developed [7–10]. However, a dataset has not been developed that includes various sounds played with both appropriate and inappropriate blowing styles with annotations of their subjective evaluation.

In this paper, we describe a dataset we have created for a flute performance support system. This dataset is a combination of flute-playing sounds and their ratings.

## 2 Dataset

Because this dataset aims to develop a support system of flute practice, the dataset has to include various sounds ranging from novice-level to advanced ones. In addition, each sound should have an annotation representing how it sounds well. Therefore, we can summarize the issues in designing the dataset as follows:

- **The skill level of players**

Various players ranging from novices to experts should participate in our recording. In particular, asking novice players to participate is essential because such players may hesitate to record their flute sounds, even though it is crucial to analyze sounds played by such players.

- **Playing styles**

The dataset should include sounds played in inappropriate styles, such as too large mouth, too small mouth, too upward breath, and too downward breath. In particular, it would be adequate to ask skillful players to play in such styles intentionally.

- **The number of sounds to be collected**

The dataset should include as many sounds as possible. It is helpful to collect sounds on the Web because participants record their sounds without restrictions on the place and time. It was also significant because our lifestyle was strongly influenced by COVID-19 when we made the dataset.

- **Annotations of subjective evaluation**

To evaluate the appropriateness of flute sounds played by various players on a computer, we need an annotation of the subjective evaluation of each sound. We have to ask sufficiently advanced players to do subjective evaluations to keep the evaluations reliable.

This strategy has some limitations. One is a lack of uniformity in the recording quality. Because participants record their sounds and send them to us via the Web, they are assumed to be recorded via their own devices (such as smartphones). Also, the



Fig. 1. Note performed by participants

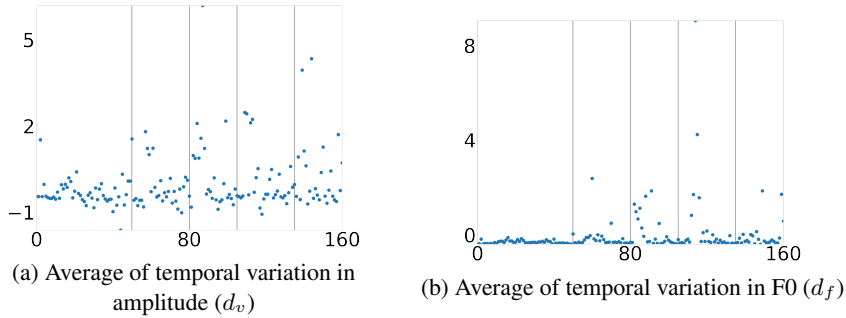


Fig. 2. Acoustic features of the collected flute sounds. Horizontal: sound ID (from left to right: [Normal], [Large mouth], [Small mouth], [Breath upward], [Breath downward]), Vertical: standardized feature values

recording environment (e.g., the distance between the microphone to the flute) cannot be unified. The lack of uniformity may have negative influences on sound analysis.

Another is that we cannot check if participants follow our instructions. Even if they are asked to play in the "too large mouth" style, no one can check if they are genuinely opening their large mouth.

Even though it has such limitations, we made a dataset based on this strategy. Below, we mention the details of the dataset. The dataset is available at the following URL:

<https://github.com/5418010saiohshita/dataset>

## 2.1 Audio recordings

We collected flute sounds played in various blowing styles, including both appropriate and inappropriate ones. As inappropriate ways, we focused on mouth size and breath direction. Due to COVID-19, we asked performers to record their performances themselves and collected them on a crowdsourcing site. The performers were asked to play the score shown in Figure 1 without vibrato. To reduce the burden on individual performers, we asked either of the following two patterns:

- 1 [Normal] [Large mouth] [Small mouth] [Breath upward]
- 2 [Normal] [Breath downward]

The details of the performers and the number of collected sounds are listed in Tables 1 and 2, respectively. To compensate for the fact that the sound volume varies depending on the recording conditions, we corrected the amplitudes so that the temporal mean values of the amplitudes are equal.

Figure 3 shows acoustic features of the collected flute sounds: the averages of temporal variations in amplitude and fundamental frequency (F0). Regarding both the amplitude and F0, sounds played in non-normal blowing styles tend to have more considerable temporal variations.

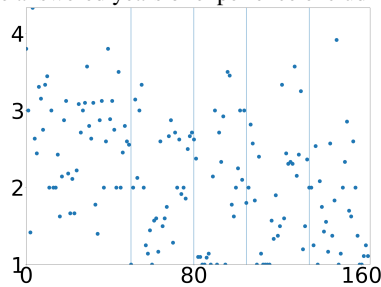
**Table 1.** Details of musical experience etc. of performers

Per-formers	Age	Exp.* [yrs.]	Gap in exp. [yrs.]	Non-flute experience	Max non-flute exp. [yrs.]	Self-determined flute level
P01	46	2	5	Trumpet	9	Beginner
P02	34	10	3	piano, Guitar	5	Intermediate near beginner
P03	55	2	0	Piano	7	Beginner
P04	29	3	0	Piano	5	Intermediate near beginner
P05	33	2	1	Tenor sax, soprano sax, alto sax	12	Beginner
P06	51	0.8	2	Piano, alto sax, clarinet, bass clarinet	11	Almost no experience
P07	21	3	10			Intermediate near beginner
P08	21	0.1				Intermediate near beginner
P09	46	5	10	Piano	12	Beginner
P10	16	8				Intermediate near advanced
P11	24	10	2			Intermediate near beginner
P12	30	10	5	Piano	16	Intermediate near advanced
P13						
P14						

Exp.: experience

Empty cells mean unanswered.

\*Some performers may have answered years of experience excluding the gap.



**Fig. 3.** Distribution of subjective evaluation 1 (overall quality) of collected sound. Horizontal: sound ID (from left to right: [Normal], [Large mouth], [Small mouth], [Breath upward], [Breath downward]), Vertical: ratings

## 2.2 Subjective evaluation

To each sound collected above, we annotated its blowing appropriateness. To obtain such annotations, we conducted subjective evaluations of the collected sounds using a web-based crowdsourcing service. Participants were limited to current or former students of flute majors in music colleges or high school music departments and those who have played the flute for at least 12 months. As a result, six participants listed in Table 3 participated. The number of participations is different among the participants because we allowed them to participate several times as they would like.

When each participant opened the designated web page, 20 randomly selected sounds were displayed. They listened to them individually and entered their answers to the questions in Table 4. The choices for choice-type questions are listed in Tables 5 and 6.

Figure 3 shows the distribution of subjective evaluation 1 (overall quality) for the collected sounds. In general, sounds played in the normal-blowing style were given higher ratings than those in the non-normal-blowing style.

**Table 2.** Number of Experiments for flute sound collection

Performers	Blowing styles					Total
	Normal	Larger	Smaller	Upward	Downward	
P01	1	1	1	1	0	4
P02	2	1	1	1	1	6
P03	10	5	5	5	5	30
P04	1	0	0	0	1	2
P05	1	1	1	1	0	4
P06	10	5	5	5	5	30
P07	2	1	1	1	1	6
P08	7	4	4	4	4	23
P09	7	7	7	7	6	34
P10	3	3	3	3	3	15
P11	2	1	1	1	1	6
P12	1	1	1	1	0	4
P13	1	0	0	0	1	2
P14	1	0	0	0	1	2
Total	49	30	30	30	29	168

**Table 3.** Participants (evaluators) for subjective sound evaluation

Participant	# of participation	Flute experience [yrs.]	Non-flute experience
S01	1	3	Sax, piano
S02	10	5	Piano, harp
S03	1	6	Electric organ, percussion, etc.
S04	1	22	Piano, percussion, piccolo, etc.
S05	1	28	Piano
S06	1	28	Piano

**Table 4.** Questions used in the subjective evaluation

1 Overall quality	Response type 1
2 Clearness of the tone	Response type 1
3 Stability of the intensity	Response type 1
4 Stability of the pitch	Response type 1
5 Smallness of the breathy noise	Response type 1
6 Which in the blowing problems apply? (one or more)	Response type 2
7 Write anything else you noticed	Description

**Table 5.** Response type 1 for subjective evaluation

1 Below beginner level. Seen as just starting level.
2 Beginner level. There are some areas that need improvement.
3 Intermediate level. Some improvement is needed. In general, the student's performance is satisfactory.
4 Intermediate to advanced level. There are some points to be improved, but the performance is acceptable for an amateur concert.
5 Advanced level. There is nothing to be improved at all.

**Table 6.** Response type 2 for subjective evaluation

1 Breathing too strong
2 Breathing too weak
3 Mouth size too large
4 Mouth too small
5 Breath too upward
6 Breath too downward
7 No problem
8 I don't know

**Table 7.** Acoustic features extracted from flute sounds

Feature	Feature description
$d_v$	Average of time variation of amplitude
$d_f$	Average of time variation of fundamental frequency
$r_v$	Amplitude range
$r_f$	Fundamental frequency range
$o_s$	Number of harmonic components (including fundamental frequency components) at the beginning of blowing
$f_s$	Percentage of overtones (non-fundamental components) in all harmonics at the beginning of blowing
$n_s$	Percentage of overtones in the entire spectrum at the beginning of blowing
$o_c$	Number of overtones (calculated from the middle interval)
$f_c$	Percentage of non-fundamental frequency components in all overtones (calculated from the middle interval)
$n_c$	Percentage of overtone components in the whole spectrum (calculated from the middle interval)

### 3 Examples of the use of this dataset

In this section, we present examples using the dataset we created <sup>1</sup>.

#### 3.1 Predicting subjective evaluation from acoustic features

We conducted the prediction of subjective evaluation from acoustic features. This would help develop support system for flute practice. Here, we used linear regression. From each audio signal included in the dataset, 10 acoustic features listed in Table 7 are extracted. Then, these features are applied to linear regression. In linear regression, the objective variable is subjective evaluation 1 (overall quality), while the explanatory variables are those features. Half data were assigned to the training data and the rest to the test data.

Figure 4 compares the subjective evaluation's predicted and actual values. The figure shows that even though the actual value of the highest subjective evaluation is 4.75, and its predicted value is 2.38. When the outliers are removed, the sounds where the actual subjective evaluation is greater than 3 have lower predicted values than the actual evaluation. The root mean square error (RMSE) of the prediction is 0.670. When the outliers are removed, the RMSE is 0.642.

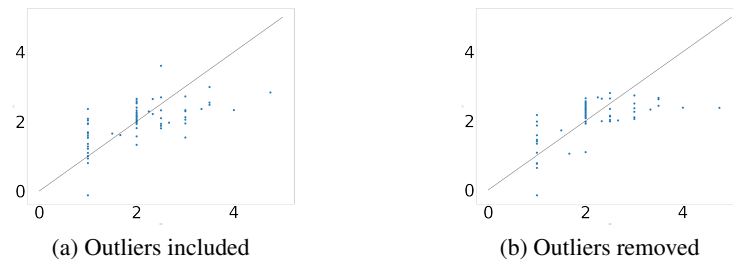
We also attempted the same prediction with decision trees (DTs) after the subjective evaluation was discretized into two or three classes (that is, we conducted it as two-class or three-class classification). The objective feature and explanatory features are the same as above. Table 8 lists the classification accuracy and the depth of the trees acquired. An example of the trees is shown in Figure 5.

#### 3.2 Predicting blowing styles from acoustic features

When the performers recorded a sound, they were asked the blowing style from [Normal], [Large mouth], [Small mouth], [Breath upward], and [Breath downward]. We

<sup>1</sup> These have been presented in our previous paper [11].





**Fig. 4.** Actual subjective evaluation (horizontal) and its prediction (vertical) with linear regression

**Table 8.** Accuracy of predicting subjective evaluation with DT (in parentheses: outliers removed)

Classification	Maximum(Depth 2)	Maximum
Two-class (Lower than 2 / 2 or higher)	0.93 (0.94)	0.93 (0.94)
Three-class (Lower than 2 / 2 to 3 / 3 or higher)	0.83 (0.84)	0.86 (0.84)

attempted the prediction of this blowing style from the acoustic features. We conducted different classification tasks with DTs: two-class [Normal / Other], three-class [Normal / Mouth-size-related / Breath-direction-related], and five-class: each style. Table 9 lists the classification accuracy. An example of the acquired trees is shown in Figure 6.

## 4 Conclusion

We presented a flute sound database consisting of sounds played in appropriate and inappropriate blowing styles. This dataset is intended to be used for developing a support system of flute practice by analyzing how inappropriate the user’s sounds are and why. To help such analysis, we annotated the subjective evaluation to each sound.

In addition, we presented examples of flute sound analysis using our dataset. Even though the prediction of subjective evaluation using linear regression and DTs showed promising results to some extent, the accuracy for predicting blowing styles was low. One possible reason could be that the performer could not strictly control the mouth size and breath direction.

In the future, we would like to improve how to collect sounds. For example, we will ask advanced players to control their mouth size and breath direction strictly and will check them via video recordings. Through this, we would like to develop technologies that help novice flute players improve their skills.

## References

1. Yoonchang, H. and Kyogu, L.: Hierarchical Approach to Detect Common Mistakes of Beginner Flute Players, ISMIR, 2014
2. Jin, K. and Gou, K.: Sensing Control Parameters of Flute from Microphone Sound Based on Machine Learning from Robotic Performer, Sensors, 22, 2022
3. Romani, O., Parra, H., Dabiri, D., Tokuda, H., Hariya, W., Oishi, K., and Serra, X.: A Real-time System for Measuring Sound Goodness in Instrumental Sounds, 138th Audio Engineering Society Convention, 2015

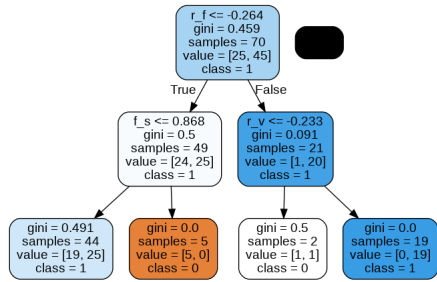


Fig. 5. DT for predicting subjective evaluation

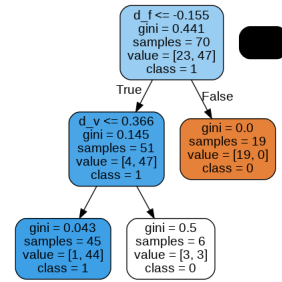


Fig. 6. DT for predicting blowing styles

Table 9. Accuracy of predicting blowing styles with DT (in parentheses: outlier removal)

Classification	Maximum (Depth 2)	Maximum
Two-class (Normal, Others)	0.70 (0.41)	0.71 (0.75)
Three-class (Normal, Oral, Breath)	0.39 (0.31)	0.49 (0.44)
Five-class (Each blowing style)	0.36 (0.19)	0.36 (0.19)

4. Brum, J. P. B.: Traditional Flute Dataset for Score Alignment, 2018, <https://www.kaggle.com/jbraga/traditional-flute-dataset>
5. Elena A. C.: Flute Audio Labelled Database for Automatic Music Transcription, 2018, <https://doi.org/10.5281/zenodo.1408985>
6. Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R. : RWC Music Database: Music Genre Database and Musical Instrument Sound Database, ISMIR, 2003.
7. Ito, K. and Johnson, L.: The LJ Speech Dataset, 2017, <https://keithito.com/LJ-Speech-Dataset/>
8. Vassil, P., Guoguo, C., Daniel, P., and Sanjeev, K.: LibriSpeech: an ASR Corpus Based on Public Domain Audio Books, ICASSP, 2015
9. John, K. and Alan, W. B.: CMU ARCTIC Databases for Speech Synthesis, CMU-LTI-03-177, 2003
10. Paul, B.: Automatic Prosodic Analysis for Computer Aided Pronunciation Teaching, PhD thesis, University of Edinburgh, UK, 1994
11. Oshita, S. and Kitahara, T.: Automatic Classification of Blowing Properness in Flute Sounds, ICA, ABS-0467, 2022

# Melody Blending: A Review and an Experiment

Stefano Kalonaris<sup>1</sup> and Omer Gold<sup>2\*</sup>

<sup>1</sup> Music Information Intelligence Team, RIKEN AIP, Japan

<sup>2</sup> Blavatnik School of Computer Science, Tel Aviv University, Israel

**Abstract.** The blending of two melodies into a third is a creative process useful for exploring a search space and can be employed in compositional or improvisational tasks. Two melodic blend tropes are considered: *hybridization* (recombination of features) and *morphing* (generation of intermediate feature values). After reviewing the approaches that have been used to this end, a bespoke implementation of common methods for both tropes is undertaken, and excerpts demonstrating some use case scenarios are provided. A set of evaluation metrics is then put forward and selected blending modes are tested accordingly in a melodic blending task, for comparison.

## 1 Introduction

In this paper, the task of obtaining a melody  $C$  by blending two melodies  $A$  and  $B$  is considered. The goal is to produce  $C$  so to retain perceptual properties of both input melodies, to different degrees and according to different methods. This procedure relates to conceptual blending [1] whereby two input spaces are integrated into a third by cross-mapping and projection. Conceptual blending has been hailed as a useful tool for creative exploration, and has been used in music with applications relating to harmonization [2] or emotion [3], among others. While there are precedents [4] of conceptual blending applied to melody generation, this paper narrows the scope by inheriting the distinction between *hybridization* and *morphing* originally proposed in [5] and porting it from the raw audio domain to symbolic representation. In hybridization, each attribute of  $C$  is inherited by  $A$  or  $B$ . Thus, each constituent part of  $C$  is obtained by recombining the respective parts of the input melodies. In morphing, instead, the resulting melody  $C$  is an “in between”, intermediate melody which typically maintains the shared properties of melodies  $A$  and  $B$  (if they exist), and can be closer to  $A$  or to  $B$ , proportionally to a morphing coefficient  $\lambda$ . Hereinafter, the terms *source*, *target*, and *blendoid* will be used interchangeably with  $A$ ,  $B$ , and  $C$ , respectively.

Different approaches (ranging from simple *recombination* [6], to music theory [7], or even number theory [8]) have been used to implement melodic blending, each with its own advantages and limitations. The most notable of these are reviewed in Section 2,

---

\* Part of the work on this paper was done while the second author was visiting RIKEN, Japan.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

revealing frequent misnaming and ambiguity (according to the hybridization/morphing dichotomy), as well as much diversity regarding evaluation procedures and metrics. With focus on their creative potential for music generation and computer aided composition, some of the blending methods reviewed are reimplemented ad hoc in Section 3, use case scenarios are illustrated in Section 4, and a set of metrics derived from [5] is applied for the evaluation of blended melodies, in Section 5.

## 2 Related Work

Following is an overview of some key approaches developed so far in the context of melodic blending.

### 2.1 Music Theory

Hamanaka et al. [9, 10] proposed melody blending methods based on the *Generative Theory of Tonal Music* [11] (GTTM) whereby, after computing the intersection between the time-span trees<sup>3</sup> for melodies  $A$  and  $B$ , an intermediate melody is generated by combining segments of the two melodic divisional reductions going from each melody to the intersection. Because of the difficulty in applying the (often ambiguous) GTTM preference rules, this method has suffered from a lack of automatization, and requires human expertise (*i.e.*, manual annotation of GTTM tree structures). This method also assumes that the two reference melodies are in the same key and with a non-empty intersection set. According to [9] the melodies generated using this method satisfy the condition that  $A \& C$  and  $B \& C$  are more similar than  $A \& B$ . The measure of similarity is reportedly calculated as the intersections of notes  $A \cap B$  scaled by the reciprocal of  $\max(\text{length}_A, \text{length}_B)$  and thus does not account for the interpolation of notes. Furthermore, the literature on the GTTM-based blending method only provides examples where melodies  $A$  and  $B$  are related to each other. Arguably, said examples are more akin to what in music is known as the “theme and variations” practice, rather than blending of two independent melodies. For these reasons, it is unclear whether GTTM-based melodic blending can be fully classed as a morphing method, falling somewhat in between the two blending categories.

### 2.2 Probability

The probabilistic approach proposed by Wooller & Brown [12] is also difficult to class (although its authors use the term *morphing*). According to it, the input melodies are subdivided into segments of equal duration (in quarter note length). Starting from a source segment and based on a probability value  $p$  (which determines whether to sample from either the source or target) and the *order* of the Markov process (how many steps to look back within the pitch and duration sequences), the algorithm generates the next segment, sampling from the chosen Markov chain. This repeats as many times as

<sup>3</sup> one of the four hierarchical structures used in the GTTM, the other three being: grouping structure, metrical structure and prolongational reduction.

desired. This method dissociates “musical segments with their original temporal location” [12] and ignores concerns about the alignment between the source and the target. Nevertheless, it is suitable as a creative tool for generating melodic blends and transformations. Wooller & Brown’s method was evaluated through the responses and commentaries of eleven volunteers who compared transitions (both short and long) between tracks performed by a DJ with those obtained by the Markov-based blending. While the focus was on qualitative metrics and the perceived musicality of the blending transitions, the results of this evaluation are difficult to generalize, given the size of the study.

### **2.3 Geometry**

*DMorph* [13] is Oppenheim’s proprietary system which allows the blending of two or more melodies based on Dynamic Time Warping [14] or time syncing algorithms. *DMorph* affords different methods but, while Oppenheim defines morphing as “the sensation of a natural transformation from one theme into another” [13, p.5], some of these (*e.g.*, recombination, interleaving, weighted selection) might class as hybridization, while others (*i.e.*, interpolation) abide by the formal definition of morphing found in [5]. *DMorph* is suitable for pairing sections of the source to sections of the target beyond arbitrary length sampling. It is a fully automatic method and does not depend on corpora or domain expert knowledge. Unfortunately, *DMorph* is not open source and a working version of the software is nowhere to be found. To the authors’ knowledge, *DMorph* lacks a formal evaluation.

### **2.4 Neural Nets**

*MusicVAE* [15] is a variational autoencoder model which addresses long-term structure by using the embeddings of the input musical subsequences to generate output subsequences independently. To train and generate accordingly, *MusicVAE* requires monophonic melodies or drum patterns of a specified length. The quantization is done in sixteenth notes based on the assumption that all training points are in a 4/4 meter. For the evaluation of *MusicVAE*, both quantitative and qualitative methods were used. The former included assessing the accuracy of the *MusicVAE* in reconstructing melodies and comparing the interpolations of two types of *MusicVAE* against a baseline obtained by weighted selection. The latter, instead, asked participants to indicate on a Likert scale whether they deemed the model’s or real compositions more musical.

A more recent work [16] uses VAE to connect smoothly two musical sequences, where *smoothness* relates to pitch and duration transition (*i.e.*, a few consecutive notes around the connection boundary are used to compute Markov transition matrices of each statistics as states).

## **3 Melody Blending**

Some of the techniques discussed so far are here reimplemented with in view to, in the future, developing an integrated toolbox for melody blending. This section describes the main technical details, to this end.

### 3.1 Alignment

To blend two melodies, an appropriate alignment between them must be established first. Here, priority for the alignment is given to the time dimension, and two approaches are explored: *time-sync* and *time-warp*.

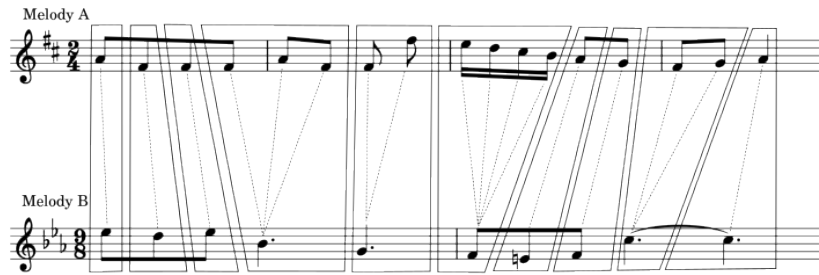
**Time-sync** In the authors implementation of time-sync alignment, it is assumed that source and target are reasonably similar in quarter length duration. If needed, the melodies are *zero-padded* (by lengthening the shortest melody with a rest of duration equal to the sequence difference) so that their length match and divide by a quarter note duration. Then, the melodies are partitioned into disjoint segments as follows. The first segment starts at the beginning of each melody and ends once an event (note/pause) in which the cumulative sum of the onset values of the source and target melodies equals to  $t$  times quarter note duration is reached, where  $t$  is an integer (it is assumed that such a  $t$  always exists after the zero-padding). The next segment starts after the end of the previous segment, and so on, until the end of the melodies. An example of this procedure is given in Figure 1.



**Fig. 1.** An example of time-sync alignment between melodic extracts from L.v. Beethoven's *6 Variations in D major, op.76* (melody A, top) and S. Foster's *Beautiful Dreamer* (melody B, bottom), using quarter-note syncing.

**Time-warp** Dynamic time warping (DTW) [14], instead, is a geometrical approach which can be used also when the source and target melodies are considerably different in length. Recall the definition of DTW between two point-sequences. Let  $A = (p_1, \dots, p_n)$  and  $B = (q_1, \dots, q_m)$  be two sequences of points in some metric space  $(X, \text{dist})$ . A DTW-coupling  $C = (c_1, \dots, c_k)$  between  $A$  and  $B$  is an ordered sequence of distinct pairs of points from  $A \times B$ , such that  $c_1 = (p_1, q_1)$ ,  $c_k = (p_n, q_m)$ , and  $c_r = (p_i, q_j) \Rightarrow c_{r+1} \in \{(p_{i+1}, q_j), (p_i, q_{j+1}), (p_{i+1}, q_{j+1})\}$ , for  $r < k$  (note that  $\max\{n, m\} \leq k \leq n + m$ ). The DTW-distance between  $A$  and  $B$  is

$$\text{dtw}(A, B) = \min_{C: \text{coupling}} \left\{ \sum_{(p_i, q_j) \in C} \text{dist}(p_i, q_j) \right\}. \quad (1)$$



**Fig. 2.** Example of a time-warp alignment between the onset series of the same melodies given in Figure 1.

A coupling  $C$  for which the above sum is minimized is called an *optimal coupling*<sup>4</sup>. Here, the two point-sequences  $A$  and  $B$  represent melodies, where each point  $a_i \in A$  and each point  $b_i \in B$  is a vector with entries corresponding to musical features (e.g., pitch, duration, velocity, etc.). The distance metric  $\text{dist}$  can be chosen among common measures. In this case, the Euclidean distance was used.

Using different feature vectors for the calculation of an optimal coupling (in the fashion of Conklin's *viewpoint sequences* [17]) might produce different results. However, in an effort to achieve better rhythmic coupling, onset series were used as the new point-sequences, to this end. The typical optimal coupling format is a sequence of tuples of indices for matching events in  $A$  and  $B$ . For example, the optimal coupling in Figure 2 would be:

$$C = (0, 0), (1, 1), (2, 2), (3, 3), (4, 3), (5, 3), (6, 4), (7, 4), (8, 5), (9, 5), \dots \quad (2)$$

### 3.2 Blend Methods

Based on the precedents seen in Section 2, several methods for melodic transformation were implemented or adapted.

**Interleaving** In this blend method, one simply alternates between source and target, using the matched events obtained either by time-sync or time-warp alignment (as specified by the user). In the example used thus far, matched events are clearly delineated using polygon contours (see Figures 1 and 2). Despite its simplicity, the interleaving method can produce some interesting blends (see Section 4 for an example).

**Weighted Selection** This blend method operates similarly to interleaving, but considers a blend coefficient between 0 and 1 as the probability  $p$  of selecting, for a given match, from either the source or the target. Weighted selection affords the ability to steer the output closer to the source or the target.

<sup>4</sup> It is possible that there is more than one optimal coupling.

**Markov Chain** Similarly to the previous method, for each match, either the source or the target is selected stochastically using the blend coefficient. Accordingly, the first event is used as the seed to generate a sequence of notes/rests based on the corresponding transition matrix (source or target), using a specified Markov order, and for as long as the duration sum of the generated events (in quarter length) does not exceed that of the original events in the match. As an example, consider the subsequence  $(3, 3), (4, 3), (5, 3)$  of the optimal coupling (2). Suppose that according to the blending coefficient the source is selected: then, the event with index 3 in the source (*i.e.*, an  $F\#$  eighth note) will be the seed for generating notes/rests based on the source's transition matrix, for as long as their duration sum does not exceed a dotted quarter note, which is the duration sum for events with indices  $(3, 4, 5)$  in the source. This process repeats until the exhaustion of matched events in the alignment.

**Interpolation** This blend procedure uses pitch and duration value interpolation over a time-warp optimal coupling. Let  $A = (a_1, \dots, a_n)$  and  $B = (b_1, \dots, b_m)$  be two melodies, where each point  $a_i \in A$  and  $b_j \in B$  is a vector with entries corresponding to musical features (*e.g.*, pitch, duration, velocity, etc.). Let  $C = (c_1, \dots, c_k)$  be an optimal coupling obtained by the DTW algorithm. For each pair  $c = (a_i, b_j) \in C$ , the musical features are interpolated so that, for each pair of points in the coupling, a new point that is “in-between” them is obtained. Although there are many interpolation techniques (piecewise constant, spline, etc.), in the authors' system, the morphed feature  $m_{i,j}$  for a pair  $c = (a_i, b_j) \in C$  is generated by applying linear interpolation using a blend coefficient to yield intermediate values closer to either the source or the target, as desired.

## 4 Use Cases

Different blending methods may be more or less appropriate depending on the musical task at hand.

### 4.1 Style Blend

For example, if one wanted to blend styles in a given musical genre, pure interpolation methods could prove problematic for idiomatic dependencies that might be expected in a scenario of this kind. Conversely, weighted selection or Markov-based methods might be better candidates. Figure 3 shows II-V-I<sup>5</sup> licks<sup>6</sup> by C. Parker's solo on *Au Privave* and from M. Brecker's solo on *Take a Walk*, and the blendoid obtained using weighted selection with a 0.3 blend coefficient.

Source and target are indicative of how the jazz idiom developed over the years, from the *enclosure* approach [18] common in the *be bop* era to the polychordal superimposition employed by more recent players, and the blendoid is an example of successful hybridization of the two.

<sup>5</sup> A standard chord progression serving as building block for larger harmonic structures.

<sup>6</sup> Idiomatic melodic patterns.



CHORDS:  $G_{min}^7$   $C^7$   $F_{maj}^{\#}$

The figure shows three staves of music in 4/4 time. The top staff is labeled 'C. Parker (source)' and contains a melodic line with a triplet. The middle staff is labeled 'M. Brecker (target)' and contains a more complex melodic line. The bottom staff is labeled 'Blendoid' and is annotated with 'Weighted selection, time-sync, 0.3'. Above the staves, the chords  $G_{min}^7$ ,  $C^7$ , and  $F_{maj}^{\#}$  are indicated.

**Fig. 3.** Blending styles over a II-V-I chord progression using weighted selection with time-sync and a blend coefficient of 0.3.

## 4.2 Theme & Variations

Another task where time-sync is suitable could be the generation of variations, as commonly done in the classical tradition. Figure 4 shows a possible variation in the context of W. A. Mozart's *7 Variations on "Willem von Nassau"*, K.25, obtained by blending the original theme with the 3<sup>rd</sup> variation.

The figure shows three staves of music in 4/4 time. The top staff is labeled 'Theme (source)' and contains a simple melodic line. The middle staff is labeled '3rd variation (target)' and contains a more complex melodic line. The bottom staff is labeled 'Blendoid' and is annotated with 'Interleaving'. The Blendoid staff shows a combination of the notes from the theme and the 3rd variation.

**Fig. 4.** A blendoid (bottom staff) generated by interleaving the theme (top staff) and the 3<sup>rd</sup> variation (middle staff) of *7 Variations on "Willem von Nassau"*, K.25 by W.A. Mozart.

## 4.3 Heterogeneous Blend

A case where time-warp interpolation methods would prove interesting is the blending of melodies from heterogeneous genres, or with different metrical structures, length, etc. As an example, Figure 5 shows an interpolation blend of *Le Cygne* by C. Saint-Saëns and *Salut d'amour* by E. Elgar, using a 0.3 coefficient.

The figure displays three musical excerpts in G major, 3/4 time. The top block, labeled 'source', is a melody from 'Le Cygne' by C. Saint-Saëns. The middle block, labeled 'target', is a melody from 'Salut d'amour' by E. Elgar. The bottom block, labeled 'blendoid', is a hybrid melody generated by interpolating the source and target melodies with a blend coefficient of 0.3. The blendoid melody exhibits characteristics of both source and target, including some of the intervals and phrasing from both original pieces.

**Fig. 5.** A blendoid (bottom block) generated by interpolating ( $\lambda = 0.3$ ) excerpts of *Le Cygne* by C. Saint-Saëns (top block) and *Salut d'amour* by E. Elgar (middle block).

## 5 Evaluation

As seen in Section 2, there is no standardized procedure for evaluating melodic blends. Given the combination of available blending methods and time alignments, a universal and exhaustive evaluation protocol might be beyond the scope of this paper. In fact, important criteria in the evaluation of morphing methods might not have a clear correspondence for hybridization techniques and viceversa, thus making the development of consistent evaluation metrics difficult. Notwithstanding, and deferring a more comprehensive evaluation framework to include qualitative metrics to future endeavors, a minimal set of objective metrics is tested. These include *similarity* and two of the three independent criteria proposed in [5]: *intermediateness* and *smoothness*. It must be noted that the latter were originally developed for raw audio and are here interpreted and implemented to reflect the different representation (symbolic) of the musical surface. *Correspondence*, originally also part of the set in [5], is not contemplated here, as one assumes it is guaranteed by virtue of the feature matching in the representation of the melodies. Only blending methods allowing a blend coefficient were considered in this study: weighted selection, Markov chain, and interpolation. These are evaluated over complete blends, going from 0.0 to 1.0 with 0.1 increments, as described below.

**Similarity** Many melodic similarity measures have been proposed and argued, the main approaches being mathematical [19–29], cognition-based [30–32], and musicological [33–35]. To account for true in-between pitch values, this study focuses on melodic contours and employs two measures. One is obtained as in [35], albeit substituting the

original  $n$ -gram similarity over the extended Implication-Realization (IR) symbols at character level with the complement of the  $n$ -gram Jaccard similarity at token level. The other similarity measure is obtained using the normalized Euclidean DTW distance between melodic contour (smoothed) series. For either of these similarity measures  $\text{sim}(\cdot, \cdot)$ , the indicator function in Equation 3 determines whether a blendoid  $b$  is appropriately more similar to the source  $s$  or the target  $t$  with respect to the blend coefficient  $\lambda$ . The weighted sum over a complete blend is taken as the final measure and indicated as SimIR or SimDTW, depending on which similarity metric was used for the indicator function.

$$\mathbf{I}(s, t, b) := \begin{cases} 1 & \text{if } (1 - \lambda) \cdot \text{sim}(s, b) \geq \lambda \cdot \text{sim}(t, b), \text{ for } \lambda \leq 0.5 \\ 0 & \text{if } (1 - \lambda) \cdot \text{sim}(s, b) < \lambda \cdot \text{sim}(t, b), \text{ for } \lambda \leq 0.5 \\ 1 & \text{if } \lambda \cdot \text{sim}(t, b) \geq (1 - \lambda) \cdot \text{sim}(s, b), \text{ for } \lambda > 0.5 \\ 0 & \text{if } \lambda \cdot \text{sim}(t, b) < (1 - \lambda) \cdot \text{sim}(s, b), \text{ for } \lambda > 0.5 \end{cases} \quad (3)$$

**Intermediateness** For intermediateness, a problem posed by the symbolic music domain is the limited choice of discrete steps for in-between notes. Another issue to bear in mind is that linear interpolation of the parametric space does not necessarily result in perceptually intermediate blends. Notwithstanding, the following procedure is proposed: first, the melodic piecewise contours for source  $s$ , target  $t$ , and blendoid  $b$ , are calculated and resampled to  $n$  points proportionally to the blending coefficient. Then, for each point  $i$  in this range, the following is checked:  $\min(s_i, t_i) \leq b_i \leq \max(s_i, t_i)$ . The weighted sum of all the TRUE values is taken as the intermediateness index for that blendoid.

**Smoothness** In [36], a melody is defined smooth simply if the intervals between consecutive notes are within a fifth (*i.e.*, seven semitones). In the context of this experiment, however, a different definition is needed to compare melodies and to quantify whether the blending from source to target is gradual and, thus, successful. In this paper, *autocorrelation* (lag-one), *roughness*, and *mean squared jerk* (MSJ) are employed. Autocorrelation with scores near 1 might imply a smoothly varying series whereas if there isn't an overall linear relationship between consecutive data points one might expect values closer to 0. Roughness in this context is considered as the smoothness penalty as defined in the cubic spline, albeit with a normalization factor that accounts for the length of the input series. The mean squared jerk measure is defined as in [37], and here adapted to the music domain (it is normally employed in movement analysis to measure how much the acceleration of a movement contour changes over time). For all three smoothness measures, the melodic contour (smoothed) series of each blendoid in a complete blend is used as input (like in the DTW-based similarity described earlier).

Using the above metrics and the same two melodic excerpts of Section 3.1, yielded the results shown in Table 1. Note that the values (mean and standard deviation) reported refer to a run of 10 instances of complete blends since all methods but interpolation are stochastic and might generate different blendoids for the same blend coefficient. For the Markov-based method, an order of  $n = 3$  was used.

**Table 1.** Comparing different blending methods based on the proposed evaluation metrics, over 10 full blends. Abbreviations for the methods are: WS (weighted selection), MC (Markov chain), and Lerp (linear interpolation), with *ts* and *tw* indicating time-sync and time-warp, respectively. Abbreviations for the evaluation metrics are: Intrm (intermediateness), Acorr (autocorrelation), Rghns (roughness), and MSJ (mean squared jerk).

	SimIR	SimDTW	Intrm	Acorr	Rghns	MSJ
WS (ts)	0.9 ± 0.3	<b>0.833</b> ± 0.373	0.405 ± 0.045	0.988 ± 0.005	2.185 ± 1.412	2.889 ± 2.045
WS (tw)	0.878 ± 0.328	0.722 ± 0.448	<b>0.43</b> ± 0.13	0.986 ± 0.007	3.95 ± 3.006	4.118 ± 2.912
MC (ts)	<b>0.911</b> ± 0.285	0.689 ± 0.463	0.372 ± 0.069	0.99 ± 0.004	2.3 ± 1.943	3.071 ± 3.054
MC (tw)	0.878 ± 0.328	0.822 ± 0.382	0.359 ± 0.06	0.99 ± 0.002	2.295 ± 1.528	2.654 ± 2.189
Lerp	0.778 ± 0.416	0.444 ± 0.497	0.402 ± 0.041	<b>0.994</b> ± 0.001	<b>0.527</b> ± 0.352	<b>0.507</b> ± 0.413

## 6 Conclusion

This paper offered a brief review of melodic blending approaches, presented an original appropriation for some of these, and proposed objective metrics, in an effort to move towards a more standardized evaluation procedure. The blending operations implemented by the authors are prototypical, and much remains to be improved upon. The morphing methods, particularly, do not handle diatonic perceptual imperatives, and, in cases with a strong “tonal” or “idiomatic” expectation, linear interpolation of features is likely to violate it. Additional features (*e.g.*, dynamics, articulation), could also be included to enhance the blended melody’s musical quality. It is also important to note that, while this experiment dealt with standard symbolic representation, there are other approaches, such as the Tonal Interval Space [38], which merit consideration in future implementations, as they might yield different and more nuanced intermediate values for interpolation. Despite the system’s current limitations, this experiment’s results suggest that a toolbox packaging of the blending functionality described in this paper could be a useful addition to one’s creative workflow, either as a module in a larger generative music system or, conditioned upon further development, as a standalone application.

## References

1. Gilles Fauconnier and Mark Turner. *The way we think: Conceptual blending and the mind’s hidden complexities*. Basic Books, New York, NY, 2003.
2. Maximos Kaliakatsos-Papakostas, Marcelo Queiroz, Costas Tsougras, and Emiliios Cambouropoulos. Conceptual blending of harmonic spaces for creative melodic harmonisation. *Journal of New Music Research*, 46(4):305–328, 2017.
3. Michael Spitzer. Conceptual blending and musical emotion. *Musicae Scientiae*, 22(1):24–37, 2018.

4. Maximos Kaliakatsos-Papakostas. Generative conceptual blending of high-level melodic features: Shortcomings and possible improvements. In *International Conference on New Music Concepts (ICNMC)*, 2019.
5. Marcelo Caetano and Naotoshi Osaka. A formal evaluation framework for sound morphing. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 104–107, 2012.
6. David Cope. *Computer Models of Musical Creativity*. The MIT Press, Cambridge, MA, 2005.
7. Masatoshi Hamanaka and Keiji Hirata anSatoshi Tojo. Applying melody morphing method to composition. In *Proceedings of the 3rd Conference on Computer Simulation of Musical Creativity (CSMC)*, 2018.
8. Jay Hardesty. A self-similar map of rhythmic components. *Journal of Mathematics and Music*, 10(1):36–58, 2016.
9. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Melody Morphing Method Based on GTTM. In *Proceedings of the International Computer Music Conference (ICMC)*, 2008.
10. Keiji Hirata, Satoshi Tojo, and Masatoshi Hamanaka. Melodic Morphing Algorithm in Formalism. In Carlos Agon, Moreno Andreatta, Gérard Assayag, Emmanuel Amiot, Jean Bresson, and John Mandereau, editors, *Mathematics and Computation in Music*, pages 338–341, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
11. Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, MA, 1983.
12. René Wooller and Andrew R. Brown. Note sequence morphing algorithms for performance of electronic dance music. *Digital Creativity*, 22:13–25, 2011.
13. Daniel Oppenheim. ‘DMorph’: An Interactive System for Compositional Morphing of Music in Real-Time. Technical report, IBM, 1995.
14. Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *ACM Trans. Algorithms*, 14(4), 2018.
15. Adam Roberts, Jesse H. Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. *CoRR*, abs/1803.05428, 2018.
16. Taketo Akama. Connective fusion: Learning transformational joining of sequences with application to melody creation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pages 46–53, Montreal, Canada, October 2020.
17. Darrell Conklin and Christina Anagnostopoulou. Representation and discovery of multiple viewpoint patterns. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 479–485, 2001.
18. Jerry Coker. *Elements of the jazz language for the developing improviser*. Alfred Publishing Company, 1991.
19. Luigi Logrippo and Bernard Stepien. Cluster analysis for the computer-assisted statistical analysis of melodies. *Computers and the Humanities*, 20(1):19–33, 1986.
20. Donncha Ó Maidín. A geometrical algorithm for melodic difference. In Walter B. Hewlett and Eleanor Selfridge-Field, editors, *Melodic Similarity - Concepts, Procedures and Applications, Computing in Musicology II*, chapter 2, pages 65–72. MIT Press, Cambridge, Massachusetts, 1998.
21. Rodger J. McNab, Lloyd A. Smith, Ian H. Witten, Clare L. Henderson, and Sally Jo Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proceedings of the First ACM International Conference on Digital Libraries, DL '96*, pages 11–18, New York, NY, USA, 1996. ACM.
22. Kjell Lemström. *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Faculty of Science, Department of Computer Science, 2000.

23. Godfried Toussaint. A comparison of rhythmic similarity measures. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 242–245, 2004.
24. Rainer Typke, Panos Giannopoulos, Remco C. Veltkamp, Frans Wiering, and René van Oostrum. Using transportation distances for measuring melodic similarity. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 107–114, 2003.
25. Shyamala Doraisamy and Stefan Rüger. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70, 2003.
26. Anna Lubiw and Luke Tanur. Pattern matching in polyphonic music as a weighted geometric translation problem. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004.
27. Rainer Typke, Frans Wiering, and Remco C. Veltkamp. Transportation distances and human perception of melodic similarity. *Musicae Scientiae*, 11(1\_suppl):153–181, 2007.
28. Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. Melodic similarity through shape similarity. In Sølvi Ystad, Mitsuko Aramaki, Richard Kronland-Martinet, and Kristoffer Jensen, editors, *Exploring Music Contents*, pages 338–355, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
29. Fumio Hazama. Mathematical analysis of melodies: Slope and discrete Frechet distance. *ArXiv e-prints*, 2014.
30. Ana de Carvalho Junior and Louis Batista. Sms identification using PPM, psychophysiological concepts, and melodic and rhythmic elements. In *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 2012.
31. Carles Roig, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Submission to Mirex 2013 symbolic melodic similarity. In *Proceedings of the Annual Music Information Retrieval Evaluation Exchange*, 2013.
32. Naresh N. Vempala and Frank A. Russo. An empirically derived measure of melodic similarity. *Journal of New Music Research*, 44(4):391–404, 2015.
33. Maarten Grachten, Josep Lluís Arcos, and Ramón López de Mántaras. Melody retrieval using the implication / realization model. In *Proceedings of the Annual Music Information Retrieval Evaluation Exchange*, 2005.
34. Nicola Orió and Antonio Rodà. A measure of melodic similarity based on a graph representation of the music structure. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2009.
35. Sakurako Yazawa, Masatoshi Hamanaka, and Takehito Utsuro. Subjective melodic similarity based on extended implication-realization model. *International Journal of Affective Engineering*, 15(3):249–257, 2016.
36. Rui Pedro, Paiva Teresa, and Mendes Amílcar Cardoso. Exploiting melodic smoothness for melody detection in polyphonic audio. In *Proceedings of the International Computer Music Conference (ICMC)*, 2005.
37. Neville Hogan and Dagmar Sternad. On rhythmic and discrete movements: reflections, definitions and implications for motor control. *Experimental Brain Research*, 181(1):13–30, Jul 2007.
38. Gilberto Bernardes, Diogo Cocharro, Marcelo Caetano, Carlos Guedes, and Matthew E.P. Davies. A multi-level tonal interval space for modelling pitch relatedness and musical consonance. *Journal of New Music Research*, 45(4):281–294, 2016.

## Balancing Musical Co-Creativity: The Case Study of Mixboard, a Mashup Application for Novices

Ottolin, Thomas, Sankaranarayanan, Raghavasimhan, Lei, Qinying, Hugar, Nitin, and Weinberg, Gil \*

<sup>1</sup> Georgia Institute of Technology

tottolin3@gatech.edu

<sup>2</sup> violinsimma@gmail.com

<sup>3</sup> qlei33@gatech.edu

<sup>4</sup> nitin.hugar@gatech.edu

<sup>5</sup> gilw@gatech.edu

**Abstract.** Recent developments in generative AI have posed a challenge for developers who attempt to maintain an effective balance between the system’s generative input and user’s sense of creativity and control. In this paper, we present a longitudinal study of a web/mobile application we developed, Mixboard, which allows novice music lovers to create and share personalized musical mashups in a co-creative manner. Different balances between the role of system automation and user creative input have been developed and studied over a period of two years. Findings from users studies indicate that while novices appreciate the system’s AI driven automation and suggestion, they seek to expand their level of control and creative input into the final product over time. Future developments may therefore include a personalized level of control balance based on continuous assessment of user behaviour.

**Keywords:** Co-creativity, Musical AI, Longitudinal User Studies, Mobile Applications, Novices

### 1 Introduction

Systems that use Artificial Intelligence (AI) to aid in creative processes have recently increased in popularity, partly driven by OpenAI’s suite of Generative Pre-Trained Transformer (GPT) releases starting in 2018 [13]. One of the main challenges facing developers of co-creative systems is how to provide automation and content in a manner that would maintain a sense of creative control and agency for the user. User satisfaction may be negatively impacted if the system prompts the user to contribute too much or too little to the creative outcome. This also comes at a time where music production and

---

\* Thanks to Hardik Goel for his developmental work.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

consumption is, or at least appears to be to novices, more widely accessible. Popular social media, like TikTok or Instagram, empower users to select and edit music and sounds to go along with their planned content. As Jenkins et al. describe, these technologies and new forms of consumption "signals a movement toward a more participatory model of culture one which sees the public not as simply consumers of pre-constructed messages, but as people who are shaping, sharing, reframing and remixing media content in ways which might not have been previously imagined" [8].

We developed Mixboard [15] to allow music lovers to "shape and remix" any set of songs into high-quality musical mashups, assisted by AI. A mashup, in this context, can be defined as a blend of elements from 2+ songs. Aimed at novices, the application acts as a co-creative agent that contributes to the musical decision making, rather than giving the user full control over the final outcome. The AI handles both low-level computational tasks such as source separation, segmentation, tempo and key detection, stretching, and transposition, as well as high-level artistic decisions such as selecting appropriate musical segments and suggesting compositional structures. A previous set of comprehensive user studies with the app identified a clear desire for further user control. This motivated our team to rewrite the system's software infrastructure to provide a more effective balance between user control and system automation. In this paper, we provide a short summary of the original system, describe the new features developed to address the control balance, and present newly conducted research studies that indicate a higher level of user satisfaction and productivity while working with the app.

## 2 Related Works

Recent generative audio systems rely on artificial intelligence and machine learning for creation and manipulation of sound data. Certain products depend on Digital Audio Workstations (DAW), such as Avid Pro Tools [11] or iZotope's mixing product suite [7] to support professional musicians who are familiar with advanced musical concepts. Such systems require layered knowledge and experience with waveform editing, rendering the musical outcome to be fully dependent on the user's abilities and talent. Conversely, applications designed for novices such as Splash Music [6], Amper [2], OpenAI's MuseNet [14] or Jukebox [5], allow little creative input for the users in constructing the musical outcome. With these kinds of systems, the user only provides high level input such as mood, length, or style, while the AI generates the music without supporting ongoing creative input for the users. Santo et al. [16] identified that users would like a co-creative to provide some control over the output. As Tanaka et al. [17] found with their co-creative musical systems, "The ability of the listener to distinguish his own contribution within the total resulting music is a crucial element in granting musical agency to individual users."

For mashup applications, too, recent efforts tend to simplify the interaction design, which limits the creative expression and control of the user. MixMash [10], for example, presents users with a song proximity map but does not provide an interface for users to creatively generate full songs. Other systems such as AutoMashUpper [3] and PopMash [19] pose creative constraints, whether it is limiting the songs a user can work with or limiting the user's creative potential by providing a overly technical user inter-



face. These systems also do not allow users to choose any song of their liking, which limits personalization and engagement. DropMix [12], on the other hand, does provide commercial songs for users to mashup. However, DropMix's song library is limited and the system does not allow the user to engage creatively in constructing the final product. Mixboard was designed to address these challenges, providing users with ongoing AI-driven creative input during the construction of their songs.

### 3 Web Application Overview

The first implementation of Mixboard was designed for the Web [15]. The application allows users to select any four songs from Spotify and organize them over a visual canvas. The users can drag song album art onto the canvas, positioning them over four lanes: Vocals, Instruments, Bass, and Drums. These stems have been source separated using Demucs [4]. Users can then edit the length and location of each segment by dragging and dropping segments over the canvas. The system selects the optimal key and tempo for the mashup, and stretches and transposes all songs segments to the optimal tempo and key using Elastique by Zplane [20]. It also makes high level creative suggestions such as providing templates for songs and selecting the particular audio segments for each placed segment on the canvas. In a set of comprehensive user studies [15], we found that the majority of users asked for more control over their creations. Additionally, we found that while users may have started their mashup process by leaning on the AI-powered features to select random songs with (*Choose for Me*) or determine the placement of their songs with (*Surprise Me*), no user exclusively used the AI features; this indicates that even novice users were capable and willing to explore more nuanced AI-powered features, but they still wanted to exert their own creative goals themselves.

### 4 iOS Application Overview

To address our initial evaluation findings, we developed a new iOS version of the app. The iOS app interface can be seen in Figure 1. A video demonstration of the application can be viewed here: <http://bit.ly/mixboard>. Three main features were added to the application in an effort to provide more control to the users, while still providing meaningful AI input. To allow users to better decipher between the components of the mashup, *Mute* and *Solo* functionalities were added to each lane. To provide users with more control over which audio segment is chosen by the AI for each section, we added a *Shuffle* function:

*Mute*: Turning on *Mute* for a lane will silence corresponding sound pulled from songs placed in that lane. This lets the user silence lanes while listening to live playback, enabling the user to zero in on sounds they want to highlight or remove.

*Solo*: Turning on *Solo* will only play sound generated from that corresponding lane.

*Shuffle*: After clicking a placed segment, a *Shuffle* button appears next to the *Delete* button. When *Shuffle* is clicked, the system will sort through all available relevant segments to pull another segment that matches the length of the placed segment. Given the high volume of requests for users to select specific segments from songs, the *Shuffle*

function is designed to grant the user more control choosing the segment, while still allowing the AI to make an informed decision on which audio segment would fit well.

In addition to these changes, the iOS version also prompts users to log in with their Spotify accounts, which leads to their most recently played songs to display in the *Spotify* window of the song selection window. This allows for easier and faster personalization, which was requested by many users. The iOS version also removed the *Generate* button, and replaced it with *Play/Pause* to reduce wait time in listening to a mashup.

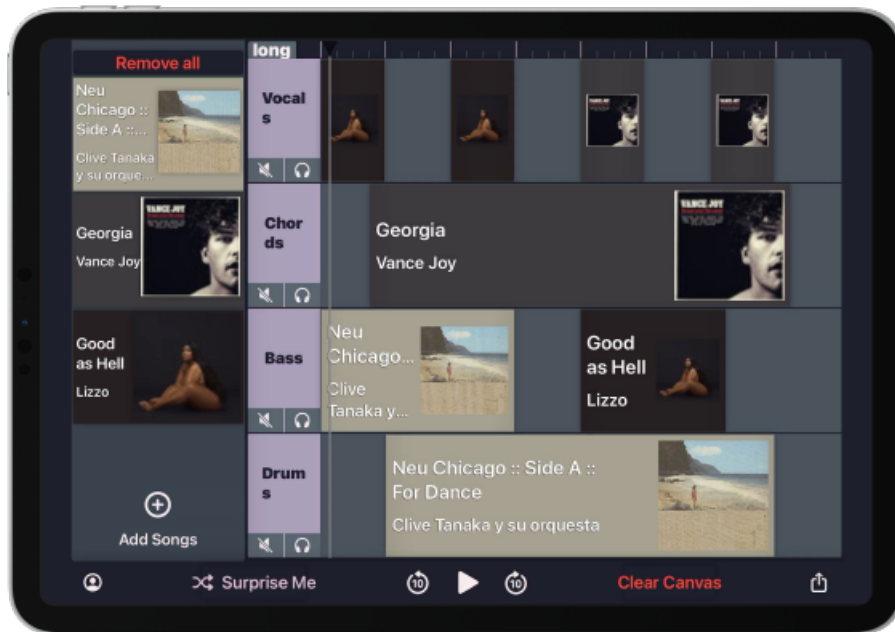


Fig. 1. The iOS version of Mixboard, rendered on an iPad 2 in Dark mode

## 5 Evaluation

We conducted one study on the iOS version of Mixboard using a tablet to maximize screen size. We recruited 20 participants, 16 of which participated in research of the web version. The participants ranged from 22 - 27 years of age, and no one held more than a year of professional or recreational music mixing experience. Participants were given up to 30 minutes to interact with the system; the audio and visual content of the device was recorded throughout the experiment. After the experimentation phase ended, subjects participated in a semi-structured interview and survey. The interview included questions that explicitly asked about how the participant liked and used the

three new control-granting features. The survey asked users about their experience using 20 Likert-scale questions, some of which were adapted from previous musical AI experiments [9]. Two survey questions asked participants to rank potential features in terms of how interested they were in trying the feature and how effective they perceived the features could be in helping them create better mashups. The 27 features included in this section all came from previous participants' desires or misconceptions of Mixboard; these ideas both further expanded existing functionality, e.g. lane labels to set expectations on what to hear, and generated functionality, e.g. a song recommendation system based on the selected songs' tempo or key. The survey data was aggregated to generalize findings quantitatively by assessing the measures of central tendency of this study against previous studies conducted.

## 6 Results

Results from the 20 Likert-scale measures are shown in Figure 2. Two major system bugs were identified during research, one of which broke *Shuffle* and the other frequently broke the *Play/Pause* button. The team was able to identify these issues and fix them after the 8th study. As such, survey means were calculated across all studies (labeled as "Study 3 Mean"), as well as specifically for participants 9-20 (labeled as "Study 3 Post-Fix Mean"). ANOVA tests were conducted on all 20 measures across these three groups, and each measure was proven to be statistically significant between groups. After these fixes, the iOS version of Mixboard proved to be more **consistent** ( $mean(\mu) = 1.62$  (decrease of 0.63 from previous research),  $standard\ deviation(\sigma) = 0.8$ ), **well-integrated** ( $\mu = 4.31, +.30, \sigma = 0.74$ ), and **easier to use** ( $\mu = 4.69, +0.27, \sigma = 0.5$ ) than the web version. The iOS version also scored better in the **control** ( $\mu = 3.69, +0.60, \sigma = 1.15$ ) and **need for more learning** ( $\mu = 2.23, -0.35, \sigma = 1.02$ ) measures than the previous version of the system. Interestingly, the average for **automation** ( $\mu = 2.85, +0.47, \sigma = 1.18$ ) moved closer to 3, meaning more participants "neither agreed nor disagreed" with the statement, "The system should automate more of the composition process for me." There was minimal difference in the **creative expression, trust, learnability, and user confidence** measures, which demonstrated that the new version's changes were not noticeably detrimental to the well-favored user experience of the system.

Technical errors impacted 9 screen recordings. The team decided to only analyze recordings that captured the full experimentation period, so 11 screen recordings were analyzed. *Mute* was the most commonly used feature, with only 2 of the 11 users observed choosing not to interact with the feature at least once during production. It is possible that returning participants were more drawn to interact with the feature given its newness; some returning participants requested this feature previously, which could have further motivated its use. In the features aspect of the survey, only 7 features received strictly positive remarks, meaning no participant stated they were "Not interested" in trying the feature, and no one believed the feature worsen the experience. Each of these features would grant the user more control and improve the quality of the final product. All 20 participants chose to use the full 30 minutes to experiment with the system, and each participant stated they would want more time with the system.

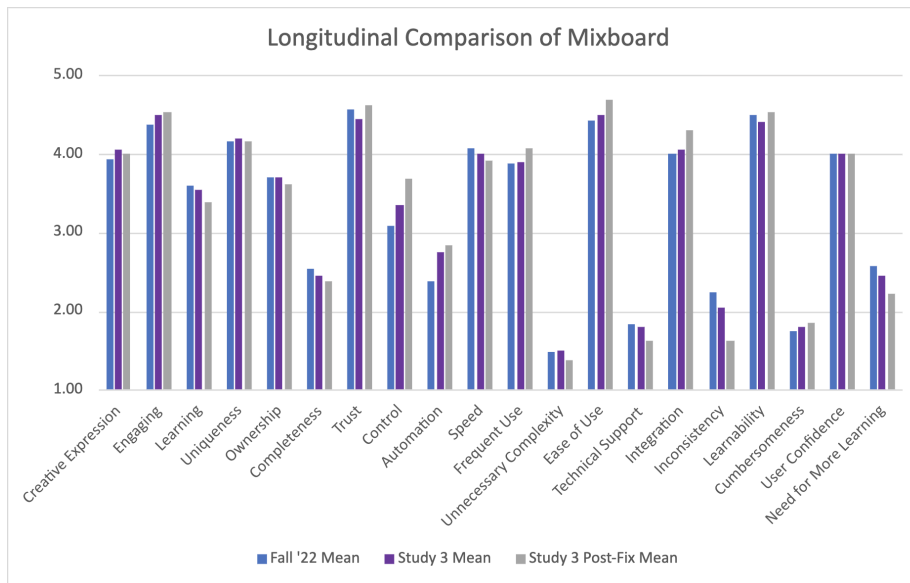


Fig. 2. Longitudinal Comparison of Mixboard

## 7 Observation and Discussion

### 7.1 Version Comparison

We asked participants "Do you think the system provides you too much, too little, or just enough control over your creations?" 8 participants stated "Just enough", and another 10 qualified their "Just enough" responses by saying they would want more control as time went on. This data paired with the improved **control** measure score indicate that this version achieves a more desirable balance between automation and user control; however, there is more to be desired. One returning participant shared, "I like that I can now change it (using *Shuffle*), but I think it'd be more of a unique experience if I could choose which part of the song. Whenever I think of a song I want, I have a specific part I want to add, not just the entire song" (P17.7). Furthermore, participants generally used *Choose for Me* and *Surprise Me*, two AI-driven features, less frequently than in the web version, which could be due to returning participants having a clearer vision of what they want to create or because their pre-existing system knowledge meant they had less to explore. Nearly all returning participants supported the transition from web to tablet, yet 7 of the 16 participants stated they felt less precise without using a mouse or bigger screen. One of these 7 participants stated she felt she had less control over her mashups in this version compared to the web, making her the only returning participant who said they lost control in a negative sense. Participants often stated *Solo* and *Mute* made it easier to identify sounds they wanted to accentuate or eliminate, which granted more control. *Shuffle* likely should remain, even if more data should be gathered around the feature when it works.

## **7.2 Achieving Long-Term Co-Creativity**

The juxtaposition captured by our results demonstrates how difficult it is to provide a universal co-creativity balance that would be appreciated by all users in longitudinal studies. One participant stated throughout the study that his expectations had changed due to capabilities and limitations experienced in the previous study, "I can't necessarily choose the exact seconds of a track, so knowing that means I have to be very open with the vision going into this...if the system's already going to choose the parts of the track for me, then I feel like trying to put specific tracks down is in conflict with that." This reflects gradual user trust can also prepare the user for more advanced features. 13 of the 20 participants requested Mixboard's AI to expand to influence their work further; participants most commonly requested suggesting songs or placement based on what they already had selected (10 participant requests) and more information about the AI's decision making process (8 participant requests). It is worth noting that the latter request did not emerge in the first iteration of research studies, again showing how user expectations can evolve. Users were more likely to request these advanced features when they had previously participated in our studies, reaffirming Turchet et al.'s [18] finding that "personalization mechanisms (should be) based on the expertise level of the user." More control could allow these users to evolve their abilities over time, which increases the likelihood of creating works they are happy to claim as their own.

## **7.3 Ethical Standards**

This project was developed by Georgia Tech students for academic purposes. The human subjects research was approved by the Georgia Tech Institutional Review Board. Informed consent was collected verbally and in writing at the beginning of each research study. No compensation was offered to participants. Anonymized data was stored in a secure drive only accessible to the researchers included on the IRB protocol.

The ethics of remixing and redistributing musical works will be addressed in future work of this system. Since Mixboard is not publicly available, there is minimal risk regarding copyright infringement or improper compensation for the artists. Mixboard could greatly increase the number of works that could be used to generate revenue outside of proper royalty structures, namely if a user were to use a mashup on sponsored social media content or to sell to other social media users. Furthermore, since participating in streaming music slightly increases the likelihood to participate with music piracy [1], we must be especially careful that our users understand the consequences of illegal usage of copyrighted music.

## **8 Conclusions and Future Work**

Mixboard will continue to evolve to address the wealth of user feedback we have collected. The team plans to evaluate whether the system should intentionally scaffold learning via unlockable features or advanced tutorials. While it is clear that different users will have different expectations and different preferences, we will explore variation that would personalize the level of control based on assessing users interaction with the system.

## References

1. Karla Borja and Suzanne Dieringer. Streaming or stealing? the complementary features between music streaming and music piracy. *Journal of Retailing and Consumer Services*, 32:86–95, 2016.
2. Wayne Cheng. Amper: Custom music in seconds, April 2021.
3. Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1726–1737, December 2014. Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing.
4. Alexandre Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.
5. Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music, April 2020. arXiv:2005.00341 [cs, eess, stat].
6. Stuart Dredge. What’s the real end-game for ai music? popgun’s ceo has ideas..., December 2019.
7. iZotope team. Izotope mixing plug-ins.
8. HENRY JENKINS, SAM FORD, and JOSHUA GREEN. *Spreadable Media: Creating Value and Meaning in a Networked Culture*. NYU Press, 2013.
9. Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Honolulu HI USA, April 2020. ACM.
10. Catarina Maçãs, Ana Rodrigues, Gilberto Bernardes, and Penousal Machado. Mixmash: a visualisation system for musical mashup creation. In *2018 22nd International Conference Information Visualisation (IV)*, pages 471–477. IEEE, 2018.
11. Mark Marrington et al. Composing with the digital audio workstation. *The singer-songwriter handbook*, pages 77–89, 2017.
12. Nick Mudry and Riley Davis. Harmonix music systems: Dropmix, April 2017.
13. OpenAI. Introducing ChatGPT, November 2022.
14. Christine Payne. MuseNet, April 2019.
15. Raghavasimhan Sankaranarayanan, Nitin Hugar, Qinying Lei, Thomas Ottolin, Hardik Goel, and Gil Weinberg. Mixboard - a co-creative mashup application for novices. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2023.
16. Luis Espirito Santo, André C Santos, and Marcio Lima Inácio. Focusing on artists’ needs: Using a cultural probe for artist-centred creative software development.
17. Atau Tanaka, Nao Tokui, and Ali Momeni. Facilitating collective musical creativity. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA ’05, page 191–198, New York, NY, USA, 2005. Association for Computing Machinery.
18. Luca Turchet, Carlo Zanutto, and Johan Pauwels. “give me happy pop songs in c major and with a fast tempo”: A vocal assistant for content-based queries to online music repositories. *Int. J. Hum.-Comput. Stud.*, 173(C), may 2023.
19. Baixi Xing, Xiang Zhang, Kejun Zhang, Xinda Wu, Hui Zhang, Jun Zheng, Lekai Zhang, and Shouqian Sun. Popmash: an automatic musical-mashup system using computation of musical and lyrical agreement for transitions. *Multimedia Tools and Applications*, 79(29):21841–21871, 2020.
20. Elastique pro v3 by zplane. [https://licensing.zplane.de/uploads/SDK/ELASTIQUE-PRO/V3/manual/elastique\\_pro\\_v3\\_sdk\\_documentation.pdf](https://licensing.zplane.de/uploads/SDK/ELASTIQUE-PRO/V3/manual/elastique_pro_v3_sdk_documentation.pdf). Accessed: 2023-01-30.

# Global Prediction of Time-span Tree by Fill-in-the-blank Task

Riku Takahashi<sup>1</sup>, Risa Izu<sup>1</sup>, Yoshinari Takegawa<sup>1</sup> and Keiji Hirata<sup>1</sup>

Future University Hakodate  
g2122038@fun.ac.jp

**Abstract.** Time-span trees in A Generative Theory of Tonal Music (GTTM) have global and local relationships. However, no analysis based on global relationships has been done, and higher-order mechanisms have not been clarified. Therefore, in this research, we will clarify this mechanism by masking the time-span tree as it is and using it as a fill-in-the-blank task. To experiment with the fill-in-the-blank task, we vectorized and embedded the tree structure. We also extended the data by changing the pitch and masking. As a result of experiments, it is possible to predict higher layers when masking a small maximum time-span.

**Keywords:** Generative theory of tonal music (GTTM), time-span tree, LSTM, Seq2Seq, blockview, skip-thought

## 1 Introduction

In cognitive science, how humans listen to music is an important factor. When listening to music, people do actively listen while predicting the next melody, harmony, and rhythm. Emotional arousal during listening to music is said to be related to betrayal of the listener's conscious and unconscious predictions[1]. It has been reported that when an unpredictable change in pitch occurs while listening to a melody, people react psychologically and physiologically[2]. It has also been found that the memorability of a melody is related to the predictability [3]. For these reasons, we find that human perception of music is simultaneously analyzing and predicting. Therefore, music analyzers are required to be able to perform multiple analysis and prediction.

In addition, music theory shows that there are not only local relations but also global relations. A Generative Theory of Tonal Music (GTTM) [4], which is a cognitive music theory, describes local and global relationships, and is implemented on computers. In particular, GTTM's time-span tree implementation has been attempted [5, 6]. So far, however, there has been little research about how to implement higher-order mechanisms. Specifically, the current algorithm inputs scores in specific batches and analyzes the time-span tree based on local relationships. However, when humans listen to music, they analyze and make predictions at some point. We never listen to the whole music



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



and analyze it in batches. In fact, referring to Fig. 1, the subtree of C C and the subtree of G G are composed one subtree, so they have a local relationship. Similarly, there is a global relationship between C C and A A. On the other hand, the subtree of G G and the subtree of A A have not local relationship and global relationships. However, the subtree of C C, G G, and A A can be seen as three consecutive syllables of the same two sounds. Therefore, there is also a relationship between G G and A A as syllables. However, due to the reduced subtrees of the GTTM rules, it is difficult to find hidden global relations for subtrees of such syllables and phrases.

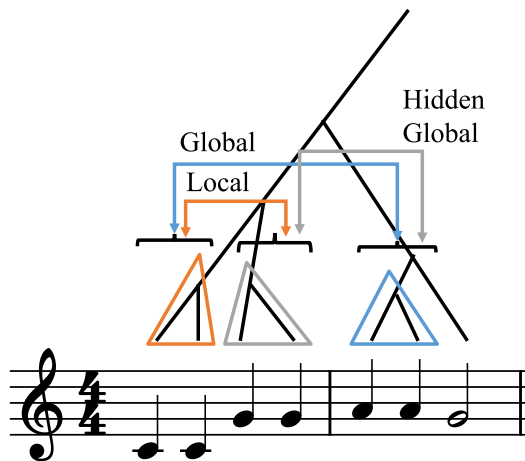


Fig. 1. Local and global relations in time-span tree

In this paper, we consider local and global analyzes of time-span trees. In detail, this paper reproduces the analysis and prediction of human music listening by taking missing time-span trees as inputs and complemented time-span trees as outputs. It enables local and global analysis and prediction using Sequence to Sequence (Seq2Seq) models and fill-in-the-blank tasks. The Seq2Seq model uses LSTM, and the recursive processing of LSTM enables analysis and prediction at the same time. Also, the input must be a tree structure. However, although previous research has discussed the analysis of tree structures[7, 8], using the tree structure itself as data has not been previously considered. In this research, based on the representation proposed in ON-LSTM, a tree structure input is realized by factoring in the hierarchical direction and horizontal directions. It predicts complemented time-span trees from missing time-span trees. In conclusion, this paper proposes local and global time-span tree filling tasks for analyzing time-span trees based on human music perception.

This paper is divided into five sections as follows. First, the definition of the time-span tree used is explained. Section 2 reviews previous studies using time-span tree and deep learning models for tree-structured data. Section 3 details the dataset and data augmentation and embedding used in this paper. Section 4 introduces the Seq2Seq model used for the fill-in-the-blank task in this paper. Section 5 presents results from



experiments to predict subtrees from subtrees. Section 6 presents a discussion of the results. Section 7 describes conclusions and future prospects.

## **2 Prerequisites and Related Work**

Hamanaka et al. [5] analyzed time-span tree obtained by GTTM using deep learning. At low-level boundaries, the accuracy is 0.03 points higher than the conventional method, and the metrical structure has almost the same performance as the conventional method. However, this method is analyzed based on the theory of GTTM, and it shows that it cannot be analyzed without analyzer of GTTM. Hamanaka et al. [6] realized time-span analysis and melody morphing using deep learning. The research makes it possible to analyze time-span trees by learning the order of melody reduction. However, it does not take a tree structure as input, and also requires a melody input.

Ordered Neurons LSTM (ON-LSTM) [7] showed a function called Cumax. This function allows deep learning to learn the composition of the hierarchical structure. Then, the paper demonstrated that the function is effective in unsupervised learning. It shows that it is similar to analyzing a tree structure by a person who does not know the rules. However, it is not extended from tree structure to learning tree structure. Pyraformer [8] investigated whether hierarchical attention mechanisms are effective for long-term dependencies on time-series data. In addition, Pyraformer reduced the amount of calculation compared to the conventional method. However, it requires a large amount of data, well exceeding the number of pieces of music that a human being can listen to in a lifetime.

In conclusion, there is no model that predicts a tree structure from another tree structure, and either rules are used, or a large amount of data is required to acquire the hierarchical structure. Therefore, it is necessary to research a model that predicts and analyzes tree structure with a small amount of data without knowing the theory, similar to people listening to music.

## **3 Create the Experimental Dataset**

The size of the tree structure varies greatly both vertically and horizontally. This tendency is particularly strong in the case of binary trees. There are two cases of extreme tree structures. One is when every possible reduction always occurs at each layer. In this case, since there are many tones, the sequence length is long, but the number of layers is relatively small. On the other hand, there are cases where the reduction is done only once at each Layer. In this case, even if the number of tones is small and the sequence length is short, the number of layers increases in proportion to the number of tones. As a result, when a tree structure is used as input, the size of the data differs greatly. Therefore, by using different learning methods for the vertical and horizontal directions, the effects of each are reduced and, with the implementation of deep learning, learns rules related to time-span trees. In the following, four procedures for solving the subtree filling problem of a time-span tree with Seq2Seq.

### 3.1 Data Collection

A dataset analyzing time-span trees of 300 songs has been published [9]. The time-span tree of the dataset is 8 bars long. Time-span trees are excluded that are ambiguous and subject to multiple analyzes due to preference rules. Furthermore, excluding the time-span tree whose sequence is long and sparse, the number of songs analyzed in this research is 279. The number of minimum layers is 5 and the maximum is 10. Also, the number of minimum notes is 10 and the maximum is 80.

### 3.2 Splitting the Dataset and Data Augmentation with Transposition

279 songs are divided into 8 to 2, and divided into 223 songs as learning data and 56 songs as test data. After that, the learning data is split 8 to 2 into training data and validation data, resulting in 178 songs in training data and 45 songs in validation data. Additionally, as a data augmentation, the training and validation data pitches are changed +2, +4, +5, +7, +9, +11. This multiplies the original data by 7.

### 3.3 Vectorization of Time-span Tree

The tree structure is constructed so that it is easy for humans to understand. However, it is a format that is difficult to handle for Artificial Intelligence such as deep learning. Therefore, blockview was proposed by ON-LSTM [7] as a vector representation of the tree structure. This research extended blokview, which normally analyzes in parse trees, to time-span trees. This research extends time-span trees to blockview. Also, smaller parts are zero-padded. In detail, vectorization is performed so that the maximum lengths in the vertical and horizontal directions can be obtained.

In the original time-span tree, the duration becomes the total value as it is simplified. But in this research, it is not the total value in order to avoid the prediction becoming deterministic due to the duration. In addition, when using the total value, it is possible that the accuracy of the prediction will be greatly affected. The reason is that all durations that appear must be labeled, and with the current number of data, the data will be sparse. For these two reason, duration is not a total value in this research.

Fig. 2 shows the original time-span tree and the time-span tree vectorized by the blockview. Note id 1 wins the most, so only notes with note id are vectorized in fourth layer. Originally, the simplification of note id 4 and 5 is ambiguous whether it is the second layer or the first layer. But for the sake of simplification, it is the first layer that can be considered. For note id 1, the duration is 0.75, 1.5, 3, but as We said earlier, we can't consider all combinations of numbers, so we consider it only as 0.75.

### 3.4 Vertical Embedding

To predict the tree structure, the tree structure needs to be embedded into the latent space. Time-span trees vectorized by blockview are split vertically. This split vector is called a timestep. Timesteps have more similarity in elements as they go up in the layer. To take advantage of this feature, we use skip-thought [10]. Skip-thought captures a latent space from some sentence that predicts the sentences before and after it. Fig. 3

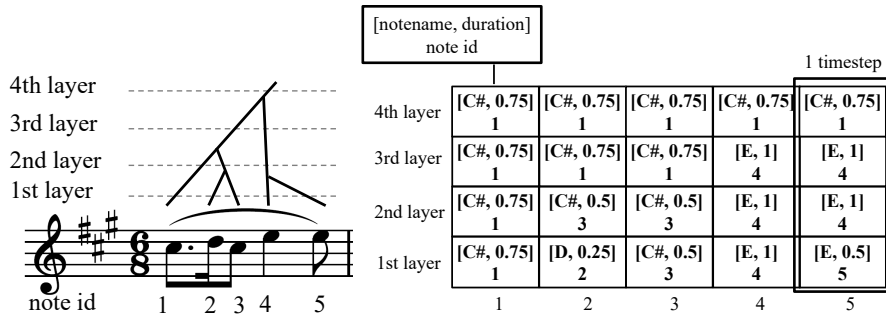


Fig. 2. The original time-span tree and corresponding blockview

shows the skip-thought architecture for embedding timestep vectors. In addition, it is a Seq2Seq model that outputs a timestep vector with the latent space by skip-thogfht as input. Therefore, information above the maximum time-span must also be masked for a complete gap-filling task.

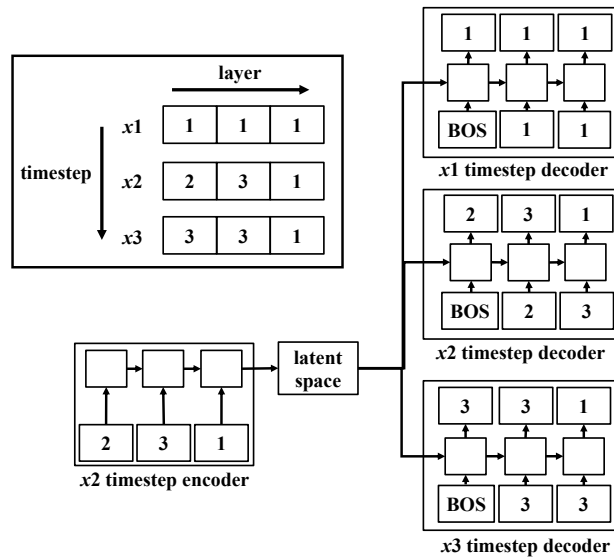


Fig. 3. Embedding x2 timestep by skip-thought

### 3.5 Data Augmentation and Create Fill-in-the-blank Tasks by Masking Blockview

By creating the experimental dataset, the training data is 1246 songs and the training data is 315 songs. Also, the test data is 56 songs. However, if one mask is applied to one song, the data is insufficient for deep learning. Therefore, we create all possible masked time-span trees. For example, if a time-span tree has  $n$  notes,  $n-1$  time-span trees are obtained as masked time-span trees by subtracting the head of the time-span tree. Using Fig. 2 as an example, mask all but the maximum time-span with note id=1. Two examples are shown below.

If the note id is 2 in Fig. 4, the maximum time-span is only the first layer, so only the time step with note id 2 is masked. On the other hand, if the note id is 4 in Fig. 4, the maximum time-span is the second layer, so the timestep with note id 4 and the timestep with note id 5 that is reduced to 4 will be masked. In addition, in this research, since the latent space by skip-thought is input and the time step vector is output, it is necessary to mask the information above the maximum time-span tree in order to complete the blank filling task.

By masking, we obtained 43253 training data, 10780 validation data, and 1833 test data. For training and validation data, we obtained 34 times as many as original ones, and for test data, 32 times as many as original one.

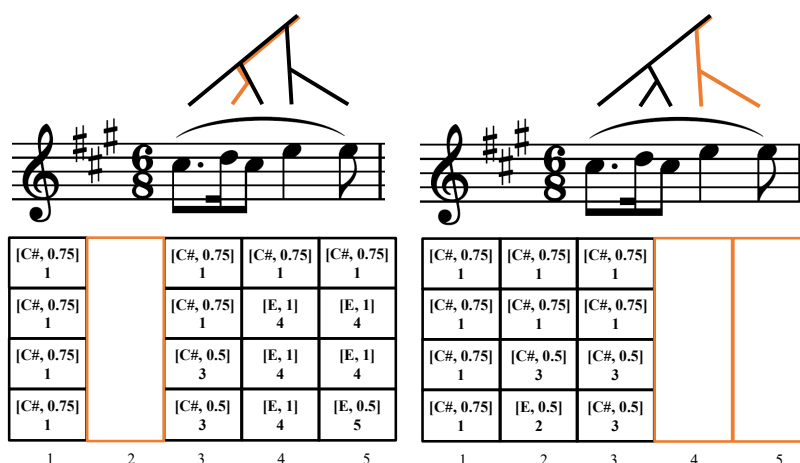


Fig. 4. (left) Masking the maximum time-span with note id 2, (right) Masking the maximum time-span with note id 4.

## 4 Overview of the Seq2Seq model

The purpose of this paper is to clarify the local and global relationships of time-span trees as a high-level analysis. To realize the implementation of this relationship, it is

necessary to do analysis and prediction at the same time. To analyze and predict at the same time, we propose a Seq2Seq model with LSTM. The Seq2Seq model realizes analysis and prediction at the same time by recursive processing of LSTM. In addition, it learns local and global relationships through a time-span tree fill-in task. Also, by using single-head attention, learn the relationship between local and global. Specifically, the decoder in the Seq2Seq model adds attention and computes the relationship between the masked time-span tree and the hole-filled time-span tree.

The Seq2Seq model used in this experiment takes skip-thought embedding as input and outputs a timestep vector as output. At this research, by solving the fill-in-the-blank task, it learns the global tree structure relationships. The LSTM used for the Seq2Seq model is bidirectional in the encoder and unidirectional in the decoder. Also, decoder use attention to help them learn global relationships. Fig. 5 shows the outline of the Seq2Seq model when attention is used.

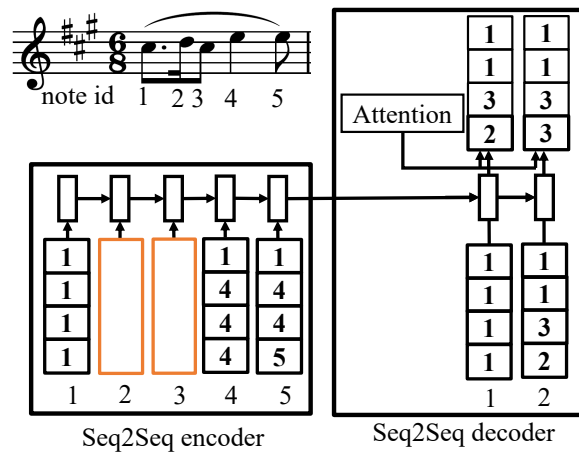


Fig. 5. Outline of Seq2Seq model using attention

Conventionally, attention takes all correspondences, but since fill-in-the-blank task, it does not calculate the masked timesteps. Fig. 6 shows the attention calculation in this research. This avoids the state where the correct answer data paired with the masked timesteps are visible due to attention. Self-correspondence is not used because the same problem can occur and affect the prediction of other masked time-span trees.

Also, this research, each note is a multi-hot vector by combining a one-hot vector for duration, a one-hot vector for octave, a one-hot vector for note name, a label indicating padding, and a label indicating mask. In addition, when outputting, padding labels are combined with duration, octave, and note name, and softmax is calculated as a one-hot vector. After that, the loss is calculated using the categorical cross-entropy as the loss function. Details of each category are shown in Table 1. The masked parts are note id 2 and 3 in Fig. 5, backpropagation is performed to reduce the loss of note id 2 and 3.

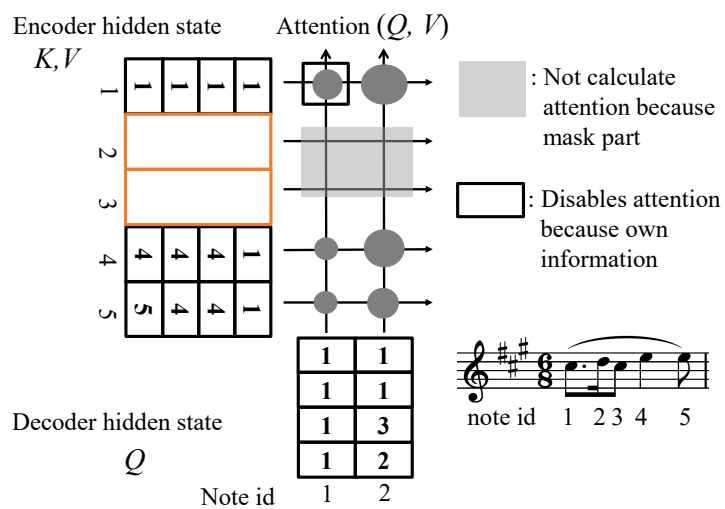


Fig. 6. Calculating the attention of the timestep vector in Fig. 5

Table 1. multi-hot vector ravel

category	contents of label	number
mask	mask or not	1
padding	BOS, EOS, padding for sequences, padding for layers	4
duration	0.125, 0.1667, 0.25, 0.3333, 0.375, 0.5, 0.625, 0.6667, 0.75, 0.875, 1.0, 1.167, 1.25, 1.333, 1.5, 1.625, 1.75, 2.0, 2.25, 2.5, 2.75, 3.0, 3.5, 3.75, 4.0, 4.5, 5.0, 6.0	28
octave	4, 5, 6, 7, 8	5
note name	C, C#, D, Eb, E, F, F#, G, G#, A, Bb, B	12

## 5 Experiment Results

Using the data set prepared in section 3, we show the results of training with the model in section 4. To see if attention is valid for local and global information, we compare it with a normal Seq2Seq model without attention.

### 5.1 Parameter tuning of Seq2Seq model

Deep learning requires learning with optimal parameters, but learning with all data requires a huge amount of computation for skip-thought and learning with Seq2Seq models. Therefore, we decided to find the optimal parameters for 178 training data and 45 validation data before data augmentation, and use those parameters when learning with all data.

To determine the parameters of the Seq2Seq model, the parameters were tested four dimensions: 200, 300, 400, 500, in the latent space of skip-thought and the hidden layer of the Seq2Seq model. Also, the seq2seq parameter tried three learning rates: 0.001, 0.0001, 0.00001. We trained Seq2Seq over 50 epochs and compared the loss function with the validation data. As a result, the loss value was lowest at 2.697 with a skip-thought dimension of 300, a hidden layer dimension of 200, a learning rate of 0.0001, and a batch size of 64. The top 10 losses are shown in Table 2. 6 of the top 10 losses no longer had a loss update within 10 epochs. Also, the loss was not updated when the learning rate was high, and the loss value was large when the learning rate was low.

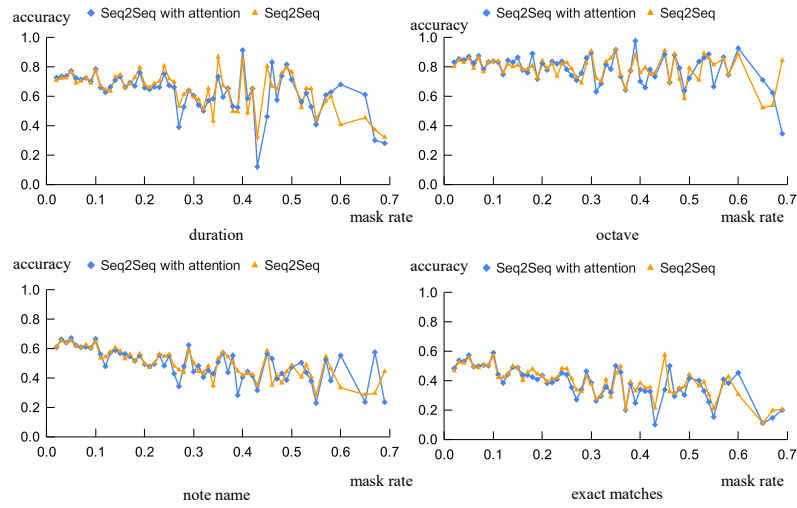
**Table 2.** Validation loss value

thought dim	hidden layer	learning rate	batch size	epoch	validation loss
300	200	0.001	32	5	<u>2.730</u>
		0.0001	32	39	2.732
			64	29	<u>2.697</u>
400	400	0.0001	32	17	2.748
	500	0.001	32	3	2.765
500	200	0.0001	32	43	2.759
		0.0001	64	8	2.742
	500	0.001	32	2	2.760
			64	2	<u>2.7576</u>
		0.0001	64	7	2.761

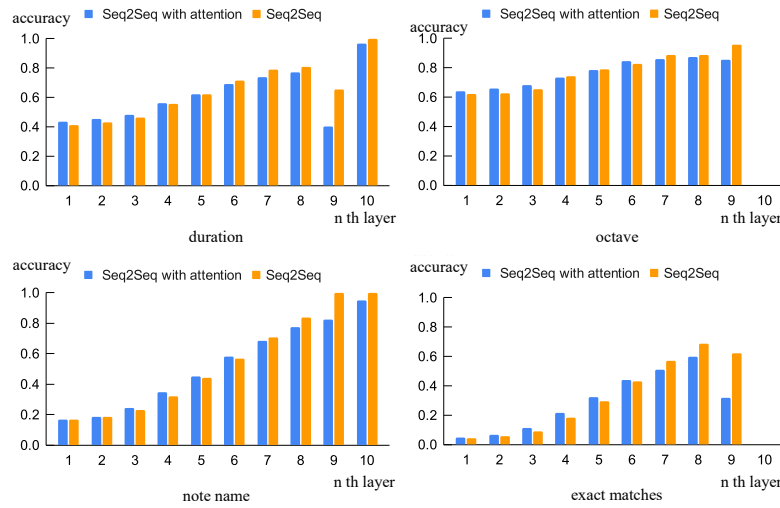
As a result, for the fill-in-the-blank task, the best parameter skip-thought dimension was 300, the hidden layer was 200, and the learning rate was 0.0001. The batch size was 256, that it will be multiplied by 7.

### 5.2 Fill-in-the-blank task results

First, since the timestep was masked as a fill-in-the-blank task, we calculated the accuracy rate for the test data for each masking rate of the timestep including padding.



**Fig. 7.** Accuracy of mask rate in four conditions



**Fig. 8.** Accuracy for each n th layer in four conditions



In other words, in a time-span tree with a maximum layer of 5, the remaining 5 layers are padding. The mask rate is calculated by dividing the masked timestep by the total timestep. Fig. 7 four accuracy for each mask rate. As a result, the prediction for the pitch name had the lowest accuracy without exact matches. The duration was the worst when the mask rate was 0.43, the accuracy was 0.1208 with attention, and the accuracy was 0.3208 with the normal model.

Next, we show the result of whether the prediction was correct as a layer. Fig. 8 shows the four accuracy for each  $n$ th layer. In the graph, padding has been removed to show how well the predicted timesteps restore the original time-span tree. Therefore, the total number of predictions made decreases with each layer. As a result, we were able to confirm the pure accuracy for each layer. In all four results from first to fifth layer, Seq2Seq with attention is equal to or better than normal Seq2Seq. For sixth layer, Seq2Seq with attention gave better results for octaves and note names. However, normal Seq2Seq performed better in all results from seventh to tenth layer. For ninth layer, the duration was 0.4036 for Seq2Seq with attention and 0.6539 for normal Seq2Seq. The result was 0.3677 lower with attention and 0.1515 lower with normal, than eighth layer. In particular, Seq2Seq with attention, the accuracy was the lowest for all layers. Octaves were not predictable at all for tenth layer. Along with that, the exact matches was also completely unpredictable in the case of tenth layer.

## 6 Discussion

Let us discuss the following four issues.

- The parameter tuning of the Seq2Seq model : The top 10 losses were presented, and 6 of them completed training within 10 epochs. The learning has not converged from the general case of deep learning. It is necessary to devise more evaluation functions.
- The treatment of duration : In GTTM rules, simplification is performed using the total value of duration. Due to the number of labels, this study did not use total values. However, this method is not to learn the higher the hierarchy. To discuss this issue, we implement a loss function that recursively predicts the duration and compares the lengths.
- The data size is small : In fact, the most predicted factor was octave, and the next was duration. This is thought to be due to data augmentation for the pitch during learning. It is necessary to increase the total number of time-span trees and learn the data with a suitable distribution. In addition to the number of data, it is also necessary to discuss for labeling. This time, the note names are simply labels, but we should also consider the difference in how many notes are separated from each other.
- Using attention : The reason why there was no difference between Seq2Seq with attention and normal Seq2Seq is that attention was single-headed. Single-headed attention takes only one relationship. On the other hand, multi-head attention takes multiple relationships. Therefore, multi-head attention may improve subtree-to-subtree prediction.

## 7 Conclusion

We proposed learning a global tree structure by a fill-in-the-blank task of a time-span tree. As a result, Small subtrees within a time-span tree can now be predicted based on local and global relationships. However, the larger the subtree, the lower the accuracy. In addition, it is now possible to make predictions about the upper layers. However, the lower the layer, the more difficult the prediction, and the first layer could hardly be restored. Also, using attention did not show any significant improvement in accuracy.

The points to be improved in the future are as follows. Since the blockview is divided vertically as a timestep and the skip-thought is used as an embedded expression, it is possible that the constraints before and after are not completely predicted. This could be improved by changing the Seq2Seq decoder activation function to Cumax, which is used in ON-LSTM. Also, by taking the relationship between attentions like pyraformer, it may be possible to take the relationship in units of subtrees.

## 8 Acknowledgements

This work was supported by JSPS KAKENHI Grant number 21H03572.

## References

1. Meyer, L.B.: Meaning in music and information theory. *The Journal of Aesthetics and Art Criticism*, 15(4), pp.412—424. (1957)
2. Egermann, H., Pearce, M. T., Wiggins, G. A., McAdams, S.: Probabilistic models of expectation violation predict psychophysiological emotional responses to live concert music. *Cognitive, Affective, & Behavioral Neuroscience*, 13(3), pp.533—553. (2013)
3. Agres, K., Abdallah, S., Pearce, M.: Information-theoretic properties of auditory sequences dynamically influence expectation and memory. *Cognitive Science*, Vol. 42, pp.43—76. (2018)
4. Lerdahl, F., Jackendoff, R.: *A Generative Theory of Tonal Music*, The MIT Press, Cambridge (1983)
5. Hamanaka, M., Hirata, K., Tojo, S.: deepGTTM III: Multi task Learning with Grouping and Metrical Structures. *the 13th International Symposium on Computer Music Multidisciplinary Research*, pp.161—172. (2018)
6. Hamanaka, M., Hirata, K., Tojo, S.: Time-span Tree Leveled by Duration of Time-span. *the 15th International Symposium on Computer Music Multidisciplinary Research*, pp.155—164. (2021)
7. Shen, Y., Tan, S., Sordoni, A., Courville, A.: Ordered Neurons: Integrating Tree Structures into Recurrent Neural Network. *In Proceedings of International Conference on Learning Representations*, New Orleans (2019)
8. Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., Dustdar, S.: Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. *International Conference on Learning Representations*, Online, (2022)
9. GTTM Database, <https://gttm.jp/gttm/ja/database/>
10. Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., Fidler, S.: Skip-Thought Vectors. *Advances in Neural Information Processing Systems* 28, Montreal (2015)

# Music Emotions in Solo Piano: Bridging the Gap Between Human Perception and Machine Learning

Emilia Parada-Cabaleiro<sup>1,2,3</sup>, Anton Batliner<sup>4</sup>, Maximilian Schmitt<sup>4</sup>,  
Björn Schuller<sup>4,5</sup>, and Markus Schedl<sup>1,2</sup>

<sup>1</sup> Institute of Computational Perception, Johannes Kepler University Linz, Austria

<sup>2</sup> Human-centered AI Group, Linz Institute of Technology (LIT), Austria

<sup>3</sup> Department of Music Pedagogy, Nuremberg University of Music, Germany

<sup>4</sup> Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany

<sup>5</sup> GLAM – Group on Language, Audio & Music, Imperial College London, UK  
emiliaparada.cabaleiro@hfm-nuernberg.de

**Abstract.** Emotion is an important component of music investigated in music psychology. In recent years, the use of computational methods to assess the link between music and emotions has been promoted by advances in music emotion recognition. However, one of the main limitations of applying data-driven approaches to understand such a link is the scarce knowledge of how perceived music emotions might be inferred from automatically retrieved features. Through statistical analysis we investigate the relationship between perceived music emotions (rated by 41 listeners in terms of categories and dimensions) and multi-modal acoustic and symbolic features (automatically extracted from the audio and MIDI files of 24 pieces) in piano repertoire. We also assess the suitability of the identified features for music emotion recognition. Our results highlight the potential of assessing perception and data-driven methods in a unified framework.

**Keywords:** Music emotion recognition, multi-modal features, perception

## 1 Introduction

Following decades of research about music emotions in psychology [1], an increasing interest in investigating music emotions through computational methods has been driven by advances in music emotion recognition (MER) [2]. However, despite music being a multifaceted channel characterised by a variety of communication modalities, such as acoustic cues, music syntax, or lyrics, multi-modal MER is still under-investigated, in part due to the scarcity of corpora [3, 4]. In addition, since emotions are subjective concepts for which a *ground truth* does not exist, emotion recognition systems rely on a *gold standard*, i. e., labels based on some consensus annotation [5]. Still, the validity of MER labels is often questioned due to the limited number of annotators [6]. Note that, throughout the article, we will refer to *gold standard*, a standardised term in *affective computing* [7], which is more appropriate than *ground truth* [8].



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

To assess how perceived music emotions can be mapped onto machine-readable features, we present a perceptual and data-driven study based on 24 classical piano pieces. Through statistical analysis, we identify the acoustic and symbolic features most suited to infer a categorical and dimensional gold standard, based on ratings by 41 listeners. Finally, to evaluate the generalisability of our results, we assess the machine learning (ML) performance obtained with different feature sets on EMOPIA [4], a multi-modal pop piano music corpus for MER. In sum, we assess two research questions (RQs):

**RQ1:** Which are the most appropriate multi-modal features to automatically identify emotions perceived in piano music?

**RQ2:** Can the suitability of these features be generalised to other dataset?

## 2 Materials and Methods

### 2.1 Musical data and emotion models

We concentrate on classical western compositions for piano solo, by that minimising the influence of genre and scoring diversity. As we aim to assess both acoustic and symbolic features, the dataset introduced by Poliner and Ellis [9], containing both recordings and MIDI files, was considered for the perception study and the feature assessment. Although developed for automatic music transcription, this dataset was chosen due to its suitable repertoire and considering the limited multi-modal corpora for MER. From the 29 files available, 24 with a homogeneous musical discourse, i. e., without contrasting sections that may lead to several perceived emotions, were selected. Although we perform the feature evaluation on a reduced data-set of classical piano compositions—which was needed in order to perform a reliable user study, the generalisability of our results will be assessed in RQ2 on EMOPIA, a well-established piano dataset for MER. EMOPIA contains 1 087 clips from 387 songs and is annotated at clip-level according to the 4 quadrants derived from the circumplex model of emotions [10].

We employ the two models predominantly used in research on music and emotion [6]: the dimensional and the categorical one. For dimensions, we employ the circumplex model [10] representing emotions in a 2-dimensional space delimited by arousal (intensity) and valence (hedonic value), generally used in MER [4, 3]. Although research on MER often refers to basic categories, such as those described by Ekman [11], arguments in favour of moving beyond the *Basic Emotion paradigm* when working with musical emotions have been presented [12]. Thus, for categories, we use the *Geneva Emotion Music Scale* (GEMS) [13], a domain-specific categorical model specially developed to investigate music emotions, already used for MER in western classical music [14]. As we investigate perceived emotions, the 10-factorial version of GEMS<sup>6</sup>, used in Study 2 in [13] to assess perceived emotions, was preferred to the original GEMS (developed to assess felt emotions). Note that GEMS has proven to be as suitable to evaluate perception as felt emotions (see Study 2 [13] as well as [14]). In addition, as typical in MER [4, 3] and in order to assess RQ2, the four quadrants derived from the intersection of the two emotional dimensions will be considered as target categories for the ML experiments. The quadrants are defined as in [15]: Q1 (high arousal, positive valence); Q2

<sup>6</sup> The 10-factors (i. e., emotional categories) are: Activation, Amazement, Dysphoria, Joy, Power, Tenderness, Tranquility, Transcendence, Sadness, and Sensuality.

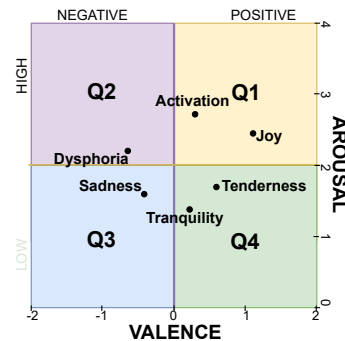


Fig. 1: Emotional categories distributed according to the 4 quadrants. The dots indicate the gold standard, i. e., the mean valence/arousal coordinate across samples per emotion.

(high arousal, negative valence); Q3 (low arousal, negative valence); Q4 (low arousal, positive valence); cf. Figure 1 (positions of categories are explained in Section 2.2).

## 2.2 Annotation process

41 male students participated in the listening experiment as a requirement of a course.<sup>7</sup> The musical samples, each with a duration of 59 seconds, were presented in randomised order over headphones; the responses were given in a forced-choice format through a web-based interface. For each musical sample, the participants had to choose one of the 10 emotional categories, a level of arousal (from 0 to 4), and a level of valence (from  $-2$  to  $2$ ). Note that valence (unlike arousal) can have negative values; thus the scale is not the same but more adequate. We used static annotations instead of continuous, i. e., each annotation was given at sample level. Despite the length of the samples, this was considered the best choice in order to be consistent with the annotations from EMOPIA, the dataset used to validate our results. As already mentioned, to prevent annotation ambiguity due to samples' length, those with a homogeneous musical discourse were selected. Finally, since liking and familiarity have played a role in previous works [16, 17], participants were also requested to indicate in binary form (yes/no) whether they were familiar with the evaluated repertoire and whether they liked it.

To create a gold standard for valence and arousal, we computed the mean across ratings per sample and dimension, as typical in MER [6]. In addition, we also computed the Evaluator Weighted Estimator (EWE), an standard method to compute a gold standard in affective computing [18] that takes into account an individual evaluator-dependent weight for each annotator. The evaluator-dependent weights are the normalised correlation coefficients obtained between each listener's responses and the average ratings across all listeners [18]. As both Spearman and Pearson correlations between mean and EWE are at 99%, we use the mean in the following. To create the categorical gold standard, the emotional factor showing the highest agreement was considered as target category, as typical in MER [6]. In Figure 1, the categories chosen most frequently

<sup>7</sup> Although considering only males' ratings might affect the results, responses by the only three females who took part in the experiment had to be discarded to preserve a coherent cohort.

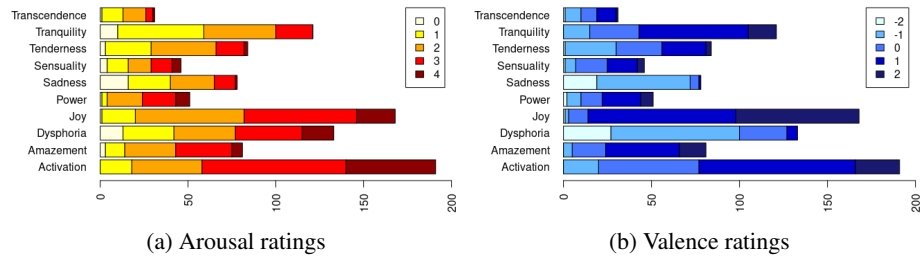


Fig. 2: Distribution of the 984 ratings (41 listeners  $\times$  24 samples) for each dimension.

across samples are shown within the quadrants. The mean arousal and valence ratings across all samples identified with the given categories are shown. For the distribution of all the listeners' ratings across factors and dimensions, see Figure 2.

### 2.3 Feature extraction and processing

Symbolic and acoustic features were extracted from the MIDI and audio files and subsequently concatenated in a feature vector. Concerning the symbolic data, we extracted the features of `jSymbolic 2.2` [19], which include a variety of statistical descriptors related to pitch, rhythm, melody, chords, texture, and dynamics (related to MIDI velocity), i. e., musical properties suitable to automatically capture emotional content from MIDI [15]. Since we aim to evaluate the features in relationship to the perceptual results, we choose `jSymbolic`, whose features are highly interpretable in musical terms. As acoustic representation, we considered the `openEAR emobase` feature set extracted with the default parameters of `openSMILE` [20], which is tailored to model emotions in audio and has been used in the context of MIR as well [21]. `OpenEAR emobase` contains statistical descriptors related to intensity, loudness, pitch, envelope, and spectrum.

After excluding irrelevant features, e. g., those related to the Music Encoding Initiative format for the symbolic and the delta coefficients for the acoustic modelling, 188 symbolic and 494 acoustic features were retained for analysis and subsequently z-score normalised. In order to prevent collinearity [22], redundant features, i. e., those showing a pair-wise correlation of  $r \geq 0.7$ , were automatically identified; the one showing the largest mean absolute correlation was subsequently removed. For this, the correlations were recomputed at each step with the R function `findCorrelation`. This yielded a total of 91 features—68 symbolic and 23 acoustic. From now on, these constitute the 91-dimensional feature vector representing each sample.

### 2.4 Statistical methods

To explore which features might be suitable to predict perceived arousal and valence, Pearson correlation was computed between each feature and the gold standard for each dimension. Since features might also be suitable in combination, two multiple regression models were fitted separately for each dimension. In addition, to assess individual ratings instead of the gold standard, all *raw* responses were directly taken as outcome variable for these models. Note that, as every listener co-occurs in the design with every

song, the variables user-ID and song-ID were considered crossed random effects. The need of applying a multi-level analysis was confirmed by the decreased *Akaike's information criterion* (AIC) of the intercept model with crossed random effects w. r. t. those with only one random effect: for both dimensions,  $p < .001$ . Suitable predictors were automatically recognised through a *Genetic Algorithm* (GA), implemented in R with default parameters and 100 iterations. Subsequently, forward selection was applied in order to evaluate if additional predictors might yield a lower AIC. Given the inherent problems of  $p$ -values [23], in particular for linear mixed models [24], we will interpret the role of the fixed effects according to the regression coefficients.

After identifying suitable features through correlation and multiple regression, in order to visually interpret the suitability of such a features in mirroring the listeners' ratings, we compare perception and classification results. For this, we used *Non-Metric Multi-Dimensional Scaling* (NMDS) solutions [25], which aim at representing the optimal distances between items. To find the optimally scaled data, NMDS is initialised with a random configuration of data points and subsequently finds the optimal monotonic transformation of the proximities. This search for a new configuration is performed iteratively until Kruskal's normalised stress1 criterion or its gradient is below a threshold of  $10^{-4}$ . Since our goal is not to achieve the best possible result through fine-tuning, but to compare classification performance across feature sets while keeping hyperparameters constant, for this experiment, the classification framework (described in Section 2.5) was implemented with default parameters and without optimisation.

## 2.5 Machine learning models and optimisation

Four classifiers, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Random Forest (RF), and k-Nearest Neighbour (k-NN), were implemented. To leverage the advantages of all models, we created a hybrid classifier using late-fusion of results via majority voting, i. e., the class most frequently chosen by the four models was taken as final prediction. We do not concentrate on pushing the approaches towards their limits, but aim at baseline results with 'standard' settings, by this encouraging generalisation of the outcomes. As evaluation metric, we use Unweighted Average Recall (UAR) [7].

The data were randomly split into train, validation, and test. We targeted a similar distribution between classes across quadrants; samples from the same song did not occur in different sets. To increase validity, five different splittings were generated; we report the average results across experiments. The models were built on the scikit-learn python library [26] with the default hyperparameters, except for the following set-up: For the SVM, we use linear kernel and evaluate five different complexities [0.0001, 0.001, 0.01, 0.1, 1.0]. For the MLP, we use batch size 8, two hidden layers, and evaluated the same number (N) of neurons per layer from the following five N [25, 50, 100, 175, 300]. For the RF, we evaluate five different N of estimators [10, 50, 100, 150, 200]. For the k-NN, we evaluate five different N of neighbours [3, 5, 7, 9, 11]. All hyperparameters were optimised independently for each of the five splits via grid search.

## 3 Gold Standard Assessment

As first step to create the gold standard, we evaluated the role of familiarity and preference. For this, multiple regression was performed considering both variables as cat-

egorical predictors and the perceived valence and arousal individually as dependent variables. Our results show that neither preference nor familiarity play a role in the model, neither for arousal, nor for valence ( $p \geq .084$ ). This is also confirmed for within song evaluation: the models yielded  $p \geq .286$  for arousal,  $p \geq .353$  for valence.<sup>8</sup> Thus, in the following, all listeners' responses will be taken into account for our experiments.

The gold standard computed from listeners' responses shows that joy is mainly associated with Q1 (5 songs) and to some extent with Q4 (1 song); activation with Q1 (5 songs) and to some extent with Q2 (2 song); dysphoria with Q2 and Q3 (2 songs each); sadness is clearly associated with Q3 (2 songs); tenderness with Q4 (1 song); tranquility with Q3 and Q4 (2 songs each). This distribution of emotional categories across the bi-dimensional space (cf. Figure 1) is consistent with the one described in previous works (cf. [10] and [1, p. 113]), where joy/dysphoria are associated with positive/negative valence; activation/tranquility are associated with high/low arousal; tenderness/sadness are related to low arousal and to positive/negative valence. This is displayed by the distribution of the dimensional ratings. For sadness, in particular, the ratings are mostly distributed across the lowest and intermediate arousal (cf. 0 to 2 in Figure 2a), and almost all display negative valence (cf.  $-2$  and  $-1$  in Figure 2b).

To gain more insights on the perceptual results, we investigated the relationship between both dimensions. For this, each of them was considered as outcome and predictor, respectively, in a linear model, disregarding the categorical ratings. The positive slope indicates that there is a direct relationship between both variables:  $F = 83.56$ ,  $\beta = 0.30$ ,  $r = 0.28$ ,  $p < .001$ . In other words, as perceived ratings increase in one unit for a given dimension, the model predicts that the perception for the other one will also increase in 0.30 units. Still, the correlation of  $r = 0.28$  indicates only a weak tendency.

Subsequently, to evaluate if the relationship between valence and arousal might be associated with categorical perception, for each emotion, an individual linear model was fitted with the corresponding dimensional ratings. The results show that the positive relationship between both dimensions is only marked for some emotions: the linear regression yields  $p \leq .046$  for amazement, joy, sensuality, and tranquility, i. e., those generally associated with a more positive valence, cf. Figure 2b; for the others,  $p \geq .346$ . Indeed, fitting again the model with the dimensional ratings of only these emotions increased the correlation coefficient ( $r = 0.48$ ), which confirms the positive association between valence and arousal but only within the positive half of the dimensional space, i. e., Q1 and Q4. To reproduce the gold standard and results, please visit our repository.<sup>9</sup>

## 4 Results

### *RQ1: Which are the most appropriate multi-modal features to automatically identify emotions perceived in piano music?*

**CORRELATION ANALYSIS:** To investigate the relationship between the automatically extracted features and the perceived emotional dimensions, correlation analysis was performed. In Table 1, only the top ranked features ( $|r| \geq 0.4$  in at least one dimension), i. e., those showing a moderate correlation, are displayed. Since a relationship between

<sup>8</sup> Bonferroni correction was applied for multiple testing throughout the results.

<sup>9</sup> [https://github.com/SEILSdataset/FeatureEval\\_MER/](https://github.com/SEILSdataset/FeatureEval_MER/)



Table 1: Top ranked correlation with the mean ( $\mu$ ) perceived arousal and valence.

Arousal		Valence	
Feature	$\mu$	Feature	$\mu$
<i>Common Rhythm</i>	-.65	<i>m/M Triad Rat.</i>	-.54
<i>ZCR Skewness</i>	-.57	<i>F0 Quartile3</i>	.53
<i>Note Density</i>	.54	<i>Intensity abs. min.</i>	-.48
<i>Mel. Large Int.</i>	-.49	<i>m/M Mel. 3rd Rat.</i>	-.46
<i>N. Strong Pulses</i>	-.48	<i>Arousal</i>	.46
<i>Standard Triads</i>	-.46	<i>F0 Skewness</i>	-.44
<i>Valence</i>	.46	<i>Similar Motion</i>	-.43
<i>Rat. Strong Pulses</i>	-.42	<i>Rat. Strong Pulses</i>	-.41
<i>BPM</i>	.42	<i>Dynamic Range</i>	-.40
<i>Prev. Dotted Notes</i>	-.41	<i>Dim. Aug. Triads</i>	-.40

both dimensions was shown in the gold standard assessment, these are also included in the correlation analysis. In the following, the correlation results will be interpreted according to [1, p. 113], which summarises the outcomes from music psychology.

**Arousal.** The experimental results are consistent with the general believe that slow and fast mean tempo correspond to music expressing low and high arousal, respectively. This is shown by the positive correlation of arousal with Beat Per Minute (BPM,  $r = .42$ ) as well as by the negative one with common rhythm and prevalence of dotted notes ( $-.41 \leq r \leq -.65$ ), indicating that music characterised by a fast tempo and a prominent use of short (not dotted) notes is associated with higher arousal. Similarly, the use of accents on unstable notes (typically used to express highly aroused music) is shown by the negative correlation of arousal with number and ratio of strong pulses ( $-.42 \leq r \leq -.48$ ): As perceived arousal increases, the amount of strong beat peaks decreases and is diversified towards non-beat ones.

High arousal is also associated with a high sound level, which is confirmed by the positive correlation of arousal with note density ( $r = .54$ ) and the negative one with Zero-Crossing Rate (ZCR) skewness ( $r = -.57$ ). While note density is implicitly related to sound level, a low ZCR skewness can be interpreted as a ‘constant’ (not skewed) distribution of frequency density over time:  $ZCR = 0$  indicates no sound. Besides being consistent with outcomes from music psychology [1, p. 113], our experimental results for arousal also show that an increase in this dimension goes along with a decrease in the use of standard triads w. r. t. other vertical intervals ( $r = -.46$ ). This can be interpreted as an association of high arousal with a more ‘empty’ (without third) sonority.

**Valence.** The small sound level variability typically associated with positive valence is shown by the negative correlation of this dimension with dynamic range ( $r = -.40$ ). Our results are also consistent with the believe that minor/Major music expresses negative/positive emotions [27], as shown by the negative correlation of valence with m/M triad and melodic third ratio ( $-.46 \leq r \leq -.54$ ). Similarly, positive valence goes along with a detriment in augmented and diminished triads ( $r = -.40$ ), which indicates that negative valence is associated with a higher use of dissonant chords. Our results suggest that positive valence is linked to the use of a lower variety of pitches concentrated around high pitch, something that can be related to the common association of joy with bright timbre. This is shown by the positive correlation of valence with the Fundamen-

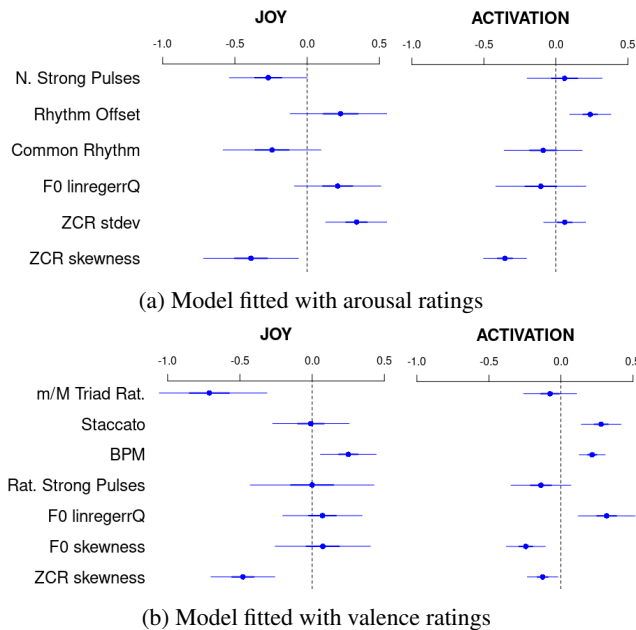


Fig. 3: Fixed effects' regression coefficients (blue dot) and confidence intervals (blue line) for the two models: one for arousal, the other for valence.

tal frequency (F0) quartile 3 ( $r = .53$ ) and by the negative one with the F0 skewness ( $r = -.44$ ): Low F0 skewness indicates a similar distribution of frequencies over time.

Arousal and valence are positively correlated ( $r = .46$ ). Still, the low sound level typically used to express emotions with positive valence and low arousal is also shown by the negative correlation of valence with absolute minimum intensity and dynamic range ( $-.40 \leq r \leq -.48$ ). This indicates that, despite the positive correlation between both dimensions in the investigated samples, the extracted features are also suitable to identify emotions with a positive valence and low arousal.

**MULTIPLE REGRESSION:** To investigate the interplay between the automatically extracted features and the categorical as well as dimensional ratings, the best fitting models, separately identified for each dimension, were also fitted with the subset of dimensional ratings corresponding to each emotional category (cf. Section 2.4). Using the general models tailored to each dimension was preferred to retrieving an individual model per category, to enable comparability. Due to space limitations, in Figure 3, only results for joy and activation, i. e., the two categories with the highest number of observations—joy 168, activation 191, thus showing most robust results—are shown.

The features of the model tailored to recognise arousal include three symbolic, related to rhythm, and three acoustic ones, related to F0 and ZCR. Indeed, both note duration, related to rhythm, as well as intonation and spectral noise, related to F0 and ZCR, are relevant properties for the expression of arousal in music [1, p. 113]. In particular, the higher positive slope of ZCR standard deviation for joy indicates that unlike for activation, an increase in arousal goes along with a higher variability of silent and

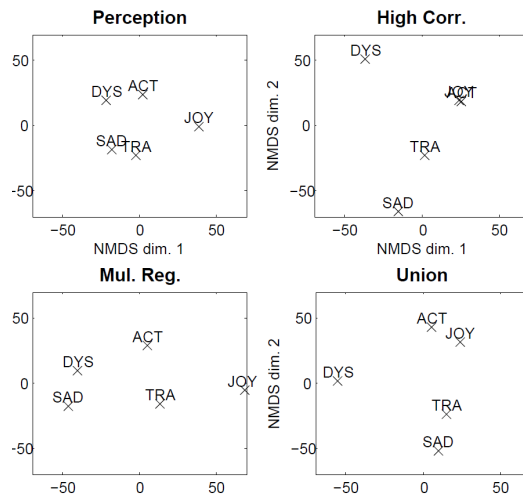


Fig. 4: NMDS for the perception and classification (High Correlation, Multiple Regression, and Union features) of JOY, ACTivation, DYSphoria, SADness, and TRAnquility. Kruskal’s stress: Perception (.097); High Cor. (.093); Mul. Reg. (.024), Union (.006).

dense frames over time. Again, as shown in the correlation analysis, the m/M triad ratio is relevant to predict valence, as clearly displayed for joy. Interestingly, BPM and staccato are meaningful features for the valence model but not for the arousal one. The fact that these features show a relatively marked positive slope—for activation both, for joy only BPM—might again be an indicator of the positive relationship between both dimensions, as shown by the listeners’ association of these two factors with high arousal and positive valence (cf. Q1 in Figure 1).

**PERCEPTION VS CLASSIFICATION:** To further explore the suitability of the identified features for discriminating between the perceived emotions, we compare classification performance with the perceptual results (cf. Figure 4). As there is a relationship between the emotional factors and specific regions of the bi-dimensional space (cf. Figure 1), the features tailored to arousal and valence are both considered for the classification of emotional categories. Three feature sets are assessed: the features with top correlation (High Corr., 17 features), shown in Table 1; the ones used for the Multiple Regression (Mult. Reg., 11), shown in Figure 3; and the union of both (Union, 21). As some features are part of both High Corr. and Mult. Reg., Union contains less features than the sum of these sets. For a description of the features see Table 2. More details are given in the official documentation of `jSymbolic2.2` and `openSMILE.Tenderness` (cf. Figure 1) is not considered, as attributed to only one sample.

The Union feature set, showing the best fit (Kruskal’s stress .006), is the one best mirroring the Perception NMDS: Joy and activation are shown towards Q1; dysphoria towards Q2; sadness and tranquility are close to each other. Although for perception, sadness is more clearly displayed in Q3 than for the Union feature set, this set, combining High Corr. and Mult. Reg., is a less condensed version of the Perception results; cf. Union in Figure 4. Thus, from now on, the Union feature set will be used.

Table 2: Description of the symbolic and acoustic features of the Union set.

Symbolic Features			
<i>Common Rhythm</i>	Most common rhythm in quarter note units	<i>Similar Motion</i>	Fraction of similar movements, e. g., parallel
<i>N. Strong Pulses</i>	N. of beat peaks with magnitudes over 0.1	<i>Staccato</i>	Fraction of notes shorter than 0.1 seconds
<i>Rat. Strong Pulses</i>	Ratio of the two highest beat magnitudes	<i>Note Density</i>	Average number of notes per second
<i>Rhythm Offset</i>	Median absolute duration offset	Acoustic Features	
<i>m/M Mel. 3rd Rat.</i>	Ratio of the minor/Major melodic thirds	<i>Intensity abs. min.</i>	Frame-based absolute minimum intensity
<i>m/M Triad Rat.</i>	Ratio of the minor/Major vertical triads	<i>BPM</i>	Beat per minute
<i>Standard Triads</i>	Fraction of minor or Major triads	<i>ZCR stdev</i>	Standard deviation of the zero-crossing rate
<i>Mel. Large Int.</i>	Fraction of melodic intervals > octave	<i>ZCR Skewness</i>	Skewness of the zero-crossing rate
<i>Dynamic Range</i>	Highest loudness value minus the lowest	<i>F0 Skewness</i>	Fundamental freq. (F0) contour's skewness
<i>Prev. Dotted Notes</i>	Fraction of dotted notes	<i>F0 linregerrQ</i>	Quadratic error of the F0 contour
<i>Dim. Aug. Triads</i>	Fraction of diminished or augmented triads	<i>F0 Quartile3</i>	Third quartile of the F0 contour

**RQ2: Can the suitability of the identified features be generalised?**

To assess the generalisability of the identified features, we performed the classification experiments (optimising the models as described in Section 2.5) on the EMOPIA dataset. To interpret confusion patterns across the dimensional quadrants, i. e., the target categories in EMOPIA, besides the Union dataset (used to assess the RQ1), we now investigate the performance of the Union features tailored to recognise each dimension individually as well. In addition, since the size of EMOPIA enables to carry out a real evaluation of the results beyond NMDS interpretation, the ML models were also trained with all the features (i. e., the 91 described in Section 2.3). Thus, the experiments on EMOPIA were performed with four feature sets: all features (91), Union features tailored to arousal and valence (12 each), and the Union feature set (21).

The results on EMOPIA indicate that training the models with all the features shows a clear differentiation of the arousal dimension: Q1 and Q2 (both with high arousal) are clearly distinct from Q3 and Q4 (both with low arousal) while confused with each other (Q1 with Q2, Q3 with Q4); cf. *All features* in Table 3. As expected, this pattern is enhanced for the features tailored to arousal, which do not contain features tailored to recognise valence information and display a much more pronounced confusion between quadrants of the same arousal level (cf. dark cells of *Arousal selection* in Table 3). In contrast, besides a relatively high recall for Q4 and its confusion towards Q1 (both with positive valence), no clear distinction/confusion pattern is shown for the features tailored to recognise valence; cf. *Valence selection* in Table 3. This feature set yields the worst UAR (39.2 %), and the recall for Q1 and Q4 does not outperform the one achieved by the other feature sets either, which suggests its low capability in capturing information relevant to the target dimension. Finally, the *Union* features (without dimension selection, i. e., A + V) slightly outperform the *Arousal selection* (UAR = 52.5 % vs UAR = 50.7 %), but without reaching the performance of *All features* (UAR = 64.1 %). Again, a differentiation in terms of arousal is displayed.

The experimental results suggest that the arousal dimension is more prominent in the evaluated data, something also observed in emotional speech, where arousal is better represented by acoustic cues than by linguistic ones [28]. The lower efficiency of the features tailored to model valence might be interpreted, to some extent, according to previous works which had shown the difficulties, from a listeners' point of view, of assessing valence, even in music expressing sadness [29], a basic emotion which is, however, clearly associated to negative valence. The classification results achieved with

Table 3: EMOPIA: confusion matrices averaged across splits. Columns show ‘classified as’. UAR for each feature set: All (64.1 %); Arousal (50.7 %); Valence (39.2 %); Union (52.5 %), i. e., Arousal and Valence (A + V).

%	All features				Arousal selection				Valence selection				Union (A + V)			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Q1	81.2	14.1	1.7	3.0	67.1	23.5	3.8	5.6	51.7	27.8	6.8	13.7	65.4	24.8	3.0	6.8
Q2	34.2	55.8	3.8	6.2	33.8	51.2	7.7	7.3	39.2	37.7	8.8	14.2	37.7	50.0	3.5	8.8
Q3	7.6	8.6	61.1	22.7	14.1	8.6	43.4	33.8	29.3	26.8	20.7	23.2	13.1	7.1	48.0	31.8
Q4	12.8	7.8	21.0	58.4	15.5	11.4	32.0	41.1	25.1	19.6	13.2	42.0	14.2	10.5	30.1	42.2

all the features yielded the highest UAR, suggesting that the usability of the *Union* set for MER might be limited. Still, the identified features show reasonable results with a much lower dimensionality, something that might be beneficial for some MER systems.

## 5 Conclusion and Future Work

Besides confirming some of the outcomes presented in music psychology literature, our data-driven approach shows that automatically extracted multi-modal features might be suitable to infer perceived musical emotions. For instance, the statistical analysis suggests that in the evaluated repertoire, empty sonorities might be an indicator of perceived high arousal, while high pitch is related to positive valence. The machine learning experiments show that the features identified to model arousal lead to competitive classification results concerning the quadrants related to the target dimension. In contrast, those identified to model valence are considerably less efficient, which might be explained by the lower characterisation of this emotional dimension in music. Finally, the importance of a multi-modal approach becomes clear when evaluating the feature sets, which despite being selected in a fully automatic manner, encompass both symbolic and acoustic features. In future work, besides investigating a larger dataset from a more varied repertoire, we also plan to assess music with lyrics, by this assessing the suitability of linguistics in the identification of the valence dimension.

## Acknowledgements

This work received support from the Austrian Science Fund (FWF): P33526 and DFH-23.

## References

1. Juslin, P.: Musical Emotions Explained. Oxford University Press., Oxford, UK (2019)
2. Han, D., et al.: A survey of music emotion recognition. *Frontiers of Computer Science* **16** (2022) 1–11
3. Panda, R., et al.: Multi-modal music emotion recognition: A new dataset, methodology and comparative analysis. In: Proc. of CMMR, Marseille, France (2013) 1–13
4. Hung, H.T., et al.: EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. In: Proc. of ISMIR, Virtual (2021) 318–325
5. Cardoso, R., et al.: What is gold standard and what is ground truth? *Dental Press Journal of Orthodontics* **19**(5) (2014) 27–30

6. Gómez-Cañón, J.S., et al.: Music emotion recognition: Towards new robust standards in personalized and context-sensitive applications. *IEEE Signal Processing Magazine* **38** (2021) 106–114
7. Schuller, B., Batliner, A.: *Computational paralinguistics: Emotion, affect and personality in speech and language processing*. John Wiley & Sons, Sussex, UK (2014)
8. Parada-Cabaleiro, E., et al.: Perception and classification of emotions in nonsense speech: Humans versus machines. *PLoS ONE* **18**(1) (2023) e0281079
9. Poliner, G., Ellis, D.: A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing* (2006) 1–9
10. Russell, J.A.: A circumplex model of affect. *Journal of Personality and Social Psychology* **39**(6) (1980) 1161–1178
11. Ekman, P.: Basic emotions. In: *Handbook of emotion*. John Wiley & Sons (1999) 226–232
12. Cespedes-Guevara, J., Eerola, T.: Music communicates affects, not basic emotions—A constructionist account of attribution of emotional meanings to music. *Frontiers in Psychology* **9** (2018) 1–19
13. Zentner, M., Grandjean, D., Scherer, K.: Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion* **8** (2008) 494–521
14. Schedl, M., et al.: On the interrelation between listener characteristics and the perception of emotions in classical orchestra music. *IEEE Transactions on Affective Computing* **9**(4) (2017) 507–525
15. Panda, R., et al.: Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing* **11**(4) (2018) 614–626
16. Schubert, E.: The influence of emotion, locus of emotion and familiarity upon preference in music. *Psychology of Music* **35**(3) (2007) 499–515
17. Pereira, C.S., et al.: Music and emotions in the brain: Familiarity matters. *PloS one* **6** (2011)
18. Grimm, M., et al.: Primitives-based evaluation and estimation of emotions in speech. *Speech Communication* **49**(10-11) (2007) 787–800
19. McKay, C., et al.: jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. In: *Proc. of ISMIR, Paris, France* (2018) 348–354
20. Eyben, F., et al.: Opensmile: The Munich versatile and fast open-source audio feature extractor. In: *Proc. of ACM Multimedia, Florence, Italy* (2010) 1459–1462
21. Shen, T., et al.: Peia: Personality and emotion integrated attentive model for music recommendation on social media platforms. In: *Proc. of the AAAI Conf. on AI, New York, NY, USA* (2020) 206–213
22. Dormann, C., et al.: Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography* **36** (2013) 27–46
23. Wasserstein, R.L., Lazar, N.A.: The ASA’s statement on p-values: Context, process, and purpose. *The American Statistician* **70** (2016) 129–133
24. Baayen, R.H., et al.: Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language* **59**(4) (2008) 390–412
25. Kruskal, J., Wish, M.: *Multidimensional Scaling*. Sage University, London, U.K. (1978)
26. Pedregosa, F., et al.: Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* **12** (2011) 2825–2830
27. Gabrielsson, A., Lindström, E.: The role of structure in the musical expression of emotions. In: *Handbook of Music and Emotion*. Oxford Uni. Press, Boston, MA, USA (2010) 187–221
28. Atmaja, B.: Predicting valence and arousal by aggregating acoustic features for acoustic-linguistic information fusion. In: *Proc. of TENCON, Osaka, Japan* (2020) 1081–1085
29. Eerola, T., Vuoskoski, J.K.: A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music* **39**(1) (2011) 18–49

## Listeners' Perceived Emotions in Human vs. Synthetic Performance of Rhythmically Complex Musical Excerpts

Ève Poudrier<sup>1</sup>, Bryan Jacob Bell<sup>1</sup>, Jason Yin Hei Lee<sup>2</sup>, and Craig Stuart Sapp<sup>3</sup>

<sup>1</sup>University of British Columbia, School of Music

<sup>2</sup>McGill University, Schulich School of Music

<sup>3</sup>Stanford University/CCARH/PHI

eve.poudrier@ubc.ca

**Abstract.** Research on listeners' perceived emotions in music draws on human and synthetic stimuli. Although research has shown that realistic synthetic audio can convey emotions, studies that compare listeners' experience of synthetic audio and human performances are limited. Using short musical excerpts, we investigate the effect of performance (human vs. synthetic) and instrumentation (piano vs. string quartet) as well as the influence of twelve musical features on participants' ratings of five emotional dimensions (mood, energy, movement, dissonance, and tension). Findings show a small main effect of performance and a large main effect of instrumentation. Synthetic audio was perceived as more positive in mood and less tense than human performances. Piano excerpts were also perceived as more positive and as conveying less tension and energy than synthetic excerpts. Several rhythmic and pitch measures were reliably predictive of participants' perceived emotions, supporting the need for considering finer-grain structural features when using naturalistic stimuli.

**Keywords:** empirical aesthetics, perceived emotion, computational musicology, music performance, synthetic audio generation

### 1 Introduction

Research on perceived emotion in music generally relies on listeners' judgments of aesthetic qualities based on audio excerpts of varying lengths. Such stimuli may involve pre-recorded human performances or synthetic audio generated by a computer following a set of instructions. Eerola and Vuoskoski (2013) report that a majority (75%) of studies in music and emotion research used human performances [1]. Although performance medium and source are usually reported along with the results, it is not clear whether the methods used to produce musical excerpts have an effect on listeners' experience. One disadvantage of using human performances as compared with synthetic audio generation is the lack of experimental control on the stimuli, which may limit researchers' ability to manipulate source materials and generalize findings.

#### 1.1 Perceived Emotion in Human Performances versus Synthetic Audio

Research related to audio generation in terms of performance medium tends to focus on two aspects: timbral differences and expressive differences. Studies on timbral



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

differences typically ask participants to identify and categorize single-note stimuli in terms of instrument type as well as perceptual dimensions such as “nasality,” “brilliance,” and “naturalness” [2]. Other studies investigate the effect of timbre on emotion by comparing excerpts played on different instruments (e.g., electronic synthesizer vs. human performances on piano, violin, and trumpet), with observed effects on listeners’ perceived emotions interpreted as being related to acoustical factors [3], rather than performance medium.

Studies on expressive differences investigate expressive performance actions (micro-differences in terms of tempo, dynamics, articulation, intonation, and vibrato) applied in a human performance [4]. Most of the research in this area focuses on observed differences between the notated score and a human performance [5], or between different human performances of the same notated score [6]. Some studies have explored the effects of such differences on listeners’ experience by manipulating human performances. For example, listeners have been shown to be able to distinguish between original and tempo-transformed versions of the same human performance [7]. Synthetic stimuli with different levels of timing manipulations have also been used to explore perceived “expressiveness” and “liveliness” [8], and the addition of other human-like expressive performance actions to synthetic audio, such as expressive dynamics, has been shown to result in higher ratings of “likeability” and “emotional expressiveness” [9]. Still, very few studies have explored listeners’ ratings of emotional expression in synthetic audio as compared to human performances. On one hand, listeners have been found to have a negative bias in their ratings of expressive qualities of human performances presented as synthetic (i.e., “pseudo-synthetic” performances) [8]. On the other hand, synthetic versions of short melodies with human-like expressive differences in tempo, sound level, spectrum, articulation, attack, vibrato, and timing has been shown to convey discrete emotions such as happiness, sadness, anger, fear, and tenderness as effectively as human performances of the same melodies [10]. Nevertheless, direct comparisons of listeners’ perceived emotion in human and synthetic performances of multi-part music are still needed.

In this study, we use rhythmically complex musical excerpts characterized by concurrent rhythmic patterns that cannot readily be mapped onto a single metric grid (i.e., “polyrhythm”). These materials were selected because they provide a naturalistic and rich environment within which listeners’ perceived emotions can be tackled. To-date, very little research has looked into how rhythmically complex music is aesthetically evaluated by listeners. When real music is used (as opposed to controlled “lab” stimuli), attention is devoted to global aspects of the musical compositions, such as tempo, loudness, timbre, and mode, among other factors. However, most studies do not offer sufficient fine-grained information on the rhythmic structure of the selected music to allow for generalization over a wider range of music.

## **1.2 Aims**

The goal of the present study is twofold. First, we aim to determine whether performance (human vs. synthetic) has an effect on listeners’ judgment of five emotional dimensions (mood, energy, movement, dissonance, and tension) for two different instrumentation types (piano and string quartet). Second, because we used excerpts from musical compositions that feature complex rhythmic and harmonic



structures, we also aim to explore the effect of features specific to the musical style on listeners' perceived emotions. To that end, we used a set of computational measures of rhythmic structure (duration, density, alignment, contrast, and regularity) as well as pitch organization (pitch range, pitch mean, register, and sonority dissonance).

We assumed a null hypothesis for the influence of performance, but predicted a main effect of instrumentation. With regard to musical features, we expected event density to be positively correlated with mood, energy, and tension [11, 12, 13], pitch range to be positively correlated with mood and energy, and pitch mean to be negatively correlated with energy [14]. We also expected sonority dissonance to be predictive of perceived dissonance and tension.

## 2 Methods

### 2.1 Participants

Participants were recruited using an online survey implemented in Qualtrics. The survey was approved by the Ethics Review Board of the University of British Columbia, and shared through social media postings, email notifications to institutional and professional listservs, and the UBC Psychology SONA platform. 162 participants with normal hearing completed the study; two datasets were excluded from analysis due to reported difficulty with English in everyday life. Gender distribution was uneven, with a large proportion of participants self-identifying as women (76%) as compared to men (21%); two participants self-identified as non-binary persons and three participants selected “prefer not to answer.” Participants' age ranged from 18 to 59 years old ( $M = 23.3$ ;  $SD = 7.3$ ), and self-reported years of formal musical training ranged from 0 to 20 years ( $M = 5.9$ ;  $SD = 5.1$ ). A greater proportion of participants reported familiarity with the musical style represented by the excerpts (43%, as compared with 23% and 34% for no familiarity and “not sure”), but much fewer participants reported familiarity through listening or performance of a specific excerpt (23%, as compared with 61% and 17% for no familiarity and “not sure”). Finally, most participants listened to the excerpts using built-in speakers (41%), followed by standard and noise-canceling headphones or earbuds (26% and 25%); a small proportion of participants reported using external speakers (8%), while only one participant reported using a phone speaker.

### 2.2 Materials

Sixteen musical excerpts from 12 different composers were selected from the *Suter (1980) Corpus* [15], ranging from 1893 to 1965 in terms of composition year (see Table 1).<sup>1</sup> Based on the availability of realistic audio synthesis and for contrast in timbre, we

---

<sup>1</sup> A full list of examples from the *Suter (1980) Corpus* and associated metadata is available at: <https://polyrhythm.humdrum.org>. The examples used in this experiment are available in kern format at: <https://github.com/polyrhythm-project/rds-scores/tree/master/experiment-lmfl>.

selected an equal number of short piano and string quartet examples with a duration of 5 to 9 s ( $M = 7.2$ ;  $SD = 1.1$ ).

**Table 1.** Source musical compositions for experimental stimuli listed alphabetically by composer’s last name. There are eight examples for each instrumentation type.

Composer	Work Title	Instrumentation	Year
Bartók, Béla	Romanian Folk Dances	Piano	1915
	Piano Sonata	Piano	1926
	String Quartet No. 3	String Quartet	1927
Berg, Alban	Lyric Suite	String Quartet	1926
Britten, Benjamin	String Quartet No. 2, op. 36	String Quartet	1945
Debussy, Claude	String Quartet, op. 10	String Quartet	1893
Falla, Manuel de	“Jota”, from Seven Spanish Songs	Piano	1914
Gershwin, George	Rhapsody in Blue	Piano	1924
Hindemith, Paul	String Quartet, op. 10	String Quartet	1918
Ives, Charles	String Quartet No. 1	String Quartet	1909
Martin, Frank	Prelude No. 8	Piano	1948
	Esquisse	Piano	1965
Martinů, Bohuslav	String Quartet No. 7	String Quartet	1947
Prokofiev, Sergei	Piano Sonata No. 7, op. 83	Piano	1942
	Piano Sonata No. 9, op. 103	Piano	1947
Ravel, Maurice	String Quartet	String Quartet	1903

Two audio versions of each example were prepared: a human performance extracted from a commercial recording randomly selected from available recordings in the Naxos Music Library, and a high-quality musical instrument digital interface (MIDI) rendering using the EastWest sound library. Audio files extracted from recorded examples were trimmed using Audacity to allow excerpts’ duration to be more precisely measured. Synthetic examples were fine-tuned in terms of MIDI note velocity (i.e., volume of each note) and articulations (legato vs. staccato for piano, but a wide variety of options for strings) to match the human performances as closely as possible. The precise timing of raising and lowering the sustain pedal was also fine-tuned for piano excerpts. The tempo of synthetic examples was set to match the average tempo of the human recorded performances, but without rubato or expressive microtiming (i.e., the timing of individual note onsets or releases). To match the acoustics of the human recordings as closely as possible, reverb was added to the piano examples using Logic’s Space Designer; it was not deemed necessary to add reverb to the string quartet examples, which fairly closely matched the acoustics of the human performances. A 0.2 s fade-out was applied to the end of each example to reduce the abruptness of the ending, and both audio file versions were then amplified or attenuated to a peak volume of -1 dB.

### 2.3 Procedure

The experiment was conducted online using Qualtrics, with participants instructed to complete the tasks in one sitting, focusing only on doing the experiment, and in a quiet location or wearing noise-canceling earphones. The order of the experimental trials was randomized across musical excerpts so that each participant heard one performance version (human or synthetic) of each of the sixteen excerpts. To avoid bias, participants were not informed of the type of performance they would hear. Participants were instructed to listen to the excerpt in its entirety at least once, and then rate the excerpt using five seven-point Likert scales. Participants rated the perceived mood (negative–positive), energy (low–high), movement (very little–very much), dissonance (low–high), and tension (low–high), with “movement” referring to how much the participant felt that they could move along to the music.

First, participants provided consent, and read the survey instructions. Participants completed a pre-experiment questionnaire on which they reported their age, gender, formal musical training, and English-language fluency. Prior to listening to the experimental stimuli, participants heard a short audio file during which they were instructed to adjust their volume to a comfortable level, and then completed two practice trials (one of each instrumentation type). At the end of the survey, participants were asked to report what listening device was used to complete the survey, and whether they were familiar with the musical style of the excerpts or with the excerpts themselves through listening or performance. Lastly, participants were given the opportunity to provide feedback and read a debriefing document.

### 2.4 Measures

In addition to participants’ ratings of the five dependent variables using seven-point Likert scales, we selected several measures derived from rhythmic and pitch structures to explore the relationship between musical features and participants’ perceived emotions. Rhythmic features required visualization and analysis of each excerpt and assessment of the differences between concurrent rhythmic patterns. Instrumental parts were divided into two contrasting rhythmic groups (A and B) based on rhythmic similarity within the group and dissimilarity across groups, with the lowest part on the score assigned to Group A by default. The experimental excerpts include up to four instrumental parts; note that although piano excerpts are notated on two staves, each staff could include more than one part. Because the rhythmic design of a given instrumental part may vary over time, group attribution was performed at the measure level. To allow for comparison between examples with a different number of instrumental parts, we use composite rhythms, i.e., the sequential presentation of event onsets across parts. Figure 1 illustrates the visual analytic markup for a sample used in the experiment. Group A notes are colored in red, while Group B notes are in blue. The top two staves are the original score and underneath are the extracted rhythmic patterns and number of event onsets for Group A only, Group B only, Groups A and B combined (“composite”), and the intersection of Groups A and B (“coincidence”). The analytic markup shown in the musical example is automatically generated by the *composite* filter in Verovio Humdrum Viewer [16]; full documentation for the composite filter is available at: <https://doc.verovio.humdrum.org/filter/composite>.



Fig. 1. Visual analytic markup for Gershwin, Rhapsody in Blue (1924), mm. 91–94.

Table 2 presents the six features used to characterize rhythmic structure. Four additional pitch features were also used. *Pitch mean* (average pitch) and *pitch range* (interval between lowest and highest pitch) are calculated using MIDI note values. *Register* corresponds to the proportion of events in each of three ranges: *low* (below C3), *middle* (C3 to C5), and *high* (above C5). To measure *sonority dissonance*, each sonority was assigned a score based on its most dissonant interval (octave/unison = 0; P4/P5 = 1; M/m 3/6 = 2; M2/m7/M9 = 3; A4/d5/m9 = 4; m2 = 5); these values were then averaged and weighed by duration.

Table 2. Calculation and interpretation of rhythmic features

Feature	Calculation	Interpretation
Duration	Total duration of audio file in seconds	N/A
Composite event density	Total number of composite events divided by audio file duration	Rate of presentation of events in time (global information load)
Event density ratio	Number of events in smaller group divided by number of events in larger group	Potential for metric ambiguity or conflict across groups
nPVI group difference <sup>2</sup>	Absolute difference between the nPVI scores of the two rhythmic groups	Contrast in note-to-note regularity across rhythmic groups
Nested ratio	Total number of coinciding event onsets across rhythmic groups divided by total number of composite events	Potential for integrated percept of two rhythmic groups
Polarity ratio	Absolute difference between number of events in rhythmic groups divided by total number of composite events	Relative activity within rhythmic groups (salience)

<sup>2</sup> This measure is an extension of the *normalized pairwise variability index*, a measure of the average durational variation between successive pairs of events.

## 2.5 Analysis

We conducted statistical analysis in RStudio, with R version 4.1.1 and used the `rstatix` package for summary statistics and the `broom` package for summarizing linear models.<sup>3</sup> Although piano and string quartet examples were different in terms of musical materials, they belong to the same historical period. The relative stylistic homogeneity of these musical excerpts was supported by a series of *t* tests: there was no statistically significant difference between piano and string quartet for each of the twelve musical features. To test the effect of performance and instrumentation on participant ratings, we conducted a two-way Multivariate Analysis of Variance (MANOVA) on the combined five dependent variables with performance and instrumentation as the independent variables. Point biserial correlations were used to explore the linearity between the five dependent variables and the two independent variables of performance and instrumentation. To explore the effects of our twelve musical features on participants' ratings, we performed multiple regression analyses. A linear model was constructed between each dependent variable and the twelve musical features. Because participants' ratings were done on a seven-point Likert scale, dependent variables were log-transformed using  $\log(1+x)$ .

## 3 Results

Participants rated 16 excerpts on five Likert scales ( $N = 2,560$ ). The average rating for energy was the highest ( $M = 4.8$ ;  $SD = 1.5$ ), while those for dissonance ( $M = 3.5$ ;  $SD = 1.7$ ) and movement ( $M = 3.7$ ;  $SD = 1.8$ ) were the lowest. The average ratings for mood and tension were in the 4–5 range ( $M = 4.3$  and  $4.0$ ;  $SD = 1.5$  and  $1.7$ ).

### 3.1 Performance and Instrumentation

The main effect of performance was statistically significant, but small,  $F(1, 2556) = 2.89$ ,  $p = .013$ , while the main effect of instrumentation was statistically significant and large,  $F(1, 2556) = 15.97$ ,  $p < .001$ . There was also a significant, although relatively small, interaction between performance and instrumentation,  $F(1, 2556) = 3.32$ ,  $p = .005$ .

Point biserial correlations were calculated between each dependent variable and performance (Human = 1; Synthetic = 2) as well as instrumentation (Piano = 1; String Quartet = 2). Performance was positively correlated with mood,  $r_{pb}(2558) = .05$ ,  $p = .01$ , but negatively correlated with tension,  $r_{pb}(2558) = -.06$ ,  $p = .002$ . Instrumentation was negatively correlated with mood,  $r_{pb}(2558) = -.09$ ,  $p < .0001$ , but positively correlated with energy and tension,  $r_{pb}(2558) = .12$ ,  $p < .0001$ , and  $r_{pb}(2558) = .13$ ,  $p < .0001$ . In other words, participants perceived synthetic excerpts as more positive in mood and as conveying less tension than human performances. Piano excerpts were also perceived as more positive in mood as well as lower in energy and tension than string quartet excerpts.

---

<sup>3</sup> RStudio, `rstatix`, and `broom` are available at: <https://www.R-project.org/>, <https://CRAN.R-project.org/package=rstatix>, and <https://CRAN.R-project.org/package=broom>.

### 3.2 Effects of Musical Features

A summary of the parameter estimates for each of the five dependent variables and the twelve musical features is presented in Table 3 (rhythmic features) and Table 4 (pitch features). All twelve musical features were predictive of participants' ratings for one or more emotional dimensions, with the best model accounting for more than a third of the variance in participants' ratings of energy ( $R^2 = .398$ ).

**Rhythmic Features.** Event density ratio, nested ratio, and polarity ratio were the most reliable predictors for four of the five emotional dimensions with significance levels of  $p < .001$ . Event density ratio and polarity ratio were negatively correlated with mood and movement, and positively correlated with tension. Excerpts with a higher potential for metric ambiguity or conflict and greater contrast in the number of events within each rhythmic group were perceived as less positive in mood, less likely to induce movement, and as conveying more tension. On the other hand, although both factors were also predictive of participants' ratings of dissonance, higher event density ratio was predictive of higher perceived dissonance, while higher polarity ratio was predictive of lower perceived dissonance. In contrast, nested ratio was positively correlated with mood, movement, and dissonance, but negatively correlated with tension. Excerpts that featured more coinciding events were perceived as more positive, more likely to induce movement, more dissonant, but less tense. nPVI group difference was predictive of participants' ratings for three of the five emotional dimensions. A greater contrast between groups in note-to-note rhythmic regularity was correlated with a more positive mood, higher energy, and lower perceived tension. On the other hand, duration and composite density had a relatively limited effect on participants' ratings. Excerpts' duration was negatively correlated with mood and positively correlated with tension. Longer excerpts were perceived as less positive in mood and as conveying more tension. Composite event density was predictive of participants' ratings for energy, with higher composite density predictive of higher energy ratings.

**Pitch Features.** The influence of pitch-related features on participants' ratings of perceived emotions was small, but not negligible. Pitch mean was negatively correlated with energy and dissonance, with higher pitch mean being predictive of lower perceived energy and dissonance. Pitch range was also reliably predictive of participants' ratings for mood, energy, and movement with significance levels of  $p < .001$ . Larger range was correlated with a more positive mood, higher energy level, and a greater impulse to move along with the music. Register (low, middle, and high) was predictive of participants' ratings of mood, energy, and tension with significance levels of  $p < .001$ . A larger proportion of events in any one of the three registers was positively correlated with mood, but negatively correlated with energy and tension. In other words, the concentration of events in one register, rather than a specific register or a more balanced dispersion of pitch activity, was perceived as more positive in mood, but as conveying lower energy and less tension. As expected, sonority dissonance was positively

correlated with perceived dissonance, but the correlation with tension was small and not significant.

**Table 3.** Parameter estimates for rhythmic features and each dependent variable. Significance levels are as follows: ‘\*\*\*’  $p < .001$ ; ‘\*\*’  $p < .01$ ; ‘\*’  $p < .05$ .

Dependent Variable	Parameter	Estimate	Std. Error	<i>t</i> value	Pr(>  <i>t</i>  )
Mood ( $R^2 = 0.282$ )	(Intercept)	-417.70	39.85	-10.48	< .001***
	Duration	-0.14	0.01	-9.72	< .001***
	Composite event density	0.01	0.00	1.67	.10
	Event density ratio	-5.62	0.61	-9.17	< .001***
	nPVI group difference	0.01	0.00	5.03	< .001***
	Nested ratio	1.32	0.14	9.59	< .001***
	Polarity ratio	-5.09	0.54	-9.48	< .001***
Energy ( $R^2 = .398$ )	(Intercept)	212.80	34.97	6.09	< .001***
	Duration	-0.01	0.01	-0.53	.60
	Composite event density	0.03	0.00	7.08	< .001***
	Event density ratio	0.71	0.54	1.32	.19
	nPVI group difference	0.00	0.00	-3.23	.001**
	Nested ratio	-0.06	0.12	-0.49	.62
	Polarity ratio	0.50	0.47	1.07	.29
Movement ( $R^2 = .107$ )	(Intercept)	-107.50	59.39	-1.81	.07
	Duration	-0.04	0.02	-1.94	.05
	Composite event density	0.00	0.01	0.47	.64
	Event density ratio	-4.16	0.91	-4.55	< .001***
	nPVI group difference	0.00	0.00	0.47	.64
	Nested ratio	1.24	0.20	6.07	< .001***
	Polarity ratio	-4.19	0.80	-5.24	< .001***
Dissonance ( $R^2 = .113$ )	(Intercept)	105.50	55.96	1.89	.06
	Duration	0.04	0.02	1.68	.09
	Composite event density	0.00	0.01	-0.15	.88
	Event density ratio	4.22	0.86	4.90	< .001***
	nPVI group difference	0.00	0.00	-0.19	.85
	Nested ratio	-1.32	0.19	-6.87	< .001***
	Polarity ratio	4.14	0.75	5.49	< .001***
Tension ( $R^2 = .260$ )	(Intercept)	557.70	49.97	11.16	< .001***
	Duration	0.14	0.02	7.28	< .001***
	Composite event density	0.00	0.01	0.59	.56
	Event density ratio	5.00	0.77	6.51	< .001***
	nPVI group difference	-0.01	0.00	-4.47	< .001***
	Nested ratio	-1.35	0.17	-7.84	< .001***
	Polarity ratio	4.40	0.67	6.54	< .001***

**Table 4.** Parameter estimates for pitch features and each dependent variable. Significance levels are as follows: ‘\*\*\*’  $p < .001$ ; ‘\*\*’  $p < .01$ ; ‘\*’  $p < .05$ ; intercept and  $R^2$  values are the same as in Table 3.

Dependent Variable	Parameter	Estimate	Std. Error	$t$ value	Pr(>  $t$  )
Mood ( $R^2 = .282$ )	(Intercept)	-417.70	39.85	-10.48	< .001***
	Pitch mean	0.00	0.01	-0.34	.73
	Pitch range	0.01	0.00	8.30	< .001***
	Register low	423.90	40.05	10.58	< .001***
	Register middle	424.90	40.14	10.59	< .001***
	Register high	424.60	40.22	10.56	< .001***
	Sonority dissonance	0.00	0.02	-0.01	.99
Energy ( $R^2 = .398$ )	(Intercept)	212.80	34.97	6.09	< .001***
	Pitch mean	-0.03	0.00	-5.87	< .001***
	Pitch range	0.01	0.00	11.64	< .001***
	Register low	-211.00	35.14	-6.00	< .001***
	Register middle	-210.20	35.22	-5.97	< .001***
	Register high	-210.60	35.29	-5.97	< .001***
	Sonority dissonance	-0.04	0.02	-1.79	.07
Movement ( $R^2 = .107$ )	(Intercept)	-107.50	59.39	-1.81	.07
	Pitch mean	-0.01	0.01	-1.36	.18
	Pitch range	0.01	0.00	7.55	< .001***
	Register low	112.70	59.69	1.89	.06
	Register middle	113.70	59.81	1.90	.06
	Register high	112.80	59.94	1.88	.06
	Sonority dissonance	-0.12	0.04	-3.38	< .001***
Dissonance ( $R^2 = .113$ )	(Intercept)	105.50	55.96	1.89	.06
	Pitch mean	-0.02	0.01	-2.32	.02*
	Pitch range	0.00	0.00	-1.52	.13
	Register low	-107.50	56.24	-1.91	.06
	Register middle	-107.30	56.36	-1.90	.06
	Register high	-106.50	56.47	-1.89	.06
	Sonority dissonance	0.15	0.03	4.20	< .001***
Tension ( $R^2 = .260$ )	(Intercept)	557.70	49.97	11.16	< .001***
	Pitch mean	0.00	0.01	-0.02	.98
	Pitch range	0.00	0.00	-1.23	.22
	Register low	-560.30	50.23	-11.15	< .001***
	Register middle	-561.00	50.33	-11.15	< .001***
	Register high	-561.50	50.44	-11.13	< .001***
	Sonority dissonance	-0.01	0.03	-0.46	.65

#### 4 Discussion

The first goal of our study was to investigate more directly the influence of synthetic generation, as compared to, human performance on listeners’ perceived emotions in



rhythmically complex music excerpts that contrasted in acoustics (piano and string quartet). Performance medium was found to have a small but significant effect, with synthetic performances being perceived as more positive in mood and as conveying less tension. This finding extends previous research that showed that synthetic generation of short melodies can convey discrete emotions effectively [10], and further qualifies the effect of synthetic audio on listeners' perceived emotions. As expected, instrumentation also had a significant and large effect, with piano excerpts giving rise to higher valence and arousal judgments as well as lower ratings for tension. The main effect of instrumentation is consistent with research on the influence of timbre and the influence of acoustical factors on listeners' perceived emotions [3]. While musical excerpts varied across piano and string quartet, they were very similar in terms of the specific musical features considered. Nonetheless, the presence of some hidden factor related to musical excerpts cannot be fully discounted and should be taken into consideration in future experiments (i.e., using musical examples that afford both piano and string quartet performances).

Our second goal was to explore the influence of a number of rhythmic and pitch features on listeners' perceived emotions. Many of our findings are novel and open avenues of investigation on the role of rhythmic structure on perceived emotion. Most notably, event density ratio, a measure of the probability of metric ambiguity or conflict between parts, and polarity ratio, a measure of the contrast in the number of events across rhythmic groups, were predictive of perceived mood, movement, and tension. Similarly, the degree of coinciding event onsets between rhythmic layers (i.e., nested ratio) had reliable, but contrasting effects on participants' ratings of mood, movement, dissonance, and tension. Taken together, these results suggest that rhythmically more integrated musical parts are perceived as more positive and are more likely to induce movement. This is consistent with findings that higher levels of rhythmic complexity have a negative effect on entrainment, which may result in reduced enjoyment [13]. Our results also point to an interaction between rhythmic structure and perceived dissonance by which a lower degree of integration of concurrent rhythmic streams may reduce listener's sensitivity to sonority dissonance between parts. To our knowledge, this is a yet unexplored area that warrants further investigation.

Pitch features had a smaller, but not negligible effect on listeners' ratings. In addition to the expected links between pitch range and higher mood and energy as well as between pitch mean and lower arousal, a wider pitch range was found to be correlated with induced movement. Also of note is the observed relationship between the concentration of pitch in one register, rather than one specific register, being predictive of a more positive mood as well as lower levels of perceived energy and dissonance. Both of these findings warrant further study using a wider selection of music and more controlled stimuli.

Several of our findings show that the interaction of rhythmic patterns in multi-part music plays a significant role in listeners' emotional experience, which calls into question the ecological validity of studies that focus on relatively simple musical sequences to study perceived emotion. Overall, findings based on rhythmic and pitch features suggest that more attention should be devoted to finer-grain musical features and their potential effect on listeners' emotional experience.

## 5 Acknowledgments

The authors thank the members of the UBC Rhythm Computation and Cognition Lab for the encoding of the musical excerpts and Walker Williams for audio preparation. This research is supported in part by funding from the Social Sciences and Humanities Research Council of Canada.

## 6 References

1. Eerola, T., Vuoskoski, J. K.: A review of music and emotion studies: Approaches, emotion models, and stimuli. *Music Perception: An Interdisciplinary Journal* 30(3), 18–49 (2013).
2. Kendall, R. A., Carterette, E. C., Hajda, J. M.: Perceptual and acoustical features of natural and synthetic orchestral instrument tones. *Music Perception: An Interdisciplinary Journal* 16(3), 327–363 (1999).
3. Hailstone, J. C., Omar, R., Henley, S. M. D., Frost, C., Kenward, M. G., Warren, J. D.: It’s not what you play, it’s how you play it: Timbre affects perception of emotion in music. *Quarterly Journal of Experimental Psychology* 62(11), 2141–2155 (2009).
4. Kirke, A., Miranda, E.: A survey of computer systems for expressive music performance. *ACM Computing Surveys* 42(1), Article 3 (2009).
5. Oore, S., Simon, I., Dielemen, S., Eck, D., Simonyan, K.: The time with feeling: Learning expressive musical performance. *Neural Computing and Applications* 32, 955–967 (2020).
6. Repp, B. H.: A microcosm of musical expression. I. Quantitative analysis of pianists’ timing in the initial measures of Chopin’s Etude in E major. *Journal of the Acoustical Society of America* 104(2), 1085–1100 (1998).
7. Honing, H.: Is expressive timing relational invariant under tempo transformation? *Psychology of Music* 35(2), 276–285 (2007).
8. Hähnel, T., Berndt, A.: Synthetic and pseudo-synthetic music performances: An evaluation. In: *Proceedings of the 3rd International Conference of Students of Systematic Musicology (SysMus10)*, Cambridge, United Kingdom (2010).
9. Kamenetsky, S. B., Hill, D. S., Trehub, S. E.: Effect of tempo and dynamics on the perception of emotion in music. *Psychology of Music* 25(2), 149–160 (1997).
10. Juslin, P. N.: Perceived emotional expression in synthesized performances of a short melody: Capturing the listener’s judgment policy. *Musicae Scientiae* 1(2), 225–256 (1997).
11. Gabrielson, A., Juslin, P. N.: Emotional expression in music performance: Between the performer’s intention and the listener’s experience. *Psychology of Music* 24(1), 68–91 (1996).
12. Fernández-Sotos, A., Fernández-Caballero, A., Latorre, J. M.: Influence of tempo and rhythmic unity in musical emotion regulation. *Frontiers in Computational Neuroscience* 10, Article 80 (2016).
13. Labbé, C., Grandjean, D.: Musical emotions predicted by feelings of entrainment. *Music Perception: An Interdisciplinary Journal* 32(2), 170–185 (2014).
14. Gomez, P., Danuser, B.: Relationships between musical structure and psychophysiological measures of emotion. *Emotion* 7(2), 377–387 (2007).
15. Poudrier, È., Shanahan, D.: Modeling rhythmic complexity in a corpus of polyrhythm examples from Europe and America, 1900–1950. In: *Proceedings of ICMPC15/ESCOM10*, ed. R. Parncutt and S. Sattman, pp. 355–360. Centre for Systematic Musicology, University of Graz, Austria (2018).
16. Poudrier, È., Sapp, C. S.: Polyrhythm analysis using the *composite* tool. In: *9th International Conference on Digital Libraries for Musicology (DLfM2022)*, July 28, 2022, Prague, Czech Republic. ACM, New York, NY, United States (2022).

# deepGTTM-IV: Deep Learning Based Time-span Tree Analyzer of GTTM

Masatoshi Hamanaka<sup>1</sup>, Keiji Hirata<sup>2</sup>, and Satoshi Tojo<sup>2</sup>

<sup>1</sup> RIKEN

<sup>2</sup> Future University Hakodate

<sup>3</sup> Asia University

masatoshi.hamanaka@riken.jp

**Abstract.** This paper describes our development of a deep learning based time-span tree analyzer of the Generative Theory of Tonal Music (GTTM). Construction of a time-span tree analyzer has been attempted several times, but most previous analyzers performed very poorly, while those that performed relatively well required parameters to be manually adjusted. We previously proposed stepwise reduction for a time-span tree, which reduces the branches of the tree one by one, and confirmed that it can be learned by using the Transformer model. However, stepwise reduction could not obtain a time-span tree because it does not know to which notes the reduced notes were absorbed. Therefore, we improved the encoding for learning stepwise reduction and specified which notes are absorbed by which notes. We also propose a time-span tree acquisition algorithm that iterates stepwise reduction by representing the time-span tree as a matrix. As a result of experiments with 30 pieces, correct time-span trees were obtained for 29 pieces.

**Keywords:** Generative theory of tonal music (GTTM), time-span tree, melody reduction, Transformer model

## 1 Introduction

We have developed a time-span tree analyzer that is based on the Generative Theory of Tonal Music (GTTM) by using deep learning called deepGTTM-IV. The GTTM was proposed by Leardahl and Jackendoff in 1983, and the time-span tree is a binary tree with each branch connected to each note [1].

Many time-span tree analyzers have been proposed, but most have many analytical errors [2–6]. The time-span tree analyzer that had the highest analytical performance required parameters to be manually adjusted [7].

The reason previous time-span tree analyzers performed insufficiently is that they analyzed in a bottom-up manner using only local information. [2–7]. Therefore, we considered learning the raw data of the entire piece by deep learning. Our deepGTTM-IV has four features.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

**Stepwise reduction:** The Ground Truth data of a time-span tree is insufficient to directly learn the relationship between a piece and its time-span tree. To enable learning, we set the learning target as the process of reducing one note.

**Branch priority:** To make possible the stepwise reduction, the priority order of branches needs to be defined. The maximum time span is used as the branch priority.

**Encoding:** By encoding the score into text, stepwise reduction can be learned in the framework of automatic translation. This makes it possible to reduce notes at designated positions in a piece as if words are omitted in a sentence.

**Time-span-tree matrix:** The time-span tree has been handled in XML and Json formats, making coding difficult [8]. We made coding easier by expressing the information necessary for reduction (i.e., pitch, duration, time-span-tree shape, and branch priority) in a matrix.

We performed an experiment in which 270 items from a GTTM analysis corpus consisting of 300 pieces and their time-span trees were used to learn the Transformer model with the remaining 30 used for evaluation and found that our analyzer was able to obtain correct time-span trees for 29 out of 30 pieces. The remainder of the paper is as follows. Section 2 presents problems of time-span tree analysis based on deep learning, Section 3 describes the data for learning and evaluation, and Section 4 describes the implementation of the analyzer. Section 5 describes the experimental results, and Section 6 gives a summary and mentions future plans.

## 2 Problems of Time-span Tree Analysis based on Deep Learning

In GTTM analysis, the relationship between structurally important notes and other notes in a score is expressed by a binary tree called a time-span tree. The time-span tree in Fig. 1 is the result of analyzing Melody A on the basis of GTTM. Reduced melodies can be extracted by cutting this time-span tree with a horizontal line and omitting the notes connected below the line. In melody reduction with GTTM, decorative notes are absorbed by structurally important notes.

There are the following three problems in the deep learning of time-span tree analysis.

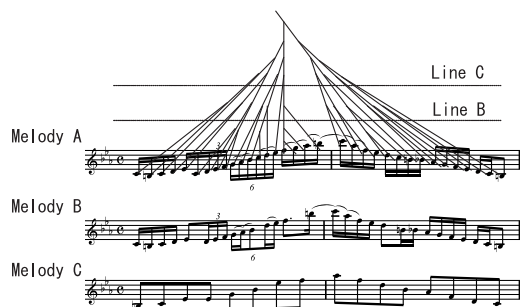


Fig. 1. Time-span tree

## 2.1 Low Number of Ground Truth Data

As ground truth data of the time-span tree, 300 classical melodies and their time-span trees are published in the GTTM database [8, ?]. However, the number of datasets (300) is extremely small for learning deep neural networks (DNNs) [10]. In the case of a small number of pieces of learning data, over-fitting is inevitable, and an appropriate value cannot be output when unknown data is input.

In the time-span analysis by musicologists, the entire time-span tree cannot be acquired at once but is gradually analyzed from the bottom up. Therefore, the minimum process of analysis is set as one dataset, and then the number of datasets is increased. For example, if the DNN directly learns the relationship between a four-note melody and its time-span tree, the number of datasets is only one. On the other hand, if we consider the process of reducing one note to one dataset, the number of datasets will be three, as shown in Fig. 2(a).

The trained DNN estimates the melody consisting of  $n - 1$  notes that is reduced to one note when a melody consisting of  $n$  notes is input. A time-span tree for a melody consisting of four notes can be constructed by estimating four to three notes, three to two notes, and two notes to one note, and combining the results (Fig. 2(b)).

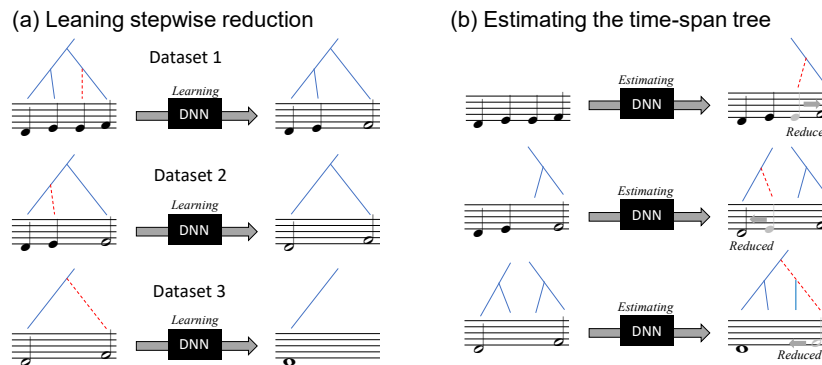


Fig. 2. Stepwise reduction

## 2.2 Ambiguity of Reduction Process

Time-span reduction removes decorative notes by pruning from the leaves at the tip of the tree, leaving only structurally important notes in the melody. To implement the stepwise reduction, the priority of branches must be obtained in a total order.

However, when it comes to GTTM itself, there are only a few examples of reduction using the time-span tree, and there is no detailed explanation on the reduction procedure [1]. For example, in Fig. 1, we can see two levels of reduction results, but it is not clear how many levels are actually necessary.

Marsden *et al.* [11] suggested a way to determine the salience of two note events (a and b), neither of which are descendants of the other. They proposed defining the salience of an event as the duration of the maximum of the time spans of the two children at the branching point when the event is generated, or where it is reduced.

In contrast, in this study, the DNN needs to learn the relationship before and after the reduction than it is to reduce the order of the notes to close to that of human cognition. We use a time-span tree leveled by the duration of the time span for a simple reduction order that it is easy for the DNN to learn [12].

### **2.3 Long Note Sequence**

The previous time-span tree analyzers performed poorly because they analyzed in a bottom-up manner using only local information [2, 5–7]. In contrast, we propose using the entire note sequence before and after stepwise reduction for learning the DNN.

When a recurrent neural network (RNN) [13] or long short-term memory (LSTM) [14] is used as the DNN, the DNN can learn using note sequence, but when a long note sequence is input, the DNN forgets the beginning of it, and then the DNN cannot make use of the whole information of the note sequence.

The Transformer model [15] can learn and predict using the information of the entire note sequence. Moreover, the Transformer model has an additional layer of position information independently and uses the absolute position.

## **3 Data for deepGTTM-IV**

This section describes the data for training the Transformer model. The Transformer model, which is an automatic translation tool, uses text for both input and output. Also, the Transformer model can learn the task of adding two values [16]. The duration of a note after reduction is the sum of the durations of the two notes before reduction, and we thought that this task could also be done with Transformer.

### **3.1 Learning and Evaluation Data**

The preparation of the dataset for stepwise reduction is as follows. First, the priority of each branch of the time-span tree is evaluated on the basis of the duration of the maximum time span [12]. We refer to the longest temporal interval when a given pitch event becomes most salient as the maximum time span for the event. Next, stepwise reduction is applied to the least important note. A learning dataset of stepwise reduction is then created using the data before stepwise reduction as input data and the data after reduction as output data.

### **3.2 Encoding**

Learning data are created from MusicXML and time-spanXML in the GTTM database. Since all melodies in the GTTM database are monophonic, the reduction method is limited to monophony. The notes in the melodies are made into a one-character string

with the pitch and duration concatenated. The pitch is represented as 12 types without distinguishing between different octaves. By multiplying by 4, the duration of most notes becomes an integer, but since there are melodies containing only a few triplets, quintuplets, sextuplets, and septuplets, the duration is rounded up to an integer. The placeholders "l" or "r" are inserted at positions where notes disappeared due to the reduction. The "l" (left) is inserted when the reduction is absorbed into the left note, and "r" (right) is inserted when it is absorbed into the right note. In our previous work, we were unable to reconstruct the time-span tree because we did not distinguish between "l" and "r" [12]. Figure 3 is an example of learning data.

```

Before reduction.  →  After reduction.
c14 c16 d30 c14 c12 c16 d20 c16 . → c14 c16 d30 c26 l c16 d20 c16.
c14 c16 d30 c26 c16 d20 c16 . → c14 c16 d30 c26 r d36 c16.
c14 c16 d30 c26 d36 c16 . → c14 c16 d30 r d62 c16.
c14 c16 d30 d62 c16 . → c30 l d30 R2 d62 c16.
c30 d30 d62 c16 . → c60 l d62 c16.
c60 d62 c16. → c62 r c78.
c60 c78. → r c138.
    
```

Fig. 3. Learning data for melody reduction

As a result of preparing the datasets, 7362 stepwise reduction training datasets are generated from 270 music pieces from the GTTM database consisting of 300 pieces and 849 stepwise reduction evaluation datasets are generated from the remaining 30 pieces for evaluation.

### 3.3 Data Augmentation

The 7362 training datasets are not enough to train the Transformer model, so we carry out data augmentation. Each note is shifted 11 times by a semitone and the amount of training data is augmented by 12 times. The durations of notes are 2-16 times and rounded up to the nearest integer, then the amount of data is augmented by 16 times. Finally, we prepare 1,432,704 (= 7362 x 12 x 16) learning datasets.

## 4 Implementation of deepGTTM-IV

A time-span tree is obtained by iterating stepwise reduction. We expressed time-span trees in XML or Json, but they were difficult to handle with programs because of their deep hierarchical tree structure. Representing a time-span tree as a matrix makes melody reduction easier to implement in a program.

### 4.1 Matrix Representation of Time-span Tree

In Fig 4(a), the first row of the matrix is the encoded pitch and duration and the second row is the connected parent branch number. The root branch has no parent branch to

which to connect, so the parent branch number is set to 0. Both the 2nd and 4th branches are connected to the 1st branch, but the branches of the time-span tree do not cross [1], indicating that the 4th is connected to the 1st at a position closer to the root. Notes that are missing due to reduction have blank pitches and durations on the matrix.

The 3rd row of the matrix is the branch priority. Since the branch priority is obtained from the time-span tree and the note duration, it is redundant information, but it is differentiated in this paper for clearer explanation.

#### 4.2 Generation of Stepwise Reduction Data

Stepwise reduction data is generated by performing stepwise reduction in the order from the lowest priority branch. Applying a step-wise reduction to Fig. 4(a) reduces the notes in the 3rd branch, which has the lowest priority, to the 4th branch. The input of the Transformer model is "d8 e8 e8 f8." The output is "d8 e8 r f16." because the third note is absorbed on the right side. Next, when stepwise reduction is applied to Fig. 4(b), the note on 2nd branch with the second lowest priority is reduced to the note on 1st branch, and the input and output of the Transformer model are "d8 e8 f16 - d16 l f16." Stepwise reduction data is created by repeating stepwise reduction until there is only one note left.

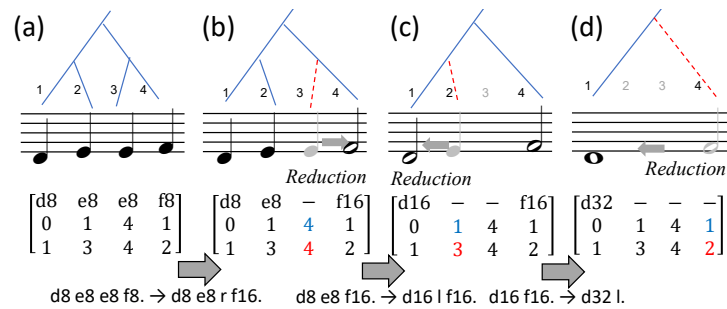


Fig. 4. Generation of stepwise reduction data

#### 4.3 deepGTTM-IV: Reduction System

Figure 5 shows an overview of the reduction system. First, the input melody is converted into Matrix Representation of the time-span tree. In the initial state, no branches are connected, so the matrix has all 0 in the second row (Fig. 5(a)). Then the note sequence in the first row is sent to the Transformer model (Fig. 5(b)). The output of the Transformer model is reflected in the matrix in which the 3rd note is absorbed in the 4th note (Fig. 5(c)). Then (a) to (c) are iterated until there are no notes for reduction (Fig. 5(d)). Finally, a time-span tree is output (Fig. 5(e)).

The Transformer model may produce unexpected outputs from untrained inputs. In such a case, it may be difficult to proceed with the reduction process and our method



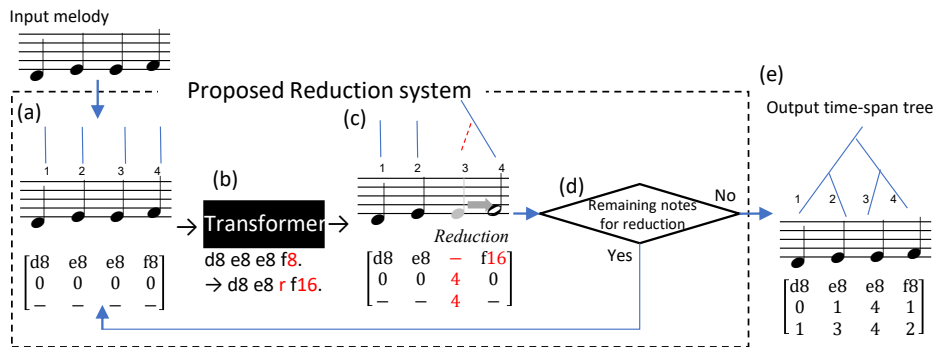


Fig. 5. Overview of reduction system

multiplies the initial duration of one note by a randomly chosen value between 2 and 16 and restarts the reduction process.

## 5 Experimental Results

We trained the Transformer model using 1,432,704 (= 7362 x 12 x 16) learning datasets created by data augmentation of 7362 stepwise reductions made from 270 pieces out of 300 pieces in the GTTM database. Accuracy was 0.99 when evaluated with 849 Stepwise reductions made from the remaining 30 pieces. Learning was carried out using Nvidia Quadro RTX5000 for laptops [17], and the learning time was seven hours.

We tried to acquire time-span trees for the remaining 30 pieces with deepGTTM-IV using the trained Transformer model, and we were able to acquire time-span trees for 29 pieces. The one remaining piece contained quintuplets and the output of the Transformer model was unexpected, so the notes could not be reduced.

## 6 Conclusion

Previous time-span analyzers could hardly obtain time-span trees without analysis errors, but we dramatically improved the analysis performance by learning step-wise reduction with the Transformer model. At the time of encoding the training data, by specifying which note to be reduced to the left or right will be absorbed, decoding becomes possible and a time-span tree can be obtained. As a result of experiments with 30 pieces, all time-span trees were obtained except one piece that contained quintuplets. We plan to conduct evaluation experiments with more pieces. In the case of quintuplets, septuplets, and higher multituplets, there is little data in the GTTM database and it is difficult to learn by the Transformer model, so we plan to increase the data of multituplets by data augmentation to improve performance.

**Acknowledgements** This work was supported by JSPS KAKENHI Grant number 21H03572.

## References

1. Lerdahl, F., and Jackendoff, F.: *A generative theory of tonal music*. The MIT Press, Cambridge, MA (1983)
2. Hamanaka, M., Hirata, K., and Tojo, S.: Implementing "A Generative Theory of Tonal Music". *Journal of New Music Research*, 35(4), 249–277 (2006)
3. Hamanaka, M., Hirata, K., and Tojo, S.: ATTA: Automatic Time-span Tree Analyzer Based on Extended GTTM. In: *Proceedings of the 6th International Conference on Music Information Retrieval Conference (ISMIR2005)*, pp. 358–365 (2005)
4. Hamanaka, M., Hirata, K., and Tojo, S.: FATTA: Full Automatic Time-span Tree Analyzer. In: *Proceedings of the 2007 International Computer Music Conference (ICMC2007)*, Vol. 1, pp. 153–156 (2007)
5. Nakamura, E., Hamanaka, M., Hirata, K., and Yoshii, K.: Tree-Structured Probabilistic Model of Monophonic Written Music Based on the Generative Theory of Tonal Music. In: *Proceedings 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 276–280 (2016)
6. Groves, R.: Automatic Melodic Reduction Using a Supervised Probabilistic Context-Free Grammar. In: *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR2016)*, pp. 775–781, New York (2016)
7. Hamanaka, M., Hirata, K., and Tojo, S.: Sigma GTTM III: Learning based Time-span Tree Generator based on PCFG. In: *Proceedings of The 11th International Symposium on Computer Music Multidisciplinary Research (CMMR 2015)*, pp. 303–317 (2015)
8. Hamanaka, M., Hirata, K., and Tojo, S.: Musical structural analysis database based on GTTM. In: *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR2014)*, pp. 325–330 (2014)
9. Hamanaka, M., Isono, Y., Hirata, K., and Tojo, S.: Web-based time-span tree editor and analysis database. In: *Proceedings of the 17th Sound and Music Computing Conference (SMC2020)*, pp. 338–343 (2020)
10. Amari, S., Ozeki, T., Karakida, R., Yoshida, Y., and Okada, M.: Dynamics of Learning in MLP: Natural Gradient and Singularity Revisited. *Neural Computation*, 30(1), 1–33 (2018)
11. Marsden, A., Hirata, K., and Tojo, S.: No Longer 'Somewhat Arbitrary': Calculating Saliency in GTTM-Style Reduction. In: *Proceedings of the 5th International Conference on Digital Libraries for Musicology (DLfM '18)*, pp. 26–33 (2018)
12. Hamanaka, M., Hirata, K., and Tojo, S.: Time-span Tree Leveled by Duration of Time-span. In: *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR2021)*, pp. 155–164 (2021)
13. Pineda, F. J.: Generalization of Back-propagation to Recurrent Neural Networks. *Physical Review Letters*, 19(59), 2229–2232 (1987)
14. Hochreiter, S. and Schmidhuber, J.: Long Short-term Memory. *Neural Computation*, 9(8), 1735–1780 (1997)
15. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention Is All You Need. In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS2017)*, pp. 6000–6010 (2017)
16. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, A., and Amodei, D., Language models are fewshot learners, arXiv preprint arXiv:2005.14165, 2020.
17. Nvidia, "NVIDIA RTX in professional laptops.", Available: <https://www.nvidia.com/en-us/design-visualization/rtx-professional-laptops/>

# Music and Logic: a connection between two worlds

Matteo Bizzarri<sup>1\*</sup>

Scuola Normale Superiore  
matteo.bizzarri@sns.it

**Abstract.** *Music and mathematics have a long-standing relationship, but what about music and logic? Only recently have some authors started to explore the relationship between logic and music analysis, thanks to developments in both fields. The aim of this paper is to analyze this relationship, by developing a system capable of analyzing chord sequences using a logical presentation as well as create new harmonic structures. The logical presentation draws heavily from proof theory and its dual, i.e. tableaux. Also if music is not a proof, its adaptability makes it effective for this purpose. The attempt here proposed will try to apply proof theory to a brief, but important part of music: chord sequence analysis.*

**Keywords:** Music Analysis; Logic; Proof theory; Chords Analysis

## 1 Introduction

Logic is primarily used in mathematics to formalize human reasoning, and the study of the relationship between mathematics and music has a long-standing tradition. This work aims to explore the connection between logic and music, which has not been studied extensively. Specifically, the idea arose from my personal interest in proof theory and its ability to simplify complex sentences into simpler propositions. The objective of this paper is to make a preliminary attempt in this direction, by exploring the possibility of applying proof theory techniques to chord analysis.

The method presented in this paper is inspired by Neo-Riemannian and Tonfeld theories, which are systematic approaches to the musical structures' analysis, albeit not in a formal logical sense. The main goal of this paper is to introduce a rule-based logic method for chord analysis, which shares some similarities with structural proof theory (e.g., Troelstra's Basic Proof Theory [11]) in its logical foundations.

Something similar to this method has been presented in various articles, such as Rohrmeier: extended harmony [9], Granroth-Wilding, Mark and Steedman, Mark: Statistical Parsing for Harmonic Analysis of Jazz Chord Sequences [3] and Satoshi: modal

---

\* I would like to express my sincere gratitude to Professors Satoshi Tojo, Mario Piazza, and Fabio De Sanctis De Benedictis for their invaluable assistance in writing and developing this essay. Additionally, I extend my appreciation to the two reviewers for their insightful suggestions, which greatly contributed to the enhancement of both the method and the overall paper. Their recommendations on potential research directions have also been immensely valuable.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

logic music [10], but this contribution takes a different approach. Instead of decomposing a chord progression through grammar syntax, it presents a set of rules that can reduce the total number of chords, making the analysis simpler.

The method is as follows:

- we begin with a given chord progression;
- we apply rules that can *reduct* a certain set of chords into a smaller one;
- repeat the process until no further reductions are possible.

The corresponding method in proof theory is the decomposition of a proposition into a set of simpler ones to understand easier if the proposition is a tautology or not. But in the case of music analysis the final set of chords, that can't be simplified, will be called the *core set* and it will be associated with a particular type of chord set according to Tonfeld theory. The rules, once explained, can be analyzed both bottom-up and top-down, revealing the application of each rule at each level of the decomposition. The invertibility of each rule can also be used to compose new chord progressions, providing a mechanical method for choosing between different chords.

The paper is organized as follows: in the second section, we provide a description of Tonfeld and the Circle of Fifths; in the third section, we present our motivations for choosing structural proof theory and a very brief presentation of it is given; in the fourth section, we present the method for analysis in detail; in the fifth section, we outline the inverse process of composition using the rule-based method. Throughout the paper, we provide examples to help illustrate the motivations and the methods being discussed.

## 2 Tonfeld and circle of fifths

The Tonfeld theory [7] provides a visual depiction of the relationships between chords in tonal harmony using an infinite plane graph where each node represents a unique pitch class. Notes are understood as points in the graph, and chords are depicted by their relationships. The theory identifies three fundamental types of relationships: octatonic, hexatonic, and stacks of fifths, which are cyclic and sufficient to describe all other cyclic groups.

Instead of explicitly defining the octatonic and hexatonic sets for each note (see for example [8]), it is possible to specify only three octatonic and four hexatonic sets, because the others are permutations of these, thanks to the limited transposition modes ([6]). This allows for a more efficient and compact representation of the harmonic relations within the Tonfeld theory:

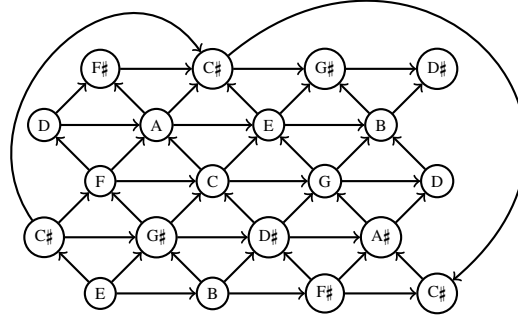
$$Oct_0 = \{C, D\flat, E\flat, E\sharp, G\flat, G\sharp, A, B\flat\} \quad (1)$$

$$Oct_1 = \{D\flat, D, E\sharp, F, G, A\flat, B\flat, C\flat\} \quad (2)$$

$$Oct_2 = \{C, D, E\flat, F, G\flat, A\flat, A\sharp, B\} \quad (3)$$

$$Hex_0 = \{C, E\flat, E, G, A\flat, B\} \quad (4)$$

$$Hex_1 = \{C\sharp, E, F, G\sharp, A, C\} \quad (5)$$



**Fig. 1.** A part of the Tonfeld. The lines outside the figure outline the relation between the same chords throughout the plane.

$$Hex_2 = \{D, F, F\#, A, Bb, C\#\} \quad (6)$$

$$Hex_3 = \{Eb, F\#, G, Bb, B\flat, D\} \quad (7)$$

The stack of fifths is not a mode of limited transposition, so it is necessary to enumerate them for each note and for each expansion. This means that all possible stacks of fifths need to be explicitly listed, unlike the octatonic and hexatonic sets which can be represented with a little number of sets due to the limited transposition modes.

$$Fif_{C,1} = \{C, G\} \quad (8)$$

$$Fif_{C,2} = \{C, G, D\} \quad (9)$$

⋮

This enumeration of every component, as is well-known, can be easily deduced thanks to the circle of fifths. Furthermore, this system simplifies the work with the proof-theoretic platform that will be presented later.

### 3 Why structural proof theory?

Proof theory is a branch of mathematical logic that studies the nature of mathematical proofs and their properties. The central questions in proof theory concern the nature of proof, the relationship between syntax and semantics, and the role of proofs in the development of mathematics. It is a fundamental component of mathematical logic, and has important applications in computer science, philosophy, and other fields. Proof theory aims to understand the nature of formal systems and develop techniques for analyzing and manipulating them; there are several approaches to proof theory, but the one we want to emphasize here is *structural proof theory*.

The origins of structural proof theory [11] can be traced back to the early 1930s when Gerhard Gentzen (1909-1945) introduced the concept in his doctoral thesis titled “Untersuchungen über das logische Schließen” [4], in 1933. In this thesis, Gentzen presented two primary systems of logical rules: natural deduction and sequent calculus. The former system aimed to closely align with the way theorems are typically proven in practice, while the latter system provided the framework through which Gentzen arrived at his main finding, often referred to as Gentzen’s “Hauptsatz”. This theorem states that any proof in classical logic can be transformed into a specific “cut-free” form, which means that the proof can be obtained without detours. Additionally, the cut-free proof has the subformula property, which states that all the premises used in the proof are contained in the conclusions. From this, general conclusions about proofs can be drawn, such as the consistency of the system of rules. The method has the following structure: the top formulas, also called *leaves*, represent the *starting point* of the proof ( $q \vdash q$  and  $p \vdash p$ ), while on the bottom we find the proven formula (i.e.,  $\vdash (p \wedge q) \rightarrow (q \wedge p)$ ).

$$\frac{\frac{\frac{q \vdash q \quad p \vdash p}{q, p \vdash q \wedge p} (\wedge_{R,I})}{p, q \vdash q \wedge p} (ex.)}{p \wedge q \vdash q \wedge p} (\wedge_{L,I})}{\vdash (p \wedge q) \rightarrow (q \wedge p)} (\rightarrow_{R,I})$$

In this context,  $R$  and  $L$  denote the side of the rule to be applied;  $I$  represents the *introduction* of a rule;  $ex.$  represents the exchange rule, which enables swapping of the terms in the proof. The symbol  $\wedge$  represents conjunction, which can be interpreted in English as *and*, and  $\rightarrow$  represents implication, interpreted as *if... then...*

We believe that this system’s clarity and duality make it an effective way to represent not only propositions but also chords. In proof theory, it is possible to use not only trees like the one presented earlier but also trees constructed *from the bottom* known as *tableaux*. These trees are constructed from the proposition to be proved, as shown below:

$$\begin{array}{c} \vdash (p \wedge q) \rightarrow (q \wedge p) \\ | \\ p \wedge q \vdash q \wedge p \\ | \\ p, q \vdash q \wedge p \\ / \quad \backslash \\ p, q \vdash q \quad p, q \vdash p \end{array}$$

The dual approach of the system, allowing for progression from the axioms or the propositions, will be useful in presenting the two-fold perspective we aim to convey in this article: analyzing chords from one direction and creating new harmonic structures from the other.

## 4 Rule based presentation

In proof theory, specifically in the style developed by Gentzen [11], a set of rules is used to introduce and eliminate certain logical connectives ( $\wedge, \vee, \rightarrow, \neg$ ) in order to determine whether the topmost nodes of a proof correspond to axioms, i.e., whether  $p \vdash p$  ( $p$  proves  $p$ ). The purpose of this section is to provide a more systematic account of the application of certain harmonic rules, using the rules of harmony, the Tonfeld theory, and a proof-theoretic framework.

While music is not a proof, a set of fundamental rules can still be outlined and adapted by adding or removing rules. This article focuses on jazz tonal harmony and presents a construction using a limited set of rules, but it can be hopefully expanded. Our attempt is to find a way to combine the generative theory of tonal music (GTTM) [9] with the ability of proof theory to explicitly indicate when and where a certain rule must or can be applied. This aims to provide a more systematic and logical approach to understanding and analyzing harmony in tonal music. It must be stressed that this attempt will not adhere to all of the structural rules typically found in proof theory. In fact, the only structural rule that we will use is the Contraction Rule, which will play a crucial role.

$$\frac{\vdash p, p}{\vdash p} \text{ (Contraction)}$$

but there is no place for weakening, because we don't want that new chords can appear spontaneously:

$$\frac{\vdash p}{\vdash p, q} \text{ (Weakening)}$$

and exchange, because we don't want that chords change position:

$$\frac{\vdash p, q}{\vdash q, p} \text{ (Exchange)}$$

### 4.1 The first rules

In a Gentzen's style presentation a rule has the following form:

$$\frac{\vdash p \quad \vdash q}{\vdash p \wedge q} \text{ (}\wedge\text{I)}$$

where the letter  $I$ , indicates the *introduction* rule. The reason for erasing a chord in the presented version of the rule is to identify the essential components of the harmonic sequence during harmonic analysis. Therefore, the objective is to isolate the *core* set of harmonies. In fact here the presentation is as follows:

Authentic Cadence:

$$\frac{V7}{IMA^7} (Fif_{I,1})$$

This kind of cadence can be expanded with the stack of fifths:

$$\frac{II m7 \quad V7 \quad IMA^7}{IMA^7} (Fif_{I,2})$$

$$\frac{VI m7 \quad II m7 \quad V7 \quad IMA^7}{IMA^7} (Fif_{I,3})$$

⋮

Another rule is the *Plagal Cadence*, which is extensively used in ancient as well as modern pop music. *Plagal Cadence* is a type of cadence that goes from the fourth scale degree to the first one. It can be schematized in three main ways, to also explicitly show the movement from the fourth minor scale degree to the first one. It can be interpreted as a particular case of the circle of fifths: from the last instances to the first one.

Plagal cadence :

$$\frac{IVMA^7}{IMA^7} \text{ P.C.} \quad \frac{IVm7}{IMA^7} \text{ P.C.} \quad \frac{IVMA^7 \quad IVm7}{IMA^7} \text{ P.C.}$$

To explicitly explain the other types of cadences, like the *Deceptive Cadence*, it must be noted that the tonic and the submediant have a lot of notes in common, which allows for their mutual substitution. For example, an authentic cadence can be transformed into a deceptive one by substituting the tonic with the submediant. Something similar occurs between the tonic and the mediant.

Inversions:

$$\frac{VI}{I} (i.) \quad \frac{III}{I} (i.) \quad \frac{I}{III} (i.) \quad \frac{I}{VI} (i.)$$

In jazz and classical music, another rule is deduced: the tritone substitution. This rule allows for the substitution of a dominant chord with its relative tritone. This is particularly useful in the context of jazz improvisation and the creation of complex harmonic progressions in classical music. The tritone substitution adds more dissonance to the progression, and is one of the most important features of jazz harmony. It can be used to create tension and dissonance and it is an essential tool to understand the harmonic language of jazz.

$$\frac{V}{I\#} (tr)$$

From the Authentic Cadence and its inversions, it is possible to also obtain the Deceptive Cadence and the Authentic Cadence with the subdominant scale degree instead of the supertonic:



- Deceptive Cadence: V-VI;
- Authentic Cadence with subdominant: IV-I;

These variations help to create a more rich and complex harmonic language and can be used to create a different emotional or stylistic effect in the music.

$$\frac{V7}{\frac{\frac{VIIm7}{IMA7}^{(i)}}{IMA7}^{(Fif_{I,2})}}$$

$$\frac{(IVMA7)}{\frac{IIIm7}{IMA7}^{(i)}} \frac{V7}{IMA7} \frac{IMA7}{(Fif_{I,2})}$$

## 4.2 Examples

*Example 1.* The following example is taken from “But not for me” by George Gershwin. The structure is divided into sections, because the tree was too long.

$$\frac{F7}{\frac{Bb7}{\frac{EbMA7}{E bMA7}} \frac{EbMA7}{Fif_{Eb,3}}} \frac{Gm7}{EbMA7}^{(i.)} \frac{F7}{\frac{Bb7}{\frac{EbMA7}{E bMA7}} \frac{EbMA7}{Fif_{Eb,3}}}$$

$$\frac{Bbm7}{\frac{Eb7}{\frac{AbMA7}{E bMA7}} \frac{AbMA7}{Fif_{Ab,3}}} \frac{Db7}{\frac{Gm7}{EbMA7}^{(i.)}}^{(tr.)} \frac{EbMA7}{P.C._{Eb,3}} \frac{Fm7}{Bb7} \frac{Bb7}{Fif_{Bb,2}}$$

$$\frac{Bbm7}{\frac{Eb7}{\frac{AbMA7}{E bMA7}} \frac{AbMA7}{Fif_{Ab,3}}} \frac{Db7}{\frac{Gm7}{EbMA7}^{(i.)}}^{(tr.)} \frac{EbMA7}{P.C._{Eb,3}} \frac{Fm7}{Bb7} \frac{EbMA7}{Fif_{Eb,3}}$$

As we can see the main chord is  $EbMA7$ , that is always the main chord.

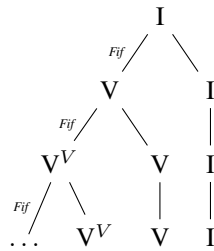
## 5 Composition of new musical structures through a rule based presentation

This method uses invertible rules to create novel harmonic tonal structures. The process is straightforward and involves selecting a fundamental node, determining the desired length of the structure, and applying the rules to expand the system until the required number of chords is reached. By using this systematic and logical approach based on the invertibility of the rules, it's possible to compose original harmonies.

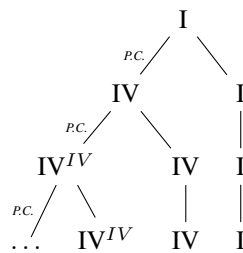
### 5.1 Tableux

The inversion of the rules made in section 4.1 can be inverted thanks to the dual of proof theory: tableux.

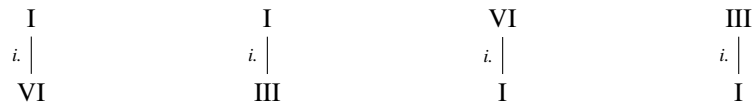
Generation of the Authentic Cadence:



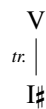
Generation of the Plagal Cadence:



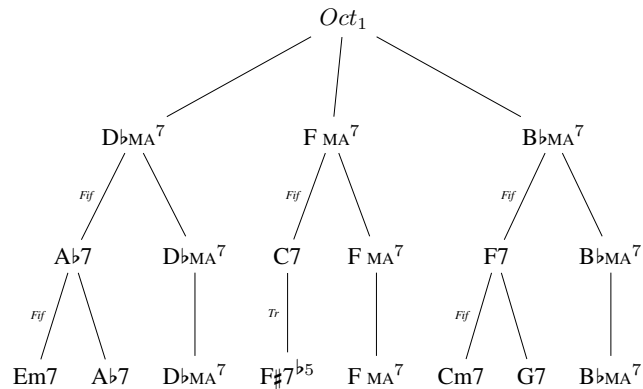
The inversions:



Triton substitution:



*Example 2.* Firstly, we'll form a basic arrangement of three notes taken from the initial Octatonic set. Then, we'll establish a preferred duration for the arrangement, say 8 measures. We'll utilize various techniques, such as stack of fifths, plagal cadence, and tritone substitution, to extend the pattern following certain guidelines until we attain the desired number of chords. Consequently, we'll obtain 11 distinctive chords, resulting in an 8-measure arrangement with increased intricacy and musical appeal.



This way of composition can be automatized to create new and different harmonic structures, always remaining into a tonal configuration, but what if we want to create a non-tonal structure?

### 5.2 New rules

One of the interesting thing about our system is that it is possible to add new rules giving flexibility to the system. Music and harmony, in fact, can change between ages and the rule-based system can be expanded with new rules if they are considered useful for a certain kind of analysis or a certain kind of composition. Suppose that the analyzed piece is taken from the baroque period and so it is important to explicit the *picardy third*. Then it is easy to add a rule that could be something like:

$$\frac{\text{Im} \quad \text{V} \quad \text{I}}{\text{Im} \quad \text{I}} \text{ P.T.}$$

This could seem redundant in respect of the rule of the stack of fifths, but the attempt here is to create something general that could be useful also in specific cases. In the baroque chorals, for example, understand when there is an authentic minor cadence or a picardy third could be useful, because a picardy third indicates the end of a phrase or of the piece.

*Example 3.* Let's consider, for example, J.S. Bach's Jesu, meine Freude (figure 2), this is an example of picardy third.

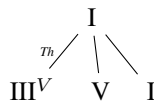
To analyze these bars it is possible to use the new rule:

$$\frac{\text{Em} \quad \text{Am} \quad \text{Em}}{\text{Em}} \text{ P.C.} \quad \frac{\text{F\#7} \quad \text{B7} \quad \text{E}}{\text{E}} \text{ P.T.}$$



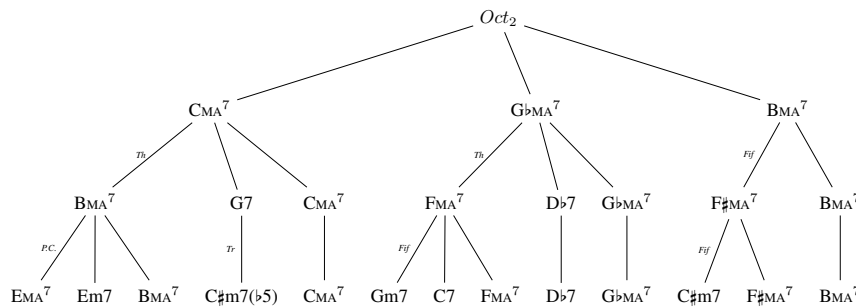
Fig. 2. J.S. Bach's Jesu, meine Freude; mm. 11-13

*New rules in composition* In addition to their traditional use in tonality, add rules can also be employed in composition to generate unconventional results. For instance, a new rule could be introduced to mandate the inclusion of the third degree of a chord in any dominant chord that appears. This rule might read as follows: “Whenever a dominant chord is encountered, it must contain also the third degree” and let’s call that *Th.*.



Incorporating this additional layer would introduce an added level of intricacy and diversity to the harmonic arrangements produced by the system, leading to a greater potential for novel and unforeseen outcomes. It is crucial to acknowledge that the regulations you integrate will shape the final composition to align with your specific requirements.

*Example 4.* For example try to write a new harmonic form using this rule with also some other rule:



### 5.3 Proof theory method and CCG

The presented method shares similarities with the one presented in [3] that uses Combinatorial Category Grammar (CCG), but there are some notable differences. On one hand, the reduction method we propose is more flexible than the one presented in [3]. On the other hand, our method is not currently linked to machine learning or automatic

analysis, which are areas that we plan to explore in the future. It's worth noting that the two methods are not in conflict and can ideally be combined in the future.

The contraction method we propose is particularly useful because it's malleable: we can add new rules to analyze and stress different musical aspects, and we can even invert the method to create new harmonic structures. In contrast, the method presented in [3] relies on a statistical machine learning technique that may not be as straightforward to implement using our proof theoretic presentation. Moreover, our proof theoretic presentation can be inverted to create new harmonic structures, which CCG can only achieve by working on the rules. However, this inversion process is not as straightforward as it is with our method. A common point between CCG and our method is the philosophical and musicological idea that chord progressions are driven by the listener's *expectations* of progression, based on the same harmonic Riemannian concept. However, our method can be expanded due to its ability to incorporate new rules.

## 6 Conclusions

The presented method is a different approach to understanding chord progressions, drawing inspiration from proof theory. While music cannot be proven, this rule-based method simplifies and clearly demonstrates the invertibility of the rules between analysis and composition. The approach's advantage is that new rules can be added to the system to emphasize specific structures or introduce new ones.

This approach to harmonic analysis offers several advantages for students and professionals alike. Firstly, its visual representation can aid in better understanding the underlying principles of harmony. By breaking down complex harmonic structures into their component parts and representing them as a tree-like structure, students can more easily grasp the relationships between chords and the rules that govern them. Secondly, this approach can be helpful in promoting creativity when composing new harmonic structures by providing the opportunity to choose new rules and leading to unexpected solutions and unique harmonic progressions. Lastly, this method can help shed light on a particular harmonic structure that may otherwise go unnoticed. By approaching a harmonic structure several times, each time with a different lens, an analyst can highlight a particular composer's choice over another, revealing the nuances and subtleties of their harmonic language. Overall, this approach to harmonic analysis offers a powerful tool for understanding and creating harmonic structures, and its potential applications are wide-ranging, from the classroom to the automatic composition.

Future researches should explore practical applications, incorporating additional examples and use cases. A deeper understanding of cut admissibility within this method is imperative. Moreover, the method presented holds potential for automated harmonic generation, which could be harnessed for an automatic theorem prover centered on harmonic structures. In summary, this method transforms chord comprehension through proof theory insights. While music eludes formal proof, this approach helps in understanding of rule dynamics in analysis and composition. Its adaptability and visual lucidity aid learners and creators alike, finding relevance from education to composition.

## References

- [1] Chew, E. Slicing It All Ways: Mathematical Models for Tonal Induction, Approximation, and Segmentation Using the Spiral Array. *INFORMS Journal on Computing*. **18** pp. 305 (2006,8)
- [2] Douthett, J. & Steinbach, P. Parsimonious Graphs: A Study in Parsimony, Contextual Transformations, and Modes of Limited Transposition. *Journal of Music Theory*. **42**, 241-263 (1998)
- [3] Granroth-Wilding, M. & Steedman, M. Statistical Parsing for Harmonic Analysis of Jazz Chord Sequences. *ICMC 2012: Non-Cochlear Sound - Proceedings of the International Computer Music Conference 2012*. pp. 478-485 (2012,1)
- [4] Gerhard Gentzen, Untersuchungen über das logische Schließen. *I. Math Z* 39. pp. 176–210 (1935)
- [5] Krumhansl, Carol L., E. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys.. *Psychological Review*. **89**, 334-368 (1982)
- [6] Messiaen, O. *Technique de mon langage musical*. (Alphonse Leduc,1944)
- [7] Polth, M. The Individual Tone and Musical Context in Albert Simon’s Tonfeldtheorie. *Music Theory Online*. **24** (2018)
- [8] Rohrmeier, M. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*. **5**, 35-53 (2011)
- [9] Rohrmeier, M. & Moss, F. A Formal Model of Extended Tonal Harmony. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. pp. 569-578 (2021)
- [10] Tojo, S. Modal Logic for Tonal Music. *Perception, Representations, Image, Sound, Music: 14th International Symposium, CMMR 2019, Marseille, France, October 14–18, 2019, Revised Selected Papers*. pp. 113-128 (2019)
- [11] Troelstra, A. & Schwichtenberg, H. *Basic Proof Theory*. (Cambridge University Press,2000)

# A Novel Local Alignment-Based Approach to Motif Extraction in Polyphonic Music

Tiange Zhu<sup>1</sup>, Danny Diamond<sup>2</sup>, James McDermott<sup>2</sup>, Raphaël Fournier-S'niehotta<sup>1</sup>,  
Mathieu Daquin<sup>3</sup>, and Philippe Rigaux<sup>1</sup> \*

<sup>1</sup> CNAM Paris

<sup>2</sup> University of Galway

<sup>3</sup> Université de Lorraine

tiange.zhu@lecnam.net

d.diamond1@nuigalway.ie

**Abstract.** The paper provides a novel approach to musicologically-informed intra-opus motif detection within polyphonic music scores. We extract diatonic interval sequences from each voice of a score; sequence segmentation is performed via pairwise local alignment between each pair of voices. From the output of this step, string-based approaches are used for motif discovery.

Specifically, a weighted directed acyclic graph is constructed, giving a custom measurement of motif importance. A selection and filtration procedure is applied according to a set of rules and music structural information, to generate a final selection of music motifs.

The ground truth annotated JKUPDD dataset is used for evaluation of the proposed methodology. The results demonstrate that this algorithm is capable of extracting musically meaningful motifs with high precision and recall.

**Keywords:** Music Information Retrieval, Pattern Discovery, Computational Musicology

## 1 Introduction

A musical motif is “*the smallest structural unit possessing thematic identity*” within a piece of music [1]. The detection of frequent musical patterns is a long-standing area of work in the field of Music Information Retrieval (MIR). The existing pattern discovery research covers both audio and symbolic music, adopting methods generally falling into three broad categories of 1) string or sequence-based [2, 3], 2) geometric pattern discovery, [4], and 3) machine-learning based methods [5].

---

\* This work is part of a project that has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004746 (Polifonia: a digital harmoniser for musical heritage knowledge, H2020-SC6-TRANSFORMATIONS).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

The main goal of this work is to develop a methodology that, while informed by musicological knowledge, is not specialized to a single genre or musical tradition. A second aim is to produce a short and focused set of output motifs, reducing the requirement for time-intensive human validation of the results.

We designed a novel approach to achieve these aims, by working with interval sequences extracted from digital music scores, and segmenting a composition based on pairwise local alignments [6] between all possible voice pairs. Alignment has been commonly applied to the task of similarity between pieces of music [7, 8], but not in works on detection of local patterns. The outputs of the segmentation are taken as the input for a string-based motif discovery process. The overall importance of a pattern is measured based on its frequency of occurrence, using a graph that represents the relationship between patterns. The top-ranked patterns are further analyzed and filtered according to their musical (metrical) structure, generating a final set of motifs. In the context of this paper, motifs are defined as short recurring melodic patterns within a piece of music which contain important or characteristic thematic material; it must repeat at least two times throughout a composition, and contain at least three intervals. The proposed method is proven to generate satisfying results for an intra-opus pattern detection task based on the JKUPDD dataset [9], discussed in Section 4. The results exhibit a high degree of accuracy, broadly comparable to state-of-the-art pattern detection algorithms.

Identified motifs are of importance in use cases which range from thematic analysis of the piece of music, or musicological study of the body of work of a composer [10], to characterisation of a musical tradition, genre or period. Apart from being applied to polyphonic melodies as in this paper, the introduced methodology can also be applied to detect motifs between multiple related monophonic scores, which is potentially of use in the study of tune families or regional styles within folk traditions [11].

## 2 State of the art

Musical pattern detection tasks in MIR can be either “*intra-opus*” (within a single piece of music) and/or “*inter-opus*” (across multiple pieces of music). Input data is typically either audio or symbolic music representation. The following discussion mainly covers work on symbolic music inputs, with the exception of [12] and [13].

Pattern detection studies on symbolic music tend to break down into string-based, geometric or machine learning approaches. String-based pattern detection studies are the most common of all approaches. They range from  $n$ -grams and NLP-based work such as [14] to tree models of pattern relationships, subsuming and compressing many unique pattern instances to a smaller set of ‘maximal’ patterns. The latter approach to pattern detection has been influential on the work presented in this paper. It has most commonly been used in monophonic inter-opus applications [15, 16]; some polyphonic applications exist in the literature [2] but differ to our work significantly in the specific structural model applied.

Best-in-class geometric work includes the family of “point-set” geometric compression algorithms set out in [17, 18]. This family of algorithms have performed well on tasks ranging from intra-opus pattern detection in the JKUPDD dataset [17], to an intra-

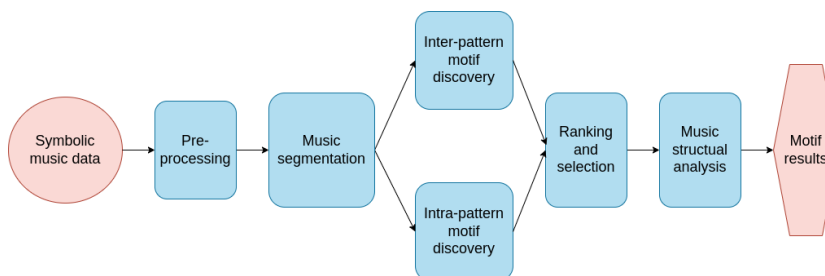


opus tune family classification task in [18]. Other interesting work, which both builds on and evaluates the “point-set” approach includes [4, 19] and [20].

Works based on machine-learning are increasingly prominent in recent years. Chai Wei [13] uses self-similarity and Dynamic Time Warping (DTW) to detect repeating structural sections in audio corpora. Unsupervised machine learning is adopted by Jacopo de Berardinis et al. [12] to build graph-based music structure hierarchies adapted for segment audio-derived feature sequences into structural sections. Matevž Pesek et al. [5] uses unsupervised machine learning, in order to construct a compositional hierarchical model for analysis and discovery of pattern in symbolic music.

### 3 Methodology

#### 3.1 Framework



**Fig. 1.** A framework of motif discovery from polyphonic symbolic music

The framework of the proposed method for discovering motifs in polyphonic symbolic music is illustrated in Figure 1. Taking a symbolic music score as an input, key-invariant diatonic interval sequences are extracted from each voice, and encoded. Music segmentation is then applied to the encoded sequences via local alignment [6]. A set of patterns are gathered for further discovery, to find a set of potential motifs. The motifs are ranked and filtered based on specific rules and a customized measure of importance, and then analyzed by their music structure information, to select a final list of results.

#### 3.2 Pre-processing

A polyphonic music score is taken as the input for pre-processing. Using music21 [21], we extract the melodic pitch sequence from each voice of the score, represented as a sequence of MIDI note numbers. From these pitch sequences we calculate diatonic intervals, then normalise them to the range of a single octave.

**Definition 1 (Melody).** Let the  $k$ th voice in a score be  $v_k$ . Let  $M(v_k) = [m_1, \dots, m_n]$  be the melody of  $v_k$ , in which  $m_i$  denotes the pitch of the  $i$ th note of  $v_k$ .



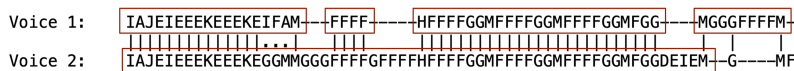


Fig. 3. Alignment between the openings of voice 1 and 2 of Bach BWV889 fugue

this set of segments to take possible patterns. Furthermore, a filter is implemented to remove all segments of less than 3 elements in length, according to the definition of motif mentioned in Section 1.

**Definition 4 (Pattern).** Let the filtered set of segments outputted by the alignment between  $v_x, v_y$  be  $A(enc(v_x), enc(v_y))$ . An element in  $A(enc(v_x), enc(v_y))$  is a pattern.

**Definition 5 (Pattern set).** Let a score of  $m$  voices be  $S = [v_1, \dots, v_m]$ . From the alignment between every possible pair of voices in  $S$ , we construct a set of all possible patterns

$$P(S) = \bigcup_{(v_x, v_y) \in S \times S} A(enc(v_x), enc(v_y)) \quad (3)$$

As a valid pattern in  $P$  may appear multiple times in the course of the segmentation process, the sum of its occurrences is defined as  $occ(p)$ .

### 3.4 Intra-pattern discovery of motifs



Fig. 4. Intra-pattern discovery example (from Bach BWV889 fugue)

String-based approaches are used to uncover additional motifs which are not well-captured in the segmentation process, including those which occur exclusively in one voice. For patterns of greater than 11 intervals in length in  $P$ , we identify and extract from them the longest frequent substring which occur two or more times. The choice of 11 as a length threshold is informed by previous use of a maximum pattern length of 12 notes in the literature on  $n$ -gram-based Music Information Retrieval [23], which is equivalent to 11 intervals. It also follows the definition of motif in this paper, favouring relatively short motivic patterns over longer patterns, which potentially correspond to musical sections or themes. The lengths of such musical structural units are not defined in absolute terms, so the length threshold of 11 elements is proposed as a working heuristic rather than a formal definition of maximum motif length. Figure 4 illustrates a case where a pattern repeatedly appears in a sequence of intervals.

The longest frequent substring extracted from a long pattern may become a substitution of the long pattern, according to rules defined as follows:

**Definition 6 (Pattern substitution).** Let  $p$  be a pattern of length  $|p|$ , and let  $sub(p)$  of length  $|sub(p)|$  be the longest substring that repeated at least two times in  $p$ . Let  $r_{sub(p),p}$  be the number of times  $sub(p)$  appears in  $p$  without overlapping.  $sub(p)$  takes place of  $p$  in  $P$  if certain conditions are met, such as:

$$p = \begin{cases} sub(p), & \text{if } |sub(p)| > 3 \text{ and } |sub(p)| * r_{sub(p),p} \geq 0.6|p| \\ p, & \text{otherwise} \end{cases} \quad (4)$$

In which,  $|sub(p)| > 3$  ensures that  $sub(p)$  is a non-trivial substring, following the logic discussed above in section 3.3, with the aim of removing frequent-but-insignificant short patterns.  $|sub(p)| * r_{sub(p),p} \geq 0.6|p|$  ensures that  $sub(p)$  meets the required threshold to substitute for  $p$ .

Consider that  $sub(p)$  could substitute for more than one pattern in  $P$ , let  $p_i$  be a pattern in  $P$  that is substituted by  $sub(p)$ , then

$$occ(sub(p)) = \sum_{p_i \in P} occ(p_i) \times r_{sub(p),p_i} \quad (5)$$

### 3.5 Inter-pattern discovery of motifs

**Definition 7 (All possible pairs of long patterns).** Let  $(p_i, p_j)$  be a distinctive pair of patterns in  $P$ , and the set of all possible pairs of patterns in  $P$  that are longer than 11 elements be  $longpairs(P)$ , then

$$longpairs(P) = \{(p_i, p_j) | p_i, p_j \in P \text{ and } |p_i| > 11, |p_j| > 11\} \quad (6)$$

We take the longest common substrings (LCSS) of each pattern pair in  $longpairs(P)$ . Unique substrings discovered in this step are added as patterns for further selection in Section 3.6.2.

### 3.6 Ranking and selection of patterns

**Graph-based pattern importance measure for ranking** A weighted directed acyclic graph is used to capture the relationship between patterns in  $P$ , in order to measure the overall importance of a pattern based on its frequency of occurrence. The graph is constructed from the set of patterns  $P$ , in which each pattern in  $P$  is a node of the graph, and the directed edges represent substring relationships between patterns. The weights on the edges reflect the strength of the relationship.

Let a graph be  $G = (P, E, w)$ ,  $P$  be a set of nodes,  $E$  be a set of directed edges, and  $w$  be the weight function. For each pair of patterns  $p_i$  and  $p_j$ , if  $p_i$  is a substring of  $p_j$ , we add a directed edge from  $p_i$  to  $p_j$  and a directed edge from  $p_j$  to  $p_i$ , weighted according to the weight function.

The weight function  $w$  is defined as follows:

If  $p_j$  is a substring of  $p_i$ , then the weight of the edge  $e_{ij}$  from  $p_i$  to  $p_j$  is 1, denoted as  $\pi(e_{ij})$ . The weight of the edge  $e_{ji}$  from  $p_j$  to  $p_i$  is the frequency of non-overlapping

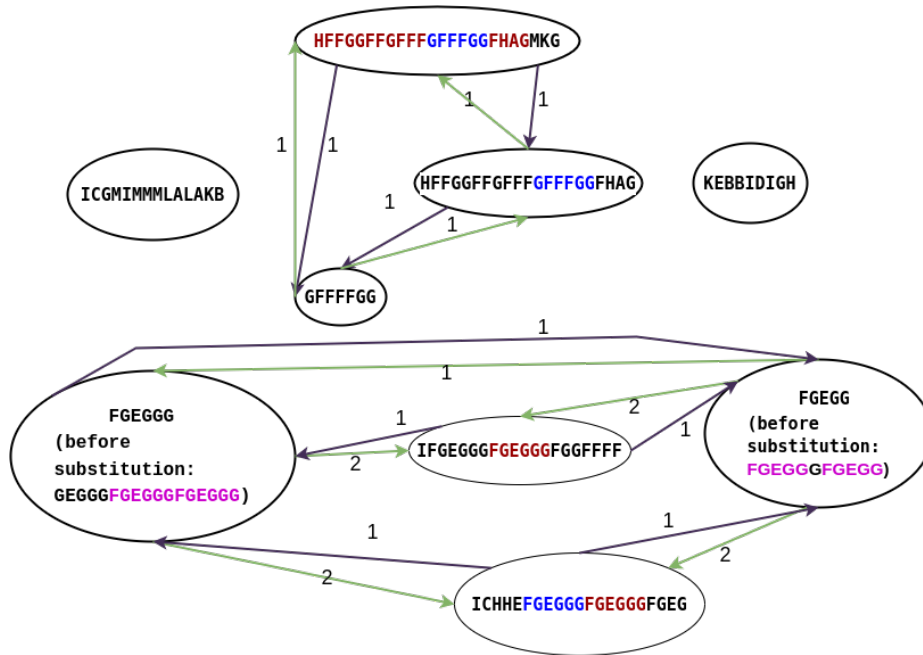


Fig. 5. A weighted directed acyclic graph of 9 patterns

occurrence of  $p_j$  in  $p_i$ , denoted as  $\pi(e_{ji})$ . Let  $I(p_i)$  be the custom importance measure of a pattern  $p_i$ , and  $[e_{i1}, \dots, e_{in}]$  be the set of directed edges from  $p_i$ , then

$$I(p_i) = occ(p_i) + \sum_{k=0}^n \pi(e_{in}) \tag{7}$$

Figure 5 shows an example of a DAG constructed from a  $P$  of 9 patterns. Edges are added when two patterns have a substring relationship. The green edges are edges from substrings to their parent string, with a number representing their weight, while the purple edges are from parent strings to their substring. Pattern “FGEGGG” is the substring of “IFGEGGGFGEGGGFGGFFFF” which appeared 2 times, thus the edge from the former to the latter is weighted as 2. Both “FGEGGG” and “FGEGG” have the highest out-degree of 4, which indicates their importance in this set of patterns.

**A set of rules for selection** The patterns in  $P$  are ranked by their importance measure. The top-20 ranked patterns are selected. The set of longest common substrings generated in inter-pattern discovery of motif process are not ranked along with patterns in  $P$ , as they are extracted substrings of patterns in  $P$ . Instead, we consider those which are longer than 3 elements and repeated more than twice as valid patterns. The patterns outputted in this selection process are retained and inputted to the music structural analysis step.

### 3.7 Music structural analysis

In addition to fundamental attributes such as duration and pitch, we use music21 to extract a *beatStrength* value for every note in an input musical score. *beatStrength* is an encoding of the degree of rhythmic emphasis associated with each note or item in a score. It takes the form of a float value between 0 and 1. The first note of every bar can be assumed to be heavily rhythmically accented, and is assigned a value of 1 by default. *beatStrength* values are extracted for all first notes of each pattern occurrence.

If the first note of a pattern has a *beatStrength* value of 1, it indicates the pattern on-set coincides with the beginning of a bar, i.e. the pattern aligns with the metric structure of the piece of music. Such patterns are retained, while patterns which begin on less rhythmically-emphasised notes are filtered out of the results.

There is one exception to this rule: As it is often the case that motifs occur at or near the beginning of a score, the above metric filtering step is not applied to patterns which occur in the opening 8 bars of a score. A threshold of 8 bars is chosen as this is the most common length for the opening period (the opening two phrases) in common western musical practice. Within this subsection of the score, patterns which begin on a less-heavily emphasised note (i.e. which are not coincident with the metric structure) are retained.

## 4 Results

### 4.1 Evaluation process

The algorithm is tested on the JKUPDD dataset [9], a set of 5 polyphonic classical scores with ground-truth annotation of repeating patterns drawn from academic sources [24–26]. This database has been previously used for testing and evaluation of other pattern detection work, notably as input data for the Music Information Retrieval Evaluation eXchange (MIREX) 2017 Discovery of Repeated Themes & Sections task [19].

Diatonic interval sequences are extracted from the labelled ground truth patterns for evaluation. The scores are manually checked to identify and annotate the exact diatonic interval occurrences of the ground truth patterns.

The pattern annotation in the JKUPDD dataset covers a wide variety of pattern types, including motifs, themes, phrases, and sections. They range in length from three elements to more than 150 elements. As our method specializes towards detection of short motif patterns, we elected to omit ground-truth annotated periods and sections from our results scoring. For the same reason, we also chose to score incomplete matches of at least 4 pattern elements as positive results in the precision and recall calculations.

According to documentation, patterns are labelled with alphabetic identifiers: “A”, “B” and so on for each score, named in order of their importance. We will make reference to this hierarchical ordering in the discussion.

### 4.2 Results

Precision and recall scores of the testing are presented in Table 1.

Work	Precision (%)	Recall (%)
Bach: BWV889 fugue	66.7	61.5
Beethoven: Op. 2, No. 1, Mvt. 3	100.0	45.0
Chopin: Op. 24, No. 4	50.0	50.0
Gibbons: The Silver Swan	87.5	84.6
Mozart: K282, Mvt. 2	60.0	100.0
<b>Average</b>	<b>72.8</b>	<b>68.2</b>

**Table 1.** Results: precision and recall for all JKUPDD scores

### 4.3 Discussion



**Fig. 6.** Exact match of ground truth pattern “A”, occurrence 4, in Bach BWV889 fugue



**Fig. 7.** Pattern “B” from Bach BWV889 fugue with two overlapping partial matches highlighted and boxed in red.

**Bach: BWV889 fugue** Patterns A and B are the most frequent and most significant patterns in this score. The algorithm returned A exactly. It is the opening musical motif of the entire piece and the key musical idea behind the composition. The result is illustrated in Figure 6, and detailed in the following sections. Although B matched only partially, the matching subsequence repeats twice within B. This may suggest we are capturing a core or fundamental motif within pattern B, per Figure 7.

**Beethoven: Op. 2, No. 1, Mvt. 3** We fail to identify pattern A but match the opening 11-element subsequence of pattern B, which is the second-most important musical pattern in the piece per annotation.

**Chopin: Op. 24, No. 4** We found a robust partial match to ground truth pattern A. The found motif occurs at the start of pattern A and repeats twice within it, in a similar manner to 7 above. Thus, the motif may be core content within pattern A, which is the most musically important/distinctive in the piece.

**Gibbons: The Silver Swan** Pattern A, which occurs early and repetitiously in 4 of the 5 voices, has been detected in full. Overall, our precision and recall scores are very high for this composition. It is possible that the shortness of the ground truth patterns for *The Silver Swan* play to the strengths of our tool, as it is tailored towards shorter motif pattern detection.

**Mozart: K282, Mvt. 2** The sole ground truth pattern detected is a significant subsequence of pattern A. This is an incomplete but positive result, capturing the last 6 notes of this significant 10-note pattern. The detected pattern does not appear in the ‘definitive’ opening occurrence of pattern A, but occurs in 10 of the 11 other noted variant occurrences of pattern A in the course of the score.

Study	Precision (%)	Recall (%)
VM1 [27]	84.0	89.0
VM2 [27]	76.0	80.0
SymCHM [5]	67.9	45.4
SymCHMMerge [5]	68.0	51.0
Chen & Su [28]	50.0	69.6
Zhu & Diamond	<b>72.8</b>	<b>68.2</b>

**Table 2.** Average establishment precision and recall results for a selection of work evaluated on the JKUPDD database. Standard precision and recall results for our work included for comparison.

**Comparison with other studies** Table 2 compares our scores against establishment precision and recall values reported in other studies tested on the JKUPDD database. The establishment precision and recall defined in MIREX task guidelines [29] allows for the validity of a partial match, which is similar to our use of standard precision/recall with positive scoring of partial matches.

Although the results in Table 2 allow informal comparison of our results against similar work, it is important to note that our use of diatonic interval sequences rather than MIDI pitch sequences, our omission of sections from the ground truth, and our use of standard precision/recall all differ from the approach set out in the MIREX task documentation.

In Table 2 our approach compares favorably against all studies other than Velarde and Meridith’s VM1 and VM2 studies [27]. Both VM1 and VM2 extract short pitch sequence ‘segments’ directly from MIDI; VM2 also filters these sequences via wavelet transform. Contiguous segments are concatenated, clustered via city block distance and ranked by the length of their occurrences in the ground truth. This building up from an initial set of short patterns contrasts against our work in which long patterns are compressed in multiple passes to produce shorter motific output patterns.



## 5 Contribution and future work

This paper introduces a motif extraction approach that makes novel use of local alignment for string segmentation. Patterns are detected by employing string-based methods, and a custom graph model for similarity scoring has been developed, combined with a musicologically-informed analysis and filtering step. The results presented exhibit a high degree of accuracy, broadly comparable to best-in-class pattern detection algorithms. To aid reproducibility, the source code is available on GitHub [30].

The proposed method supports related musicological tasks, such as the analysis of characteristic motifs in composition styles, or the classification of music corpora. It also has potential applications in various domains in MIR including music generation.

In the future, we plan to improve the algorithm via encoding more musicological knowledge. We also intend to apply the algorithm to inter-opus pattern detection in a corpus of monophonic Irish traditional folk tunes on *The Session* [31], which will help gain greater insight into the role of motifs in defining *tune families* [32] within the corpus.

## References

1. J. D. White, *The Analysis of Music*. Prentice-Hall.
2. D. Conklin and M. Bergeron, “Feature set patterns in music,” *Computer Music Journal*, vol. 32, no. 1, pp. 60–70.
3. B. Janssen, W. de Haas, A. Volk, and P. Van Kranenburg, “Finding repeated patterns in music: State of knowledge, challenges, perspectives,” in *Sound, Music, and Motion*. Springer International Publishing, vol. 8905, pp. 277–297.
4. 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 *International Society for Music Information Retrieval Conference*.
5. M. Pesek, A. Leonardis, and M. Marolt, “SymCHM—an unsupervised approach for pattern discovery in symbolic music with a compositional hierarchical model,” *Applied Sciences*, vol. 7.
6. T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences.” *Journal of molecular biology*, vol. 147 1.
7. P. van Kranenburg, “A computational approach to content-based retrieval of folk song melodies,” 2010.
8. R. Hillewaere, B. Manderick, and D. Conklin, “Alignment methods for folk tune classification,” in *Data Analysis, Machine Learning and Knowledge Discovery*. Springer International Publishing, pp. 369–377.
9. T. Collins, “The johannes kepler university patterns development database.” [Online]. Available: [https://www.music-ir.org/mirex/wiki/2017:Discovery\\_of\\_Repeated\\_Themes\\_%26\\_Sections](https://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections)
10. M. Giraud, R. Groult, and F. Levé, “Subject and counter-subject detection for analysis of the well-tempered clavier fugues,” in *From Sounds to Music and Emotions*, M. Aramaki, M. Barthelet, R. Kronland-Martinet, and S. Ystad, Eds. Springer Berlin Heidelberg.
11. A. Volk and P. Van Kranenburg, “Melodic similarity among folk songs: An annotation study on similarity-based categorization in music,” *Musicae Scientiae*, vol. 16.
12. J. de Berardinis, M. Vamvakaris, A. Cangelosi, and E. Coutinho, “Unveiling the hierarchical structure of music by multi-resolution community detection,” *Transactions of the ISMIR*, vol. 3, no. 1, pp. 82–97.

13. W. Chai, "Semantic segmentation and summarization of music: methods based on tonality and recurrent structure," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 124–132.
14. T. Nuttall, M. G. Casado, A. Ferraro, D. Conklin, and R. C. Repetto, "A computational exploration of melodic patterns in arab-andalusian music," *Journal of Mathematics and Music*, vol. 15, no. 2.
15. D. Conklin and C. Anagnostopoulou, "Comparative pattern analysis of cretan folk songs," *Journal of New Music Research*, vol. 40, no. 2, pp. 119–125.
16. Jia-Lien Hsu, Chih-Chin Liu, and A. Chen, "Discovering nontrivial repeating patterns in music data," *IEEE Trans. Multimedia*, vol. 3, no. 3.
17. D. Meredith, "COSIATEC and SIATECCompress: Pattern discovery by geometric compression," in *Music Information Retrieval Evaluation eXchange (MIREX 2013)*. International Society for Music Information Retrieval.
18. —, "Compression-based geometric pattern discovery in music," in *2014 4th International Workshop on Cognitive Information Processing (CIP)*, pp. 1–6.
19. T. Collins, B. Janssen, and Y. Hao. Mirex 2017: Discovery of repeated themes & sections results. [Online]. Available: [https://www.music-ir.org/mirex/wiki/2017:MIREX2017\\_Results](https://www.music-ir.org/mirex/wiki/2017:MIREX2017_Results)
20. 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, 2017*.
21. 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)*, pp. 637–642.
22. M. Breese, "swalign." [Online]. Available: <https://github.com/mbreese/swalign>
23. T. Zhu, R. Fournier-S'niehotta, P. Rigaux, and N. Travers, "A framework for content-based search in large music collections," *Big Data and Cognitive Computing*, vol. 6, no. 1, p. 23.
24. S. Bruhn, *J.S. Bach's Well-Tempered Clavier: in-depth analysis and interpretation*. Mainer International.
25. H. Barlow and S. Morgenstern, *A dictionary of musical themes*. Crown Publishers.
26. A. Schoenberg, *Fundamentals of Musical Composition*. Faber and Faber.
27. G. Velarde and D. Meredith, "A wavelet-based approach to the discovery of themes and sections in monophonic melodies."
28. T.-P. Chen and L. Su, "Discovery of repeated themes and sections with pattern clustering." [Online]. Available: <https://www.music-ir.org/mirex/abstracts/2017/CS3.pdf>
29. T. Collins. Mirex 2017: Discovery of repeated themes & sections results. [Online]. Available: [https://www.music-ir.org/mirex/wiki/2017:Discovery\\_of\\_Repeated\\_Themes\\_%26\\_Sections](https://www.music-ir.org/mirex/wiki/2017:Discovery_of_Repeated_Themes_%26_Sections)
30. T. Zhu and D. Diamond, "motif\_extraction: A novel local-alignment-based approach to motif extraction in polyphonic music." [Online]. Available: [https://github.com/TiangeZhu/motif\\_extraction](https://github.com/TiangeZhu/motif_extraction)
31. J. Keith. The session. [Online]. Available: <https://thesession.org>
32. 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, publisher: University of Illinois Press.

# Predicting Audio Features of Background Music from Game Scenes

Ryusei Hayashi<sup>1</sup> and Tetsuro Kitahara<sup>1\*</sup>

<sup>1</sup>Graduate School of Integrated Basic Sciences, Nihon University  
Setagaya-ku, Tokyo, Japan  
ryusei@kthrlab.jp

**Abstract.** *We propose a system to retrieve background music (BGM) for game scenes. BGM plays an important role in creating a particular atmosphere in game scenes, so studies have investigated the relationship between game scenes and BGM. However, none of the existing studies attempted to predict the audio features of BGM directly from a sequence of images expressing game scenes. In our system, the user inputs a sequence of images of a game scene, then our machine learning model, trained with gameplay videos, predicts the audio features from the input. Finally, the system retrieves the closest musical piece to the predicted audio features. Experimental results show both positive and negative tendencies: the predicted audio features for fight scenes are closer to the features of actually used BGM in fight scenes than those in other scenes (positive); the same musical piece was retrieved for different scenes (negative).*

**Keywords:** CNN-LSTM, Video Game Music (VGM), Role-playing Game (RPG), Speedrun Video, Copyright-free Music

## 1 Introduction

Background music (BGM) plays a role in creating the atmosphere of a video game. In particular, musical pieces used in different scenes (e.g., talk, fight) would be carefully composed to make different atmospheres that different scenes have. Therefore, we suppose a strong relationship exists between the atmosphere of a scene, and the feature of the BGM used there. For example, the BGM in a fight scene of a role-playing game (RPG) may tend to create a tense atmosphere with strong beats, while the BGM used in a talk scene may tend to create a calm atmosphere with soft timbres and rhythm.

The final goal of our study is to establish a technology that makes it possible to recommend the BGM that fits each of the various scenes in a game. This technology is intended to be used by indie game creators. After they create various scenes of their own game, they will give each scene (a sequence of screen images) to our system. Then, the

---

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

system recommends the musical piece that fits each scene as BGM based on a machine learning model.

Some researchers have investigated the relationship between game scenes and BGM. Yamauchi et al.[1] developed a system that retrieves music from game scenarios. Nemoto et al.[2] investigated the relationship between the emotional state of a character and BGM. Choi et al.[3] created a game music database with emotion labels and a model to generate a baseline. Kim et al.[4] developed a game that records BGM's and the player's moods simultaneously and analyzed their relationship. Ishikawa et al.[5] developed a system that retrieves BGM from visual scenes through impression words. Zeng et al.[6] developed a system that retrieves movies from music. "AmBeat"[7] is an application that adds generated music to a video when the video is input. "Deep12"[8] searches for similar music when music is input. They did not deal with directly predicting musical features of BGM for games from those games' visual scenes.

In this paper, we develop a method for predicting audio features of BGM that fits given game scenes from a sequence of screen images. Because there are many gameplay videos on Web-based video hosting services, we can quickly obtain large-scale data consisting of pairs of a sequence of screen images of a game and audio signals of its BGM. By learning those data, we will achieve the prediction of the audio features of BGM from screen images.

## 2 Proposed Method

We propose a system that outputs the audio features of BGM suitable for the input game scene. The system is intended to be used when the user creates his/her own game and finds musical pieces for adding to the game as BGM. First, the user inputs a sequence of images of a scene (e.g., Fight, Talk) included in the created game. Then, the system predicts a sequence of the audio features that are considered to fit the given scene. Finally, the system outputs the musical piece with the closest audio features to the predicted ones from the music collection prepared in advance.

It is generally challenging to find a universal relationship between scenes and BGM. We assume that the user creates a game referring to an existing game (called a *referred game* here), and they are similar to each other. Therefore, we let the user specify the referred game and learn the relationship between scenes and BGM.

### 2.1 Input and Output Data

The input and output data were taken from speedrun videos posted on YouTube. First, we saved videos in MP4 format. Next, we classify the video frame by frame using k-means[9]. Finally, we extracted one hundred 12-second segments from the video to avoid including multiple classes.

Then, we applied the following pre-processing. We divide the input and output data obtained by these processes in half and use them as training and test data.

**Input Data** We loaded a 12-second video using the OpenCV library and converted the color space of images from BGR to HSV. The image size was also changed to  $80 \times 80$ . We accordingly obtained tensor data of dimensions  $80 \times 80 \times 3$ .

**Output Data** We extracted the audio tracks from the videos mentioned above and saved them in WAV format. The audio features described in Table 1 were extracted using the LiBROSA library. Some of these audio features can be selected and used.

**Table 1.** Audio features to be extracted

	Feature	Outline
01	cqt	Semitone power spectrogram using constant-Q transform
02	iirt	Semitone power spectrogram using a multirate filter bank consisting of IIR filters
03	chroma_stft	12-dimensional features representing the power of each pitch class, calculated from the STFT-based power spectrogram
04	chroma_cqt	12-dimensional features representing the power of each pitch class, calculated from the CQT-based power spectrogram
05	chroma_cens	12-dimensional features with smoothed temporal variations in chroma_cqt
06	melspectrogram	Mel-scaled spectrogram
07	mfcc	Mel-frequency cepstral coefficients
08	mfcc_delta2	Temporal second-order differentials of MFCCs
09	nmf	Activations obtained by non-negative matrix factorization from the spectrogram

## 2.2 Model Architecture

Our model is based on CNN-LSTM[10][11], in which the CNN[12] part reduces the image data of the given game scene video while the LSTM[13] part models the temporal features contained in the scene video and BGM. The overview of this model is shown in Fig. 1. The CNN part consists of multiple convolution layers and max pooling layers. The LSTM part consists of two LSTM layers to make it possible to consider long temporal dependencies.

As mentioned above, we assume that the user has a *referred game*, an existing game that he/she referred to when creating a game. Therefore, our model is trained individually on each training game, and the model trained on the referred game is intended to be selected by the user.

This model has been implemented with the Keras library of Tensorflow. We use ADAM[14][15] as an optimizer and the mean squared error as the loss function. The batch size is 16. The number of epochs is 500 for chromagrams and 100 for other audio features. They were experimentally determined to make the loss function less than 0.01.

## 2.3 Retrieval of musical pieces from predicted audio features

After the audio features for BGM are predicted, the system retrieves the musical piece with the closest audio features to the predicted ones from a pre-made music collection. This process includes the following two phases.

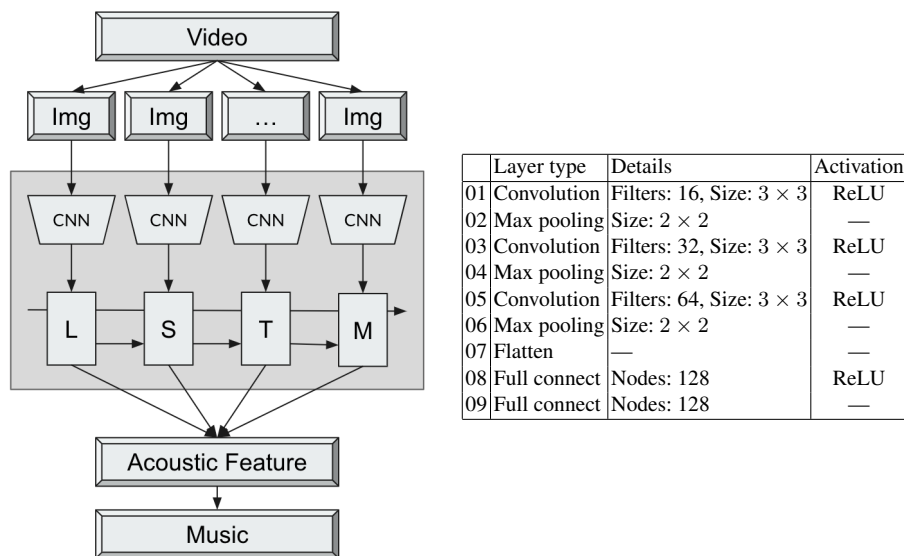


Fig. 1. Architecture of our model. The right-hand table shows the details of the CNN layers

**Extraction of 12-second representative segment** First, the system extracts a 12-second representative segment from each piece included in the collection. This is because a sequence of audio features extracted from each piece in the collection should have the same duration as the predicted audio features. Although extracting the first 12-second segment may be the simplest way, it may not capture the characteristics of the entire music. Therefore, we extract the segment that captures the characteristics of the music as follows:

1. A sequence of MFCCs is extracted from the target audio signal.  
Let  $\mathbf{x} = [x_1, x_2, \dots, x_N]$  be the sequence of the MFCCs.
2. All MFCC vectors  $x_1, x_2, \dots, x_N$  are clustered with the  $k$ -means algorithm[16].  
The number of clusters is set to 4. Let  $c_i$  be the cluster ID of  $x_i$ . Then, the most frequent cluster ID,  $c_{\text{mode}}$ , is obtained.
3. Let  $\mathbf{x}_i = [x_i, x_{i+1}, \dots, x_{i+n}]$  be a 12-second segment beginning at  $x_i$ , where  $n$  is the number of elements for a 12-second segment. Then we compute  $\hat{i}$  that satisfies the following equation:

$$\hat{i} = \underset{i \in [0, N-n]}{\operatorname{argmax}} \operatorname{count}(c_{\text{mode}}, [c_i, c_{i+1}, \dots, c_{i+n}]),$$

where  $\operatorname{count}(a, A)$  counts how many elements in a sequence  $A$  equals  $a$ .

4.  $\mathbf{x}_{\hat{i}}$  is regarded as the 12-second representative segment.

**Search of musical piece** Next, we extract audio features from the extracted 12-second representative segment for every piece in the collection. The audio features to be ex-

tracted, listed in Table 1, are the same as those used in the prediction with the CNN-LSTM model. Then, the Earth Mover’s Distance[17][18] of the extracted audio features from the predicted ones is extracted for every piece in the collection. Finally, the piece that has the minimal distance is searched.

### **3 Experiments**

We conducted the following experiments.

1. Determination of referred and test games
2. Prediction of audio features
3. Retrieval of musical pieces from the predicted audio features

#### **3.1 Dataset**

We made a dataset from speedrun videos of the games in Table 2 posted on YouTube. We divided them into 12-second scenes and extracted two fight scenes, two walk scenes, and two talk scenes. We created a music collection for BGM from the copyright-free music sites in Table 3. We downloaded 99 WAV files from free music sites.

#### **3.2 Determination of referred and test games**

As mentioned above, we assume that the user selects a referred game and uses the model trained with that game. To simulate this situation, we adopt the following three-step approach. To reduce the computation time for learning models, we first choose a referred game and then decide the test game which is closest to the referred game.

1. Choose a referred game.
2. Find the game with the most similar visual scenes to the chosen one. This game is regarded as a test game.
3. The model trained with the referred game’s data is used for predicting audio features.

Step 2 is calculated based on the average of the image hash value differences. The average hash value difference is calculated in the following steps.

- 2-1 Load the videos of the two games as images.
- 2-2 Compute hash values of all images.
- 2-3 Compute the difference between the hash values of the two games for all combinations.
- 2-4 Compute the average of hash value differences.

The results of the calculated dissimilarities are shown in Fig. 2. For “Undertale” (referred game), “OMORI” (test game) was selected. For “Chrono Trigger” (referred game), “Romancing Saga 3” (test game) was selected. Fig. 3 shows some excerpts of the visual scenes of those games.

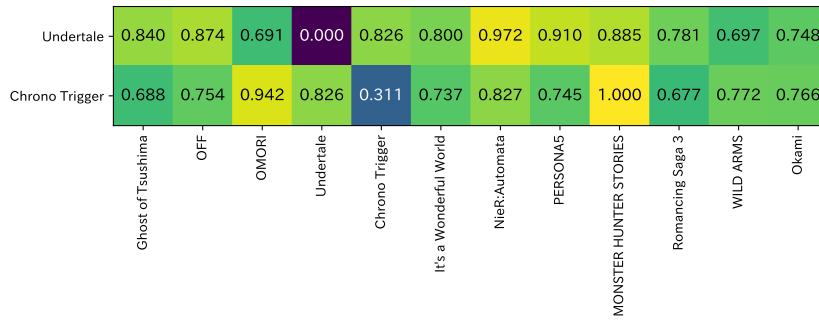
**Table 2.** Games used in the experiment as referred and test games

	Game	Usage	Outline
01	Ghost of Tsushima	Test game	A samurai joins a battle on a quest to protect Tsushima Island during the first Mongol invasion of Japan
02	OFF	Test game	An enigmatic humanoid entity called <i>Batter</i> travels the world on a <i>sacred mission</i> to <i>purify</i> the world
03	OMORI	Test game	The player explores both the real world with Sunny and the dream world with his alter-ego “OMORI” in the dream, overcoming his secrets
04	Undertale	Referred game	The player controls a child who has fallen underground and adventures back to the surface while meeting various monsters
05	Chrono Trigger	Referred game	The player controls a group of adventurers on a journey through time to prevent a global catastrophe
06	It’s a Wonderful World	Test game	Players are deprived of what is most precious to them and forced to participate in the Reaper’s Game for the survival of Shibuya
07	NieR:Automata	Test game	Players take on the role of human-made androids in a proxy war against an invading army of Machines from another world
08	PERSONA5	Test game	The player and his friends awaken their persona abilities and become the Phantom Thieves of Hearts to steal malevolent intent from the hearts of adults
09	MONSTER HUNTER STORIES	Test game	Players explore the world after the village where they live with the monsters they were born into is hit by a disaster
10	Romancing Saga 3	Test game	Rise of Morastrum occurs again, and the player ends up involved in the hunt for the Child of Destiny as eight main characters
11	WILD ARMS	Test game	Players control a boy who wields ARMS to prevent an otherworldly threat from reviving their lost leader and destroying the world
12	Okami	Test game	The player becomes Amaterasu and embarks on a journey to fulfill people’s wishes to defeat Yamata no Orochi and restore the world



**Table 3.** Copyright-free music websites used to create a music collection

	Site	URL
01	bensound	<a href="https://www.bensound.com/">https://www.bensound.com/</a>
02	DOVA-SYNDROME	<a href="https://dova-s.jp/">https://dova-s.jp/</a>
03	MusMus	<a href="https://musmus.main.jp/">https://musmus.main.jp/</a>
04	PeriTune	<a href="https://peritune.com/">https://peritune.com/</a>
05	Solitary Sound	<a href="https://az-ho.org/">https://az-ho.org/</a>
06	Devil Soul	<a href="https://maou.audio/">https://maou.audio/</a>



**Fig. 2.** Dissimilarity of visual scenes between games to determine test games

### 3.3 Evaluation on prediction of audio features

We experimented with evaluating audio feature prediction through our CNN-LSTM model. This evaluation compares the dissimilarity between the predicted audio features and those of actually used BGM. Because the effectiveness of the prediction would be different among audio feature categories, the effects of each feature category are evaluated individually, as well as their combinations.

**Method** We trained models with each feature category to evaluate the effects of each audio feature category individually. Because we have nine feature categories (Table 1) and two games (“Undertale” and “Chrono Trigger”) as referred games, we trained 18 models. Then, we gave the models six scenes  $S$  (two fight, two walk, and two talk scenes) from each of the two games (“OMORI” and “Romancing Saga 3”) as test games. Hence, we obtained the predicted audio features  $y_i^{\text{pred}}(s)$  and compared them with the audio features  $y_i^{\text{true}}(s)$  of actually used BGM ( $i$ : audio feature type,  $s \in S$ ): scene).

Because BGM for different scenes should have different audio features, we identified that the audio features have inter-scene variations. Specifically, we calculate  $i$  that maximizes the following equation:

$$\sum_{s \in S} \sum_{s' \in S \setminus \{s\}} \{\text{dist}(y_i^{\text{true}}(s'), y_i^{\text{pred}}(s)) - \text{dist}(y_i^{\text{true}}(s), y_i^{\text{pred}}(s))\}$$

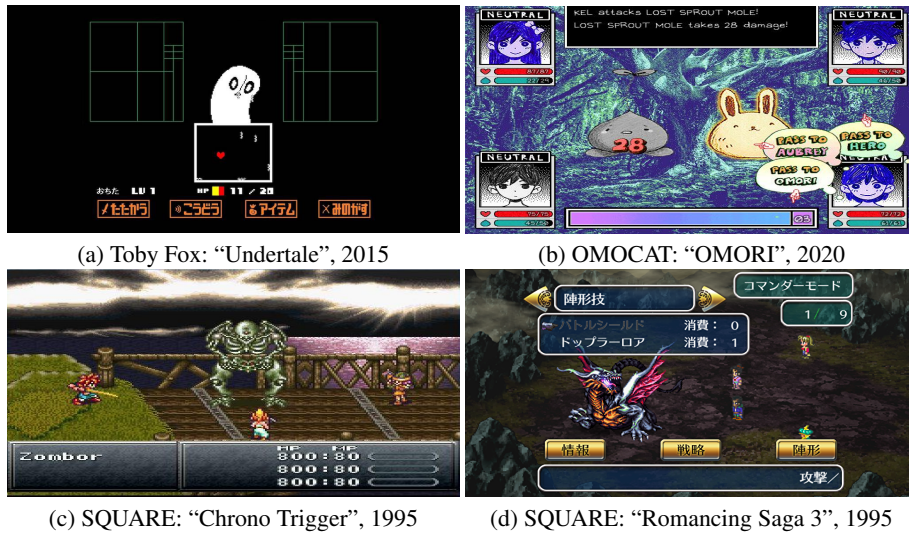


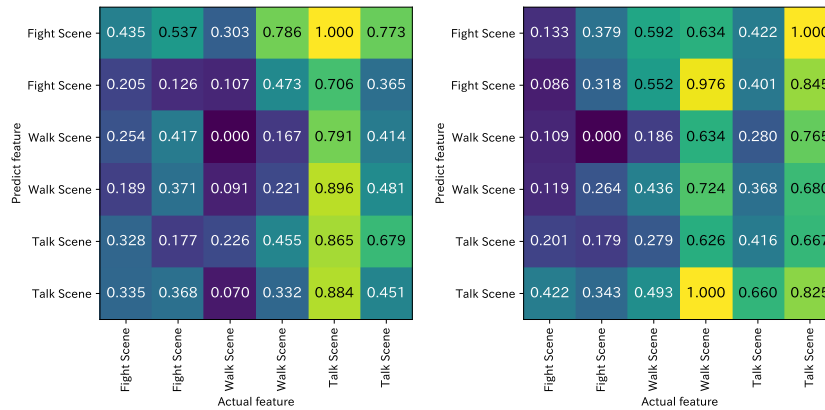
Fig. 3. Excerpts of visual scenes in the used games

where  $\text{dist}(y_i^{\text{true}}(s), y_i^{\text{pred}}(s))$  represents the distance between the predicted and actual audio features of the same scene.

**Results** For "OMORI" (referred game: "Undertale"), the use of only chroma\_cqt maximized the inter-scene variations of the audio features. For "Chrono Trigger" (referred game: "Romancing Saga 3"), the combination of chroma\_stft, chroma\_cqt, chroma\_cens, and mfcc\_delta2 maximized the inter-scene variations.

**Discussion** Fig. 4 (Left) shows the distance between the predicted features and actual features of each scene for "OMORI" (referred game: "Undertale"). Observations from this figure can be summarized as follows:

- When we focus on Fight Scene 1's predicted features, the distance from the actual features of the same scene should have been the smallest, but the distance from Walk Scene 1's actual features was the smallest. Also, for Fight Scene 2, the distance of its predicted features from Walk Scene 1's actual features was the smallest. These results imply that "Undertale" fight scenes and "OMORI" walk scenes may have similar features in BGM.
- When we focus on the two walk scenes, both Walk Scene 1's and Walk Scene 2's predicted features had the smallest distance from Walk Scene 1's actual features. It means that our models well predicted the walk scenes' audio features.
- For the two talk scenes, the distances between their predicted and actual features were the largest. In general, talk scenes' actual features tended to have large distances from all scenes' predicted features. This could be why "Undertale" tended to have few talk scenes.



**Fig. 4.** Distance matrix of predicted vs. actual audio features (Left: “Undertale”, Right: “Chrono Trigger”)

Fig. 4 (Right) shows the distance between the predicted features and actual features of each scene for “Chrono Trigger” (referred game: “Romancing Saga 3”). Observations from this figure are summarized as follows:

- When we focus on the two fight scenes, the scenes with the actual features with the minimal distance from the fight scenes’ predicted features were fight scenes. These results imply that “Chrono Trigger”’s fight scenes may have similar features BGM to each other.
- The walk scenes’ predicted features were closer to the fight scenes’ actual features than the walk scenes’ ones. Similarly, the talk scenes’ predicted features were also closer to the fight scenes’ actual features than the talk scenes’ actual features. As well as “OMORI”, the distance from the talk scenes’ actual features tended to be large in general. This could be why “Chrono Trigger” tended to have many fight scenes.

### 3.4 Evaluation on retrieval of musical pieces from the predicted audio features

We experiment with music output. If the same music is output, even if different scenes are input, it goes against the purpose of the research. Therefore, we verify which audio features are suitable for outputting different music.

**Method** We prepared six scenes from each of the 12 games listed in Table 2 (72 scenes in total). Let  $S = \{s_1, s_2, \dots, s_J\}$  be a set of the prepared scenes. For BGM retrieval, we used the music collection described in Section 3.1, which consists of 99 musical pieces taken from copyright-free music collection websites. Here,  $M = \{m_1, m_2, \dots, m_K\}$  be the music collection. For each scene  $s_i$  in  $S$ , we retrieved the musical piece that best fits the given scene. Here, the retrieved musical piece for scene  $s_i$  is represented by  $\text{output}(s_j)$ .

The important point is that retrieved musical pieces should differ for different scenes. In other words, for  $s_i$  and  $s_j$  ( $i \neq j$ ), it should be  $\text{output}(s_i) \neq \text{output}(s_j)$ . Therefore, for each musical piece  $m_k$ , we calculated the number of scenes,  $s_j$  ( $1 \leq j \leq J$ ) satisfying  $m_k = \text{output}(s_j)$ . This number is denoted by  $X_i(m_k)$  ( $i$ : the audio feature category). When the condition mentioned above is satisfied,  $X_i(m_k)$  should equal 0 or 1 (as long as  $M$  has a sufficiently large number of pieces compared to the number of scenes), and its expected value is  $J/K$ . Therefore, we evaluated the appropriateness of the BGM retrieval by calculating the mean squared error between  $X_i(m_k)$  and  $J/K$ . That is, we identified the most effective audio feature category  $\hat{i}$  that minimizes the following equation:

$$\hat{i} = \underset{i \in I}{\operatorname{argmin}} \sum_{k=1}^K \left( X_i(m_k) - \frac{J}{K} \right)^2$$

**Results** Experimental results show that `chroma_cqt` is the most effective audio feature category for learning “Undertale”. Table 4 lists the retrieval results obtained by inputting the scenes of “OMORI” into the model trained by “Undertale” with `chroma_cqt`. This shows that the same musical piece was output for different scenes (the two walk scenes and one talk scene).

For “Chrono Trigger”, `chroma_stft` is the most effective audio feature category. Table 5 lists the retrieval results obtained by inputting the scenes of “Romancing Saga 3” into the model trained by “Chrono Trigger” with `chroma_stft`. It shows that the same musical piece was output for Walk Scene 1 and Talk Scene 2. Otherwise, different musical pieces were output for different scenes.

**Table 4.** Musical pieces retrieved for scenes from “OMORI” (features: `chroma_cqt`, referred game: “Undertale”)

	Scene	EMD	Music title	Artist	URL
01	Fight Scene 1	0.132	Catch!!	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
02	Fight Scene 2	0.075	And then we ran	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
03	Walk Scene 1	0.044	Pursuer	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
04	Walk Scene 2	0.024	Pursuer	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
05	Talk Scene 1	0.125	And then we ran	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
06	Talk Scene 2	0.051	Pursuer	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>

**Table 5.** Musical pieces retrieved for scenes from “Chrono Trigger” (features: `chroma_stft`, referred game: “Romancing Saga 3”)

	Scene	EMD	Music title	Artist	URL
01	Fight Scene 1	0.078	And then we ran	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
02	Fight Scene 2	0.067	Pursuer	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
03	Walk Scene 1	0.033	Sonorously Box	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
04	Walk Scene 2	0.036	The Chuckling Witch	Hibiki Abe	<a href="https://az-ho.org/a-smiling-witch">https://az-ho.org/a-smiling-witch</a>
05	Talk Scene 1	0.088	Mid-range Strength	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>
06	Talk Scene 2	0.086	Sonorously Box	watson	<a href="https://musmus.main.jp/music_game.html">https://musmus.main.jp/music_game.html</a>

## 4 Conclusion

In this paper, we proposed a system that retrieves BGM that fits game scenes given as a sequence of screen images. Using gameplay videos taken from YouTube, we learned a CNN-LSTM-based transformation model from a sequence of screen images to audio features of BGM. Next, the system uses this model to predict the audio features that match the given game scene as BGM. Finally, the system retrieves the musical piece with the closest audio features to the predicted ones.

To confirm the effectiveness of this system, we conducted some experiments. In particular, the comparisons of the predicted audio features and those used in actual BGM show that the predicted features for fight scenes are close to those of the actual BGM. In contrast, the predicted features of walk scenes and talk scenes are not close to those of the same scenes' actual BGM. Also, we discussed retrieved musical pieces for each scene. Retrieved musical pieces should be different for different scenes. It was partly achieved, even though the same musical piece was output for some scenes.

This research is based on a strong assumption that screen images and BGM in games have explicit dependencies on each other. We believe this assumption is partly true but has not yet been fully confirmed. In the future, we will verify the appropriateness of our ideas with larger-scale data as well as the system's usability tests.

## References

1. T. Yamauchi, S. Nemoto, K. Nagano, S. Nakamura, A. Uda, Y. Saito, H. Murai, E. Tayanagi, K. Mukaiyama, and K. Hirata: "Game BGM Selection Based on Scenario and Emotional State", The 34th Annual Conference of the Japanese Society for Artificial Intelligence, pp.1–3, 2020 in Japan.
2. S. Nemoto, K. Ishikawa, A. Uda, T. Shiraishi, S. Nakamura, K. Nagano, T. Yamauchi, H. Murai, K. Hirata, K. Mukaiyama, and E. Tayanagi: "Extraction of Relationship between Character's Emotional State and BGM in Story Scene", JSIK, Vol.30, No.2, pp.263–269, 2020 in Japan.
3. E. Choi, Y. Chung, S. Lee, J. Jeon, T. Kwon, and J. Nam: "YM2413-MDB: A Multi-Instrumental FM Video Game Music Dataset with Emotion Annotations", ISMIR, arXiv:2211.07131, 2022.
4. Y. E. Kim, E. Schmidt, and L. Emelle: "Moodswings: A collaborative game for music mood label collection", ISMIR, pp.231–236, 2008.
5. T. Ishikawa: "Background Music Search System to an Input Video Using Factor Analysis for Impression Words", IIEEJ, Vol.9, No.2, pp.69–77, 2023 in Japan.
6. D. Zeng, Y. Yu, and K. Oyama: "Audio-Visual Embedding for Cross-Modal Music Video Retrieval through Supervised Deep CCA", IEEE ISM, pp.143–150, 2018.
7. AmBeat, <https://tollite.yamaha.com/Ambeat/>
8. Deep12, <https://www.sonycs1.co.jp/tokyo/14621/>
9. J. MacQueen: "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.1, pp.281–297, 1967.
10. N. Somu, R. Gauthama, and R. Krithivasan: "A deep learning framework for building energy consumption forecast", Renewable and Sustainable Energy Reviews, Vol.137, 2021.

11. R. Rial, A. R. R. Adhitya, and L. Hyun-Jin: “A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power”, *Energies*, Vol.13, 2019.
12. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner: “Gradient-based learning applied to document recognition”, *IEEE*, Vol.86, No.11, pp.2278–2324, 1998.
13. S. Hochreiter and J. Schmidhuber: “Short-Term Memory”, *Neural Computation*, Vol.9, No.8, pp.1735–1780, 1997.
14. D. P. Kingma and J. Ba: “Adam: A Method for Stochastic Optimization”, *ICLR*, arXiv:1412.6980, 2015.
15. S. J. Reddi, S. Kale, and S. Kumar: “On the Convergence of Adam and Beyond”, *ICLR*, arXiv:1904.09237, 2018.
16. B. McFee and Daniel P. W. Ellis: “Analyzing Song Structure with Spectral Clustering”, *ISMIR*, 15th International Society for Music Information Retrieval Conference, pp.405–410, 2014.
17. B. Logan and A. Salomon: “A music similarity function based on signal analysis”, *IEEE ICME*, pp.745–748, 2001.
18. Q. Xiao, S. Tsuge, and K. Kita: “Music retrieval method based on filter-bank feature and earth mover’s distance”, *Seventh International Conference on Natural Computation*, pp.1845–1849, 2021.

# A Music Exploration Interface Based on Vocal Timbre and Pitch in Popular Music

Tomoyasu Nakano<sup>1</sup>, Momoka Sasaki<sup>2</sup>, Mayuko Kishi<sup>2</sup>, Masahiro Hamasaki<sup>1</sup>,  
Masataka Goto<sup>1</sup>, and Yoshinori Hijikata<sup>2</sup> \*

<sup>1</sup> National Institute of Advanced Industrial Science and Technology (AIST)  
[t.nakano, masahiro.hamasaki, m.goto]@aist.go.jp

<sup>2</sup> School of Business Administration, Kwansai Gakuin University  
contact@soc-research.org

**Abstract.** This paper proposes an interface that enables music exploration focusing on two factors of singing voices, vocal timbre and pitch, that are useful in finding singing voices that match users' preferences. The proposed interface uses a two-dimensional color map to visualize songs being explored and locates them according to timbre or pitch similarities of their singing voices. Since similar songs are located closely on the map, users can visually find singing voices similar to their favorite singing voices. In addition to the location, the interface uses the color of each song on the map to visualize an additional factor related to characteristics of singing voices, such as acoustic features or words describing singing voices (*e.g.*, "Clear"). Prior to developing the interface, we conducted a questionnaire survey with 20 participants and confirmed that both vocal timbre and pitch are important when listening to music. The proposed interface was implemented with 102 songs, and a user study was conducted with 60 participants.

**Keywords:** Music information retrieval, vocal timbre, pitch histogram, singing descriptors, music exploration interface

## 1 Introduction

Since vocals are one of the major parts in music [1], music information retrieval (MIR) technologies focusing on singing voices are beneficial to a wide range of users [2]. In fact, MIR methods and interfaces that focus on various factors of singing voices — such as vocal timbre [3–6], vocal range profile (*i.e.*, pitch and intensity) [7], lyrics [5, 8–11], singing style [12–14], and gender [15] — have been proposed for the purpose of listening to songs or the purpose of finding songs to sing. In order to provide a new direction for such a series of academic studies, this paper proposes a novel interface that enables exploratory music retrieval focusing on multiple factors of singing voices.

\* This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number JP21H04917.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

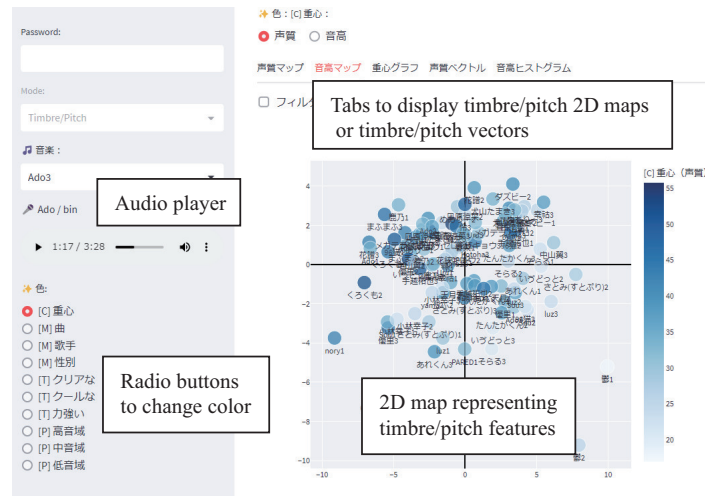


Fig. 1. Screenshot of the proposed interface.

Since singing voices have various factors, a music exploration interface that allows switching the visualized factors to be focused on is convenient for users with different purposes. For users who are interested in finding singers having a similar vocal timbre, visualizing the vocal timbre is useful, and for users who are interested in finding songs having a similar vocal pitch distribution, visualizing the vocal pitch is useful. The factors to be focused on thus depend on the purpose of the exploration.

We target vocal timbre and pitch for our interface. We consider these two factors to be effective in music exploration for two reasons. First, as a result of our survey explained later in which participants were asked to describe their favorite singing voices, many of the answers described vocal timbre and pitch. Second, it is helpful for users who want to find songs with their favorite singing voices to use or combine vocal timbre and pitch. Recently, there has been a culture in which a lot of people enjoy singing existing songs as cover versions and share their cover songs online. Users who enjoy such songs could find and enjoy songs having their favorite singing voices even if they do not know those songs or singers.

We therefore developed a music exploration interface that visualizes the two factors, vocal timbre and pitch, and enables users to switch the visualized factors to find songs having their favorite singing voices. A screenshot of our interface is shown in Figure 1. On the right side, each song is depicted as a circular dot on a two-dimensional color map representing the similarity of vocal timbre or vocal pitch factors, which can be interactively switched by a user. Since similar songs are located closely on the map, the user can easily find a song having a vocal timbre similar to that of the user's favorite singer on the map focusing on the vocal timbre similarity. The user can see the song title and singer name by mousing over a song. Each song has an identifier (ID) based on the singer name (e.g., *Ado3* means the third song of singer *Ado* in our dataset used for the interface), and the user can play back a song by specifying its ID from a pull-down menu on the left sidebar of the screen. The interface uses the color of the song to indicate one of the following: singer name, song title, singer gender, center of gravity



of the average mel spectrum, center of gravity of the pitch histogram, and singing descriptors (e.g., “Clear”). This additional color helps users understand the characteristics of singing voices in finding their favorite songs, and the combination of the location and color on the two-dimensional map gives high flexibility in visualizing multiple factors of singing voices. To the best of our knowledge, such a flexible music exploration interface that leverages both vocal timbre and pitch factors has not been proposed.

## 2 Related work

Related to this research are studies on music visualization interfaces for finding one’s favorite singers or lyrics. Fujihara *et al.* [3] proposed VocalFinder, an interface that retrieves songs having similar singing voices by modeling vocal timbre and singing style using a Gaussian mixture model. Hamasaki *et al.* [15] proposed Songrium, a music browsing assistance interface that has a function to analyze and visualize singing voices. It uses a circle to visualize a song, and the color and size of the circle indicate the singer’s gender and the number of play counts, respectively. Sasaki *et al.* [8] proposed LyricsRadar, an interface that estimates topic distributions from lyrics text using latent Dirichlet allocation and locates lyrics on a two-dimensional map using t-SNE [16]. Tsukuda *et al.* [10] proposed Lyric Jumper, an interface that visualizes lyric topic distributions for each singer as a donut chart to let users find singers with similar topics. Watanabe *et al.* [11] proposed Query-by-Blending, an interface that enables users to find songs by a query combining lyrics, song acoustic signals, and artists.

Map-based music browsing interfaces that locate songs on a two- or three-dimensional map have also been proposed [17]. In addition, as MIR methods targeting pitch, Tzanetakis *et al.* [18] used a pitch histogram to automatically classify music genres. Moreover, to recommend songs appropriate for the user’s singing ability, a feature called vocal range profile (VRP) has also been studied (e.g., [7]). The VRP indicates the range of intensity for each pitch that a singer can sing.

Words that describe singing voices help determine vocal characteristics that people are likely to pay attention to when listening to songs. There have been studies on emotional expressions of singing voices [19,20]. Scherer *et al.* [20] studied the correlation of acoustic features to “anger”, “fear”, “tenderness”, “joy”, “sadness”, and “pride” when eight professional opera singers sang musical scales. There have also been studies that determined a set of words that express impressions of singing voices and annotated them to songs [21,22] for their automatic estimation. Kanato *et al.* [21] defined a set of 47 impression words of singing voices. The factor analysis revealed three factors, “power,” “politeness,” and “brightness,” as well as 12 words (e.g., “clear” and “cute”) that comprise the singing impression scale. Kim *et al.* [22] defined 70 vocabulary words to describe solo singers. From results of five semi-experts’ annotations of actual songs using those 70 vocabulary words, 42 vocabulary words were obtained and classified into five categories: pitch (range), timbre, gender, genre, and technique.

Compared with the above studies, the key contribution of this paper is to develop a music exploration interface that uses a combination of vocal timbre and pitch. Another contribution is that we show the appropriateness of using vocal timbre and pitch as factors in music exploration by conducting a questionnaire survey.

### **3 Survey of preference for singing voice when listening to music**

Prior to the interface development, a questionnaire survey was conducted with 20 participants, males and females in their twenties. The purpose of the survey was to investigate users' impressions of vocals and their needs when listening to music. Although there were previous studies [21, 22] that defined words to describe singing voices in popular music, they did not focus on preference for singing voice when listening to music.

The questionnaire consisted of the following three sections:

- S1: a section to measure the participants' musical ability based on the Goldsmiths Musical Sophistication Index (Gold-MSI) [23],
- S2: a section in which participants were asked to write freely about their favorite singing voices, favorite artists/songs, and the reasons for their favorites, and
- S3: a section in which participants were asked to rate 3 vocal aspects (pitch height, pitch range, and timbre) on a 7-point scale.

#### **3.1 S1: Musical sophistication of participants**

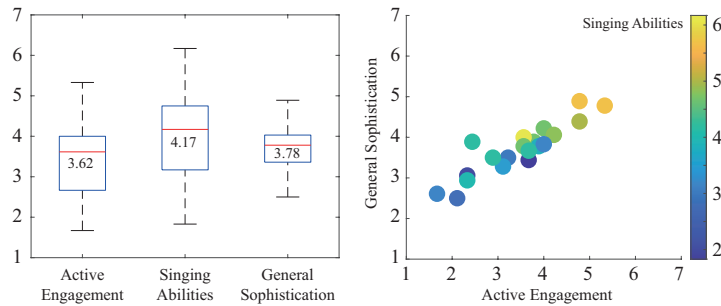
In the Gold-MSI, participants answer questions such as "I am able to hit the right notes when I sing along with a recording." on a 7-point scale from 1 (Completely Disagree) to 7 (Completely Agree). In this paper, we asked participants to answer questions on "Active Engagement" and "General Sophistication" because we thought they are relevant to music appreciation in general. Since the survey focused on singing, participants were also asked to answer questions about "Singing Abilities."

The scores for "Active Engagement," "Singing Abilities," and "General Sophistication" are shown in Figure 2. Each score was obtained by averaging the raw score values for the relevant questions for comparisons independent of the number of questions. In the scatter plot, differences in the Singing Abilities scores are represented by different colors. The results show that all median values of the scores were around 4. The correlation between Active Engagement and General Sophistication was high at 0.90, and their medians were slightly below 4, indicating a slightly lower score distribution. On the other hand, the median for Singing Abilities was slightly above 4, with a balanced distribution of high and low scores. The correlation between Singing Abilities and Active Engagement was 0.59, and that between Singing Abilities and General Sophistication was 0.74. These results indicate that the participants had an average interest and ability in music, and the high correlation between Active Engagement and General Sophistication indicated a certain degree of reliability in their answers.

#### **3.2 S2: Preference for singing voice when listening to music**

Here, analysis focused on answers to the following two open-ended questions.

- Q1 "Please describe as many characteristics as possible of your favorite singing voice when you listen to music."
- Q2 "Please describe the artist whose voice you like to listen to. Please also describe what you like about that artist's voice."



**Fig. 2.** Distribution of scores for Active Engagement, Singing Abilities, and General Sophistication in Gold-MSI. In the scatter plot, differences in Singing Abilities scores are indicated by different colors. The higher the scores, the more sophisticated with regard to those factors.

For the answers to Q1 and Q2, the words used by participants to describe the singing voice are shown below. These words are hereafter referred to as “singing descriptors.” The number of people who used them is also shown in parentheses. According to Kim *et al.* [22], each singing descriptor was classified into four categories: pitch, timbre (voice quality, singing style, emotion), gender, and singing ability (singing technique).

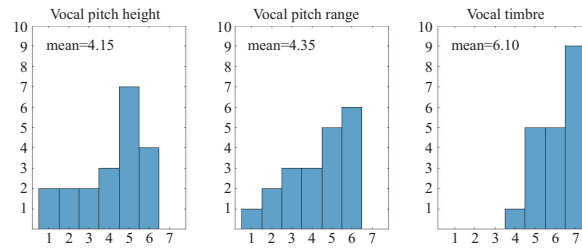
- **Pitch:** High-pitched (10), Low-pitched (6), Not too high (1), Mid-low range (1), Not too low (1), Very low (1), Wide range (1)
- **Timbre (voice quality, singing style, emotion)**<sup>3</sup>: Clear / Transparent (14), Beautiful (6), Unique (6), Tender (4), Powerful (4), Calm (4), Fluffy / Airy / Floating (3), Cute (3), Comfortable (3), Cool (2), Sexy (2), Sweet (2), Cheerful / Energizing (2), Likable (2), Rough (2), Soft (2), Deep (2), Delicate (2), Distinctive (2), Healing (2)
- **Gender:** Female (3), Male (2), Neutral (1)
- **Singing ability (singing technique):** Expressive (3), Falsetto (2), Large inflection (2), Vibrato (2), Strong (2), Accurate pitch control (2), Long tones without hoarseness (1), Comfortable high tone (1), Breathily (1), Head voice (1), Sound on inhalation (1), Precise control (1), Not labored (1), Emotional variation (1), Steady (1), Kobushi (1), Growl (1), Sing out from the stomach (1)

The above results show that singing descriptors related to pitch and timbre were frequently used. Pitch-related “High-pitched” and “Low-pitched” were included in 10 and 6 answers, respectively. Timbre-related “Clear / Transparent” and “Beautiful” were included in 14 and 6 answers, respectively. On the other hand, singing descriptors related to gender and singing ability (singing technique) were not frequently used. These results suggest that pitch and timbre are important in describing favorite singing voices.

### 3.3 S3: Factors of singing to be aware of when listening to music

Using a 7-point Likert scale, we asked participants to rate three vocal aspects (pitch height, pitch range, and timbre) that they are aware of when listening to music, without limiting themselves to specific songs.

<sup>3</sup> Since there were too many singing descriptors answered for the timbre category, only singing descriptors answered by two or more participants are shown.



**Fig. 3.** S3: Answers to questions related to vocal timbre and pitch on a 7-point Likert scale. The higher the score value, the more aware of the factor when listening to music.

The numbers of participants who answered each of the rating points (scores) are shown in Figure 3. The average scores for vocal pitch height and vocal pitch range were 4.15 and 4.35, respectively, indicating that the degree of awareness was higher than 4. The average score for vocal timbre was 6.1, which was also high.

### 3.4 Discussion

The results of this survey suggest that vocal pitch and timbre play important roles in determining a favorite singer's voice when participants with an average musical sophistication listen to music. This is also supported by previous studies in which pitch and timbre categories were used as vocal tags defined by Kim *et al.* [22] and music tags defined by Turnbull *et al.* [24]. We therefore believe that developing a music exploration interface focusing on vocal pitch and timbre is worthwhile and effective.

## 4 Interface

In order to visualize the similarity of vocal timbre and pitch and to enable exploratory search, we implement the interface (Fig. 1) as a map-based interface [17], which has been proposed widely in the past. The proposed interface estimates the timbre and pitch feature vectors of vocals from audio signals of each song and uses them to locate each song as a single circular point on a two-dimensional color map.

### 4.1 Data and back-end processing

The songs used for interface development are 51 songs for 17 female singers (3 songs for each singer), and 51 songs for 17 male singers (3 songs for each singer), for a total of 102 songs. All the songs had at least 10,000 views on YouTube as of December 2022 even though they are cover versions of 36 original songs of Japanese popular music (2 or 3 covers per song).

An overview of the back-end processing is shown in Figure 4. First, the singing voices were separated from all 102 songs using Hybrid Demucs [25]. To estimate pitch histograms, note-level pitch sequence was estimated by using Omnizart [26]. The pitch histograms were standardized by song to eliminate the effect of song length and then standardized by dimension and referred to as pitch vectors.

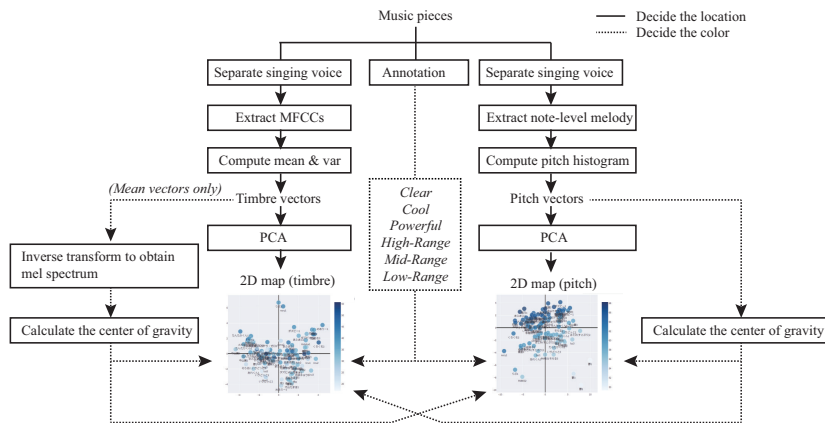


Fig. 4. Overview of the back-end processing of the proposed interface.

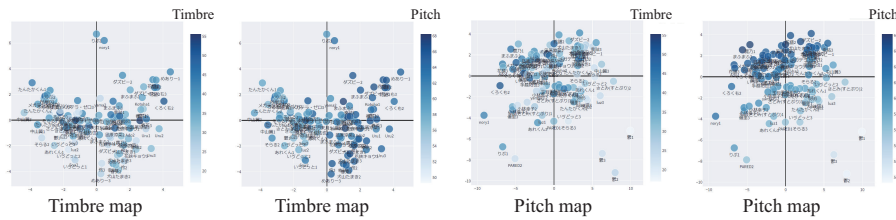
Timbre features were obtained by calculating the mean and variance of each dimension of MFCCs from the separated singing voice. To calculate MFCCs, STFT was calculated for a music signal with a sampling frequency of 22,050 Hz, with a window length of 2048 and a shift width of 512. The number of mel frequency bins was 128 and the MFCC dimension was 12, excluding DC components. Here, the vocal activity segments were determined by utilizing the note-level pitch information from Omnizart, and only the MFCCs for those vocal segments were used to calculate the mean and variance. Finally, the mean and variance of the MFCCs for all 102 songs were standardized by dimension and referred to as timbre vectors.

Finally, principal component analysis was performed on these timbre and pitch vectors, and we located them in a two-dimensional timbre map and a two-dimensional pitch map. The performance of Hybrid Demucs was high enough, but even if there were errors, they were unlikely to affect the histograms and mean vectors.

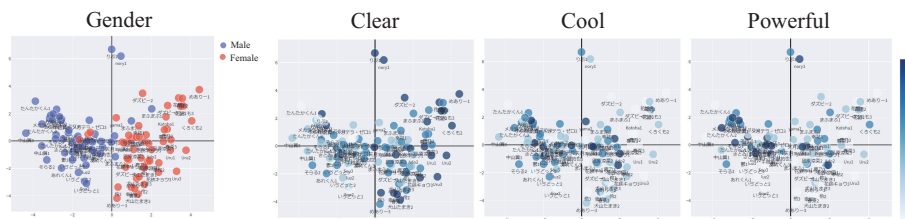
## 4.2 Annotate singing descriptors

For the purpose of improving the user’s understanding of the map, singing descriptors from human annotation are also used for coloring. To determine appropriate singing descriptors for each song, the 102 songs were tagged by six annotators, three male and three female. Three annotators per song were assigned to tag the singing voice, and at least one of the three was of a gender different from that of the singer of the song.

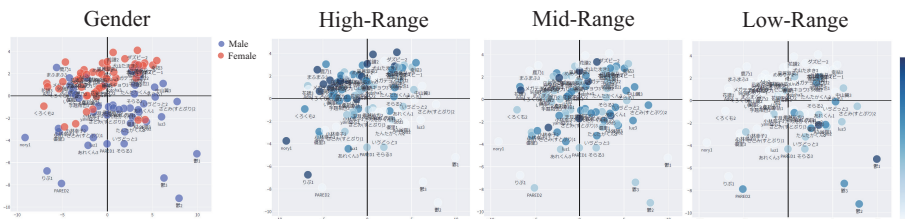
The singing descriptors used in this paper were determined based on previous studies [21, 22, 27] in which inter-annotator agreement, intelligibility, or synonymity were taken into account. First, 33 descriptors were selected from the tags used in the KVT dataset [22], 3 descriptors related to pitch range and 30 descriptors related to timbre. Then nine descriptors were added, including seven descriptors — Powerful, Nasal, Calm, Weak, Sexy, Resonant, and Dosu (Threatening / Frightening) — that were selected from previous studies on singing impression [21] and speech timbre [27], and two descriptors — Beautiful and Cool — that were from previous studies of singing im-



**Fig. 5.** Timbre maps and pitch maps colored using either the center of gravity of the timbre vector or that of the pitch vector.



**Fig. 6.** Timbre maps colored based on gender and three singing descriptors, “Clear,” “Cool,” and “Powerful.” Continuous coloring for each of the three descriptors depends on the number of annotators who labeled it.



**Fig. 7.** Pitch maps colored based on gender or three singing descriptors, “High-Range,” “Mid-Range,” and “Low-Range.” Continuous coloring for each of the three descriptors depends on the number of annotators who labeled it.

pression [21]. As a result, a total of 42 different descriptors were determined as singing descriptors labeled by the annotators.

Then, since using all the 42 descriptors gives too much information and is difficult, we used only the top three timbre descriptors — “Clear,” “Cool,” and “Powerful” — on the basis of how often they were annotated. As for the pitch descriptors, we used all the three descriptors for pitch ranges: “High-Range,” “Mid-Range,” and “Low-Range.”

### 4.3 Interaction

The user can select either the timbre map or the pitch map, and can change the color of the songs by using one of the following: singer name, song title, singer gender, center of gravity of timbre vector, center of gravity of pitch vector, and singing descriptor (the number of annotators who assigned it). Discrete coloring is applied to the singer name, song title, and vocal gender, and continuous coloring (*i.e.*, gradation) is applied to the rest. The singer name, song title, and vocal gender are taken from metadata of the songs.

Figure 5 shows the timbre and pitch maps, each of which is colored using either the center of gravity of the timbre vector or that of the pitch vector. Figures 6 and 7 also show the timbre and pitch maps, respectively, colored using vocal gender and the corresponding singing descriptors. We can see that the horizontal axis of the timbre map is correlated with gender (the correlation coefficient was 0.80) in the first map of Figure 6, placing female songs on the right and male songs on the left, as well as the center of gravity of the pitch vector (0.62) in the second map of Figure 5. It is also correlated with the number of annotators of “Clear” (0.57) in the second map of Figure 6, though the vertical axis of the timbre map is weakly correlated with the number of annotators of “Powerful” (0.34) in the fourth map of that figure.

The vertical axis of the pitch map is also correlated with gender (0.51) in the first map of Figure 7 as well as the center of gravity of the pitch vector (0.84) in the fourth map of Figure 5. It is also weakly correlated with the number of annotators of “High-Range” (0.38) and “Low-Range” (−0.48) in the second and fourth maps of Figure 7, though the horizontal axis of the pitch map is weakly correlated with the center of gravity of the timbre vector (−0.37) in the third map of Figure 5.

As shown in these examples, the proposed interface enables flexible changes in location and coloring with respect to timbre and pitch as well as related singing descriptors.

## 5 Evaluation

Since the proposed interface has functions for people who like music, we evaluated the effectiveness in terms of entertainment and knowledge discovery rather than efficiency and accuracy. Sixty participants, males and females in their teens or twenties, were assigned to the following groups, G1 through G3, each with 20 participants.

- **G1 (proposed):** Using music exploration interface based on vocal timbre and pitch
- **G2:** Using music exploration interface based on pitch
- **G3:** Using music exploration interface based on timbre

G2 and G3 are comparison groups to evaluate the effectiveness of the proposed interface. Participants assigned to G2 could not use the timbre map, the center of gravity of the timbre vector, or the singing descriptors for timbre. Participants assigned to G3 could not use the pitch map, the center of gravity of the pitch vector, or the singing descriptors for pitch. Prior to the start of the experiment, the experimenter verbally explained the experiment procedure to the participants in Japanese. The experiment was conducted on a laptop computer, and participants played music using canal-type wired earphones. Participants were paid 1,800 JPY for their participation in the experiment (approximately 1 hour and 45 minutes).

Participants first completed a questionnaire that measured their level of interest in music and then they watched a video explaining the interface. The explanation was made as easy to understand as possible for participants who are not familiar with MIR, using as little technical terminology as possible. Next, while recording the screen operation, participants were asked to explore their favorite music until they got bored within the duration of the experiment. After the experiment, each participant answered a questionnaire and was interviewed. In the post-experimental questionnaire, we assessed focused attention (FA), perceived usefulness (PU), aesthetic appeal (AE), and

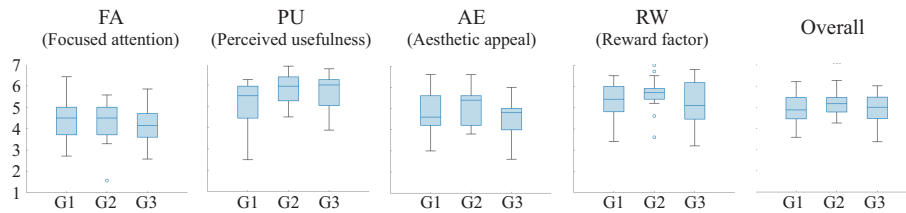


Fig. 8. Box plots of the scores in UES-LF.

reward factor (RW) using questions from the User Engagement Scale (UES-LF) [28] on a 7-point scale. Participants also answered open-ended questions about the pros and cons of the interface. In addition, participants in G1 answered whether they felt that vocal timbre or pitch was more suitable for them when exploring the music.

### 5.1 Results

First of all, the data of three participants (two in G2 and one in G3) were filtered out because the data were inappropriate (*e.g.*, the map was not used). Using the data from G1 to G3 after filtering, each score in the UES-LF was calculated. The average screen recording time for the 57 participants was 28.8 minutes (ranged from 11.8 to 53.1 minutes).

Their distribution is shown in Figure 8, where “Overall” is the overall engagement score, obtained by averaging the other four scores. A one-way ANOVA confirmed a significant difference only in PU at the 5% level ( $p = 0.037$ ). The results of Bonferroni’s multiple comparison test based on Wilcoxon’s rank-sum test showed no significant differences in all combinations. This suggested that the type of interface did not affect user engagement.

Regarding the answers to the experimental questionnaire, 13 of the 20 participants in G1 answered that the vocal timbre feature was more suitable when searching for music, while 7 participants answered that the pitch feature was more suitable. This confirmed the need for our interface that allows searching from multiple factors since the vocal timbre feature works best for some users and the pitch feature works best for others. Moreover, in the interview, seven participants in G1 commented that the combination of vocal timbre and pitch facilitated their exploration. Some participants in G1 to G3 understood their own preferences for vocal timbre and pitch, while others found that they unexpectedly liked timbres and pitches that they had thought they did not like.

The top three functions mentioned as pros by all 57 of the participants were the timbre and pitch maps by 29 participants and the timbre and pitch vectors by 16 participants. In addition, 11 participants mentioned the design and usability of the interface, and 11 participants mentioned the identification or change of their preferences. On the other hand, since the design and usability were also mentioned as cons by 50 participants, it is necessary to improve the usability of the implementation in the future. Eight participants also commented that they did not understand the meaning of the axes of the two-dimensional color map and that the differences in color according to acoustic features and singing descriptors did not match their own perception. Therefore, there is



a possibility of developing a better interface to help users grasp the meaning of acoustic features and singing descriptors.

## 5.2 Discussion

The following can be considered as reasons for the small differences in UES-LF between G1 and G2 or between G1 and G3.

- Exploring music from the visualization of pitch and timbre was a novel experience for the participants. Even for G2 and G3, the participants may have felt that it was enough for them to find preferred songs from a new point of view. In fact, some participants understood their own preferences for pitch and timbre and discovered new or unexpected preferences during the use of the interface.
- This may be due to the doubling of the amount of information and manipulation. The interface has become more complex, which probably increased the time and effort required for participants to become familiar with the interface operation.

Three participants in G2 commented that while they felt the pitch information was effective, they also wanted information on vocal timbre. Therefore, some users are expected to be more satisfied with our interface that allows for both pitch and timbre.

## 6 Conclusion

In this paper we proposed a music exploration interface that flexibly visualizes vocal timbre and pitch as well as singing descriptors. The questionnaire survey results indicated that the vocal timbre and pitch can be utilized to explore music. In the present analysis based on the UES-LF, no significant differences were identified between the proposed interface and the comparison interfaces. However, the results of the questionnaire and interviews indicated that music exploration based on vocal timbre and pitch not only provides enjoyment and fun but also leads to the discovery of preferences regarding timbre and pitch. Future work will include building an interface that improves usability and taking into account the singer's singing style.

## References

1. Demetriou, A., *et al.*: Vocals in Music Matter: the Relevance of Vocals in the Minds of Listeners. Proc. ISMIR 2018, pp. 514–520 (2018).
2. Humphrey, E.J., *et al.*: 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, pp. 82–94 (2019).
3. Fujihara, H., *et al.*: A Modeling of Singing Voice Robust to Accompaniment Sounds and Its Application to Singer Identification and Vocal-Timbre-Similarity-Based Music Information Retrieval. IEEE TASLP, vol.18, no.3, pp. 638–648 (2010).
4. Nakano, T., *et al.*: Vocal Timbre Analysis Using Latent Dirichlet Allocation and Cross-Gender Vocal Timbre Similarity. Proc. ICASSP 2014, pp. 5239–5343 (2014).
5. Nakano, T., *et al.*: Musical Similarity and Commonness Estimation Based on Probabilistic Generative Models of Musical Elements. IJSC, vol.10, no.1, pp. 27–52 (2016).

6. Nakano, T., *et al.*: Musical Typicality: How Many Similar Songs Exist?. Proc. ISMIR 2016, pp. 695–701 (2016).
7. Mao, K., *et al.*: Competence-Based Song Recommendation: Matching Songs to One’s Singing Skill. IEEE Trans. on Multimedia, vol.17, no.3, pp. 396–408 (2015).
8. Sasaki, S., *et al.*: LyricsRadar: A Lyrics Retrieval System based on Latent Topics of Lyrics. Proc. ISMIR 2014, pp. 585–590 (2014).
9. Nakano, T., *et al.*: LyricListPlayer: A Consecutive-Query-by-Playback Interface for Retrieving Similar Word Sequences from Different Song Lyrics. Proc. SMC 2016, pp. 344–349 (2016).
10. Tsukuda, K., *et al.*: Lyric Jumper: A Lyrics-Based Music Exploratory Web Service by Modeling Lyrics Generative Process. Proc. ISMIR 2017, pp. 544–551 (2017).
11. Watanabe, K., *et al.*: Query-by-Blending: A Music Exploration System Blending Latent Vector Representations of Lyric Word, Song Audio, and Artist. Proc. ISMIR 2019, pp. 144–151 (2019).
12. Ohishi, Y., *et al.*: A Stochastic Representation of the Dynamics of Sung Melody. Proc. ISMIR 2007, pp. 371–372 (2007).
13. Yamamoto, Y., *et al.*: Analysis and Detection of Singing Techniques in Repertoires of J-POP Solo Singers. Proc. ISMIR 2022, pp. 384–391 (2022).
14. Yakura, H., *et al.*: Self-Supervised Contrastive Learning for Singing Voices. IEEE/ACM TASLP, vol.30, pp. 1614–1623 (2022).
15. Hamasaki, M., *et al.*: Songrium: A Music Browsing Assistance Service with Interactive Visualization and Exploration of a Web of Music. Proc. WWW 2014 (2014).
16. Van der Maaten, L., *et al.*: Visualizing Data using t-SNE. JMLR, vol.9, no.11 (2008).
17. Knees, P., *et al.*: Intelligent User Interfaces for Music Discovery. TISMIR, vol.3, no.1, pp. 165–179 (2020).
18. Tzanetakis, G., *et al.*: Pitch Histograms in Audio and Symbolic Music Information Retrieval. JNMR, vol.14, no.2, pp. 143–152 (2003).
19. Scherer, K.R. Vocal Communication of Emotion: A Review of Research Paradigms. Speech Communication, vol.40, pp. 227–256 (2003).
20. Scherer, K.R., *et al.*: The Expression of Emotion in the Singing Voice: Acoustic Patterns in Vocal Performance. J. Acoust. Soc. Am., vol.142, no.4, pp. 1805–1815 (2017).
21. Kanato, A., *et al.*: An Automatic Singing Impression Estimation Method Using Factor Analysis and Multiple Regression. Proc. Joint ICMC SMC 2014, pp. 1244–1251 (2014).
22. Kim, K.L., *et al.*: Semantic Tagging of Singing Voices in Popular Music Recordings. IEEE/ACM TASLP, vol.28, pp. 1656–1668 (2020).
23. Müllensiefen, D., *et al.*: The Musicality of Non-Musicians: An Index for Assessing Musical Sophistication in the General Population. PLOS ONE, vol.9, no.2 (2014).
24. Turnbull, D., *et al.*: Semantic Annotation and Retrieval of Music and Sound Effects. IEEE/ACM TASLP, vol.16, no.2, pp. 467–476 (2008).
25. Défossez, A. Hybrid Spectrogram and Waveform Source Separation. Proc. MDX 2021, pp. 1–11 (2021).
26. Wu, Y.T., *et al.*: Omnizart: A General Toolbox for Automatic Music Transcription. JOSS, vol.6, no.68, p. 3391 (2021).
27. Kido, H. and Kasuya, H.: Representation of Voice Quality Features Associated with Talker Individuality. Proc. ICSLP 1998, pp. 1–4 (1998).
28. O’Brien, H.L., *et al.*: A Practical Approach to Measuring User Engagement with the Refined User Engagement Scale (UES) and new UES Short Form. Intl. J. of Human-Computer Studies, vol.112, pp. 28–39 (2018).

# Exploring Diverse Sounds: Identifying Outliers in a Music Corpus

Le Cai<sup>1</sup>, Sam Ferguson<sup>1</sup>, Gengfa Fang<sup>1</sup>, and Hani Alshamrani<sup>1</sup> \*

Creativity and Cognition Studios  
Faculty of Engineering and IT  
University of Technology Sydney

**Abstract.** Existing research on music recommendation systems primarily focuses on recommending similar music, thereby often neglecting diverse and distinctive musical recordings. Musical outliers can provide valuable insights due to the inherent diversity of music itself. In this paper, we explore music outliers, investigating their potential usefulness for music discovery and recommendation systems. We argue that not all outliers should be treated as irrelevant data, as they can offer unique perspectives to contribute to a richer musical understanding. We attempt to identify 'Genuine' music outliers, which may reveal unique aspects of an artist's repertoire and serve to enhance music exploration and discovery.

**Keywords:** Music Outlier · Music Outlier Detection · Audio characteristics · Music discovery

## 1 Introduction

In the field of music information retrieval, a primary focus is often on finding similarities among digital musical recordings, to enable recommendation systems and facilitate music discovery [10, 14, 23, 5]. Given this context, analysis of outliers has attracted less research attention [18], as they are often considered irrelevant data and removed during preprocessing, or are naturally scored lower by most similarity-focused algorithms [7]. However, outliers in the context of music can provide interesting insights and reveal unique patterns, as music inherently exhibits great diversity [20, 13].

In this paper, we explore the identification and categorization of music outliers, with an aim to ultimately enhance music discovery and recommendation systems. We propose a method to describe and discover genuine musical outliers based on audio characteristics, such as tempo and loudness. By doing so, we aim to identify outliers that can provide valuable information for music discovery while not being non-musical.

---

\* We would like to acknowledge the support received from Assoc. Prof. Sam Ferguson. In Addition, Le Cai wants to thank his partner Hanyu Meng for her unwavering patience and constant presence in his life.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

We present a definition of what constitutes a 'Genuine' music outlier and investigate its characteristics. Genuine outliers exhibit unique characteristics that set them apart from an artist's main style, providing insightful information for music discovery.

This paper is structured as follows: The introduction presents the motivation of our study, followed by a comprehensive literature review, which discusses the relevant background and prior research. Next, the aims and objectives are followed by the methodology section which outlines our proposed definition of Genuine music outliers and the subsequent dataset and algorithm developed for the detection. The results and discussion section evaluates the effectiveness of our algorithm based on the dataset and provides insights into its performance. Finally, the conclusion summarizes our findings and highlights the implications of our work, while also suggesting potential avenues for future research in the realm of music outlier detection and analysis.

## **2 Related Work**

### **2.1 What Makes a Song Different from Audio?**

Understanding outliers in the context of music recommendation systems necessitates a thorough examination of their diverse nature and the ability to differentiate actual music from other forms of audio. Müller [17] provides a comprehensive overview of music structure analysis, focusing on techniques for segmenting and organizing music into meaningful sections, laying the foundation for understanding the key aspects of music structure and demonstrating how various representations and algorithms can be used to analyze and compare music pieces. A system for finding structural descriptions of musical pieces defines the structure of a piece as segments with specific time ranges and labels, with segments sharing the same label considered occurrences of a particular structural part [21]. In another study, a multi-task deep learning framework is introduced for directly modeling structural semantic labels in music, such as "verses" and "choruses", from audio signals. This approach proposes a 7-class taxonomy that includes intro, verse, chorus, bridge, outro, instrumental, and silence, and consolidates annotations from four different datasets [27]. A large-scale analysis of songs in 315 different societies has found that songs share universal features like tonality, rhythm, and repetitive structures [16]. A study conducted by Shuqi et al [8]. analyzes the significance of repetition and structure in music, specifically in popular music, and demonstrated that deep learning models often struggle to identify these essential elements, which are crucial for generating coherent and appealing musical pieces. Subsequently, Sargent et al [22]. introduced a fourth principle: regularity. This principle posits that musical segments possess a certain degree of regularity, which can be leveraged to better understand and analyze the structure of the music.

### **2.2 Outlier Detection Approaches**

The purpose of outlier detection algorithms is to identify patterns and samples that deviate significantly from the normal characteristics of a group of data [12]. The reason to detect outliers is it can providing interesting insights contribute to a richer understanding of an artist's work. General outlier detection methods can be categorized into 4 categories based on an overview conducted by [26]:

**Clustering-based methods:** use a clustering algorithm to classify the majority of the elements of the set, while also clearly defining the outlying elements of the set.

**Density-based methods:** identifying outliers as points in low-density regions within a data set.

**Distance-based methods:** determining the distance between points, and considering outliers to be points with a large distance from their nearest neighbors.

**Statistical methods:** using measures such as mean, median and standard deviation to identify data points that fall outside of a defined range.

Clustering-based methods, exemplified by [2], utilize subspace cluster analysis to construct classification trees while addressing dataset scarcity. A deep learning model using a Clustering Augmented Learning Method (CALM) classifier improves genre classification by extracting deep time series features [11]. In contrast, density-based methods like DBSCAN [9] and OPTICS [1] identify outliers in low-density regions within a dataset, as demonstrated by the OPTICS algorithm applied to traditional Chinese folk music [28]. In the exploration of automatic outlier detection methods on music genre datasets, Lu et al [15] characterized outliers using their musical attributes, demonstrating the potential of these techniques to unveil unique insights into music structure and diversity within genre classification. Distance-based methods, including K-means [4] and CLARANS [19], examine the distance between data points and their nearest neighbors, considering outliers as points with large distances from the nearest neighbors. This approach has been successfully applied to traditional Irish music in [24]. Meanwhile, statistical-based methods employ measures such as mean, median, and standard deviation to identify data points outside a defined range to reveal patterns in cluster structure dynamics in popular music data [25]. However, while these studies focus on detecting outliers based on their statistical properties, the potential of the outliers themselves for music discovery has not been as extensively investigated.

### 3 Aims & Objectives

In this paper, our aims center around exploring the potential of music outliers for music discovery. To achieve this, we propose the following objectives:

**Propose an approach to describe musical outliers:** by examining various attributes that distinguish them from an artist's typical style, facilitating their identification and analysis. This implies establishing a clear definition to describe 'Genuine' music outliers as a distinct category of outliers that exhibit meaningful deviations from a set of existing digital musical recordings.

**Categorise music outliers:** into meaningful categories based on their distinguishing characteristics to create a meaningful interpretation, that also can help find the outliers that are helpful to understanding and discovering interesting music while excluding outliers that hold little data.

### 4 Method

We introduce the concept of genuine outliers within the context of music data and explore their potential value in recommendation systems. To accomplish this, we first

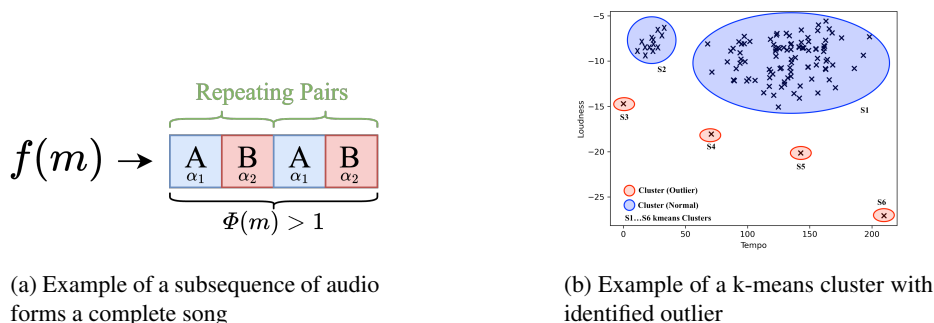


Fig. 1: The Definition of Genuine Music Outliers

propose a definition for genuine outliers, then create a labelled dataset for evaluation, and finally, apply an outlier detection algorithm to validate our definition.

#### 4.1 Definition of Genuine Music Outliers

A “Genuine” music outlier is a complete song that maintains an artist’s typical musical structure while distinctly diverging in sound and style from their predominant body of work, due to the complex nature of music, in our scenario, we consider pop music only.

To achieve this, a genuine outlier must satisfy the following constraints: 1): Forms a complete song, this distinguishes the identified outlier must be a song, not something else, e.g. speech, or sound effect, shown in fig.1a. To ensure this, audio must satisfy these conditions: a): The length of music structure  $\phi$  must be greater than one in a subsequence, this means the input recordings must at least have one or more music structures otherwise it is not music. b): The identified music structure must at least has one unique pair. e.g. a music structure in  $(A - B)$  is a song that can identify at least one unique pair, not  $(A - A)$  or  $(B - B)$ . c): The identified music structure must contain repeated parts, as repetition is a crucial element in music. A piece of audio that forms music should exhibit repeating sections, such as  $(A - B - A - B)$  patterns. The formal definition of a Genuine outlier is as follows:

Let  $M$  represents the set of all recordings produced by a specific artist. For a finite set  $S_M$ , suppose function  $\Phi : M \rightarrow \mathbb{N}^+$  that maps a recording to a finite integer and a function  $f_\Phi : M \rightarrow S_M^{\Phi(M)}$  that maps a recording  $m \in M$  to a finite length sequence. A recording  $m \in M$  is defined as a “Genuine” music outlier if the following conditions is satisfied:

**1. Forms A Complete Song:** For any recording  $m \in M$ , the sequence  $f_\Phi(m)$  and number  $\Phi(m)$  should satisfy the following:

1.  $\Phi(m) > 1$ ,
2. There exist a subsequence  $a_{k_1} \dots a_{k_p} (k_1 < \dots < k_p, p > 1)$  from  $f_\Phi(m)$  that, at least one repeated subsequence from  $f(m)$  can be found, i.e., there exist  $a_{k'_1} \dots a_{k'_p} (k'_1 < \dots < k'_p, p > 1)$  that  $k_1 \neq k'_1$  and  $k_p \neq k'_p$ .

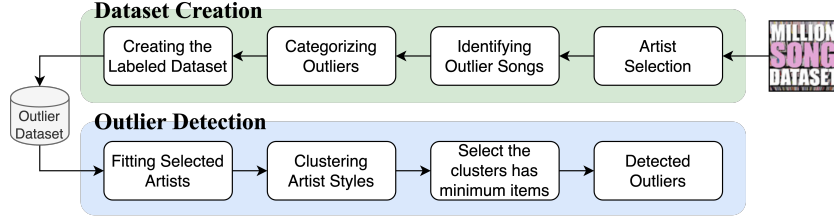


Fig. 2: Dataset Creation Process

For any given recording  $m \in M$ , if the previous condition satisfied, for given constant  $C_G \in (0, 1]$ ,  $\kappa$  and a positive integer  $N_d$ , either of the following conditions must be satisfied,

**2. Distinctiveness:** Define  $F : M \rightarrow \mathbb{R}^n$  be a function that maps each recording to an  $n$ -dimensional feature space, where  $n$  is a positive integer representing the number of musical features being considered. Define  $\text{Card}(\cdot)$  as the Cardinality of a set,  $\|\cdot\|$  as the 2-norm and  $\kappa$ -means cluster sets  $\Omega = \{\Omega_1, \dots, \Omega_\kappa\}$  ( $1 \leq \kappa \leq \text{Card}(M)$ ) that

$$\begin{aligned} \Omega \in \arg \min_{\Omega} \sum_{i=1}^{\kappa} \sum_{\mathbf{x} \in \Omega_i \subset M} \|F(\mathbf{x}) - \mu_i\|^2 \\ \text{s.t. } \mu_i = \frac{1}{\text{Card}(\Omega_i)} \sum_{\mathbf{y} \in \Omega_i \subset M} F(\mathbf{y}), \text{ for } 1 \leq i \leq \kappa. \end{aligned} \quad (1)$$

For any given integer  $N_d > 0$ , for any  $m \in M$ , if  $m \in \Omega_i$  and  $\text{Card}(\Omega_i) < N_d$ , we say that recording  $m$  is distinct with respect to  $M$ .

**3. Non-adherence:** For a given positive number  $C_G \in (0, 1]$ . We define

$$R_M(m) := \frac{\text{Card}(\{m' | f_{\Phi}(m') = f_{\Phi}(m) \text{ for all } m' \in M\})}{\text{Card}(M)}. \quad (2)$$

If  $R_M(m) < C_G$ , we say  $m$  is not adhere to set  $M$ .

In conclusion, if recording  $m$  does form a complete song, either the song  $m$  is distinct with respect to  $M$  or  $m$  is not adhere to set  $M$ , with given  $C_G$ ,  $\kappa$ , and  $N_d$ , then we say song  $m$  is an outlier.

#### 4.2 Dataset Creation and Outlier Selection Algorithm

In order to evaluate our above-proposed definition for Genuine music outliers, we created a dataset specifically designed to examine our hypothesis, To achieve this goal, the following steps were carried out:

**Artist Selection:** From the Million Song Dataset <sup>1</sup>, a smaller sample consisting of 10,000 songs extracted from the MSD, we randomly chose 20 artists using a random

<sup>1</sup> Million Song Dataset [6] is available at: <http://millionsongdataset.com>

number generator to pick artist IDs. For each of the selected artists, we identified a list of all the songs contained in the dataset with the particular artist id.

**Identifying Outlier Songs:** To identify and categorize outliers within the selected artists' music, we conducted a manual listening and labeling process for these songs by ear. First, we acquainted the artist's typical styles. To achieve that, we listened to about 20%-30% songs in this artist's main cluster. After establishing the primary styles, we listened to these songs once with selected artists again, focusing on identifying tracks that significantly deviated from the typical style. Attention was paid to musical elements, such as tempo, melody, harmony, instrumentation, and song structure.

**Categorizing Outliers:** We then listened to the characteristics of the identified outliers, focusing on specific attributes such as tempo, melody, harmony, instrumentation, and song structure that differentiate them from the artist's dominant style. By examining these properties, we classified outliers into five categories: Error, Speech, Intro, Sound Effect, and Genuine. Each identified outlier was assigned to one of the categories based on a two-step process. First, we selected songs with a Euclidean distance greater than 3 times the z-score threshold from the main cluster. Second, we manually reviewed these outliers, focusing on listening to their distinctive features such as tempo, loudness, melody, and the composition of instruments.

**Creating the Labeled Dataset:** Finally, After identifying and categorizing outliers, we compiled a list of all songs from the selected artists, along with their corresponding outlier categories. This list included each song's title, artist, genre, and other metadata. Utilizing the pre-extracted audio features from the MSD Subset, such as tempo, loudness, key, and mode, we created a dataset compiled with data including outlier categories and the MSD pre-extracted features.

We consider this to be a classification problem where each artist typically consists of 1 main distinct style. Note we only considered the case for 1. Distinctiveness and 2. Non-adherences for the current approach, the case of forms a complete song is discarded due to the complexity of the analysis of music structures.

Let  $M$  be a set of recordings under an artist. Let  $x_i \in \mathbb{R}^d$  to denote a  $d$ -dimensional feature vector for the  $i$ -th song from  $M$  ( $1 \leq i \leq \text{Card}(M)$ ). Suppose  $K$  is an integer that  $1 \leq K \leq \text{Card}(M)$ . For  $1 \leq k, k' \leq K$ , define  $C_k$  as a subset of  $M$ , such that  $C_k \cap C_{k'} = \emptyset$  for  $k \neq k'$ , and  $\cup_{1 \leq i \leq K} C_i = M$ . In our case,  $d = 2$ . We adopt k-means algorithm [3] to partition set  $M$  that satisfies such conditions. In the next section, we first delineate the outlier categorization, followed by the presentation of outlier detection results.

## 5 Results

### 5.1 Outlier Categorisations

We found 34 genuine outliers from 29 artists in 320 songs as well as 17 errors, 5 speeches, 1 intro, 5 noise, and 3 sound effects from 29 artists shown in Fig. 3, they can categorize into 5 types: *a) Error:* such as erroneous data value, e.g. doubling tempo due to feature extractor failure, incorrect data type, such as text entered as a numeric type. *b) Noise:* unintentional sounds such as non-musical noise, jitter, or glitches in a



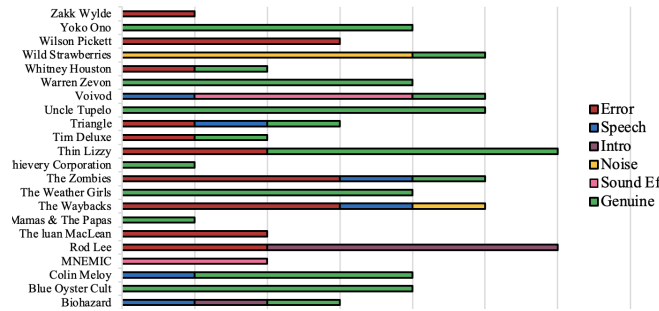


Fig. 3: Outlier Dataset: Categorisation of Outliers by Artists

recording, often due to recording issues or equipment quality. e.g. a live performance recording with excessive audience noise. c) **Speech**: such as some spoken words, or a short story integrated into the music to enhance the atmosphere of the listening experience. these types of practices commonly occur in certain music genres, such as hip and electronic music. d) **Sound Effect**: This type of recording is used to create a certain mood, it can build tension and add extra impact to certain sections of the song. Particularly, sound effect is used in certain genres of music such as electronic & rock music to add extra depth and interest to music. e) **Intro**: A very short track is often less than 30 seconds in length and serves as an introduction or a brief to establish the identity of the album. f) **Genuine**: a musical piece that deviates significantly from an artist’s typical style or the norm within a genre, exhibiting unique characteristics. These outliers are not classified as errors, noise, or other non-musical categories.

## 5.2 Automation Outlier Detection Result

The outlier detection algorithm was applied to the dataset, and the results obtained were analyzed to assess the performance of the method. The table 1 summarizes the performance of the outlier detection for various artists in the dataset. **True Positive Rate (TPR)** and **False Positive Rate (FPR)** represent the proportion of outliers that were correctly and incorrectly identified as outliers, respectively. **True Negative Rate (TNR)** and **False Negative Rate (FNR)** represent the proportion of non-outliers that were correctly and incorrectly identified as non-outliers, respectively. **Not Applicable (N/A)**: indicates no outliers being identified for a particular artist.

# 6 Discussion

## 6.1 Outlier Categorisation

Genuine outliers are complete songs that adhere to an artist’s typical musical structure but differ in sound and style. Often custom-made for specific events, they incorporate unique elements like distinct percussion. For instance, Colin Meloy’s “Lazy Little Ada” is situated at the center of Blue Oyster Cult’s cluster (Figure 4), deviating from Meloy’s

Table 1: The Result of Automatic Outlier Detection

Artist Name	TPR	FPR	TNR	FNR
Zakk Wylde	0	0.455	0.545	1
Blue Oyster Cult	1	0	1	0
Biohazard	0.5	0.381	0.619	0.5
Yoko Ono	0.25	0.049	0.951	0.75
Wilson Pickett	1	0.316	0.684	0
Wild Strawberries	1	0.087	0.913	0
Whitney Houston	0.5	0	1	0.5
Warren Zevon	1	0.356	0.644	0
Voivod	0.833	0	1	0.167
Uncle Tupelo	0.4	0.217	0.783	0.6
Triangle	0.75	0	1	0.25
Tim Deluxe	1	1	0	0
Thin Lizzy	0.429	0.37	0.63	0.571
Thievery Corporation	0.667	0.299	0.701	0.333
The Zombies	1	0.378	0.622	0
The Weather Girls	1	0.344	0.656	0
The Waybacks	0.5	0.214	0.786	0.5
The Tramps	N/A	0.136	0.864	N/A
The Subhumans	N/A	0.133	0.867	N/A
The Skatalites	N/A	0.317	0.683	N/A
THERION	0	0.275	0.725	1
The Mutton Birds	N/A	0.269	0.731	N/A
The Mission	N/A	0.324	0.676	N/A
The Mamas & The Papas	1	0.206	0.794	0
The Juan MacLean	N/A	0.324	0.676	N/A
Zee Avi	N/A	0.231	0.769	N/A
MNEMIC	1	0.043	0.957	0
Rod Lee	1	0.167	0.833	0
Colin Meloy	0.333	0.062	0.938	0.667

usual style yet resembling Blue Oyster Cult’s. In comparison, Meloy’s cluster exhibits lower loudness, potentially due to less percussion. When percussion is added to Meloy’s outliers, they show similarities to Blue Oyster Cult’s cluster. Conversely, most of Blue Oyster Cult’s songs form a well-defined cluster, but 4 outliers incorporate synthesizers and lack percussion, producing similarity to Meloy’s cluster.

In contrast, non-genuine outliers can be categorized into four types: Error, Speech, Sound Effect, and Intro. 1) Error tracks result from feature extractor misinterpretations or exceptions, such as doubling tempo or missing audio features. 2) Speech is often used in albums for storytelling, setting narrative themes, or as interludes to create a narrative flow between musical pieces, e.g., Kendrick Lamar’s “To Pimp a Butterfly.” 3) Sound effects enhance the musical narrative, especially in concept albums like Pink Floyd’s “The Trial.” 4) Intros are short tracks that set the tone, introduce themes, or provide transitions between songs. However, these tracks do not follow typical musical structures (e.g., ABAB, ABAC), so we consider them non-genuine outliers.

## 6.2 Outlier Detection Result

The overall results of the outlier detection algorithm show that it is capable of identifying outliers in an artist’s body of work. The algorithm performs exceptionally well on some artists by correctly classifying all outliers and non-outliers. However, varying performance among artists suggests the need for refining the algorithm with additional constraints. In summary, we can obtain these insights from the following results:

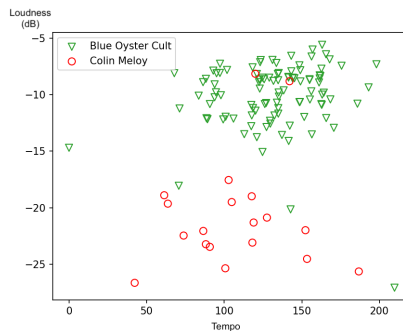


Fig. 4: Outlier Overlap: Shared characteristics between two artists' outliers.

The consideration of distinctiveness in the algorithm has worked well on artists that primarily have one style, for example, Blue Oyster Cult, The Mamas & The Papas, and Wild Strawberries had a perfect TPR and TNR, indicating that the definition works well for this artist even without considering the constraints of whether it forms a complete song. Furthermore, we find the distinctiveness constraint effective in identifying genuine music outliers in some artists. For example, the algorithm isolated outliers in artists such as Blue Oyster Cult (TPR: 1.0, FPR: 0, TNR: 1, FNR: 0) and Warren Zevon (TPR: 1.0, FPR: 0.356, TNR: 0.644, FNR: 0).

However, the varied performances across different artists suggest that discarding the consideration of constraints 1. Forms A Complete Song and 2. Non-adherence may lead to the algorithm's inability to accurately determine whether an outlier is genuine, as it lacks the capacity to assess the music structure of input recordings. The reasons causing these varying performances can be summarized as follows:

**Mixed content:** Recordings may contain various non-musical elements, including intros with predominantly speech content (e.g., "Rod Intro" by Rod Lee), live recordings featuring audience applause or speech interactions with the audience (e.g., "Dracula's Daughter" by Colin Meloy), or studio chats consisting solely of speech (e.g., "The Way I Feel Inside / Studio Chat" by The Zombies). These non-musical segments may introduce noise and impact the algorithm's performance.

**Noise and artifacts:** The presence of noise, artifacts, or other non-musical elements within a recording might lead to it being classified as an outlier, even if it does not constitute a genuine outlier in terms of musical content. Such factors can interfere with the algorithm's capability to accurately assess a song's structure.

**Transitional pieces:** Some artists release tracks with transition pieces containing sound effects or ambient sounds, which can be difficult for the algorithm to categorize as genuine outliers. Notable examples include "Catalepsy I" by Voivod and "The Audio Injection" by MNEMIC.

### 6.3 Limitations and Future Work

Overall, the findings of this study highlight the importance of considering outliers and their potential impact on the audio characteristics of different genres and artists in music production. It suggests several avenues for future research, including:

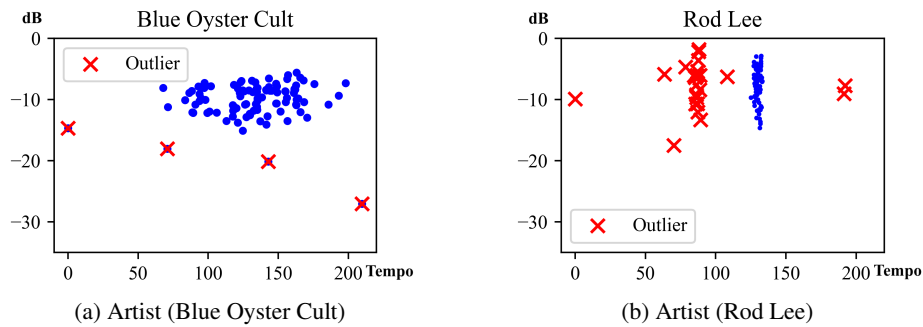


Fig. 5: Automatic Outlier Detection Using  $k$ -means: Demonstrating the algorithm works well on single-style artists (a) but fails for multi-style artists (b).

**Integration of more music features:** Expanding the range of music features integrated into the algorithm can potentially improve its performance in detecting genuine outliers. The current features, such as loudness and tempo, may not fully capture the characteristics and the style of music. Therefore, incorporating additional features like timbre, harmony, and chroma could enhance the algorithm’s effectiveness in distinguishing genuine outliers from the rest of an artist’s work.

**Consideration of “Forms A Complete Song” and “Non-Adherence” constraints:** Incorporating the “Forms A Complete Song” constraint ensures that the detected outliers are actual pieces of music. This guarantees that the outliers are indeed genuine musical outliers and not artifacts or other irrelevant audio content. The ‘Non-Adherence’ constraint ensures detected outliers distinctly deviate from an artist’s typical musical structure. This helps to identify unique songs that stand out from an artist’s typical style. Furthermore, music segmentation techniques can be considered to extract musical parts from audio pieces that may contain both speech and music. This will aid in ensuring the detected item actually is music.

**Handling artists with more than one style:** The current outlier detection approach may struggle with artists exhibiting multiple styles, such as Rod Lee (shown in fig.5). Addressing this limitation would result in a more representative understanding of an artist’s work and improve the accuracy of outlier detection. One possible solution is to consider an artist’s stylistic diversity when detecting outliers, thereby accounting for the various styles present within their body of work.

**Exploration of other clustering models:** In this study, only the  $k$ -means clustering algorithm was considered for the clustering model. However,  $k$ -means is based on circular data, which can lead to suboptimal results. Exploring other clustering models that may better handle the complexities of music data could further enhance the performance of the outlier detection algorithm and yield more accurate results.

## 7 Conclusion

In conclusion, this study has proposed a definition for genuine music outliers and explored the application of an outlier detection algorithm in music genre datasets. The

results have demonstrated that the consideration of distinctiveness is a reasonable starting point for detecting music outliers. However, the current approach lacks the ability to detect the music structure and struggles when handling artists with more than one style. To overcome these limitations, future work should focus on integrating more music features, such as timbre, harmony, and chroma, and considering the constraints of “Forms A Complete Song” and “Non-Adherence.” Furthermore, music segmentation techniques should be explored to extract musical parts from audio pieces containing both speech and music. Handling artists with multiple styles and exploring alternative clustering models, such as those that can better accommodate non-circular data, are other avenues for improvement. By addressing these limitations and incorporating these suggestions, the proposed outlier detection approach can be further refined and made more robust for detecting genuine music outliers in diverse music genre datasets.

## References

- [1] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS. *ACM SIGMOD Record* **28**(2), 49–60 (Jun 1999). <https://doi.org/10.1145/304181.304187>
- [2] Ariyaratne, H.B., Zhang, D.: A novel automatic hierarchical approach to music genre classification. In: 2012 IEEE International Conference on Multimedia and Expo Workshops. IEEE (Jul 2012). <https://doi.org/10.1109/icmew.2012.104>
- [3] Arthur, D., Vassilvitskii, S.: How slow is the k-means method? In: Proceedings of the Twenty-Second Annual Symposium on Computational Geometry. p. 144–153. SCG '06, Association for Computing Machinery, New York, NY, USA (2006). <https://doi.org/10.1145/1137856.1137880>
- [4] Azcarraga, A., Flores, F.K.: A study on self-organizing maps and k-means clustering on a music genre dataset. In: Theory and Practice of Computation. WORLD SCIENTIFIC (Oct 2017). [https://doi.org/10.1142/9789813234079\\_0017](https://doi.org/10.1142/9789813234079_0017)
- [5] Bello, J.P.: Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing* **19**(7), 2013–2025 (2011). <https://doi.org/10.1109/TASL.2011.2108287>
- [6] Bertin-Mahieux, T., Ellis, D.P.W., Whitman, B., Lamere, P.: The million song dataset. In: International Society for Music Information Retrieval Conference (2011)
- [7] Bountouridis, D., Koops, H.V., Wiering, F., Veltkamp, R.C.: Music outlier detection using multiple sequence alignment and independent ensembles. In: Amsaleg, L., Houle, M.E., Schubert, E. (eds.) *Similarity Search and Applications*. pp. 286–300. Springer International Publishing, Cham (2016)
- [8] Dai, S., Yu, H., Dannenberg, R.B.: What is missing in deep music generation? a study of repetition and structure in popular music. In: What is missing in deep music generation? A study of repetition and structure in popular music. *arXiv* (2022). <https://doi.org/10.48550/ARXIV.2209.00182>
- [9] Deng, D.: Dbscan clustering algorithm based on density. In: 2020 7th International Forum on Electrical Engineering and Automation (IFEEA). pp. 949–953 (2020). <https://doi.org/10.1109/IFEEA51475.2020.00199>
- [10] Fathollahi, M.S., Razzazi, F.: Music similarity measurement and recommendation system using convolutional neural networks. *International Journal of Multimedia Information Retrieval* **10**(1), 43–53 (Mar 2021). <https://doi.org/10.1007/s13735-021-00206-5>
- [11] Ghosal, S.S., Sarkar, I.: Novel approach to music genre classification using clustering augmented learning method (calm). In: *AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering* (2020)

- [12] Hawkins, D.M.: Identification of Outliers. Springer Netherlands (1980). <https://doi.org/10.1007/978-94-015-3994-4>
- [13] Herskind Sejr, J., Christiansen, T., Dvinge, N., Hougesen, D., Schneider-Kamp, P., Zimek, A.: Outlier detection with explanations on music streaming data: A case study with danmark music group ltd. *Applied Sciences* **11**(5) (2021). <https://doi.org/10.3390/app11052270>
- [14] Knees, P., Schedl, M.: A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications* **10**(1), 1–21 (Dec 2013). <https://doi.org/10.1145/2542205.2542206>
- [15] Lu, Y.C., Wu, C.W., Lerch, A., Lu, C.T.: Automatic outlier detection in music genre datasets. In: *International Society for Music Information Retrieval Conference* (2016)
- [16] Mehr, S.A., Singh, M., Knox, D., Ketter, D.M., Pickens-Jones, D., Atwood, S., Lucas, C., Jacoby, N., Egner, A.A., Hopkins, E.J., Howard, R.M., Hartshorne, J.K., Jennings, M.V., Simson, J., Bainbridge, C.M., Pinker, S., O'Donnell, T.J., Krasnow, M.M., Glowacki, L.: Universality and diversity in human song. *Science* **366**(6468) (Nov 2019). <https://doi.org/10.1126/science.aax0868>
- [17] Müller, M.: *Music Structure Analysis*, pp. 167–236. Springer International Publishing, Cham (2015). [https://doi.org/10.1007/978-3-319-21945-5\\_4](https://doi.org/10.1007/978-3-319-21945-5_4)
- [18] Neubarth, K., Conklin, D.: Identification and description of outliers in the densmore collection of native american music. *Applied Sciences* **9**(3) (2019). <https://doi.org/10.3390/app9030552>
- [19] Ng, R., Han, J.: Clarans: a method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering* **14**(5), 1003–1016 (2002). <https://doi.org/10.1109/TKDE.2002.1033770>
- [20] Panteli, M., Benetos, E., Dixon, S.: A computational study on outliers in world music. *PLOS ONE* **12**(12), e0189399 (Dec 2017). <https://doi.org/10.1371/journal.pone.0189399>
- [21] Paulus, J., Klapuri, A.: Music structure analysis by finding repeated parts. In: *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM (Oct 2006). <https://doi.org/10.1145/1178723.1178733>
- [22] Sargent, G., Bimbot, F., Vincent, E.: A regularity-constrained Viterbi algorithm and its application to the structural segmentation of songs. In: *International Society for Music Information Retrieval Conference (ISMIR)*. Miami, United States (Oct 2011)
- [23] Schedl, M., Gómez, E., Urbano, J.: Music information retrieval: Recent developments and applications. *Foundations and Trends® in Information Retrieval* **8**(2-3), 127–261 (2014). <https://doi.org/10.1561/15000000042>
- [24] Shingte, G., d'Aquin, M.: Unsupervised learning approach for identifying sub-genres in music scores. In: *Irish Conference on Artificial Intelligence and Cognitive Science* (2019)
- [25] Singh, R., Nakamura, E.: Dynamic cluster structure and predictive modelling of music creation style distributions. *Royal Society Open Science* **9**(11) (Nov 2022). <https://doi.org/10.1098/rsos.220516>
- [26] Smiti, A.: A critical overview of outlier detection methods. *Computer Science Review* **38**, 100306 (Nov 2020). <https://doi.org/10.1016/j.cosrev.2020.100306>
- [27] Wang, J.C., Hung, Y.N., Smith, J.B.L.: To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE (May 2022). <https://doi.org/10.1109/icassp43922.2022.9747252>
- [28] Zhang, L., Jiang, F.: Visualizing symbolic music via textualization: An empirical study on chinese traditional folk music. In: *Mobile Multimedia Communications*, pp. 647–662. Springer International Publishing (2021). [https://doi.org/10.1007/978-3-030-89814-4\\_47](https://doi.org/10.1007/978-3-030-89814-4_47)

# **Demo Papers**

## **AR-based Guitar Strumming Learning Support System that Provides Audio Feedback by Hand Tracking**

Kaito Abiki<sup>1</sup>, Saizo Aoyagi<sup>1</sup>, Akira Hattori<sup>1</sup>, Ken Honda<sup>1</sup> and Tatsunori Hirai<sup>1\*</sup>

<sup>1</sup>Komazawa University, Tokyo, Japan  
3713102k@komazawa-u.ac.jp

**Abstract.** In this study, the author developed an augmented reality (AR) system to assist beginners in learning guitar strumming. This system offers support that allow users to practice strumming anywhere using a smartphone, without the need for a physical guitar. This system utilizes hand tracking to capture the hand's coordinates and angles, effectively supporting strumming practice in the manner of a music game.

**Keywords:** Augmented reality, guitar strumming, hand tracking

### **1 Introduction**

Mastering the guitar is challenging for beginners due to the need for distinct hand techniques. The technique of strumming involves plucking the strings with the fingers and requires proper adjustment of relaxation, timing, angle, and force. According to Hosoi and Matsushita [1], skilled guitarists have been found to exhibit faster wrist rotations during strumming compared to beginners.

To practice the guitar, one needs to have a physical guitar, which can sometimes pose a limitation. This study aims to enable beginners to practice strumming even in the absence of a physical guitar. According to Fujioka [2], practicing 'air guitar' in the absence of a physical guitar is an effective means for beginners to acquire 'strumming' skills. However, the lack of feedback in air guitar poses a significant challenge when attempting to correct movements. Therefore, the proposed system conducts visual feedback in the style of a music game and analyzes hand movements to allow beginners to practice guitar strumming while enjoying the process.

Motogawa and Saito proposed a system that displays information on a display to assist in playing an actual guitar, supporting the user in playing intuitively [3]. We utilize both a vision marker and the natural features of the guitar for tracking, enabling the constant projection of support information to the appropriate position. This study focuses on fretting, but the scope of use is limited due to the use of various hardware.

---

\* This work was partially supported by JSPS KAKENHI Grant Number JP23K17023.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



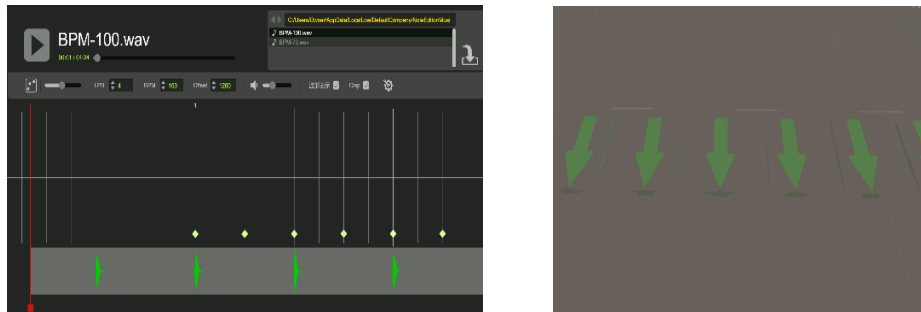


Fig. 1 NotesEditor for creating a chart (left). arrow-shaped notes (right).

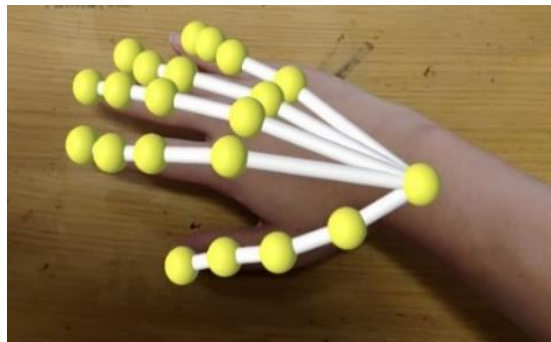
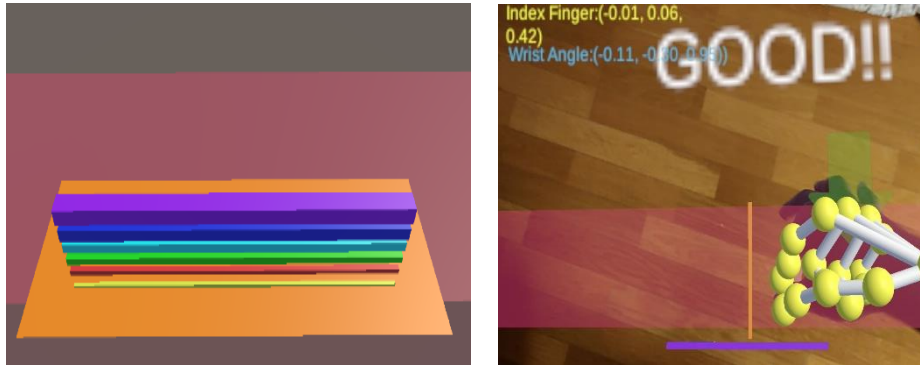


Fig. 2 The hand tracking process enabled by ToF AR.

Kashiwagi and Ochi [4] proposed a method using Kinect to detect guitar picking; however, due to the absence of real-time feedback, it is insufficient for supporting practice. This system improves right-hand strumming through AR and hand tracking, eliminating the need for a physical guitar. With the goal of providing intuitive interaction between virtual objects, musical scores, and the user's hand, we develop an AR smartphone system equipped with hand-tracking functionality for guitar strumming practice. Through hand tracking, the system displays the user's own hand movements on the UI, providing visual feedback by showing finger coordinates and wrist angle. Furthermore, its usability on smartphones renders it an excellent practice support tool for beginners.

## 2 Development of a Guitar Stroke Learning Support System

We have employed Unity and C# in the development of the proposed system, integrating ARToolKit for AR development and utilizing ToF AR for precise hand tracking. Our main challenge is to enhance the strumming techniques of the user without the physical guitar, and to address this, we have developed a game-style AR practice system.



**Fig. 3** The strings and judgement line (left), the system screen (right).

In this system, we use arrow-shaped notes created with NotesEditor. As these notes flow at a tempo of 100 BPM (beats per minute), they indicate the direction of strumming, enabling a user to move their hands in accordance and evaluate their own accuracy and timing (Fig. 1).

For hand-tracking, we utilize depth information from the ToF sensor integrated into smartphones to detect a hand within the camera's field of view (Fig. 2).

Fig. 3 shows the string object displayed on the smartphone, along with the orange judgment line used to measure the timing of strumming. As shown on the left of Fig.3, our system displays six elongated rectangular objects in different colors, representing the guitar strings. While holding the smartphone in the left hand, a user can produce the sound of an open string on an acoustic guitar by interacting with the string object using the right hand for strumming.

Fig.3 right shows the screen during the system's execution. The guitar string objects (Fig. 3 left), are displayed from a top-down perspective, similar to actual guitar strings. Upon executing the system, arrow-shaped notes flow in sync with the metronome sound set at BPM 100, and the user strums in accordance with them. When the user's hand touches the strings at the correct timing, the word "GOOD" appears.

Additionally, in the top-left corner of the screen, the user's finger coordinates and wrist angles are displayed. The wrist angle is calculated based on vector calculations using the coordinates.

### 3 Discussion

As challenges of this system, it currently has limitations in terms of the available practice tempo and stroke patterns, lacks tactile feedback, and does not possess a comparative feature for assessing hand movements against the correct reference. To address these constraints, the plan for the future is to first introduce various tempos and a range of stroke patterns for practice. For example, we plan to create multiple scores in the NotesEditor that correspond to various stroke patterns, allowing the user to freely select their preferred tempo and stroke pattern.

Currently, this system is designed to be used while holding a smartphone in one hand. In the future, we are considering the use of AR headsets to enable the use of both hands freely. Utilizing an AR headset allows for the user's non strumming hand to function as a substitute for the strings, enabling the provision of tactile feedback during strumming.

Furthermore, we plan to create 3D models that demonstrate the correct strumming motions as a reference, allowing the user to visually compare their own movements with the correct one. Additionally, in the future, we plan to implement features that provide the user with advice based on the data obtained through hand tracking.

## 4 Conclusion

In this study, we developed a guitar strumming learning support system that utilizes AR and hand tracking to provide audio feedback. We utilized Unity for the system development, supporting user's strumming practice through hand tracking technology and gamification. This gamification is achieved through interactive objects that generate guitar string sounds and arrow-shaped notes representing strumming directions. These notes appear synchronized with a metronome sound set to a specific tempo, enabling the user to practice their strumming while assessing their timing accuracy.

Through hand tracking, we visualized the user's hand movements and quantified finger coordinates and angles. Our future plans include expanding the options for tempo and strumming patterns to facilitate more versatile practice sessions. Additionally, The challenges that we plan to address and improve upon in the future include limitations in terms of tempo and stroke patterns, the absence of tactile feedback, and the lack of a feature that enables users to visually compare their hand movements. We plan to conduct evaluations of the system in the future.

## References

1. Hosoi, Y., Matsushita, S.: Evaluation of familiarity with Guitar Play using a Wearable Computer. 11<sup>th</sup> Forum on Information Technology 11 (3), pp.493-494 (2012), (in Japanese).
2. SOUND HOUSE, <https://www.soundhouse.co.jp/contents/column/index?post=2668>, last accessed 2023/9/6 (in Japanese).
3. Motokawa, Y., Saito, H.: Markerless AR Display Using Structural Feature Tracking for Supporting Guitar Play. Transactions of the Virtual Reality Society of Japan 13 (2), pp.267-277 (2008), (in Japanese).
4. Kashiwagi, Y., Ochi, Y.: Development of Guitar Performance Recognition System Using Kinect. JSiSE Research Report 32 (5), pp.115-120 (2018-1), (in Japanese).
5. Miura, S., Ando, T.: A Proposal of a Guitar Performance Learning Support System Using AR. Information Processing Society of Japan Interaction, pp.461-463 (2020), (in Japanese).

# The Demonstration of MVP Support System as an AR Realtime Pitch Feedback System

Yasumasa Yamaguchi<sup>1</sup>, Taku Kawada<sup>2</sup>, Toru Nagahama<sup>3</sup> and Tatsuya Horita<sup>3</sup>

<sup>1</sup> Sendai University

<sup>2</sup> Sendai Shirayuri Gakuen Elementary School

<sup>3</sup> Graduate School of Information Sciences, Tohoku University

ys-yamaguchi@sendai-u.ac.jp

**Abstract.** This demo paper presents and explains the system of MVP (Musical pitch Visualization Perception) support system as an AR (Augmented Reality) real-time pitch feedback system for instrumentalists who must play with correct intonation. The pitch feedback system itself uses a machine learning system called "ml5.js" and utilizes Google Glass as a feedback indicator. The system will assist not only in musical performance as a support system but also in investigating the cognitive process of intonation and musical performance as an experimental application.

**Keywords:** Realtime Pitch Feedback, ICT, Performance Support, System Development, Augmented Reality

## 1 Introduction

The proliferation of ICT (Information and Communication Technology) and the advancement of information sciences have brought about innovation in musical performance. Playing a musical instrument can be considered a perceptual-motor skill. Several studies have explored the process of perceptual-motor skill acquisition and learning. Perceptual-motor learning can be divided into three stages: the cognitive stage, associative stage, and autonomous stage [1]. On the other hand, the Acquisition of Cognitive Skill (ACT) theory consists of two stages with one section: the declarative stage, knowledge compilation, and the procedural stage [2]. In these contexts, feedback is one of the most crucial concepts in the realm of perceptual-motor learning [1].

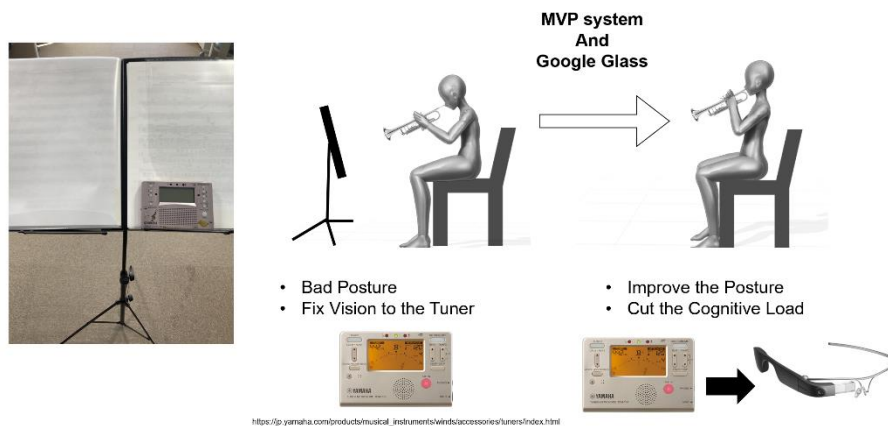
Within the context of musical performance, there has been extensive discussion about intonation. In the fields of cognitive science and musical education, Kreitman defines a listening loop for the intonation of musical performance by instrumentalists [3]. According to Kreitman, the cognitive process of instrumentalists can be classified into four sections. First, the student begins with a concept of the music in their inner ear. Second, their brain sends messages to the body to create actions. Third, these actions produce sound from the instrument. Fourth, the sound enters the ear and is sent to the



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

brain for analysis [3]. In this process, pitch feedback from a tuner greatly assists in performing with correct pitch and intonation.

Due to the benefits of using a tuner during performances, some instrumentalists tend to use a tuner placed on their music stand to receive real-time pitch feedback. While this action may contribute to good intonation, it can limit their ability to freely gaze at their music sheet, the conductor, and other musicians. Furthermore, this situation can cause a downward head posture, which is considered detrimental for instrumentalists. To address these issues, we have been developing the MVP (Musical pitch Visualization Perception) support system for smart glasses as an AR (Augmented Reality) pitch feedback system [4-5].



**Fig. 1.** The picture of the conventional tuner on the music stand and the aim of the MVP support system with a Google Glass

## 2 Development Background and System

To develop our pitch feedback system, we focused on four key aspects. Firstly, we prioritized timeliness, ensuring that the system can provide real-time pitch feedback. Secondly, we aimed for the feedback indications to be easily understandable and clear. Traditional tuners often use complex scale and needle displays, which can be overwhelming for performers during their play. Therefore, we designed our system to utilize color indicators instead. Thirdly, we incorporated recordability, a feature that sets our system apart from conventional tuners. This aspect enables the system to offer new methods and teaching materials to enhance musical education and improve performance skills. Lastly, we aimed for stability that is independent of the type of ICT equipment and environment used. Our goal is to develop a system that can work on various ICT equipment by leveraging web browsers.

The fundamental structure of our pitch feedback system is depicted in Figure 2. The system captures the sound from the participants' instrument using a condenser microphone connected to a computer. The information is then analyzed, and the pitch is

estimated using the pitch detection system. The estimated value is coloredized on a web page based on the browser's capabilities, and the Google Glass displays the coloredized webpage for the participant to view.

The computation for determining the tonal pitch is performed using the "ml5.js" library, which runs in TensorFlow [6]. This library incorporates the "Pitch Detection" package, which employs the deep-learning-based "CREPE" algorithm [7-8]. We have implemented this user-friendly system using the TensorFlow backend. In this system, the participant receives a rating based on a three-tier scale: "correct," "higher," or "lower" in comparison to the correct pitch. The feedback is sent to the participant's Smart Glass device. The display shown on the participant's Smart Glass changes interactively based on the received rating. A green display indicates a correct pitch, purple indicates a higher pitch, and blue indicates a lower pitch. This visual feedback helps the participant adjust their pitch accordingly. For our study, we defined the correct pitch range as the target, expressed in Hertz, with a tolerance of  $\pm 1\%$ . This range determines the threshold for determining whether the participant's pitch is correct, higher, or lower. We specifically utilized the "Glass Enterprise edition 2" device by Google for this study. It was selected as the platform for delivering the tonal pitch feedback. Notably, we ensured the reliability of the Google Glass system as a musical tuner by verifying its performance with a professional musician.

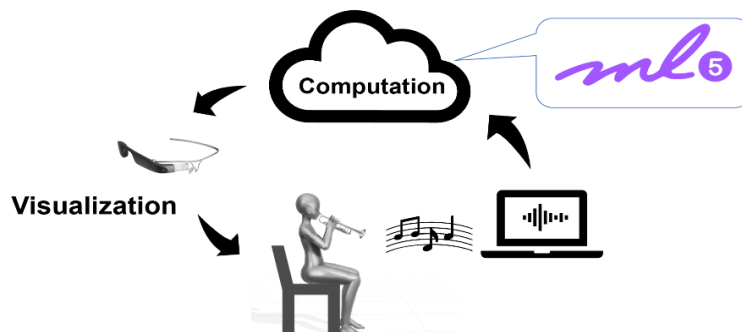


Fig. 2. The Schematic view of the MVP support system

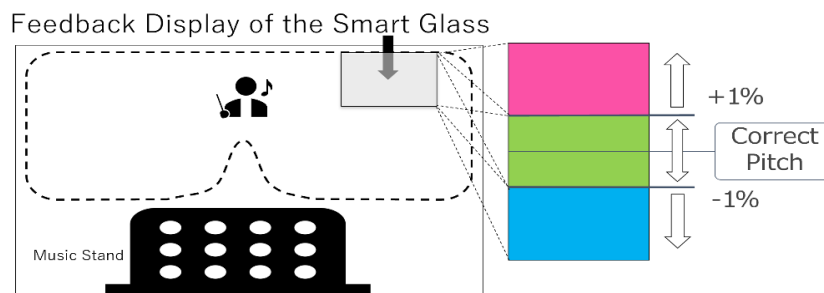


Fig. 3. The sight of user and function of MVP support system

### 3 Conclusion

#### 3.1 Future Development

In the early stages of development, we initially adopted FFT algorithms; however, the resolution rate and computation time proved insufficient for real-time pitch feedback in musical performance situations. Consequently, we decided to utilize the machine learning package provided by ml5.js, as it proves capable enough to achieve our goals. On the other hand, we encountered an issue with the smart glass. The microphone attached to the glass was not robust enough to accurately capture the tones of musical instruments. It occasionally recognized overtones of the instrument. To address this, we need to incorporate a low-pass or high-pass filter into the system.

#### 3.2 Information

The system's effectiveness was successfully presented as a pilot study [5] at the EdMedia + Innovate Learning 2022 conference in New York, held in June 2022 and further research [4] continued to display this effectiveness. In these experimental evaluations of the system, participants provided positive comments, and the system demonstrated superior performance compared to conventional tuners. These experiments highlight the potential of the browser-based system as an experimental tool for studying musical performance and pitch intonation. This demo paper is a description of the MVP support system utilizing Google Glass in [4-5] for demonstration for CMMR 2023.

### References

1. Fitts, P.M.: Perceptual-motor skill learning. In A. W. Melton (Ed.), *Categories of human learning*, 243–292. New York Academic Press (1964).
2. Anderson, J.R.: Acquisition of cognitive skill. *Psychological Review*, 89,369-406(1982).
3. Kreitman, E., *Teaching with an Open Heart*. Western Springs School of Talent Education (2010).
4. Yamaguchi, Y., Kawada, T., Nagahama, T., & Horita, T. Development and Evaluation of a Musical Instrument Performance Support System Using Smart Glasses: from the Subjective Evaluation of Form, Gaze, and Performance. *Japan journal of educational technology*, 46(Suppl.). 185-188. (2023).
5. Yamaguchi, Y., Kawada, T., Nagahama, T., & Horita, T. A Pilot Study of the MVP Support System using Google Glass. In *EdMedia+ Innovate Learning*.17-28. Association for the Advancement of Computing in Education (AACE). (2022, June).
6. NYU. ITP “*ml5js· Friendly Machine Learning for the Web.*” ml5js website. Accessed March 16, 2022. <https://ml5js.org/>, last accessed 2023/5/4.
7. Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. *Yaafe, an easy to use and efficient audio feature extraction software* In *Ismir*. (2010).
8. Kim, J. W., Salamon, J., Li, P., & Bello, J. P. *Crepe: A convolutional representation for pitch estimation*. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 161-165). IEEE. (2018).

# Melody Reduction for Beginners' Guitar Practice

Hinata Segawa, Shunsuke Sakai, and Tetsuro Kitahara\*

College of Humanities and Sciences, Nihon University  
{segawa, sakai, kitahara}@kthrlab.jp

**Abstract.** This paper describes a system that reduces a melody for novice guitar players to practice guitar solo phrases without lowering their motivation. Although there have been systems already that generate the guitar tablature for given melodies, they did not deal with melody reduction for novice guitar players. In this paper, we propose a system that generates melodies in which the difficulty of the play is reduced by using a Viterbi-like dynamic programming search. A preliminary results shows that our system can reduce melodies based on the difficulty of the play.

## 1 Introduction

Our goal is to develop a system for beginners to practice their favorite solo phrases on the electric guitar. However, it is not easy because their favorite solo phrases may be too difficult for them to play. If a system can generate melodies that are reduced but similar enough to their favorite melodies, it will be effective in maintaining their motivation.

There have been studies on generating tablatures from given melodies. For example, Hori et al. used a hidden Markov model to generate a tablature by minimizing the moving distance of fingering positions[1]. Tuohy et al. used a genetic algorithm to generate a tablature [2]. However, they did not deal with reducing melodies of the guitar solos. On the other hand, there have been many attempts of melody reduction, e.g., Ryan Groves's method using a probabilistic context-free grammar[3]. However, they do not consider the difficulty of playing melodies on the guitar.

Our system reduces melodies by introducing a playing cost and a melody modification penalty, and by finding a tradeoff between them. In particular, the playing cost becomes high when the fingering moves at upbeat, so the system generates melodies that do not move in pitch at upbeat.

## 2 Proposed System

The system inputs a monophonic tablature and generates a reduced melody tablature using a Viterbi-like search algorithm. It calculates the playing cost and melody mod-



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



ification penalty, and minimizes the sum of them to obtain an optimal sequence of fingering from the viewpoint of the ease of the play.

## 2.1 Loading MusicXML data

Given a tablature of a monophonic guitar solo melody in the MusicXML format, it is converted to a sequence of fingering positions,  $\{x_1, x_2, \dots, x_N\}$ . Here, the fingering positions for the  $n$ -th note,  $x_n$ , is defined as  $x_n = s_n + 6f_n$  where the string number  $s_n (= 0, 1, 2, \dots, 5)$  and the fret number  $f_n (= 0, 1, 2, \dots, 20)$  are combined. We also identify whether that each note is on a downbeat or upbeat.

## 2.2 Modeling Melody Reduction

The reduced melody is assumed to have the same number of notes and rhythm as the original melody, and a sequence of its fingering positions is represented by  $\{x'_1, x'_2, \dots, x'_N\}$ . The fingering position for the  $n$ -th note,  $x'_n$  is defined as  $x'_n = s'_n + 6f'_n$  using the string number  $s'_n$  and fret number  $f'_n$ .

Our system outputs a melody that is close to the original melody, but with reduced movements shift of the fingering position. For example, we set a high cost for movements of the fingering position on upbeats. This allows us to obtain the melodies such that the pitch only changes on downbeats.

**2.2.1 Playing cost** The playing cost represents the difficulty in moving the fingering position from note to note. In order to reduce pitch motions on the upbeat, we set different costs for the downbeat and upbeat. Because the use of open strings is not accepted in the current implementation, we assign a sufficiently higher cost to the movement to an open string.

(i) When  $x'_n$  is on a downbeat

We define the playing cost from  $x'_n$  to  $x'_{n+1}$  as follows.

$$C(x'_{n+1}|x'_n) = \begin{cases} \alpha & (|f'_n - f'_{n+1}| > 3) \\ \alpha & (f'_{n+1} = 0) \\ 10000 & (|s'_n - s'_{n+1}| = 3), \\ 7500 & (|s'_n - s'_{n+1}| = 2, |f'_n - f'_{n+1}| = 2, 3) \\ 5000 & (|s'_n - s'_{n+1}| = 2, |f'_n - f'_{n+1}| = 1) \\ 1000 & (|s'_n - s'_{n+1}| = 2, f'_n = f'_{n+1}) \\ 400 & (|s'_n - s'_{n+1}| = 1, |f'_n - f'_{n+1}| = 2, 3) \\ 200 & (|s'_n - s'_{n+1}| = 1, |f'_n - f'_{n+1}| = 1, 0) \\ 50 & (s'_n = s'_{n+1}, |f'_n - f'_{n+1}| = 2, 3) \\ 0 & (s'_n = s'_{n+1}, |f'_n - f'_{n+1}| = 1, 0) \end{cases}$$

Let  $\alpha$  be a sufficiently large positive value (1000000000 in the current implementation).

(ii) When  $x'_n$  is on an upbeat

We define the cost of playing from  $x'_n$  to  $x'_{n+1}$  as follows.

$$C(x'_{n+1}|x'_n) = \begin{cases} \alpha & (x'_n \neq x_n) \\ 0 & (x'_n = x_n) \end{cases}$$

As a criterion for identifying whether each note is on the downbeat or upbeat, the user can select quarter-note-level one and eighth-note-level one.

**2.2.2 Melody Modification Penalty** The melody modification penalty represents how much the output melody differs from the original melody. The melody modification penalty  $P(x'_n|x_n)$  for  $x'_n$  is defined as follows.

$$P(x'_n|x_n) = \begin{cases} 0 & (x'_n = x_n) \\ \alpha & (x'_n \neq x_n) \end{cases}$$

### 2.3 Viterbi-like dynamic programming search

The system searches  $\{x'_1, x'_2, \dots, x'_N\}$  that minimizes the following values

$$S = \left\{ \sum_{n=1}^{N-1} (P(x'_n|x_n) + C(x'_{n+1}|x'_n)) \right\} + P(x'_N|x_N)$$

This minimization can be performed using a Viterbi-like dynamic programming algorithm.

### 2.4 Tablature Output

The system finally outputs  $x'_1, x'_2, \dots, x'_N$  as a tablature in the MusicXML format.

## 3 Preliminary Results

We attempted to generate a reduced melody's tablature. Figure 1 shows the tablature used as an input (representing the original melody). This melody has a series of sixteenth notes, so requires a fast fingering movement to play. On the other hand, Figure 2 shows the generated tablature (representing the reduced melody). The fingering movements on the upbeats have been removed, and the fingering movements on the downbeats have been made smaller.

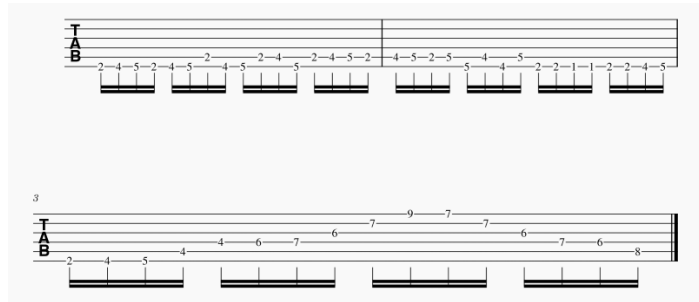


Fig. 1. A tablature used in the preliminary experiment

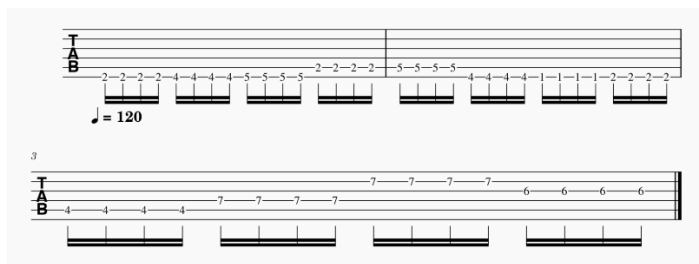


Fig. 2. A tablature of the reduced melody

## 4 Conclusion

In this paper, we developed a prototype system for generating a tablature that can be played more easily than the original melody. This system is intended to allow novice guitar players to practice solo melodies without lowering their motivations. After learning to play reduced melodies, they can try to practice the original melody.

However, there are still various issues to be addressed. First, it is necessary to evaluate the similarity of the reduced melody and the original melody to assess our melody reduction method. Because there have been many methods for melody reduction, such as GTTM-based ones, we need to compare our method with those ones.

## References

1. Gen Hori, Shigeki Sagayama: Minimax Viterbi algorithm for HMM-based guitar fingering decision, 17th International Society for Music Information Retrieval Conference,(2016).
2. D.R Tuohy and W.D. Potter: A genetic algorithm for the automatic generation of playable guitar tablature(2005).
3. Ryan Groves: Automatic Melodic Reduction Using a Supervised Probabilistic Context-free Grammar, Proceedings of the 17th ISMIR Conference (2016).

## Structural Analysis of Utterances during Guitar Instruction

Nami Iino<sup>1,2</sup>, Hiroya Miura<sup>2</sup>, Hideaki Takeda<sup>1</sup>,  
Masatoshi Hamanaka<sup>2</sup>, and Takuichi Nishimura<sup>3</sup> \*

<sup>1</sup> National Institute of Informatics

<sup>2</sup> RIKEN Center for Advanced Intelligence Project

<sup>3</sup> Japan Advanced Institute of Science and Technology  
nami-iino@nii.ac.jp

**Abstract.** The physical environments for musical instrument instruction include a mixture of various types of information such as performance sounds and speech. In our previous study, we analyzed the speech segments of audio data recorded in real one-on-one classical guitar lessons. In the current work, we annotate two types of labels for the teacher's utterance information and analyze them structurally by applying the Generative Theory of Tonal Music (GTTM) to summarize the lessons. Our findings revealed a commonality in the interpretation of utterance groupings and demonstrate that the labels for semantically categorizing the content of teachers' utterances are useful in determining the hierarchy.

### 1 Introduction

In musical instrument lessons, teachers and learners often record audio or video for later review and reflection. However, these private recordings are rarely made publicly available as a research resource. Additionally, the specific terms and instructional content utilized in real lessons have not been analyzed in detail. Therefore, we previously collected sound information from one-on-one classical guitar lessons to help clarify the features of the music teaching-learning process [1].

In the current study, we structurally analyze the utterances that occur during a guitar lesson with the aim of summarizing the overall lesson. Specifically, We annotated the teacher's utterances data with two types of labels and generated tree structures using an analysis method based on the Generative Theory of Tonal Music (GTTM) [2]. Through a comparison of the tree structures by two analysts, we examined the common structural patterns and rules for the aggregation of tree structures. Our findings revealed a commonality in the analysts' interpretations and showed that labels related to instructional content are useful in determining hierarchy.

---

\* This study was partially supported by Kayamori Foundation of Informational Science Advancement.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

A research [3] proposed an interactive information structuring method for meeting minutes utilizing the GTTM. In that study, to explicitly express the intentions contained in discussions, a tree structure and a method for its extraction were implemented to represent hierarchical importance levels based on verbal and non-verbal information. Our study aims to find a method suitable for lessons utilizing this same approach. We also take advantage of CROCUS, a publically available dataset containing performances and their written critiques [4]. While this dataset does not provide real-time performance instruction, it is applicable to our study as it focuses on the instructor's "words."

## 2 Data Preparation

**Features of Audio Segment of Classical Guitar Lessons** In our previous study, we collected audio data from a pair (teacher and student) of one-on-one guitar lessons and categorized them into groups based on musical pieces for which the lessons had been given at least three times [1]. We then manually transcribed the utterances for each teacher and student and segmented them on the basis of the time interval between utterances.

Our analysis of the segmented data showed that (i) the percentage of students' utterances tended to decrease as the number of lessons progressed, and (ii) the percentage of teachers' utterances did not change much. These findings suggest that a more detailed analysis focusing on teachers' utterances might make it possible to extract the main points of the lesson. In addition, an integrated analysis of audio segments and the content of utterances is necessary to achieve a comprehensive summary of the lesson.

**Annotation of Types of Teacher's Utterance** In this study, we focused on the teacher's utterances and annotated two types of labels representing instructional information with the transcribed data. Through these annotations, it is possible to clarify the type of instructions given at specific times during a musical lesson.

– **Instructional Topic Labels:** The Music Teacher's Ontology [5] is a knowledge system for music education that represents a hierarchy of topics on which teachers might provide feedback to students. We referred to this ontology to define instructional topic labels. First, we defined four upper-level labels: *Musical piece*, regarding the musical style of the piece (period, musical form, etc.), *Musical expression*, regarding the intentional use of expression by the teacher or learner, *Technique*, regarding physical technique, and *Other*, regarding performance, mental aspects and so on. Then, we defined 18 specific topics as lower-level labels, such as *Tempo*, *Rhythm*, *Fingering*, and *Articulation*. We annotated one upper-level label and two or fewer lower-level labels for each unit of utterance.

– **Instructional Content Labels:** SOAP is a classification framework that takes into account the semantic elements of sentences for natural language [6]. It was originally designed for scientifically describing the thought processes of doctors in medical records using natural sentences. In this study, by referring to research [4], we adapted the classification categories of SOAP to the field of music as follows:

- Subjective data (*S*): teacher providing general and/or specific conceptual information based on subjectivity.

- Objective data (*O*): teacher providing general and/or specific conceptual information based on objectively referable events or concepts.
- Assessment (*A*): teacher’s evaluation of a student’s applied and/or conceptual knowledge.
- Plan (*P*): giving a specific opinion or recommendation to guide the student’s action towards the achievement of a specific musical aims.

We pointed that the four items can characterize the instructional content and annotated them to the teacher’s utterances. If an utterance had more than one label, we provided two or more annotations.

### 3 Structural Analysis of Teacher’s Utterances

**Generation of Tree Structures using the GTTM** The Generative Theory of Tonal Music (GTTM) is a musical framework that can generate a tree structure with a hierarchical temporal organization and a principal-subordinate relationship of branches. A key feature of GTTM is that it enables somewhat subjective musical analysis because it allows for different interpretations based on the analyst’s judgment while still adhering to the basic rules. It also enables “reduction,” which extracts abstracted groups from the upper layers of the tree structure. These mean that GTTM can be utilized to reveal the analyst’s perspective and identify structuring rules for summarizing the lesson.

In this study, two researchers independently generated a tree structure. First, we took one of one lesson data described in previous section and divided the teacher’s utterances into five sections based on the segments and the content of the utterances. Next, we applied the time-span analysis method defined in the GTTM for grouping and hierarchization.

**Comparison of Tree Structures and Discussion** Figure 1 shows an example of utterances, annotated labels, and two generated tree structures, where structures A and B correspond to the results of analysts A and B. As we can see, the groupings of the two tree structures tended to be similar. Specifically, the branches were divided into three groups, suggesting a commonality in the analysts’ perceptions of speech cohesion. On the other hand, there were differences in the hierarchy within the groups. This was due to the fact that (i) analyst A was conscious of summarization and placed *S* in the upper-level labels of the instructional content labels, representing the problems of the student’s performance, whereas (ii) analyst B focused on the number of low-level labels and the latter half of the utterance.

The breakdown of instructional content labels in the five sections was *S*: 16.7%, *O*: 18.6%, *A*: 2.1%, *P*: 37.5%, *S&A*: 2.1%, and *None*: 22.9%. The fact that about 23% of the utterances could not be annotated (*None*) indicates that semantically significant utterances were limited. Examples of such unannotated utterances include rhythmic counting and responses using short words phrases as “I see.” Furthermore, *P* tended to be located at a higher level of hierarchy in both structures. These results indicate that instructional content labels are useful for identifying hierarchies.

We need to collect additional tree structure data because the samples in this study were very small. Additionally, we need to define aggregation rules for the tree structures by applying GTTM: specifically, (1) Grouping Preference Rules, which group

Speaker	Utterances	Instructional topic label			Instructional content label
		Upper-level	Low-level	Low-level	
Teacher	Yes, about this much. One-two-three.	<i>Musical expression</i>	<i>Tempo</i>		
Teacher	Well, you tend to rush with the portamento.	<i>Musical expression</i>	<i>Tempo</i>	<i>Rendition</i>	<i>S</i>
Teacher	Yes.	<i>Musical expression</i>	<i>Tempo</i>	<i>Rendition</i>	
Teacher	Please keep the sound properly to the note value. This too.	<i>Musical expression</i>	<i>Rendition</i>	<i>Note value</i>	<i>P</i>
Student	Oh, I see.				
Teacher	Also, it is fast after "ti."	<i>Musical expression</i>	<i>Tempo</i>		<i>S</i>
Teacher	Keep the note value tight.	<i>Musical expression</i>	<i>Note value</i>		<i>P</i>
Teacher	Yes, I think that would make it in time. You are rushing too much about this.	<i>Musical expression</i>	<i>Tempo</i>		<i>S · A</i>
Student	Yes, around here, I'm in a hurry, so my performance is getting squished, here.				
Teacher	Flop.	<i>Musical expression</i>	<i>Tempo</i>		
Teacher	Yes, play each note properly.	<i>Musical expression</i>	<i>Tempo</i>	<i>Note value</i>	<i>P</i>
Student	I see. My performance is getting more and more messed up here.				
Teacher	Yeah, don't rush. As you get better at playing it, the speed gets faster. So, you have to be careful.	<i>Other</i>	<i>Tempo</i>		<i>A</i>

Fig. 1. Example of utterance data and tree structures.

utterances based on measures such as utterance duration, interval, and the number of utterances, and (2) Significance Preference Rules, which identify important utterances based on key words and label information.

#### 4 Conclusion

In this paper, we performed a structural analysis of teacher's utterances to summarize the content of musical instrument lessons. First, we annotated the transcribed data of a classical guitar one-on-one lesson with two types of semantic labels. Then, we hierarchically structured the utterances based on the GTTM. As a result, we were able to clarify the analyst's perspective and examine the issues with the structuring rules. In future work, we will expand the data and conduct a more in-depth analysis of individual data. This will enable us to apply the findings to support student reflection and improve the quality of teaching.

#### References

1. Miura, H., Iino, N., Hamanaka, M., Takeda, H., and Nishimura, T.: An Attempt on Modeling of Teaching Knowledge in Musical Instrument Performance Situations, JSIAI2021 (2021).
2. Lerdahl, F. and Jackendoff, R.: A Generative Theory of Tonal Music, MIT Press (1983).
3. Miura, H., Takegawa, Y., Terai, A., and Hirata, K.: Discussion Summarization Based on Hierarchical Structure Using Verbal and Non-Verbal Information, in Proceedings of the International Conference on Internet and Multimedia Technologies 2018, World Congress on Engineering and Computer Science, pp. 308–313 (2018).
4. Matsubara, M., Kagawa, R., Hirano, T., and Tsuji, I.: CROCUS: Dataset of Musical Performance Critiques Relationship between Critique Content and Its Utility, Proc. of the 15th International Symposium on CMMR, pp. 279–288 (2021).
5. Yee-King, M. J., Wilmering, T., Llano, M. T., Krivenski, M., and d'Inverno, M.: Technology Enhanced Learning: The Role of Ontologies for Feedback in Music Performance, Frontiers in Digital Humanities, 5(29) (2019).
6. Weed, L. L.: Medical records, medical education, and patient care: The problem-oriented record as a basic tool, Press of Case Western Reserve University (1969).

## Music in the Air: Creating Music from Practically Inaudible Ambient Sound

Ji Won Yoon<sup>1</sup> and Woon Seung Yeo<sup>2</sup>

<sup>1</sup> Applied Music & Sound Major, College of Music and Performing Arts,  
Keimyung University, Daegu 42601, Republic of Korea

<sup>2</sup> Department of Content Convergence, Ewha Womans University,  
Seoul 03760, Republic of Korea  
woony@ewha.ac.kr

**Abstract.** *Music in the Air* is a pioneering system that generates real-time music from the "theoretically audible but practically inaudible range" (TAPIR) ambient sound, opening new possibilities in music composition. The system captures and analyzes the ambient sound and utilizes its TAPIR portion to generate musical notes in a MIDI format. It has proved to function successfully at an audiovisual art exhibition, showing the potential to blur the line between natural and artistic by unveiling hidden melodies within "silence" for inspiring compositions. In addition to its musical significance, this paper introduces the system focusing on the mapping strategy for MIDI note generation. The demo at the conference will showcase its initial implementation and a newer version with more advanced features supporting versatile musical mappings.

**Keywords:** Real-time music generation, ambient sound, spectral analysis, TAPIR (Theoretically Audible but Practically Inaudible Range), MIDI, musical composition, musical sonification

### 1 Introduction

Throughout history, humanity has continuously sought inspiration from the world around us to create music. While natural soundscapes have often served as a source of creative influence, the concept of generating a new piece of music from hardly audible portions of the ambient sound is relatively unexplored.

In this context, we introduce *Music in the Air*, a system for real-time music generation from the almost inaudible high-frequency part of the ambient sound, thereby unveiling the melodies hidden within "silence." The system provided the music (or a monophonic melody) for an audiovisual art exhibition with the same title held at *Gallery Insa Art*, Seoul, Republic of Korea, in May 2023, which was presented by the authors (see Fig. 1).



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).





Fig. 1. Photo of *Music in the Air* exhibition held at *Gallery Insa Art* in May 2023.

The motivation behind creating new music from the hardly audible part of the ambient sound of a space lies in the desire to push the boundaries of music composition and foster a deeper appreciation for the intricacies of sound; by extracting, analyzing, and transforming the inaudible ambient sounds, we aim to unlock a new realm of musical possibilities, allowing composers to explore uncharted auditory landscapes. Not only can this be understood as an effort to extend the range of human audibility, but also a unique and innovative method to generate original sounds or compose new music based on something that is not audible.

This paper introduces a brief technical background to the practically inaudible audio, describes not only the overall structure of the system but also the technical details of the implementation and the demo scenarios, and discusses its musical significance and possible future enhancements.

## 2 System and Methodology

Fig. 2 illustrates the overall setup and the signal flow of the system. Here, only the TAPIR components of the captured ambient sound get analyzed in the frequency domain to provide musical information.



Fig. 2. System Overview.

## 2.1 (In)audible Range

Human audible frequency range spans from 20 Hz to 20 kHz, theoretically. However, in reality, very few individuals possess the ability to hear sounds at the extreme ends of this. Several factors, including age, genetics, and the environment (e.g., exposure to loud noises over time), can affect an individual's hearing sensitivity and narrow the practical audible range. In practice, the upper limit of the audible frequency range at average volumes is around 18 kHz for most adults – including those in their 20s, which is lower than the theoretical value. As a result, we may call this marginal bandwidth in the upper end of human audible frequencies as the "theoretically audible, but practically inaudible range" (TAPIR), as suggested in [1].

## 2.2 Audio Capture and Spectral Analysis

Notably, with a sampling rate of 44.1 kHz or above and typical acoustic transducers (i.e., ordinary microphones without ultrasound features) covering a frequency response range up to above 20 kHz, most computers and smartphones can capture and analyze sounds in this TAPIR range with little problem. Understanding the existence of these sounds, which we can hardly hear but that machines easily can, is crucial for designing audio experiences and related technologies for *Music in the Air*. For the exhibition, we used an SM58 by Shure [2], one of the most common microphones.

The spectrum of the captured audio signal is then analyzed in real time using the fast Fourier transform (FFT) function of the `minim` library [3] in the Processing programming environment [4]. Depending on the size of the FFT, the "resolution" (or the interval between adjacent frequencies) and the number of spectral components (or frequency bins) of the resulting spectrum may vary. For example, with the FFT size of 1024 and the sampling rate of 44.1 kHz, we get about 94 bins within the range from 18 to 22.05 kHz at the interval of 43.07 Hz between the adjacent frequency bins of FFT.

## 2.3 MIDI Mappings

For the actual generation of music/sound (i.e., something we can hear), spectral analysis results must get utilized to determine the elements of musical notes or parameters for sound synthesis. In the exhibition, real-time spectrum information was "mapped" to generate musical notes in a MIDI message format, i.e., pitch by note number, duration by Note On and Note Off (or onset and release, respectively), and the type of the instrument by MIDI channel (or other program messages, if necessary), as briefly described below:

**Pitch.** To determine the pitch of the musical notes to generate, we should establish a rule that maps selected frequency bins from the FFT result to musical notes, which has the most significant impact on the overall impression of the result. Depending on the type of musical characteristics in mind, the mapping rule can change differently, involving questions on musical texture (i.e., monophonic vs. polyphonic) and scale (i.e., limited choice of pitch).

**Onset and Release.** In the case of real-time generation, the duration of a MIDI note can only be determined as the interval between the reception of Note On and Note Off messages. For this, the loudness level of each frequency bin of interest needs to be monitored continuously to detect the moment of onset and release. Note that, in addition to adjusting the threshold levels for onset and release detection, we may need to check/control the gain level of the microphone input signal.

**Instrument.** Generated note information is sent to any instruments, either hardware synthesizers or virtual instruments, that support MIDI connection. Multiple instruments can be connected simultaneously, each playing a different melody.

### 3 Demo

*Music in the Air* is a work in progress and continues to evolve. As such, the demo will feature not only the original system used in the exhibition but also a newer version that provides a graphic user interface (GUI) that allows the user to choose the desired musical mapping scheme from various options. In addition, we will share our experience obtained through the design and development process of the system in detail, primarily focusing on the MIDI mapping rules.

As for the musical instrument, the demo will showcase several software synthesizers with different timbre and tonal characteristics, including those provided on macOS via *Logic Pro*, providing a chance to experience the effect of timbral change.

### 4 Conclusion

*Music in the Air* demonstrates the ability to generate musical information from the inaudible portion of ambient sounds, suggesting a new approach in musical composition that blurs the lines between the natural and the artistic. By expanding the boundaries of traditional musical sources and unveiling the hidden melodies within silence, composers can draw inspiration from the subtle and often overlooked soundscapes and gain a fresh perspective on the infinite possibilities within our sonic environment.

In addition to performing tests in various environments to develop a versatile but robust methodology and mapping schemes, future work will include the integration of the system into interactive and immersive musical experiences with novel and engaging performances in mind.

### References

1. Yeo, W., Kim, K., Kim, S., Lee, J.: TAPIR Sound as a New Medium for Music. *Leonardo Music Journal* 22, 49–51 (2012). doi: [https://doi.org/10.1162/LMJ\\_a\\_00091](https://doi.org/10.1162/LMJ_a_00091)
2. Shure SM58, <https://shorturl.at/oOSZ4>, last accessed 2023/7/31.
3. Minim, <https://code.compartmental.net/minim/>, last accessed 2023/7/31.
4. Welcome to Processing!, <https://processing.org/>, last accessed 2023/7/31.

## **Creating an interactive and accessible remote performance system with the Piano Machine**

Patricia Alessandrini, Constantin Basica, Prateek Verma

Center for Computer Research in Music and Acoustics – CCRMA, Stanford University

patricia@ccrma.stanford.edu, cobasica@ccrma.stanford.edu,  
prateekv@stanford.edu

**Abstract.** This demo shows a work-in-progress development of an intelligent interactive system for the Piano Machine, a physical computing system that causes the strings of the piano to sound through mechanical excitation. While the first version of the Piano Machine, developed in 2017 by Patricia Alessandrini and Konstantin Leonenko, employed a simple midi-keyboard control using Aftertouch for continuous control, the Piano Machine was further developed in 2019-2020 with Machine Learning to allow for higher-level control, allowing the Piano Machine to respond interactively to inputs such as live sound and gesture. The current development will integrate the Piano Machine into AI-driven co-creative systems, such that performers/improvisors can use a variety of inputs in-person or remotely. By expanding inputs – ranging from text to gesture, tapping or humming – and providing remote access through a browser-based environment, this system will increase access to musical experiences with the Piano Machine, including for Disabled and/or non-expert music-makers.

**Keywords:** #Accessibility #Network Performance #Artificial Intelligence

### **1 Employing AI for inclusive interaction with the Piano Machine in remote or in-person music-making**

This project endeavors to bring greater accessibility to music-making with the Piano Machine for both Disabled and/or non-expert music-makers by creating an expressive, dynamic and responsive generative music system that allows a performer to dialogue and interact in real time with the Piano Machine. The Piano Machine - a “robotic” physical computing device designed and created by Patricia Alessandrini in collaboration with Konstantin Leonenko in 2017 - plays the strings of the piano directly through mechanical, sustained vibration created by a set of motors and finger-like appendages controlled by microprocessors, thus creating dynamic control of notes over time [1]. Control data for



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

the Piano Machine will be generated in real time using AI models to interpret a range of different inputs – including text, gesture, tapping and humming – to create an environment for accessible creative music-making, including composition and improvisation.

The use of accessible inputs is based on research performed by Prateek Verma, Constantin Basica, and Patricia Alessandrini principally over the past year [2], with financial and institutional support from Stanford Human-centered Artificial Intelligence (HAI), the Stanford Humanities Changing Human Experience Grant project *Considering Disability in Online Cultural Experiences*; the EU Horizon project Multisensory, User-centred, Shared cultural Experiences through Interactive Technologies (MuseIT); the Center for Computer Research in Music and Acoustics (CCRMA); and the Institute for Research and Coordination in Acoustics/Music (IRCAM), particularly through its European Research Council (ERC) Project REACH: Raising Co-creativity in Cyber-Human Musicianship.

## **2 Piano Machine developments to date**

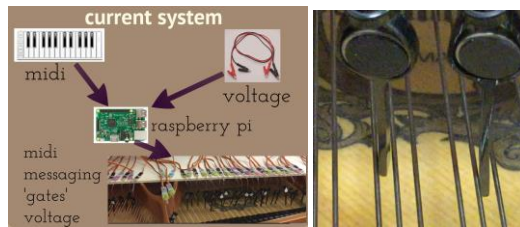
### **2.1 A MIDI-based instrument for direct control in performance**

The first Piano Machine was built in the Hatch Lab at Goldsmiths, University of London, with funding from the Arts Council of England (ACE). It was commissioned by Explore Ensemble for the première of *Tracer la lune d'un doigt* [3] at the Huddersfield Contemporary Music Festival (HCMF) in 2017 as well as for other new repertoire [4].

To facilitate composition, notation and performance in varying compositional contexts, the first Piano Machine was controlled by a MIDI keyboard, with Aftertouch used to independently control the post-attack intensity of each note. As the midi-messaging is handled by a micro-computer, no computer is required, only a (USB) MIDI keyboard.

### **2.1 Voltage handling and playing mechanism**

As illustrated in Figure 1 below, the original Piano Machine uses an external voltage source to power individual motors with added appendages – with one motor per piano note - while the voltage sent to each individual motor is determined by the MIDI messages received by the Raspberry Pi, including polyphonic pitch, attack and Aftertouch.



**Fig. 1.** Voltage and control data flow in the original Piano Machine, with a close-up of the motors and appendages forming the Piano Machine playing mechanism.

The playing mechanism consists of small, cell phone “pager” vibration motors equipped with laser-cut appendages suspended between two piano strings of the same note (or just next to low notes with a single string), which only touch the strings when voltage is sent to them (Figure 1). It is thus possible to play the piano using standard techniques without any interference while the Piano Machine is installed.

## 2.2 Higher-level control of the Piano Machine

The 2019 version of the Piano Machine was created for *Ada’s Song*, in which it is controlled in realtime using AI processes [5]. It has up to 96 notes (instead of the original 64), wireless OSC messaging, improved stability of the appendages leading to better tone quality, finer dynamic control, lighter weight, better adaptability to different pianos, improved stability, better cushioning and portability, as well as gestural control.



**Fig. 2.** The new Piano Machine, pictured in rehearsal at The Warehouse, London, October 2019

## 3 Developing accessible inputs for the Piano Machine

### 3.1 Integrating text inputs into the Piano Machine

Building on Basica et al’s previous work using AI models to respond generatively – both online and in-person – to text inputs using language

models trained with tagged musical data sets such as MTG-Jamendo [2] [6], in the course of 2022-23 these generative music systems have been further developed in collaboration with IRCAM, to further elaborate on the melodies generated by the models with the Somax2 system, which can be stylistically trained using music provided or chosen by a participant. While much of this work to date has been workshopped and performed using a Disklavier, the Piano Machine's capacity for continuous control will allow for greater expressivity.

### **3.2 Humming, tapping and gesture for music-making**

In 2022-2023, collaborative co-design workshops organized as part of the Considering Disability in Online Cultural Experiences project have explored a range of inputs to be interpreted into musical outputs by AI models, particularly for potential music-makers who may have limited language use and/or prefer humming, tapping or gesture. Humming and tapping can both be translated by AI models into music, and sonically integrated into improvisations [2]. Integrating humming and tapping, along with the existing gestural control, will thus offer more options for defining musical materials for non-expert music-makers.

## **4 Remote music-making with the Piano Machine**

The goal is to allow the above inputs to be transmitted remotely to the Piano Machine, which will then be live-streamed to the participant(s) using the low-latency, uncompressed audio environment JackTrip (7).

### **References**

1. See project page <https://patriciaalessandrini.net/piano-machine>, last accessed 2023/07/30.
2. Verma, P., Basica, C., Alessandrini, P., Berceanu, A.: Accessible Co-Creativity through Language and Voice Input. Pre-print, Proceedings of AIMC 2023: The International Conference on AI and Musical Creativity. Sussex, Brighton, UK. (2023).
3. <https://patriciaalessandrini.net/tracer-la-lune-dun-doigt>, last accessed 2023/07/30.
4. See for instance Leeming, Z., *At the Node of Ranvier*, (2019) <https://soundcloud.com/user-416324597/at-the-node-of-ranvier>, last accessed 2023/07/30.
5. See project page, <https://patriciaalessandrini.net/adas-song>, last accessed 2023/07/30.
6. Morgenstern, M., Basica, C. et al. Lost Interferences, Ars Electronica. (2021). <https://ars.electronica.art/newdigitaldeal/en/lost-interferences-event/>, last accessed 2023/07/30.
7. Cáceres, J-P., Chafe, C. JackTrip: Under the Hood of an Engine for Network Audio. Proceedings of the International Computer Music Conference, Montréal, Canada (2009)

## **A Singing Toolkit: Gestural Control of Voice Synthesis, Voice Samples and Live Voice.**

D. H. Molina Villota, C. D'Alessandro, G. Locqueville, and T. Lucas \*

Institut Jean Le Rond d'Alembert  
Equipe Lutheries-Acoustique-Musique  
Sorbonne Université - Centre National de la Recherche Scientifique  
Paris, France  
daniel.molina.villota@sorbonne-universite.fr

**Abstract.** The Singing Toolkit demo presents three approaches to real-time gestural control of voice : control of vocal synthesis using the Cantor Digitalis instruments; syllabic re-sequencing and modification of pre-recorded vocal tracks with the Voks instrument; control of real-time vocal performances, using DAFx and inertial devices. These three approaches exemplify the potential of gesture-based control to enhance vocal performances, expand the creative possibilities in vocal music production, and open up new avenues for expressive control and artistic exploration.

**Keywords:** Gestural Control of Voice, IMU, Theremin, Voks, Chironomic, Gesture, Cantor Digitalis

### **1 Introduction**

The Singing Toolkit demonstrates our recent work in three directions for real-time gesture control and modification of voice signals. The first instrument, Cantor Digitalis, is a formant synthesizer using bimanual (chironomic) gestures for melodic and formantic control with the help of graphic tablet. The second instrument, Voks, allows for syllabic resequencing using tapping gestures and chironomic control of intonation and voice quality. The third approach is real-time voice transformation through gesture-controlled vocal effects using the IMU RiOT-Bitalino inertial measurement units (an Ircam and Bitalino joint project).

---

\* This Research is funded by ANR National Research Agency: Analysis and Transformation of Singing Style ANR-19-CE38-0001 & Gepeto: GESture and PEdagogy of inTONation ANR-19-CE28-0018



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



## 2 Cantor Digitalis : Chironomic control of synthesized voice

Cantor Digitalis <sup>1</sup> is a vowel and semi-vowel singing instrument controlled by chironomic gestures [2]. It translates manual gestures into formant synthesis parameters based on the linear model speech production [1], allowing musicians to control the pitch, vocal effort, and vowel of a synthetic voice in real time. The primary gesture interface used for controlling Cantor Digitalis is the Wacom graphic tablet. Writing or drawing gestures by the preferred hand are controlling pitch and vocal effort, while the other hand control the vowel space using a 2D (2 formants) surface, as shown in Figure 1a.

The pen's low latency (5 ms) makes sound produced by Cantor Digitalis seem to exhibit a direct causality similar to that of acoustic instruments. A visual cue is also printed on the tablet to enhance usability. The graphic tablet has proven effective for controlling voice intonation and singing with Cantor Digitalis. Cantor Digitalis can also be controlled with other continuous interfaces, e.g. the Roli Seaboard RISE Multi-dimensional Polyphonic Expression interface (MPE) [6]. In this case, pitch is controlled using a chromatic keyboard, and vocal effort is controlled by pressure on the touch surface. MPE allows for continuous transitions between notes and pressure levels. Cantor Digitalis [7] [3] won the first prize in the Margaret Guthman Musical Instrument Competition (2015). Cantor Digitalis is limited to vowels or vocalic sounds, to the exclusion of most consonants.

## 3 Voks: Syllabic sequencing of a prerecorded voice

The Voks singing instrument [4] makes it possible to control any voice utterance, including consonants. As it appeared impossible to control each individual articulatory parameter in real time, the syllable is chosen as rhythmic control unit. In practice, the user first loads a sample recording of the desired text being uttered, together with a syllabic annotation of said recording. The loaded sample needs not have any particular rhythm or melody. Then, during the performance, the system resequences the loaded sample, with a rhythm, pitch and vocal quality controlled in real time by the performer's manual gestures.

*Syllabic sequencing:* Syllabic rhythm control is performed using a cyclic tapping gesture. Several interfaces can capture such gesture data, including buttons, keys, pads, and pressure sensors. Upon tapping/pressing or releasing one's finger on the interface, a one-time signal is sent to the system, triggering advancement of a virtual playhead to the next frame timestamp.

*Other gestures:* In addition to rhythm sequencing, other parameters are to be controlled by the performer: pitch, vocal effort, vocal tract stretching factor. Some of those parameters are common to Cantor Digitalis, although they are not implemented in the same way — in Cantor, synthesis parameters are controlled directly, whereas in Voks, a prerecorded sample is modified in real time based on control values.

Following Cantor Digitalis, the graphic tablet and MPE interfaces are used to control pitch and vocal effort in Voks. In addition, the theremin has been used as a control

<sup>1</sup> <https://github.com/CantorDigitalis>

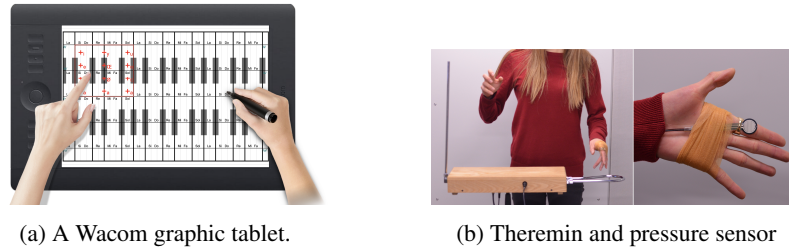


Fig. 1: Two interfaces that can be used for gestural control of vocal synthesis. (a) The Wacom tablet has been used with Cantor Digitalis (pen and finger) and Voks (pen only) (b) The theremin and pressure sensor have been used to control Voks.

interface, with one antenna controlling pitch and the other controlling vocal effort, and an added pressure sensor placed in between the thumb and index of the performer for rhythm control. T-Voks (i.e. Voks played by a Theremin and a rhythm control button) won second place in the 2022 Guthman musical instrument competition.

#### 4 Gesture Control of Digital Audio Effects with IMU

The third tool in the Singing Workshop is interactive real-time gestural control of digital audio effects (DAFx) for voice. The the BITalino R-IoT (abbreviated as R-IoT)[5] is chosen because of its lightness and powerfullness. It is a 9-axis digital IMU sensor (LSM9DS1) that provides absolute orientation in space with low latency over the OSC protocol. The data flow follows the structure indicated in Figure 2. First, R-IoT data is carried to the computer by a router through wifi. Then, data from R-IoT (orientation, quaternions, and acceleration) is received in MAX using the dedicated Bitalino object and Mubu package (by IRCAM). For each DAFx, a selection of parameters, mapping, limit conditions, and appropriate scaling must be made. The data is then sent from Max to the TouchOSC object in Ableton Live using the OSC protocol. There, another mapping is performed to assign those OSC values to different controls in the effects used.

Now we will describe briefly some effects that have been implemented. We have mapped hand rotation to panning: visually, the performer can make an opening gesture, which allows capturing an appropriate range of orientation values for the axis of rotation. Body limitations help define the scaling limits in MAX so that the movement adequately covers the maximum, minimum, and center of stereo panning. Figure 3 a) illustrates this gesture simply. The second effect is an overdrive effect. Within the specific musical piece for which it has been developed, this effect involves distortion applied to all vocal tracks, which gradually increases towards the end of the song. The backward movement of the hand, as shown in Figure 3 b), relates to the incremental distortion by tilting the arm. Finally, another performer triggers a delay effect momentarily using the same gesture. In this case, the sudden movement launches the delay effect based on the speed of the motion, making the control of the delay much more efficient than with a traditional knob. This movement can be seen in Figure 3 c).

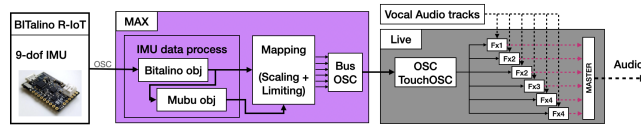


Fig. 2: Flow diagram for Interactive Vocal DAFx with R-devices using MAX and Ableton LIVE.

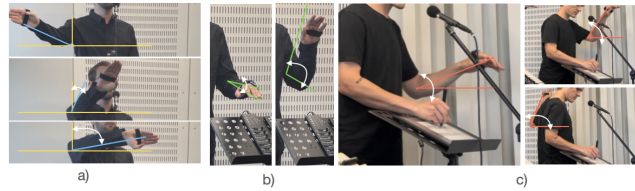


Fig. 3: Schema for the configuration of a) Panning, b)distorsion, c) delay using the R-IoT devices.

## 5 The Demo

The Singing Workshop the demo consists of a room with the three devices set up, each with its corresponding interfaces and computers. Additionally, there will a poster and three assessors who will explain how the three devices work using musical pieces as examples, within there are also included some tracks of the Chorus Digitalis project, including Cantor Digitalis, Voks and real voices.

## References

1. L. Feugère and C. d'Alessandro. Contrôle gestuel de la synthèse vocale. les instruments cantor digitalis et digitartic. *Traitement du Signal*, 32:417–442, 12 2015.
2. L. Feugère, C. d'Alessandro, B. Doval, and O. Perrotin. Cantor digitalis: chironomic parametric synthesis of singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2017.
3. L. Feugère, S. Le Beux, and C. d'Alessandro. Chorus digitalis: polyphonic gestural singing. 03 2011.
4. G. Locqueville, C. d'Alessandro, S. Delalez, B. Doval, and X. Xiao. Voks: Digital instruments for chironomic control of voice samples. *Speech Communication*, 125:97–113, 2020.
5. Plux Wireless Biosignals and IRCAM. *User Manual R-IoT Bitalino*. <https://www.bitalino.com/storage/uploads/media/manual-riot-v12.pdf>.
6. Roli. Zimphony: The seaboard rise with cantor digitalis, 2016. <https://youtu.be/mC4pmokMwRo>.
7. S. S.Le Beux, L. Feugère, and C. d'Alessandro. Chorus digitalis: Experiments in chironomic choir singing. pages 2005–2008, 08 2011.

# Sonifying Players' Positional Relation in Football

Masaki Okuta and Tetsuro Kitahara\*

College of Humanities and Sciences, Nihon University, Japan  
{okuta, kitahara}@kthrlab.jp

**Abstract.** This paper presents a prototype system that visually and aurally represents information on the positions and movements of players in a football game, with the aim of facilitating understanding of football. Understanding the positional relationship of players is important in analyzing the situation of a game. Although visualization has been used for this purpose, there are no examples of audible representation. In this paper, we use Delaunay triangulation to find the pass courses between the players, and then sonify the pass courses with sound. By extending this trial, we expect to be able to understand the match situation more effectively than with visualization alone.

**Keywords:** Football Sonification visualization

## 1 Introduction

In football, eleven players work together and yet move differently to move the ball to the goal. In order to understand the matchup, it is necessary to understand the movements of the players who do not have the ball. However, it is not easy to keep track of the standing positions and movements of all eleven players at the same time.

There have already been studies on tracking the positional relationship of football players [1][2]. On the other hand, sonification is used in other fields to facilitate to understand complex scenes [3], but there have not been attempts for analyzing footnote scenes.

In this paper, we attempt to visualize and sonify the positional relationships of players in a football game. In football, it is necessary to pass the ball in order to carry the ball to the goal. Therefore, our system analyzes available pass courses from the player holding the ball by making triangles among the players and tell them through sonification as well as visualization.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

---

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.

## 2 Proposed System

### 2.1 System overview

Our system visualizes the positional relationships between players and then represents these relationships as sound to promote understanding of football. The visualization and sonification shown below are performed only when one team (called *team A* here) is attacking.

### 2.2 Data

We use positional tracking data provided by Data Stadium Corporation, which records frame numbers, players, and the ball's position at every 1/25 second. The players' data are recorded for a total of 22 players (11 players  $\times$  2 teams) on the field (Table 1).

Table 1: Tracking Data

Data	Description
Game ID	ID that uniquely identifies a match
Frame	Tracking system frame number
HA	Flags for home and away identification. 1 : Home 2 : Away 0 : Ball
NO	Player's back number . 0 for ball
X	-5250~5220 Pitch size 105m x 68m, 105m side
Y	-3400~3400 Pitch size 105m x 68m, 68m side
Speed	Indicates the speed of the ball and the players in km/h

### 2.3 Visualization

First, every 1/25 second, the coordinates of the players and the ball for both teams are represented by a circle. Each team is distinguished by a different color, and the ball is represented by black.

Next, the system connects the points  $P_1, \dots, P_{11}$  representing each player of team A by an edge so that the following conditions are satisfied.

- Every point  $P_i$  has an edge.
- Every edge  $P_iP_j$  does not intersect any other edge  $P_kP_l$ .

This edge are drawn by using the Delaunay triangulation algorithm [4]. Suppose now that the player represented by vertex  $P_i$  is in possession of the ball (i.e., the coordinates of  $P_i$  and  $P_0$  overlap). When vertex  $P_i$  has edges  $P_iP_{j_1}, P_iP_{j_2}, \dots, P_iP_{j_N}$ , we can consider these as pass courses from  $P_i$ .

An example of the visualization is shown in Figure 1.

## 2.4 Sonification

The position of the ball and the pass courses obtained above are represented as sounds because these are highly related to the chances of scoring goals. These data are converted to MIDI note numbers and MIDI Note On messages are sent out. In the current implementation, an acoustic piano tone is used.

**Sonification of ball position** Every  $2/5$  second, the coordinates of the point  $P_0$  representing the ball are converted to a MIDI note number. The note number is determined according to Figure

**Sonification of pass courses** When a player holds the ball, the pass courses' feature obtained above is made audible. Let  $P_i$  be the vertices of the player holding the ball and  $P_j$  be the edges of  $P_iP_{j_1}, P_iP_{j_2}, \dots, P_iP_{j_N}$ . The vertices  $P_{j_1}, P_{j_2}, \dots, P_{j_N}$  can be interpreted to represent players to whom the ball can be passed from  $P_i$ . The coordinates of these vertices are then converted to note numbers according to Figure 2 and sounded. When  $N$  is large (when there are many passable players), chords with many tones are formed. When the edge  $P_iP_{j_n}$  ( $n = 1, 2, \dots, N$ ) is long or the coordinates of  $P_{j_n}$  are far from each other, chords with open voicings are formed. In this way, the user can know the occurrence and characteristics of the pass courses from the chords. This sonification process occurs only when the player holding the ball changes, unlike the sonification of the ball position.

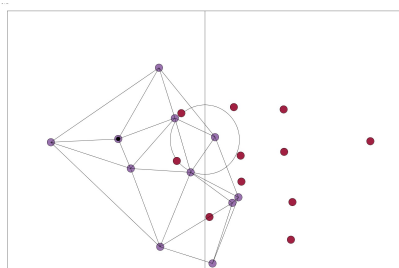


Fig. 1: Visualization Results

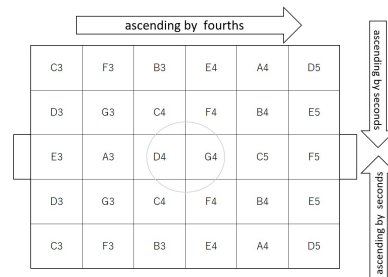


Fig. 2: How to determine note numbers

## 3 Preliminary Results

Using this system, the position of the ball and the pass courses were sonified using the positional data described in Section 2.1. Figure 3 is a spectrogram obtained by sonifying a scene in which team A connects a pass from a position close to its own goal to a player in the center of the rival team and carries the ball. As a result of the sonification, the following can be read.

- From around 0 to 4 seconds, only the sound of the ball was heard; there was no movement of the ball between blocks.
- From around 4 to 12 seconds, the sound (chord) consisting of multiple tones was heard as multiple pass courses were found
- From 12 to 18 seconds, only the ball is sounded, but the ball is moving between the blocks, so the pitch of the sound ascends
- From 18 seconds onwards, the pass courses were audible, while the ball was closer to the rival's goal, so the pitch of the sound was higher than in the 4–12 second period.

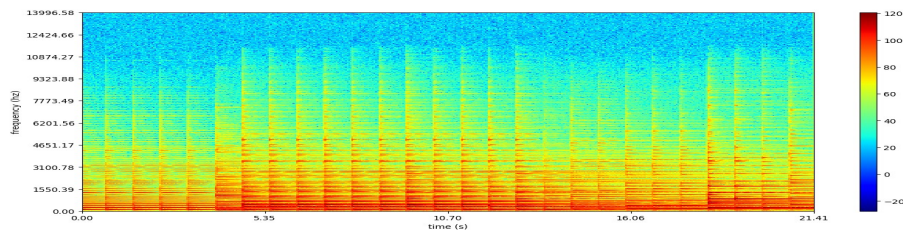


Fig. 3: Spectrogram of the sound generated by our sonification method

## 4 Conclusion

This paper described a prototype system that visually and aurally represented information on the positions and movements of players in a football game. The system facilitated the feature of pass courses in the football game by sonifying them as well as visualization.

For more precise analysis of game scenes, we have to solve many issues. If a player in the defending team is near from pass courses found through our system, this pass course cannot be considered available. If this is the case, this pass course must be excluded, but this has not yet been implemented. It is also necessary to consider how to sonify players other than those forming pass courses. Through such development, we would like to establish a technique that helps understand football games.

## References

1. M.Xu, J. Orwell, G. Jones, :Tracking football players with multiple cameras, ICIP (2004)
2. B. Gonçalves, R. Marcelino, L. Torres-Ronda, C. Torrents, J. Sampaio, :Effects of emphasising opposition and cooperation on collective movement behaviour during football small-sided games, Journal of Sports Sciences Volume 34(2016)
3. M.Wand, W.Straber , :A Real-Time Sound Rendering Algorithm for Complex Scenes, ISSN 0946-3852(2003)
4. D. T. Lee , B. J. Schachter , :Two algorithms for constructing a Delaunay triangulation, International Journal of Computer & Information Sciences volume 9, pages 219–242 (1980)

## Talking with Fish: an OpenCV Musical Installation

Gabriel Zalles Ballivian<sup>1</sup> \*

UC San Diego  
gzalles@ucsd.edu

**Abstract.** *Talking with Fish* is an interactive multi-media installation in which the movement of fish is translated to musical material using OpenCV. Specifically, the installation makes use of the centroids algorithm to track the position of multiple blobs simultaneously, and the data generated by these blobs is applied to synthesis parameters that generate sound. The X coordinate of each blob provides us with a frequency and position for the voice, the Y coordinate modulates the volume, and the area of each blob modifies the spread of the voice over the loudspeakers. The algorithm is perpetually modulating the musical key, moving clockwise along the circle of fifths, creating a constant harmonic movement for added interest.

### 1 Introduction

The idea of using fish and a sensing system to create art is not a novel concept. Walker, Kim, and Pends [1] from the Georgia Institute of Technology wrote about this idea in 2007. In this paper, the authors actually noted that sonifying elements of exhibits is not simply an artistic endeavor but actually has deeper implications. The concept the authors were attempting to convey is that visually impaired guests do not have the same experience in museums, or aquariums, as those with 20/20 vision. Therefore, as an approach to increasing inclusion in these spaces, sonifying data becomes imperative for all patrons to derive a meaningful experience. Part of our future goal, therefore, is to explore not just how this tank can be meaningfully sonified, but rather how can we add sound to the entire aquarium, to make a richer experience for visually impaired people<sup>1</sup>

FuXi [2] is another project which featured fish in a real-time music performance system. In contrast to our project, the authors of that work decided to incorporate a MIDI controller into the design, allowing the musician to collaborate with the fish in the music-making process. The authors note how the use of live animals adds indeterminacy to the composition and natural gestures. Rather than using a MIDI controller, our system tracks and reports the musical key while it generates the music. Our vision was that,

---

\* Thanks to Birch Aquarium at Scripps.

<sup>1</sup> This might include a guided audio tour or audio descriptors at our various tanks.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



habitually, professional musicians would be invited to the aquarium to accompany the fish in the music-making process<sup>2</sup>.

Baldan et al. [3] also discuss a real-time motion-tracking-based aquarium installation in their 2012 paper. The authors use four webcams, the Processing programming language, Pure Data, and Open Sound Control (OSC). The use of Free and Open Source Software (FOSS) makes this project more accessible and is something we would like to consider. Using a single application like MAX/MSP simplifies this process since a single program can be run. Baldan et al. used a number of interesting criteria for music making, such as evaluating the different blobs for color and using this data to influence timbre. They also use blob velocities which is something our system does not currently consider.

This installation, while not entirely novel, stands out due to the fact that it is permanent. Some of the other authors we mention in the literature review intervene in different contexts briefly, in contrast, this project was designed to be viewed by patrons of the aquarium 363 days a year (we close on New Year and Christmas). This means that thousands upon thousands of people will experience this material in one form or another over the course of the year - last year's attendance record for our aquarium was almost half a million people!

## 2 Technical Design

*Talking with Fish* relies on a Panasonic AW-HE2 camera mounted across our large kelp tank at Birch Aquarium at Scripps. The video signal is fed to a capture card which lets us use the feed directly in MAX/MSP. Once the video signal is recognized we modify the feed to facilitate blob tracking. To reduce GPU load the frames are resized to a much smaller resolution and these are then converted to a gray-scale before converting to a binary format. In this final conversion, the pixel data is essentially assigned a value of 0 or 1 based on the luminosity of the data point (e.g., bright pixels become white, dark pixels become black).

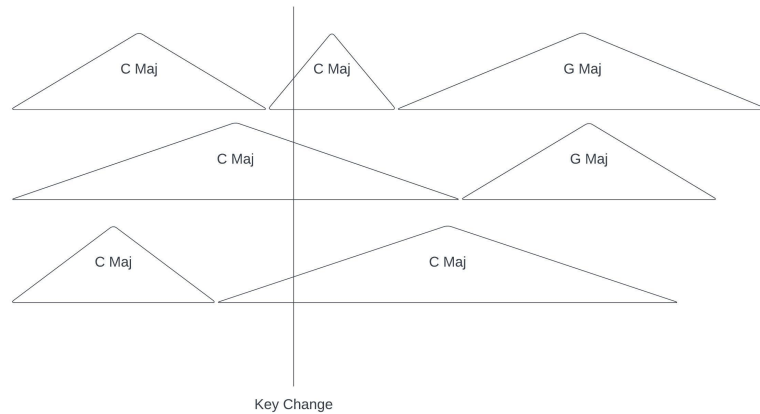
The centroid algorithm then works by looking for clusters of white pixels. A sizeable mass of grouped white pixels is considered a blob, and these blobs are evaluated moment per moment to determine their position at each frame. Our system currently allows for at most eight voices for the harmonic synthesis section. In order to create a constant sonic environment we built an operator which has a random note duration, above a specified minimum, and an envelope to control its gain. After each operator concludes producing the desired note, it uses the last available frequency value stored for the analysis to play the next chord member.

Before playing the new note, the operator also determines which musical key the system is currently in, to determine the final MIDI value to assign to the voice. As a result of this design the system is often in between musical keys and the harmonic analysis of the music would reveal that it is perpetually modulating around the chromatic scale. In other words, very often, the chords are composed of members of two musical keys, closely related by the number of incidentals. Figure 1 depicts this idea using only

---

<sup>2</sup> For example during special events.

three notes; notes before the key change are in C major, and notes after the key change are in G Major. At certain moments the two keys are overlaid one on top of the other, making the modulations smoother.



**Fig. 1.** Diagram showing harmonic structure using only 3 notes.

One of the challenges with this installation was creating something that would be informed by the data from the feed but that would remain active while fish were out of frame. Guests, however, also expected a marked sound to be played when a blob was detected. For this reason, a second instrument was developed digitally to satisfy this criterion. We call this the melodic element because the tones are short and sequenced linearly. The instrument matches the musical key of the chord generator and evaluates the number of columns in our data matrix. The OpenCV algorithm dynamically tracks the number of blobs, thus, whenever the count goes from low to high, we generate a melodic note to signify that a new blob has been detected.

### 3 Fabrication

The hardware for this project was placed inside a wooden box custom-made to fit the space we needed; the box actually shaped like a parallelepiped due to the topology of the space. Inside the box we hold the computer which runs the MAX/MSP standalone application and a mouse, to start the program each morning. The box also contains the aforementioned video card which turns the Panasonic AW-HE2 HDMI output into a serial stream recognizable by the computer. The audio output of the computer is the built-in headphone jack which connects the computer to a set of PreSonus speakers mounted on the walls of the aquarium. Above the box is a display monitor which shows the original video, the processed video (with green circles around each blob), and infor-

mation about the piece. A QR code on the GUI links to the project page on the artist's site. Please visit this temporary link for a demo.<sup>3</sup>

## 4 Future Work

One of the big criticisms of the work currently is that the kelp inside the Kelp Forrest often triggers sound more than the fish does. This is because the OpenCV program we are using does not use a trained model to identify fish, it simply searches for clusters of white pixels<sup>4</sup>. In the next version of this project, I would like to use a database of marine life species (e.g., images) to train a model. Unfortunately, the current hardware we have might not be able to handle this task in real-time, so for now we are sonifying both the fish and kelp. Even more appealing, would be the idea of identifying specific species using machine learning and computer vision. We envision a system that is trained to identify all the various species found in this tank and use a different instrument for each species - for example.

## 5 Conclusion

This paper has described the installation *Talking with Fish* which generates music material from the video analysis of a kelp forest in San Diego, CA. The MAX/MSP visual programming environment was leveraged to create a custom system that tracks centroids in the field of view. The area and coordinates of these pixel masses are used to drive a live synthesis algorithm that creates harmonic material in all 12 keys of the Western scale sequentially. The resulting program was compiled as a standalone application and copied to a dedicated computer responsible for creating the sound material from the live video feed.

## References

1. Walker, Bruce N., Jonathan Kim, and Anandi Pendse. "Musical soundscapes for an accessible aquarium: Bringing dynamic exhibits to the visually impaired." ICMC. 2007.
2. de Londres, Rua, and R. A. E. Macao. "FuXi: a Fish-Driven Instrument for Real-Time Music Performance."
3. Baldan, Stefano, Luca A. Ludovico, and Davide Andrea Mauro. "" Musica Sull'Acqua": A motion tracking based sonification of an aquarium in real time." (2012).

---

<sup>3</sup> <https://drive.google.com/file/d/17xMwLS4-QyuEB091nsoMBiWkgVyCsVFR/view?usp=sharing>

<sup>4</sup> A task made complicated due to the changing lighting conditions of the tank over the course of the day.

# The Sound Morphing Toolbox: Musical Instrument Sound Modeling and Transformation Techniques

Marcelo Caetano<sup>1\*</sup> and Richard Kronland-Martinet<sup>1</sup>

Aix-Marseille Univ, CNRS, PRISM

**Abstract.** Sound morphing requires the use of several audio processing algorithms involving the analysis, transformation, and resynthesis of sounds. The aim of this demo is to review the techniques implemented in the sound morphing toolbox (SMT). The SMT is open-source and freely available on GitHub. This demo will cover the analysis, transformation, and resynthesis steps used in standard morphing techniques applied to isolated notes from musical instrument sounds, such as sinusoidal modeling, spectral envelope estimation, time-scale modification, and resynthesis models. The demo will include sound examples to illustrate each step and a hands-on session to show the participants how to use the SMT.

## 1 Motivation and Relevance

Perceptually, sound morphing is a transformation that gradually blurs the categorical distinction between the sounds being morphed by blending their sensory attributes [4]. When morphing musical instrument sounds, the challenge lies in interpolating across dimensions of timbre to produce the perception of hybrid musical instruments. Figure 1 shows a striking example of image morphing to illustrate sound morphing with a visual analogy. For musical instrument sounds, the corresponding transformation gradually transitions from the source instrument to the target instrument. Additionally, the midpoint ( $\alpha = 0.5$ ) should give rise to the perception of a hybrid instrument that resembles both source and target instruments at the same time. To achieve that, we need a sound model that captures perceptual information and transformation strategies that allows manipulating this information in a perceptually meaningful way.

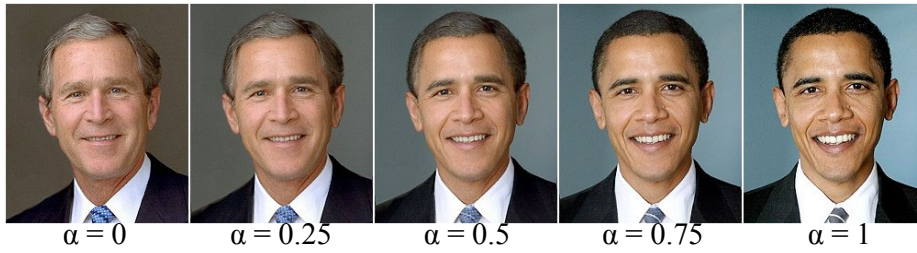
This demo will explore the methodological foundations and philosophical implications of morphing, from the conceptual formalization of morphing to the categorical perception of sounds. Morphing raises important issues in perception and cognition both in terms of research questions and the mathematical and computational means to address them. How is the morph represented in the brain given the representations of the source and target stimuli?

---

\* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 831852 (MORPH)



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



**Fig. 1.** Image morphing used to illustrate the aim of sound morphing.

The topic of the proposed demo overlaps with several of the topics of interest of the CMMR community, such as audio and music processing; representations of sound and music; timbre and musical instruments; sound and music analysis, synthesis, and transformation; philosophical implications and methodological foundations; and expression and performative aspects of music. Sound morphing uses several audio and music processing algorithms for the analysis, synthesis, and transformation of sounds. For example, sinusoidal modeling, additive synthesis, spectral modeling, and time-scale modification. Given the strong connection between musical instruments and timbre [1], interesting morphing transformations happen across dimensions of timbre perception [4] and timbre features [2] lie at the core of sound morphing.

## 2 Innovation

Sound morphing has found creative, technical, and research applications in the literature. In music composition, sound morphing allows the exploration of the sonic continuum [22,10,15] by creating hybrid sounds that are intermediate between a source and a target sounds. Sound morphing is also used in audio processing, sound synthesis, and sound design [21,6]. Additionally, sound morphing techniques have been used to investigate different aspects of timbre perception [9,5,19]. More recently, Google Magenta's *NSynth* <https://magenta.tensorflow.org/nsynth-instrument> applied machine learning techniques to sound morphing. New technologies push the boundaries of what is possible to achieve in terms of sound transformations. Particularly, the great yet unexplored potential for creative applications in music composition and performance that sound morphing possesses is capable of driving further developments in creative music systems.

The primary aim of this demo is to introduce the Sound Morphing Toolbox (SMT) to the participants. The SMT contains MATLAB® implementations of sound modeling and transformation algorithms used to morph musical instrument sounds. The SMT is open-source and freely available on GitHub<sup>1</sup>, making it highly flexible, controllable, and customizable by the user. This hands-on workshop is aimed mainly at less technically inclined participants such as composers or researchers without the technical background. During the workshop, participants will be guided on how to use the SMT

<sup>1</sup> <https://github.com/marcelo-caetano/sound-morphing>

step by step. Our aim is to provide an intuitive rather than technical understanding of the audio processing algorithms used. By the end of the workshop, the participants will be able to make informed decisions about audio processing algorithms and parameter values and use the SMT on their own. Additionally, the workshop will draft a research agenda for sound morphing that introduces technical aspects, aesthetic and perceptual issues. Finally, we will identify shortcomings of the currently available pieces of morphing software listed above and research opportunities.

### 3 Research

The SMT has open-source implementations of sound modeling and transformation algorithms based on the sinusoidal model [14,18] and the source-filter model [20,4]. The demo will review several classic audio processing algorithms widely used in sound analysis, transformation, and resynthesis, giving the participants a broad overview of musical instrument sound morphing and the tools to use them.

The demo will briefly review the analysis steps, namely parameter estimation, peak selection, partial tracking, partial selection, and harmonic selection in the SMT. In additive synthesis, the estimation of the parameters can use nearest-neighbor estimation [14] or parabolic interpolation under linear, log [18], or power scaling [3]. Peak selection [17] allows to only keep the parameters of the spectral peaks that correspond to sinusoids. Partial tracking [14] connects these peaks across frames to create time-varying sinusoids called *partials*, that are further selected according to their duration and harmonicity.

The demo will cover spectral envelope estimation with linear prediction [12] and iterative cepstral smoothing [16]. The demo will also cover the time-scaling modification (TSM) and frequency transposition techniques currently implemented in the SMT. TSM uses SOLA-FS [11]. Frequency transposition uses the sinusoids, with the frequencies transposed by intervals in cents. The amplitudes can be transposed or preserved using the spectral envelope [4]. Finally, the demo will cover both additive and source-filter resynthesis techniques. The SMT has implementations of additive synthesis by cubic-phase polynomial fitting [14], phase reconstruction by frequency integration [13], and overlap add [8,7]. Additionally, the residual from sinusoidal analysis is further modeled with the source-filter paradigm. The demo will briefly discuss residual modeling by time-varying linear prediction estimation [18].

### References

1. Anne Caclin, S. McAdams, Bennett K. Smith, and Suzanne Winsberg. Acoustic correlates of timbre space dimensions: A confirmatory study using synthetic tones. *J. Acoust. Soc. Am.*, 118:471–482, 2005.
2. M. Caetano, C. Saitis, and K. Siedenburg. Audio content descriptors of timbre. In K. Siedenburg, C. Saitis, S. McAdams, A. Popper, and R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition. Springer Handbook of Auditory Research*, volume 69, chapter 11, pages 297–333. Springer, Cham, 2019.

3. Marcelo Caetano and Philippe Depalle. On the estimation of sinusoidal parameters via parabolic interpolation of scaled magnitude spectra. In *Proceedings of the International Conference on Digital Audio Effects (DAFx20in21)*, pages 81–88, 2021.
4. Marcelo Caetano and Xavier Rodet. Musical instrument sound morphing guided by perceptually motivated features. *IEEE Trans. Audio, Speech, Language Process.*, 21(8):1666–1675, 2013.
5. Sandra Carral. Determining the just noticeable difference in timbre through spectral morphing: A trombone example. *Acta Acust united Ac*, 97:466–476, 05 2011.
6. K. Fitz, L. Haken, S. Lefvert, C. Champion, and M. O’Donnell. Cell-utes and flutter-tongued cats: Sound morphing using loris and the reassigned bandwidth-enhanced model. *Comput. Music J*, 27(3):44–65, Sep. 2003.
7. E. B. George and M. J. T. Smith. Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model. *IEEE Trans. Speech Audio Process.*, 5(5):389–406, Sep. 1997.
8. E. Bryan George and Mark J. T. Smith. Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones. *J. Audio Eng. Soc*, 40(6):497–516, 1992.
9. John M. Grey and John W. Gordon. Perceptual effects of spectral modifications on musical timbres. *J Acoust Soc Am*, 63(5):1493–1500, 1978.
10. Jonathan Harvey. “Mortuos Plango, Vivos Voco”: A realization at IRCAM. *Comput. Music J*, 5(4):22–24, 1981.
11. Don Hejna and Bruce R. Musicus. The SOLA-FS time-scale modification algorithm. Technical report, BBN, 1991.
12. John Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63:561–580, 1975.
13. R. McAulay and T. Quatieri. Magnitude-only reconstruction using a sinusoidal speech model. In *Proc. ICASSP*, volume 9, pages 441–444, March 1984.
14. R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Trans. Acoust. Speech Signal Process.*, 34(4):744–754, Aug 1986.
15. Michael McNabb. “Dreamsong”: The composition. *Comput. Music J*, 5(4):36–53, 1981.
16. Axel Röbel and Xavier Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Proceedings of the 8th International Conference on Digital Audio Effects*, pages 30–35, 2005.
17. Xavier Rodet. Musical sound signal analysis/synthesis: Sinusoidal+residual and elementary waveform models. In *IEEE Time-Frequency and Time-Scale Workshop*, 1997.
18. X. Serra. *Musical Signal Processing*, chapter Musical Sound Modeling with Sinusoids plus Noise, pages 91–122. G. D. Poli, A. Piccialli, S. T. Pope, and C. Roads Eds. Swets & Zeitlinger, Lisse, Switzerland, 1996.
19. Kai Siedenburg, Kiray Jones-Mollerup, and Stephen McAdams. Acoustic and categorical dissimilarity of musical timbre: Evidence from asymmetries between acoustic and chimeric sounds. *Front Psychol*, 6:1977, 2016.
20. M. Slaney, M. Covell, and B. Lassiter. Automatic audio morphing. In *Proc. ICASSP*, volume 2, pages 1001–1004, May 1996.
21. Edwin Tellman, Lippold Haken, and Bryan Holloway. Timbre morphing of sounds with unequal numbers of features. *J. Audio Eng. Soc*, 43(9):678–689, 1995.
22. Trevor Wishart. *On Sonic Art*. Routledge, New York, 1 edition, 1996.

# Morphing of Drum Loop Sound Sources Using CNN-VAE

Mizuki Kawahara<sup>1</sup>, Tomoo kouzai<sup>1</sup>, and Tetsuro Kitahara<sup>1\*</sup>

College of Humanities and Sciences, Nihon University, Japan  
{kawahara,kouzai,kitahara}@kthrlab.jp

**Abstract.** In this paper, we attempt a morphing technique that combines a convolutional neural network (CNN) and a variational autoencoder (VAE) in order to produce a variety of sound sources of drum loops. Although there have already been studies related to sound or music morphing, and some of them have focused on drum sound synthesis, morphing of sound sources of drum loops has not been attempted. Our system trains the spectrograms of the drum loop sound sources using CNN-VAE and generates a new source by interpolating two sources in the latent space. Preliminary experiments using commercially available sound sources show promising results.

**Keywords:** morphing, spectrogram, convolutional neural networks (CNN), variational autoencoder (VAE), drum sound source

## 1 Introduction

A loop sequencer is commonly used in music production, with which the creator concatenates and mixes various loop sound sources. However, it is often difficult to find the sound sources that they desire from a limited set of sound sources. For this reason, research has been conducted to generate a variety of sound sources by morphing. For example, Primavera et al. [1] proposed a method to achieve smooth transitions between different sound sources in sound morphing. Nistal et al.[2] and Aouameur et al.[3] proposed a method for the synthesis of drum sounds, in which they have explored methods for extracting features of drum sounds and generating new sound sources based on the extracted features.

In this paper, we propose a sound sources morphing method that combines a convolutional neural network (CNN) and a variational autoencoder (VAE). First, the features of given drum loop sound sources are extracted using a CNN and are mapped to the latent space using a VAE. Then, a new sound source is generated by morphing two given sound sources in the latent space.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



## 2 Proposed System

Our proposed method uses a model based on CNN and VAE (hereinafter referred to as CNN-VAE) to achieve morphing in a low-dimensional latent space representing the features of sound sources.

In the training phase, the sound sources are first transformed into spectrograms by the Fourier transform. Then, a convolutional layer is used to map the sound sources into a low-dimensional latent space. Then, the inverse convolution layer is used to reconstruct the spectrogram of the original sound source. The CNN-VAE model is trained so that the reconstructed spectrogram is equivalent enough to the original spectrogram.

At the generation phase, two sound sources are selected from the trained one. A new vector in the latent space is generated by interpolating the two vectors corresponding to the selected sources. Then a spectrogram is generated by using the decoder.

### 2.1 Generation of spectrograms

An input audio signal is transformed into a spectrogram with the short-time Fourier transform (STFT). A Hamming window is used with a window width of 2048 and a hop size of 1/4 of the window length. Since the sampling frequency is assumed to be 22050 Hz and the length of the audio signal is about 3.43 seconds (two measures at 140 BPM), the spectrogram is a 1025-by-148 matrix.

### 2.2 Building a CNN-VAE model

The CNN-VAE model is built and trained with given spectrograms of drum loop sound sources. This model has an encoder consisting of three convolution layers to compress spectrograms and them to the 16-dimensional latent space. The decoder consists of three deconvolution layers that reconstruct spectrograms from the 16-dimensional latent vector. The ReLU function is used as the activation function, the batch size is 64, and the number of epochs is 3000. The mean square error is used as the loss function.

### 2.3 Morphing

The user selects two sound sources (denoted as  $s_i$  and  $s_j$ ) from the trained ones. Let  $z_i$  and  $z_j$  be the latent vectors obtained from  $s_i$  and  $s_j$ , respectively. Then, a new latent vector  $z_{\text{new}} = (1 - \alpha)z_i + \alpha z_j$  is calculated, in which  $z_{\text{new}}$  represents an internally dividing point of  $z_i$  and  $z_j$  in the ratio  $\alpha : 1 - \alpha$ . Finally, the spectrogram is transformed into an audio signal by inverse Fourier transform and phase restoration.

## 3 Preliminary Experiment

### 3.1 Method

We trained our model with 74 drum loop sound sources. The morphing parameters  $\alpha$  were set to 0.00, 0.25, 0.50, 0.75, and 1.00. The 74 drum sounds were taken from a commercial loop sound dataset "Sound Pool vol.2"<sup>1</sup>(genre: Techno & Trans).

<sup>1</sup> <https://www.ah-soft.com/soundpool/>

**Table 1.** Pairs of sound sources used in the experiment

degree of similarity	0.0013	0.2505	0.5101	0.7501
sound source pair	s17, s24	s15, s28	s34, s73	s08, s51

**Table 2.** Morphing result (similarity to original source)

(a) s17 — s24						(b) s15 — s28					
$\alpha$	0.00	0.25	0.50	0.75	1.00	$\alpha$	0.00	0.25	0.50	0.75	1.00
s17	1.0000	0.8510	0.1120	0.0041	0.0013	s15	1.0000	0.9221	0.6264	0.3368	0.2505
s24	0.0013	0.0211	0.3877	0.9331	1.0000	s28	0.2505	0.4165	0.6708	0.8931	1.0000

(c) s34 — s73						(d) s08 — s51					
$\alpha$	0.00	0.25	0.50	0.75	1.00	$\alpha$	0.00	0.25	0.50	0.75	1.00
s34	1.0000	0.5736	0.5838	0.5830	0.5101	s08	1.0000	0.9492	0.8218	0.7596	0.7501
s73	0.5101	0.6714	0.6954	0.9250	1.0000	s51	0.7501	0.7818	0.7873	0.9773	1.0000

After these sound sources were trained, pairs were selected for morphing. The selected pairs are shown in Table 1. The "s + number" (e.g., s01, s02, ...) represents sound sources. The pairs of sound sources used for morphing were selected so that the cosine similarity of the spectrograms is 0.00, 0.25, 0.50, and 0.75 in order to try a wide range of pairs, including those similar and dissimilar to each other.

When  $\alpha$  is close enough to 0 (or 1), the generated sources should become similar to  $s_i$  (or  $s_j$ ) as  $\alpha$ . We confirm this by calculating the cosine similarity between the spectrograms of the generated sources and the original sources.

### 3.2 Results

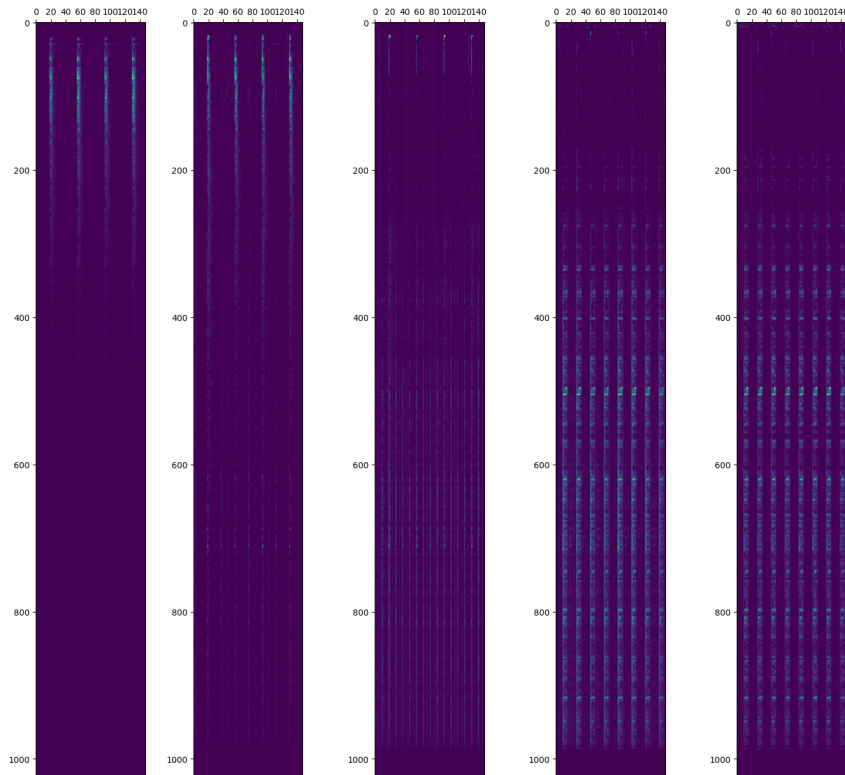
The results of the morphing are shown in Table 2. Some of the generated results are posted at the following URL: <https://sites.google.com/kthrlab.jp/en-drum-morphing>

It is confirmed that the original sound sources are reconstructed with sufficient accuracy for  $\alpha=0.00$  and  $\alpha=1.00$ . In the cases of  $\alpha=0.25$  and  $\alpha=0.75$ , the similarity is intermediate between the values when  $\alpha = 0.00$  and when  $\alpha = 1.00$ . This indicates that generated sound sources that contain both features of the two sources.

For the pairs in Table 2 (b) and Table 2 (c), when  $\alpha=0.50$ , sound sources with low similarity to both original sources were generated. It implies the possibility that our model can generate novel sources. In fact, Fig 1 shows that the spectrogram with  $\alpha = 0.50$  is different from those with  $\alpha = 0.00$  and  $\alpha = 1.00$ .

## 4 Conclusion

In this paper, we proposed a CNN-VAE model to achieve sound source morphing in the latent space. When we changed  $\alpha$  (a morphing ratio), the model generated different sound sources accordingly. Future work includes larger-scaled experiments with various sound source pairs and subjective evaluation through listening experiments.



**Fig. 1.** Spectrogram of morphed source from s17 and s24 (from left to right  $\alpha = 0.00, 0.25, 0.50, 0.75, 1.00$ )

## References

1. Andrea Primavera, Francesco Piazza, and Joshua D. Reiss: Audio Morphing for Percussive Hybrid Sound Generation, *Proceedings of the 45th AES Conference* (2012).
2. Javier Nistal, Stefan Lattner, and Gaël Richard: DrumGAN: Synthesis of Drum Sounds With Timbral Feature Conditioning Using Generative Adversarial Networks, arXiv:2008.12073 (2020).
3. Cyran Aouameur, Philippe Esling, and Gaëtan Hadjeres: Neural Drum Machine: An Interactive System for Real-time Synthesis of Drum Sounds, arXiv:1907.02637, (2019).

# Generating Tablature of Polyphony Consisting of Melody and Bass Line

Shunsuke Sakai<sup>1</sup>, Hinata Segawa<sup>1</sup> and Tetsuro Kitahara<sup>1</sup> \*

College of Humanities and Sciences, Nihon University  
{sakai, segawa, kitahara}@kthrlab.jp

**Abstract.** Our final goal is to develop a system that generates a tablature for a given lead sheet consisting of a melody and a chord progression. Generating a tablature for a lead sheet requires a complex solution search because plural possibilities exist in voicing each chord. As the first step, we address a system that generates a tablature consists of a melody and a bass line using the Viterbi algorithm. Polyphonic fingering states are modeled as 6-dimensional vectors and the playing difficulty is modeled as a cost function of such vectors. By minimizing the cost function, our system generates a playable tablature.

## 1 Introduction

Tablatures are helpful to play the guitar, so many non-professional guitarists use tablatures. Therefore, there have been attempts to automatically generate tablatures from audio signals or scores. Wiggins et al. used audio signals as inputs and estimated fingering positions using a neural network [1]. Yazawa et al. also used audio signals as inputs, and generated tablatures using fingering forms and note value-based costs [2]. Hori et al. proposed a web application that enables arrangement with transposition using hidden Markov models [3].

The solo guitar, which means playing both a melody and an accompaniment alone, is an attractive playing style for guitars, especially classic guitars. However, there are less commercially available tablatures for the solo guitar. Therefore, if one wants to play their favorite songs in the solo guitar, they must arrange those songs for the solo guitar. However, it is a difficult task for most amateur guitarists.

The final goal is to achieve a system that automatically generates such tablatures for the solo guitar. Inputs are assumed to be lead sheets, which describe melodies and chord progressions but no voicings. We need a complex solution search to generate tablatures from lead sheets because there are plural possibilities in the voicing of each chord. As the first step, we address a system that generates tablatures for simultaneously playing a melody and bass line.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.

## 2 Proposed System

Given a lead sheet describing a melody and a chord progression, our system generates tablatures for simultaneously playing the melody and the bass line on the classic guitar.

### 2.1 Importing MusicXML data

Once a lead sheet given in the MusicXML format, a sequence of the melody notes and the bass notes is represented as:

$$X = \{(x_1, r_1), (x_2, r_2), \dots, (x_N, r_N)\}$$

where  $x_n$  is the pitch (MIDI note number) of the  $n$ -th note and  $r_n$  is the chord's root note (pitch class from 0 to 11) when note  $x_n$  is being played. In the current implementation, the root note is played only at the timing of a chord change, and  $r_n$  is empty when there is no chord or the previous chord continues (represented by  $r_n = \epsilon$ ).

### 2.2 Designing the Viterbi Algorithm

Given  $X$ , our system estimates the fingering positions for each element of  $X$ .

**Set of fingering state vectors** Let  $V$  be the set of vectors representing the playable fingering states. Each element  $\mathbf{v}$  in  $V$  is represented by a 6-dimensional vector  $\mathbf{v} = (f_1(\mathbf{v}), f_2(\mathbf{v}), f_3(\mathbf{v}), f_4(\mathbf{v}), f_5(\mathbf{v}), f_6(\mathbf{v}))$ . Let  $f_m(\mathbf{v})$  denote the fret number of string  $m$  ( $f_m(\mathbf{v}) = -1, 0, \dots, 14$ ), where  $f_m(\mathbf{v}) = -1$  means not to play and  $f_m(\mathbf{v}) = 0$  means open string. To ensure that  $V$  contains only playable fingering states,  $V$  only has elements that satisfy the following conditions.

1.  $\max_m(f_m(\mathbf{v})) - \min_m(f_m(\mathbf{v})) \leq 3$ ,
2. The number of  $m$  satisfying  $f_m(\mathbf{v}) > 0$  is 2 or less,
3. Multiple strings do not correspond to the same pitch.

**Basic Mechanism** Given  $X = \{(x_1, r_1), \dots\}$  representing the main melody (+ root notes of the chord progression), the system finds the optimal sequence of fingering states,  $Q = \{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_N\}$  ( $\mathbf{q}_n \in V$ ) by minimizing the cost (degree of non-optimality) for  $Q$ . The cost is defined as a combination of the following three.

- Initial cost  $C(\mathbf{q}_1)$ : gives a slightly larger cost to the fret furthest from the neck, based on the idea that playing in a position closer to the neck is more common.
- Transition cost  $C(\mathbf{q}_{n+1}|\mathbf{q}_n)$ : Based on the idea that it is easier to play when the movement of the fingering position is less, a larger cost is given when the movement of the fingering position is larger.
- Emission cost  $C((x_n, r_n)|\mathbf{q}_n)$ : gives a sufficiently large cost if somebody cannot play the correct note at the given fingering position.

The total cost  $C(Q)$  is defined by the following equation:

$$C(Q) = C(\mathbf{q}_1) + \left\{ \sum_{n=1}^{N-1} (C((x_n|r_n)|\mathbf{q}_n) + C(\mathbf{q}_{n+1}|\mathbf{q}_n)) \right\} + C((x_N|r_N)|\mathbf{q}_N)$$

and the Viterbi algorithm finds the minimum  $Q$ .

The initial cost, transition cost, and emission cost are defined as follows:

**Initial cost** On acoustic guitars, the closer-to-the-neck position is more common than the closer-to-the-body position. Therefore, we give the closer-to-the-body position a slightly higher cost than the closer-to-the-neck position. That is, the initial cost  $C(\mathbf{q}_1)$  is defined as follows:

$$C(\mathbf{q}_1) = \begin{cases} 2.5 & (\max_m(f_m(\mathbf{q}_1)) \leq 5) \\ 5.0 & (\text{otherwise}) \end{cases}$$

**Transition cost** By giving higher costs to large movements in fingering-positions, we reduce the difficulty of playing the strings. In addition, for the same reason as above, we prioritize the position closest to the neck. Therefore, we divide the transition cost  $C(\mathbf{q}_{n+1}|\mathbf{q}_n)$  into the cost of the move itself  $C_1(\mathbf{q}_{n+1}|\mathbf{q}_n)$  and the cost to prioritize the neck side  $C_2(\mathbf{q}_{n+1})$ :

$$\begin{aligned} C(\mathbf{q}_{n+1}|\mathbf{q}_n) &= C_1(\mathbf{q}_{n+1}|\mathbf{q}_n) + C_2(\mathbf{q}_{n+1}) \\ C_1(\mathbf{q}_{n+1}|\mathbf{q}_n) &= \begin{cases} 0.0 & (\text{dist}(\mathbf{q}_n, \mathbf{q}_{n+1}) \leq 3) \\ 5.0 & (\text{dist}(\mathbf{q}_n, \mathbf{q}_{n+1}) \leq 4) \\ 30.0 & (\text{otherwise}) \end{cases} \\ C_2(\mathbf{q}_{n+1}) &= \begin{cases} 5.0 & (\max_m(\mathbf{q}_{n+1}) = 0) \\ 10.0 & (\max_m(\mathbf{q}_{n+1}) \leq 4) \\ 20.0 & (\text{otherwise}) \end{cases} \end{aligned}$$

where  $\text{dist}(\mathbf{v}_1, \mathbf{v}_2)$  ( $\mathbf{v}_1, \mathbf{v}_2 \in V$ ) is defined as follows:

$$\text{dist}(\mathbf{v}_1, \mathbf{v}_2) = \left| \max_m(f_m(\mathbf{v}_1)) - \max_m(f_m(\mathbf{v}_2)) \right|$$

**Emission cost** The emission cost indicates whether the fingering position can produce the given note. Let  $\text{note}(f_m(\mathbf{v}))$  be the pitch (MIDI note number) played at the fingering position  $f_m(\mathbf{v})$  on string  $m$ . The fingering state  $\mathbf{q}_n$  produces  $(x_n, r_n)$  when each of  $m = 1, \dots, 6$  satisfies one of the following:

- $f_m(\mathbf{q}_n) = -1$
- $\text{note}(f_m(\mathbf{q}_n)) = x_n$
- $\text{note}(f_m(\mathbf{q}_n)) = r_n + 12o$  (in the case of  $r_n \neq \epsilon$ )

$o$  is an integer to change octaves in the range satisfying  $r_n < x_n$ . The emission cost  $C((x_n, r_n)|q_n)$  is represented as follows.

$$C((x_n, r_n)|q_n) = \begin{cases} 0.0 & (\text{satisfying the above conditions}) \\ 10.0 & (\text{otherwise}) \end{cases}$$

### 2.3 Exporting Tablature

The notes and fingering positions obtained by the method described above are exported in the MusicXML format.

## 3 Preliminary Results

We tried to generate a tablature for a lead sheet shown in Figure 1. The tablature generated by the proposed system is shown in Figure 2. The melody in the lead sheet in Figure 1 was retained, and the root note of each chord was output. Because the initial and transition costs for open strings are relatively low, the generated tablature used open strings as much as possible.



Fig. 1. Input lead sheet

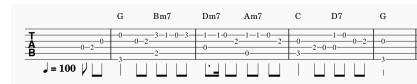


Fig. 2. Generated tablature through our system

## 4 Conclusion

In this paper, we proposed a system that generates a tablature containing a melody and bass line from a lead sheet. Our preliminary experiment shows that the system generated a tablature with which the player can play a melody and a bass line simultaneously. Future work will include the generation of more complex accompaniments such as arpeggio.

## References

1. A. Wiggins and Y. Kim: Guitar Tablature Estimation with A Convolutional Neural Network. In Proceedings of ISMIR (2019).
2. K. Yazawa, K. Itoyama, and H. G. Okuno: Automatic Transcription of Guitar Tablature from Audio Signals in Accordance with Player's Proficiency. In Proceedings of ICASSP, IEEE (2014).
3. G. Hori and S. Sagayama: HMM-based Automatic Arrangement for Guitars with Transposition and its Implementation. In Proceedings of ICMC-SMC (2014).

## Development of an easily-usable smartphone application for recording instrumental sounds

Takanori Horibe and Masanori Morise \*

Meiji University  
{cs222034, mmorise}@meiji.ac.jp

**Abstract.** We have studied the automatic performance skill evaluation in the instrumental sound based on only the recorded sound. Since many instrumental sounds are essential for statistical analysis, a tool for effectively collecting instrumental sounds is helpful. This paper introduces an easily-usable smartphone application that users can record their performance with a single tap operation. This application has several functions to appropriately reject the insufficient result based on simple acoustical analysis.

**Keywords:** recording application, instrumental sound, acoustical analysis

### 1 Introduction

Instrumental sound analysis has been carried out from several aspects [1–3], and the recording tool is essential to collect the sounds. To record the instrumental sounds with high quality, the researchers often employ a recording engineer and use a quiet environment, such as a soundproof room and recording studio. On the other hand, if the background noise in the environment does not affect the acoustic analysis, it is unnecessary to record the sound with such quality. In such a case, it is reasonable to record the instrumental sounds by each user with a smartphone application.

In this study, we developed a smartphone application for recording instrumental sounds with a fundamental frequency that the user can easily record the sound. Since the player and the recording operator use this application for recording, the user includes both of them in this study. This application has several functions to automatically reject insufficient results based on acoustic analysis. The concept of this application is that non-expert can record instrumental sounds with a simple operation.

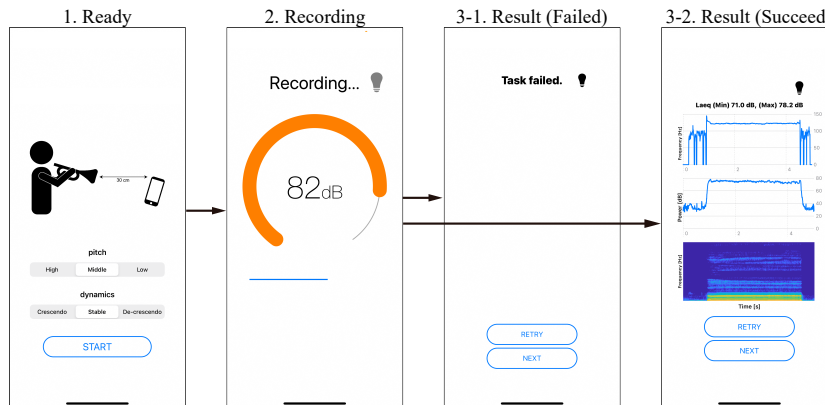
---

\* We thank Dr. T. Oku for providing the instrumental sound required for making Fig. 2. This work was supported by JSPS KAKENHI Grant Number JP21H04900.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).





**Fig. 1.** Screenshots of the implemented application. The user can record the instrumental sounds with a single tap of the START button, and the insufficient result is automatically rejected.

## 2 Concepts and implementation

### 2.1 Concepts of the application

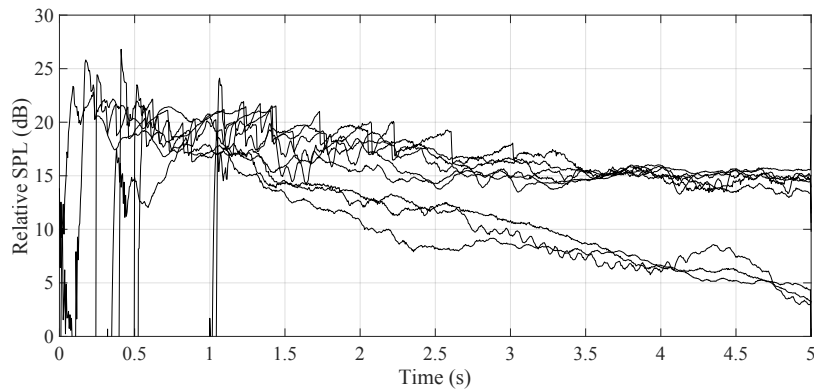
It is assumed that users cannot always record their performances in soundproof rooms or recording studios and may record in general rooms. Since background noise is contained in the instrumental sound recorded in such an environment, it is desirable to guarantee enough sound pressure level (SPL) of instrumental sounds. Based on the above, we attempted to implement an interface and functions that meet the following two points.

- When this application is launched, the recommended distance between the smartphone and the instrument is provided as a guide for recording at the sufficient SPL. The recording can be completed with a single button tap operation.
- A simple analysis is performed immediately after the recording to automatically detect errors such as missed performances, clipping, and inappropriate recording, for example, far distance from the smartphone.

### 2.2 Procedure for recording

Fig. 1 shows the procedures for recording by using the implemented application. After setting the recording condition by the left panel of Fig. 1, the user can record the instrumental sounds with only a single tap. We explain the detailed procedure as follows.

First, the user places a smartphone in an appropriate position and launches this application. On the *1. Ready* panel, the user selects the pitch and volume of the instrumental sound from the “pitch” and “dynamics” boxes and starts recording by the START button. As shown in *2. Recording* panel, the application records instrument sound for 5 seconds after the 3-second countdown. During recording, the user plays a



**Fig. 2.** Time sequences of the relative SPL of ten trumpet sounds played as the decrescendo.

long tone with monitoring the equivalent continuous A-weighted SPL. Clipping is automatically detected when the maximum absolute amplitude of the instrumental sound exceeds 0.95. If clipping, the icon in the upper right corner lights up, and the application skips the following step. If not, a simple acoustic analysis is carried out to reject the insufficient result.

When the result does not meet the required quality, it moves to *3-1. Result (Failed)* panel and instructs re-recording. If it does, it moves to *3-2. Result (Succeed)* panel. It feeds back the fundamental frequency contour, power based on equivalent continuous A-weighted SPL, and spectrogram of the instrumental sound to the user. The user can select whether to Accept or Reject by referring to these results; if Accept, the user can return to *1. Ready* panel by the NEXT button, and if Reject, the user can move to *2. Recording* panel by RETRY button and start over from the countdown. The accepted result is automatically saved with a filename based on the selected conditions.

### 2.3 Implementation

This application was developed with Swift. The instrumental sounds are recorded using the AVAudioRecorder [4] of the AVFAudio Framework. The sampling conditions are 48 kHz, 16 bits with PCM format. The equivalent continuous A-weighted SPL displayed on *2. Recording* panel was calculated with a frame length of 200 ms.

An acoustic analysis to determine the validity of the recording results is as follows. First, the fundamental frequency and spectral envelope are calculated from the recorded instrumental sounds using WORLD [5]. The spectral envelope is summed for each frame to obtain the relative SPL. The validity of the recording results is judged based on the relative SPL, and the insufficient results are rejected.

This application can reject the sound with too high SPL by detecting the clipping. On the other hand, provided the distance between the microphone and the instrument is too far, the instrumental sound is not likely to be recorded with sufficient SPL. To solve this problem, we test-recorded ten decrescendo trumpet sounds with this application with the sufficient condition and confirmed the relative SPL. Fig. 2 shows the relative

SPLs of all results. The horizontal and vertical axes show the time and relative SPL, respectively. According to this result that the relative SPLs were included from 3 to 27 dB, we calculated a median value from the relative SPL in all frames identified as voiced section. When the median value is in the range of 3–27 dB, the result is accepted. This calculation enables to reject where the SPL of the instrumental sound is too low.

### **3 Discussion**

The user can use this application from recording to analysis with a single tap operation. The application can reject the insufficient result. The clipping detection rejects the recorded sound with too high SPL. The identification by the threshold also rejects the recorded sound with too low SPL. Since this function does not require an environment such as a soundproof room, the user can record their performance in a general room.

Since our research target is the automatic performance skill evaluation by recorded instrumental sounds, the next step requires the evaluation of the application by recording many kinds of sounds with many players. We can evaluate the application in two aspects; One is whether the insufficient result is appropriately rejected. The other is whether the sufficient result is appropriately accepted. The evaluation also includes the usability evaluation of whether the user can easily use this application.

### **4 Conclusion**

In this study, we implemented a smartphone application to record instrumental sounds and confirmed that users can record, analyze, and save a file with a single tap operation. This application has several functions to obtain only reliable sounds. Informal tests confirmed that this application could automatically reject the insufficient sound.

We will statistically evaluate the performance of implemented functions by many recording results. After confirmation of the adequacy of them, we will collect the instrumental sounds by various kinds of users. Developing acoustic features related to the performance skill by using the recording sounds is also an important future work.

### **References**

1. N. Kimura, K. Shiro, Y. Takakura, H. Nakamura, and J. Rekimoto: SonoSpace: Visual Feedback of Timbre with Unsupervised Learning, 28th ACM International Conference on Multimedia, pp. 367–374 (2020)
2. T. Knight, F. Upham, and I. Fujinaga: The potential for automatic assessment of trumpet tone quality, 12th International Society for Music Information Retrieval Conference, pp. 573–578 (2011)
3. A. Lee, S. Furuya, M. Morise, P. Iltis and E. Altenmüller: Quantification of instability of tone production in embouchure dystonia, *Parkinsonism & Related Disorders*, vol. 20, no. 11, pp. 1161–1164 (2014)
4. “AVAudioRecorder”, <https://developer.apple.com/documentation/avfaudio/avaudiorecorder>
5. 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. E99-D, no. 7, pp. 1877–1884 (2016)

## **A Research on Music Generation by Deep-Learning including ornaments**

### **- A case study of world harp instruments-**

Arturo Alejandro Arzamendia Lopez, Akinori Ito, Koji Mikami

Tokyo University of Technology

g312200180@edu.teu.ac.jp, akinori@edu.teu.ac.jp,  
mikami@stf.teu.ac.jp

**Abstract.** In this research, we explore the application of deep learning techniques, including recurrent neural networks and LSTM, to traditional Paraguayan music known as "Guarana". This style is characterized by specific playing techniques and ornaments such as arpeggios and glissandos, which are executed using the Paraguayan harp. The learning and generation processes are performed individually using the TensorFlow and Keras libraries comparing the different results from different architectures to identify which one generates the most accurate or similar harp music that captures the intricacies of "Guarana" style. Furthermore, in future work, we demonstrate the capability of these techniques in other world harp music styles, such as Meiji period music employing the Koto, and western music from the 19th and 20th centuries incorporating the concert harp.

**Keywords:** LSTM, Guarania, Musical Ornaments, Glissando, Arpeggio, Tremolo.

## **1 Introduction: Paraguayan Identity**

Paraguay, a landlocked country nestled in the heart of South America, has its own unique history and culture, which may not be as widely recognized externally but is deeply rooted in the hearts of Paraguayans. This identity, referred to as "Paraguayidad" (Paraguayan-ness), is shaped by a history of war and immigration, bilingualism (Guarani and Spanish), geographical isolation, among other factors. The primary means of expressing this Paraguayan national identity is through folkloric music [1].

Despite the deep-rooted and widespread nature of this Paraguayan identity within the country's culture, its music is barely recognized outside its borders and there are few examples of scientific research dedicated to it. This situation has led to the initiation



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

of this research, which focuses on Paraguayan music and employs contemporary deep learning techniques for music generation. The aim is to raise awareness about the existence and beauty of Paraguayan culture while simultaneously exploring the capabilities of deep learning in a slightly different context.

## **2 Paraguayan Music Styles: Guaranía and Paraguayan Polka**

In the Paraguayan folkloric repertoire there are two major musical styles: Guaranía and Paraguayan Polka.

The Paraguayan Polka is a rhythmically lively song, with a 6/8-meter, diatonic harmony, and the use of hemiola and syncopation in rhythmic patterns [2]. Due to its lively nature, it is more popular in rural areas, as it is well-suited for dancing and festive celebrations.

On the other hand, Guaranía, it was created by musician José Asunción Flores in the early 20th century as a way to express the character of the Paraguayan people [3]. While it shares melodic and harmonic features with the Paraguayan Polka, Guaranía is slower and imparts a more nostalgic, sentimental feel resonates predominantly in urban areas.

Both styles, can be performed in various ways, including orchestra arrangements, guitar renditions, and vocal interpretations. However, the most cherished and traditional instrument in Paraguayan culture for playing them is the Paraguayan harp (Diatonic Harp), known for its charming melodies, driving rhythms, and rich ornamentation. Some of the best know Guaranía pieces include “Recuerdos de Ypacarai” (Memories of Ypacarai) [4] and “Mis Noches sin Ti” (My Nights Without You) [5]. For the Polka, a representative piece is “Pájaro Campana” (Bellbird) [6].

## **3 Musical Ornaments**

Musical ornaments, in music, refer to additional notes added to a melodic line to enhance interest, variety, and expressiveness in a song or musical piece. For string instruments like the harp, some common ornaments are:

- **Glissando:** A glissando is a rapid slide between two or more notes, played fast and in succession. It creates a smooth and sliding effect, producing a seamless transition between pitches.
- **Tremolo:** Tremolo is the rapid reiteration of a single musical tone or the alternation between two different tones, producing a trembling or quivering effect.
- **Arpeggio:** An arpeggio is a broken chord, where the individual notes of a chord are sounded one after the other in a progressive rising or descending order.

## 4 LSTM Model Experiment

In this experiment, the Guaranian piece "Lejania" composed by Herminio Gimenez was utilized. The model was designed with an LSTM layer containing 512 neurons, followed by a 3-neuron dense layer outputs for predicting the pitch, duration of the note, and step time.

The pitch was represented by an integer value ranging from 1 to 128, which corresponded to all possible MIDI note values. The duration of the notes was measured in seconds, while the step represented the time interval between the start of the previous note and the current note, also in seconds. Each note in the sequence was represented by these three values.

For training the model, various experiments were performed with sequences of 25, 12 and 5 notes that was fed into the neural network, which then outputted a prediction for the next note. The objective was to enable the model to learn the patterns and structures present in the Guaranian piece and generate music that resembled the style of the original composition. The following section presents the best results.

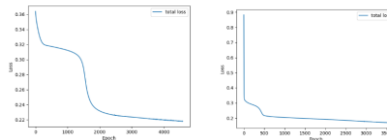
**Table 1.** Additional Hyperparameters.

Hyperparameter	Value
<b>Optimizer</b>	Adam
<b>Epochs</b>	1000
<b>Loss (Pitch)</b>	Sparse Categorical
<b>Loss (Duration, Step Time)</b>	Cross entropy
	MSE

## 5 Results

After training the model with the mentioned Guaranian music and the specified hyperparameters, a generation test was conducted to evaluate the effectiveness of the model in creating similar Guaranian music.

Throughout the training process, the loss function was monitored, and it showed a decreasing trend as the epochs increased, indicating that the model was learning from the data. However, for the 25-note sequence, the resulting MIDI file exhibited sparse notes scattered randomly, failing to form any recognizable melody or musical structure. On the other hand, the 5-note sequence displayed a greater variety in note durations and even included some short notes resembling an ornament known as an *appoggiatura* (a short note before a longer note).



**Fig. 1.** Loss Reduction for 25 and 5-Note Sequences Over Epochs.

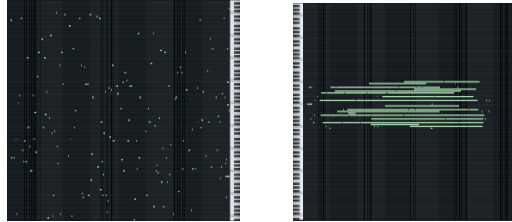


Fig. 2. Results for 25 and 5-Note Sequences.

## 6 Exploring Solutions for Improved Music Generation

One of the key challenges in deep learning for music generation is the requirement of a large dataset to capture all the intricate details, including the ornaments. These details are often scattered throughout the music, making it essential to have numerous examples for the machine to learn and replicate accurately.

Another challenge lies in the model used for music generation. The current model only considers a limited size note input during the generation process, neglecting the context of the entire music piece. This limitation can make it difficult for the model to capture and generate the intricate nuances, such as ornaments.

To tackle these challenges, we are exploring alternative models, including transformers or models with attention mechanisms, and expanding the dataset with additional Guaranía music.

## 7 Future Work

In the future, we plan to explore and compare the effectiveness of alternative models in music creation. We will investigate how different architectures perform in generating music, including harp music from the 19th and 20th centuries and Japanese koto music.

## References

1. Krüger Bridge S. 2019. Paraguay: History, Culture, and Geography of Music Sturman JL. The SAGE Encyclopedia of Music and Culture :1661-1663 Sage.
2. Krüger, S (2019) Paraguay: Modern and Contemporary performance practice. In: Sturman, JL, (ed.) Sage International Encyclopedia of Music and Culture. Sage.
3. 「The Rise of the Guaranía a musical style that is Paraguay's own」, <https://soundsandcolours.com/articles/paraguay/the-rise-of-the-guarania-a-musical-style-that-is-paraguays-own-4321/>, last accessed 2023/08/02.
4. “Recuerdos de Ypacarai”., <https://youtu.be/uYjOtUimp20>, last accessed 2023/08/02.
5. “Mis Noches sin Tí”., <https://youtu.be/ddD57JxMHaA>, last accessed 2023/08/02.
6. “Pájaro Campana “. <https://www.youtube.com/watch?v=RrsyNPxqnCI>, last accessed 2023/08/02.

# Automatic Music Composition System to Enjoy Brewing Delicious Coffee

Noriko Otani<sup>1</sup>, So Hirawata<sup>1</sup>, and Daisuke Okabe<sup>1</sup> \*

Faculty of Informatics, Tokyo City University  
otani@tcu.ac.jp

**Abstract.** It is important to control the amount of hot water poured and the timing of each operation when brewing coffee by the drip method. Since this is difficult for inexperienced people, some kind of guidance is needed to brew delicious coffee to their liking. We aimed to enable users to enjoy brewing delicious coffee regardless of their coffee brewing knowledge or experience, and proposed a method to automatically generate music to play during coffee brewing. In the demonstration, participants can generate coffee brewing music based on their personal sensibilities, brew coffee while listening to the generated piece and taste it.

**Keywords:** Coffee Brewing, Music Composition, Symbiotic Evolution

## 1 Background

The taste of coffee depends not only on the type of bean, its condition, the grinding method, and the temperature of the hot water, but also on the brewing method. There are several brewing methods, such as drip, immersion, and pressurized, but drip is the most popular as an easy way to enjoy delicious coffee. Generally, a small amount of hot water is first poured over all the coffee powder in the dripper to steep it. Then the two processes are repeated, slowly pouring hot water to about halfway up the dripper and waiting for the water to fall into the cup. Because it is difficult for inexperienced people to control the amount of hot water poured and the timing of each operation, some kind of guidance is needed to brew delicious coffee to their liking.

Music is known to have various effects, such as inducing emotions, creating an atmosphere, activating the brain, and enhancing the effects of exercise. However, the impressions and feelings that arise when listening to music are different for each individual, and therefore, the sources distributed to the general public may not be effective for each individual. The way one feels may vary depending on one's mood or situation when listening to music, and one may get bored and want to listen to different music after listening to the same music for a long period of time.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS KAKENHI Grant Number 23K11384 and the Faculty of Informatics, Tokyo City University.



**Table 1.** Music structure

Order	Operation	BPM	Part ID	Timbre	No. of bars
1	Lift the kettle	49	<i>a</i>	Marimba	1
2	Aim the spot to pour		<i>b</i>	Marimba	1
3	Pour hot water (1st)		<i>d</i>	Piano	4
4	Wait		<i>f</i>	Bass	5
5	Aim the spot to pour	54	<i>b</i>	Marimba	1
6	Pour hot water (2nd)		<i>e</i>	Piano	5
7	Wait		<i>g</i>	Bass	3
8	Aim the spot to pour		<i>b</i>	Marimba	1
9	Pour hot water (3rd)		<i>e</i>	Piano	5
10	Remove the dripper		<i>c</i>	Marimba	3

In this context, we aimed to enable users to enjoy brewing delicious coffee regardless of their coffee brewing knowledge or experience, and proposed a method to automatically generate music to play during coffee brewing. The following describes the proposed method and the details of the demonstration.

## 2 Music Composition for Brewing Coffee

### 2.1 Music Structure

In order to find out the focus of the coffee brewing process, an experiment was conducted with the expert with the Coffee Sommelier Certification and the inexperienced persons. They were instructed to brew coffee freely, and the process was filmed from above, in front of, and to the left of them. Analysis of the videos revealed that the number of pours, the number of spout rotations, and the time required for each operation were important.

In order to guide operations similar to those of the expert, the structure of the music was determined as shown in Table 1. The number of spout rotations in each pouring was equal. Since the first pouring took longer time than the second and third pourings, the tempo is changed after the fourth operation. It is assumed that the spout makes one rotation in one beat. The BPMs are calculated from the average rotation time, and the number of bars required for each operation is determined.

Part *a - f* are the parts consisting of a piece of music. The same part is assigned to the operation with the same content and the same number of bars. To make it easy to recognize the start and end of the pouring and the end of the waiting process, the timbre of each part should be different. A bell is played on a vibraphone with the key note two octaves higher for one beat to signal the end of each operation. The bell should ring on the beat before the end of pouring, so that the kettle can be returned after the bell rings during pouring.

### 2.2 Composition Method

The parts *a - g* are generated by the proposed method, that is based on the composition method adapting to individual sensibilities[2]. The composition flow of the proposed

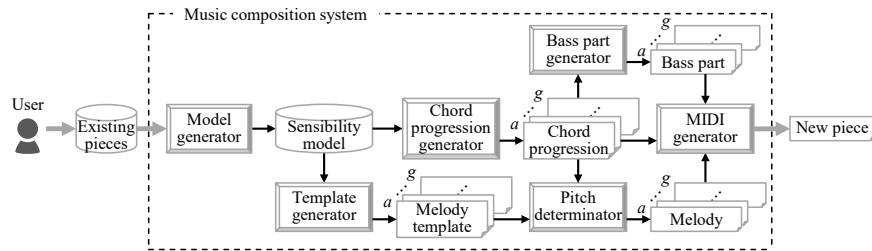


Fig. 1. Composition flow

method is illustrated in Fig. 1. Some existing pieces are needed as the training dataset. The pieces included in the training dataset and the parts generated by the proposed method consists of a chord progression, a melody and a bass part with a 4/4 time signature. The basic duration of a note or rest in a melody is defined as that of a sixteenth note. The basic duration of a chord in a chord progression is defined as that of a quarter note. The bass part is a sequence of eighth notes with the lowest pitch of the chord in the chord progression.

First, existing pieces are specified as the training dataset according to the targeted person's sensibilities, aims, and/or the purpose of the intended composition. Sensibility models for the chord progression and the melody are obtained based on the training dataset. In the next step, chord progressions  $a - g$  and melody templates  $a - g$  that adapts to the sensibility models and the basic music theory are generated depending on the numbers of bars for the parts  $a - g$ . A melody template indicates the time at which each sound in the melody is played, the length of time each sound is played in succession, and the up-and-down stream of the melody line. In other words, a melody template is a melody without the pitch of each note. Subsequently, the pitch of each note in melodies  $a - g$  is determined using the melody templates  $a - g$  and chord progressions  $a - g$ . Finally, bass parts  $a - g$  are generated and combined with the chord progressions  $a - g$  and melodies  $a - g$  to form the parts  $a - g$ . Arrange the parts  $a - g$  in the order shown in Table 1, set the BPM, add bells, and output in the form of a MIDI file.

Symbiotic evolution[1], an evolutionary computation algorithm that results in a fast, efficient search and prevents convergence to suboptimal solutions, is applied to generate a chord progression and melody template. It is characterized by maintaining two separate populations: a partial solution population, the individuals of which represent partial solutions, and a whole solution population, the individuals of which are combinations of individuals in the partial solution population and represent whole solutions. In the former population, partial solutions that may be components of the optimal whole solution are generated. In the latter population, combinations of the partial solutions that may be the optimal solution are generated.

In generating chord progressions and melody templates, a bar is represented as a partial solution and a part is represented as a whole solution. The fitness of a whole solution individual is calculated based on the degree of adaptability to the sensibility models and the basic music theory. The fitness of a partial solution individual is the fitness of the best whole solution individual that refer to the partial solution individual.



Fig. 2. Screens in the composition system

### 2.3 Effectivity

Experiments were conducted with three inexperienced people. First, they were instructed on how to brew coffee, then they brewed the coffee at their own pace and tasted the brewed coffee. Next, they selected some pieces as training data according to their own sensibilities. Using the proposed method, a new piece of music was generated for brewing coffee. They brewed the coffee while listening to this piece and tasted the brewed coffee. The total dissolved solids and extraction yield of the brewed coffee were measured, and the values of the second cup were closer to the ideal for all participants. Subjective taste ratings were also higher for the second cup, and listening to the piece while brewing coffee was also well received.

## 3 Demonstration

In the demonstration, the participants can use the system in which the proposed method is embedded to generate coffee brewing music for themselves based on their personal sensibilities. Examples of the system screen are shown in Fig.2. When they select some pieces for training and press the “Compose” button on the screen of Fig.2(a), the screen of Fig.2(b) will appear in about 10 seconds after passing through the progress indicator screen. The selected pieces are displayed on the screen and they can listen to the generated piece. If they wishes, they can actually brew coffee while listening to the generated piece and taste it.

## References

1. Moriarty, D. and Miikkulainen, R.: Efficient Reinforcement Learning through Symbiotic Evolution, *Machine Learning*, 22, 11–32 (1996)
2. Otani, N., Okabe, D., Numao, M.: Generating a Melody Based on Symbiotic Evolution for Musicians’ Creative Activities, *Proceedings of the Genetic and Evolutionary Computation Conference 2018*, 197–204 (2018)

# Expressor: A Transformer Model for Expressive MIDI Performance

Tolly Collins and Mathieu Barthet

Centre for Digital Music, Queen Mary University of London  
tollycollins@gmail.com, m.barthet@qmul.ac.uk

**Abstract.** The Transformer neural network has been used to generate new music with expressive features with significant success, but it has not previously been applied to generate an expressive performance of an existing score. We propose Expressor, a Transformer model with a novel encoder-decoder skip connection design for expressive performance rendering. The model shows promise in applying coherent temporal and dynamics expressive features based on human performance. We develop a new tokenisation scheme to overcome challenges in representing interrelated expressive performance features.

## 1 Introduction and Related Work

We outline here a work in progress on how deep learning can be used to alter temporal and dynamics properties of a MIDI score to add similar expressive properties to those present in a human performance. Previous studies have applied Recurrent Neural Networks to model expressive timing [1, 2], and while they found success in modelling periodic expressive events, they performed less well for isolated events used to convey emotion or meaning. Transformers have shown promise in music generation tasks [3], where they have been more adept at modelling the longer-term structural properties of a musical score. We propose Expressor, a new Transformer model for expressive performance rendering with skip connections between corresponding encoder and decoder layers, and a new tokenisation scheme to represent expressive features. To our knowledge, this is a novel architecture and we find that the skip connections improve performance over the original design.

## 2 Methodology

**Dataset.** We use the ASAP dataset [4], with 1067 professionally-performed classical piano pieces with paired performed and unperformed MIDI versions.

**Tokenisation.** We use a compound word tokenization [5], with a metric rather than absolute timing representation inspired by the REMI approach [6]. The perceptual, hierarchical and interdependent nature of expressive attributes poses a significant



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

challenge in determining ground truth values. For example, note onset deviations are relative to local tempo, but tempo is itself a subjective measure that continually fluctuates over time. Furthermore, preceding notes may themselves deviate from precise metrical timings. Our solution is to provide the model with ground truths calculated relative to a piecewise constant tempo function with jumps at beat times (see Fig. 1). For example, timing deviations are calculated as the difference (as a proportion of beat length) between the actual note onset and expected onset given by a linear proportion of the beat length on from the start time of the beat. Expressive features for dynamics follow a similar hierarchical classification [7], and our model also considers articulation by varying note length relative to the notated version to produce more staccato or legato phrasings.

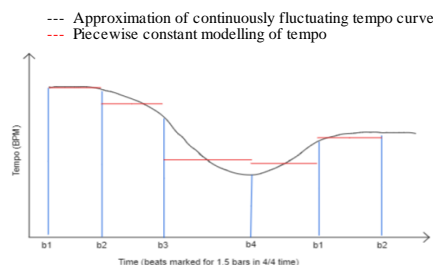


Fig. 1. Illustration of tempo modelling.

Table 1. Token Descriptions

Name	Type	Description
Type	Meta	Determines if a word is <i>meta</i> (for start- and end-of-sequence), <i>metric</i> (occurring at the start of each beat) or <i>note</i> (each word corresponds with exactly one note).
Beat	Metric	Hold the number of the beat in a bar.
IBI	Metric	Inter-beat interval. Express the tempo as a quantized beat length in seconds.
Local vel. band	Metric	Coarse measure of MIDI velocity.
Local IBI	Metric	The median IBI over a number of beats spanning closest to 4 seconds, centred on the beat relating to the given metric word.
Pitch	Note	The MIDI pitch number of a note (integer between 1 and 127).
Start	Note	Score-based start position of a note relative to the beat, given as a proportion of the beat (quantized to 1/60 beats).
Duration	Note	Number of beats a note is designated to last for in the score, quantized to 1/60 beats.
Rubato	Note	Designates any beat marked with rubato in the ASAP dataset annotations, meaning that the music departs from standard metrical timing during this beat.
Timing flux	Metric	Mean deviation in onset of notes in a beat from the precise division of the IBI.
Dynamic flux	Metric	The average number of absolute standard deviations for the velocity of each note in a given beat from the local mean.
Accent	Note	Designed to represent an accent score notation. Calculated as any performed note having a velocity of more than 2 standard deviations above the local mean.
Staccato	Note	Whether or not a note should last for < 25% of the expected IBI proportion.
Local vel. mean	Metric	The mean note velocity over a given number of beats, centred on the current beat.
Tempo difference	Metric	Difference between a beat's IBI and the local tempo, measured in BPMs.
Articulation	Note	How long a note will last for, relative to the expected duration taken from the score. The value is a number of beats, quantized to a given sub-interval.
Timing deviation	Note	The sub-interval of a beat by which the note onset differs the score.
Vel. difference	Note	Difference between a note's velocity and the local velocity mean.

**Model.** We use the Transformer with Linear Attention design [8] in an encoder-decoder format. The aim is for the encoder to create a representation of score-specific structural information such as note pitches, medium-term tempo and general dynamics. We design for additional attribute tokens input directly to the encoder output latent space, allowing for control to be imposed on the generation akin to score markings guiding a pianist. The decoder layers then output words containing tokens with expressive properties such as *timing deviation* per note or *local mean velocity*. We also introduce the

use of skip connections (see Fig. 2) between the outputs of individual encoder layers and the attention mechanism in the corresponding decoder layer. The idea is to encourage corresponding hierarchical representations of the information throughout the encoder and decoder stacks.

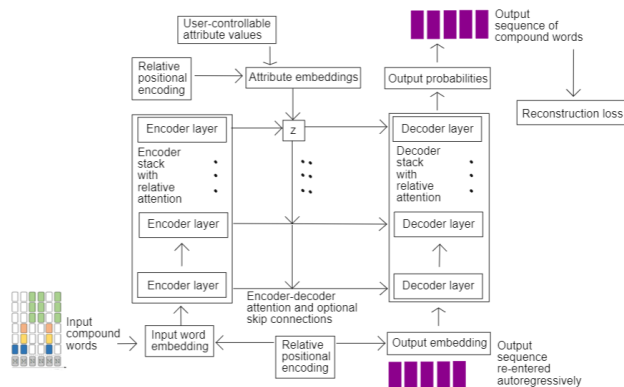


Fig. 2. Expressor Architecture

Output tokens are then combined with the input information to render back into MIDI format. This results in a version of the original piece that incorporates expressive performance features. As each compound word is made up of separate tokens, the network decoder is followed by one head for each output token which consists of a separate feed-forward network to map latent space vectors to logits for the relevant values in the token’s vocabulary. The network can therefore be viewed as a multi-task network, and the loss is made up of a linear combination of the reconstruction losses for each head.

### 3 Results Discussion and Conclusion

**Model.** With hyperparameter tuning, we found the best performing model had encoder and decoders both with dimension 256, 8 layers and 8 attention heads per layer. Fig. 3 shows the results from two training runs with these same model parameters, but one with added skip connections between corresponding encoder and decoder layers. The results suggest that training is improved by the addition of the skip connections.

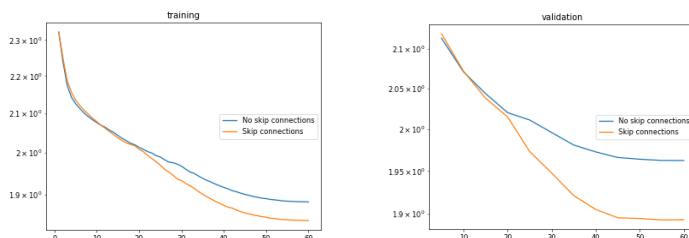


Fig. 3. Training and validation loss curves for identical Expressor models with and without skip connections between the encoder and decoder layers

Intuitively, the skip connections may encourage the model to match hierarchical levels in the music between encoder and decoder stacks.

Music Transformers often use embedding dimensions larger than vocabulary sizes [3]. The use of compound words allows for tailored embedding sizes for each token type, and we found that embedding sizes between 4 and 16 performed better than larger values. As described in Table 1, many of the measures used in Expressor represent a quantized linear scale, such as IBI or pitch, and although there may be some higher-dimensional relationships such as the chroma dimension for pitch, in general this data should not require large numbers of dimensions to represent.

**Qualitative evaluation and discussion.** Selected audio examples can be found at the link below<sup>1</sup>. While we have yet to conduct independent listening tests, we suggest these demonstrate that the model shows considerable promise in mapping general expressive performance features onto a MIDI score in a realistic manner. The features often follow locally coherent patterns such as *crescendi* or *staccato*. We did notice that the expressive features could often be inappropriate in relation to the musical period or the commonly interpreted emotional content. Additional tokens such as *composer* or *period*, alongside planned latent space semantic guidance tokens could help. We have yet to analyse statistically how well the model relates expressive features to structural features in the score (such as musical phrasing or unexpected harmonic moments), but our intuition is that pre-training the structural modelling of the encoder section may improve performance in this area. We also intend to conduct an ablation study to further understand the impact of the encoder-decoder skip connections.

## References

1. Shi, Z.: Computational analysis and modeling of expressive timing in Chopin Mazurkas. In: Proc. of the 22nd Int. Society for Music Information Retrieval Conf., Online (2021).
2. Jeong, D., Kwon, T., Kim, Y., Lee, K., and Nam, J.: VirtuosoNet: A Hierarchical RNN-based system for modeling expressive piano performance. In: 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands (2019).
3. Wu, S.-L., and Yang, Y.-H.: MuseMorphose: Full-Song and Fine-Grained Music Style Transfer with One Transformer VAE. IEEE (2021). arXiv:2105.04090v3
4. Foscarin, F., McLeod, A., Rigaux, P., Jacquemard, F. and Sakai, M.: ASAP: a dataset of aligned scores and performances for piano transcription. In: 21st International Society for Music Information Retrieval Conference, Montréal, Canada (2020).
5. Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C., and Yang, Y.-H.: Compound Word Transformer: Learning to Compose Full-Song Music Over Dynamic Directed Hypergraphs. In: Proc. of the AAAI Conference on Artificial Intelligence (2021).
6. Huang, Y.-S., and Yang, Y.-H.: Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In: Proc. 28th Int. Conf. on Multimedia (2020).
7. Oore, S., Simon, I., Dieleman, S, Eck, D.: This time with feeling: Learning expressive musical performance. In: Neural Computing and Applications 32.4, pp. 955–967 (2018).
8. Katharopoulos, A., Vyas, A., Pappas, N. and Fleuret, F: Transformers are RNNs: Fast autoregressive Transformers with linear attention. In: Proc. Int. Conf. Machine Learning (2020).

---

<sup>1</sup> [https://drive.google.com/drive/folders/1JwENHB5iOSYsl5FPYeeoWW2q46PESTfy?usp=share\\_link](https://drive.google.com/drive/folders/1JwENHB5iOSYsl5FPYeeoWW2q46PESTfy?usp=share_link)

# Real-Time Piano Accompaniment Using Kuramoto Model for Human-Like Synchronization

Kit Armstrong, Ji-Xuan Huang, Tzu-Ching Hung, Jing-Heng Huang, Yi-Wen Liu

National Tsing Hua University  
s110061892@m110.nthu.edu.tw

**Abstract.** Composition and performance in the Western classical tradition represent fields of highly sophisticated artistic endeavor which have not been mastered by AI. Machine performance, though it has become an indispensable tool to composers for creating audio mock-ups, does not appear on the concert stage, where human musicians perform. An eventual goal is a machine that plays a part of a score in real time together with live musicians playing other parts, with results indistinguishable from human efforts. This work focuses on the collaborative aspect of music-making, starting with a behavior-capturing experiment that investigates how musicians adapt their playing to that of others in an ensemble. Using the empirical data thus obtained, we train a Kuramoto model for synchronization which we adapted to the context of score-based collaborative musical performance.

**Keywords:** Classical music, interpretation, chamber music, expressive performance, automatic accompaniment, rhythmic synchronization, Kuramoto model

## 1 Introduction

Within the Western classical paradigm of Composer-Performer-Listener, the composer creates a score, and the performers convert it into a performance that the listeners can experience. Both composition and performance have not been fully mastered by AI. For machine composing in general, years of research have developed well-publicized results (e.g. [1–3]), leading to commercial applications. Machine performance is simultaneously more commonplace and yet more distant: it has become an indispensable tool to composers for creating audio mock-ups via sound synthesis tools; however, machines or virtual musicians [4] do not commonly appear on the concert stage to perform alongside with human musicians.

The richness of classical music has much to do with the ways in which performers can create different experiences for the listener out of the same compositions. To do so, an ensemble of performers, individually and simultaneously interpreting their parts of the score, must synchronize with each other in real time. An AI can approach this by starting with *audio transcription*, whose purpose is to provide a machine-friendly representation of musical acoustic information [5, 6], and *score following*, the process



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



by which the machine takes a performance and determines the point in the score to which it is most likely to correspond [7, 8]. Our motivation is to allow a machine, once in possession of these elements, to exhibit human-like behavior acting in real-time as a fellow musician of an ensemble.

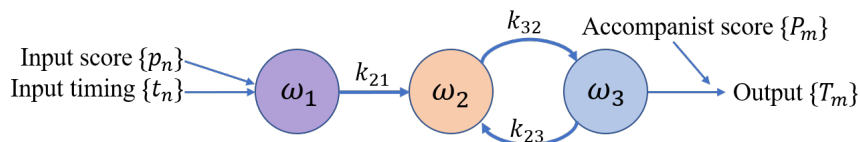
In this research, we consider an environment with a single instrument, the piano, to be played jointly by one person and our model. We focused particularly on synchronization in the time-domain, with the goal being to simulate human-like behavior, not to generate a perfectly aligned accompaniment.<sup>1</sup>

## 2 Model Design

In our approach, we assume that both musicians play exactly according to a known score, and treat each musician’s output as a sequence of discrete “note-on” and “note-off” events<sup>2</sup>, which are relayed perfectly to the other musician as soon as they occur. By relating the received events to the score, our model ascertains the timing of the other musician and adjusts in real-time the timing of its own future output.

### 2.1 Kuramoto Model

Previous work on other instances of synchronization have provided us with inspiration regarding the specific mechanism of the adjustment. We have taken the Kuramoto model [9, 10] as a basis. Our implementation follows Heggli et al. [11], who adapted Kuramoto’s approach to model human synchronization behavior in a situation where two people were faced with the task of tapping in unison.



**Fig. 1.** Adapted Kuramoto model, playing an  $m$ -note “output” part together with a  $n$ -note “input”.

The model consists of 3 oscillators  $\omega_1, \omega_2, \omega_3$  that are coupled as in Fig. 1. Their positions are determined by the coupling equations:

$$\frac{d\theta_i(t)}{dt} = \sum_{j \neq i} k_{ij} \sin(\theta_j(t) - \theta_i(t)) + \Omega_i(t), \quad (1)$$

for  $i, j \in \{1, 2, 3\}$ , where  $\theta_i$  represent the positions of the respective oscillators,  $\Omega_i(t)$  their intrinsic speed, and  $k_{ij}$  the coupling coefficients (with only  $k_{21}, k_{23}, k_{32}$  being non-zero, as per Fig. 1).

<sup>1</sup> It is in this regard that our approach distinguishes itself from applications using score-following to automatically accompany a human player.

<sup>2</sup> Our focus is on the piano; for other instruments, this assumption would be less workable.

## 2.2 Adapting the Kuramoto Model to Musical Scores

Our oscillator model naturally deals with continuous movement, but a musical score, according to our assumptions, consists of discrete events. We reconcile the two paradigms as follows: we define each beat, in the traditional musical sense, as corresponding to a rotation of the oscillator through  $2\pi$ . The score gives us a sequence  $\{p_n\}$  of positions (in beats) at which note onsets in the input part are designated. Denoting  $t_n$  the time at which the  $n^{\text{th}}$  note is actually received, we set  $\theta_1(t_n) = 2\pi p_n$ . By linear interpolation, we construct a continuous function  $\theta_1(t)$  which encapsulates the timing information of the other player.

Analogously, we obtain the output timings  $\{T_m\}$  by solving  $\theta_3(T_m) = 2\pi P_m$ , with  $\{P_m\}$  being the beat positions of the output part's notes as given by the score.

In a real-time context, the values of  $\theta_i(t)$  for all  $t$  are not known beforehand. Thus it is necessary to perform the above calculations for each interval of  $t$  at the moment when the information for that interval becomes available. It seemed reasonable to introduce a parameter  $t_r$  reflecting reaction time, that is, a delay between receiving information from the input and performing the calculations based on it for adjusting the output.

We denote by  $\{T_m^*\}$  the timings of the output events resulting from this process.

## 3 Implementation

Having set our objective as human-like musical collaboration, the first step was to investigate human behavior in a similar controlled environment. We prepared a series of MIDI recordings containing the melody of well-known music pieces, and invite subjects to accompany these recordings.<sup>3</sup> To train our model to simulate a subject's behavior, we input the same MIDI recording and search for the parameters  $k_{ij}, t_r$  that produce the output  $\{T_m^*\}$  most similar to the performance of the subject, which we denote  $\{S_m\}$ :

$$\arg \min_{k_{ij}, t_r} \sum_m (T_m^* - S_m)^2 \quad (2)$$

We proceed to set up the model for accompanying a human subject in real time. First we enter the score of the chosen music piece (i.e.  $\{p_n\}$  and  $\{P_m\}$  as per Fig. 1). With this information and the MIDI input of the subject, which provides  $\{t_n\}$ , the model determines  $\theta_1(t)$ , from which it calculates  $\theta_2(t), \theta_3(t)$  by Eq. 1, and finally  $\{T_m^*\}$ .

### 3.1 Refinements

Error-handling is outside the scope of this research, which focuses on collaborative aspect of music-making under the assumption of following the score exactly. However, we found during a pilot experiment that having to start over whenever one mistakenly touched a note was unnecessarily frustrating and time-wasting. Therefore, to make our model practically usable, we made it to ignore or automatically rectify common errors.

Furthermore, we implemented a method for following the human player's dynamics based on a running average of velocities of recent input notes.

<sup>3</sup> We use the terms "melody" and "accompaniment" loosely here; the described procedure may be applied to any piece that can be separated into two parts to be played simultaneously.

## 4 Conclusion

Taking inspiration from models of the synchronization of biological phenomena, we approached the subject of musical performance from the perspective of its collaborative aspects. Our model attempts to emulate basic interactions among ensemble members, which play an important role in classical music-making. As such, it ideally would complement score-interpreting AIs, enabling machines to listen not only to identify cues but to respond and enter musical dialogues in a human-like way.

We have observed playing together with the model to be satisfying to a surprising degree. We performed a sort of Turing test, in which subjects did not know whether they were being accompanied by a human or by our model, resulting in 49/83 (59%) correct guesses, and 18/38 (47%) by listeners present. We consider this an encouraging basis for exploring the possibilities of human-AI collaboration at an increasingly high level of musicality.

## References

1. G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: a survey, a critical view and future prospects," in *Soc. Artificial Intelligence and Simulation of Behaviour Symp. Musical Creativity*, Edinburgh, UK, 1999, pp. 110–117.
2. K. Bryden, "Developing metrics for evolving a musically satisfying result from a population of computer generated music compositions," in *Proc. IEEE Symp. Computational Intelligence in Image and Signal Processing*, Honolulu, HI, USA, 2007, pp. 171–176.
3. C.-H. Liu and C.-K. Ting, "Computational intelligence in music composition: A survey," *IEEE Trans. Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 2–15, 2017.
4. Y.-J. Lin, H.-K. Kao, Y.-C. Tseng, M. Tsai, and L. Su, "A human-computer duet system for music performance," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 772–780.
5. L. Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Signal Language Proc.*, vol. 23, no. 10, pp. 1600–1612, 2015.
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. R. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Commun. ACM*, vol. 49, pp. 38–43, 2006.
8. R. Agrawal and S. Dixon, "A hybrid approach to audio-to-score alignment," *arXiv:2007.14333*, 2020.
9. Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. New York, NY: Springer-Verlag, 1984.
10. J. A. Acebron, L. L. Bonilla, C. J. Vicente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of Modern Physics*, vol. 77, no. 1, pp. 137–185, 2005.
11. O. Heggli, J. Cabral, I. Konvalinka, P. Vuust, and M. Kringelbach, "A Kuramoto model of self-other integration across interpersonal synchronization strategies," *PLoS Comput. Biol.*, vol. 15, no. 10, e1007422, 2019.

## Intuitive Control of Scraping and Rubbing Through Audio-tactile Synthesis

Mitsuko Aramaki, Corentin Bernard, Richard Kronland-Martinet, Samuel Poirot, Sølvi Ystad

Aix Marseille Univ, CNRS, PRISM (Perception, Representations, Image, Sound, Music),  
31 Chemin J. Aiguier, 13402 Marseille Cedex 20, France

{name}@prism.cnrs.fr

**Abstract.** Intuitive control of synthesis processes is an ongoing challenge within the domain of auditory perception and cognition. Previous works on sound modelling combined with psychophysical tests have enabled our team to develop a synthesizer that provides intuitive control of actions and objects based on semantic descriptions for sound sources. In this demo we present an augmented version of the synthesizer in which we added tactile stimulations to increase the sensation of true continuous friction interactions (rubbing and scratching) with the simulated objects. This is of interest for several reasons. Firstly, it enables to evaluate the realism of our sound model in presence of stimulations from other modalities. Secondly it enables to compare tactile and auditory signal structures linked to the same evocation, and thirdly it provides a tool to investigate multimodal perception and how stimulations from different modalities should be combined to provide realistic user interfaces.

**Keywords:** sound synthesis, invariant signal structures, multimodal perception, tactile perception, continuous friction interactions

### 1 Introduction

Previous results in the field of multimodal perception have provided examples of strong perceptual influences between modalities. One well-known example is the McGurk effect in which visual stimuli influence speech perception [8]. More recent studies revealed that sounds can modify the perception of a visual trajectory and even the gestural reproduction of the visual shape [12]. In the case of touch perception, several studies have revealed a strong influence of auditory stimuli on perceived textures [2, 5, 6, 7, 10, 11].

In the present study we explore such multimodal interactions in the light of our previous works on intuitive sound control that describes the sound as the result of an action on an object. This approach presumes the existence of sound invariants responsible for the evocation of sound events [4], and has led to a synthesizer that makes it possible to



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

control sounds from semantic labels that describe the action (rubbing, scratching rolling) and the object (material, shape, size, ...). Continuous control between the different evocations makes it possible for the users to freely navigate between different actions hereby creating both realistic and virtual sound events [1,3,9].

As a first approach to multimodal synthesis, we focus on evocations of continuous friction interactions, in particular rubbing and scraping, to investigate whether tactile invariants of these actions exist and whether they resemble the corresponding auditory invariants. In the following section we describe the synthesis process of the tactile stimulation, the experimental setup and some preliminary results of ongoing perceptual tests.

## 2 Synthesis of Auditory and Tactile Stimulations Evoking Continuous Friction Interactions

As a first approach to investigate perceptual invariants for tactile structures, we focus on the evocation of two different continuous interactions namely scraping and rubbing. In the auditory domain it has been shown that these actions can be simulated by successive impacts (see Fig. 1) with different temporal intensities [13]. The impact distribution is smoother for rubbing than for scratching since scratching is considered as an action in which the interaction with each surface irregularity is encountered one after another and more intensely than in the case of rubbing. This model was perceptually validated by Conan et al [3] and confirmed that impact distributions are associated to the auditory invariant allowing for the distinction between scratching and rubbing.



Fig. 1. Phenomenological model of continuous interactions

Would this also be the case in the tactile domain? To answer this question, we designed a synthesis model based on the same features as in the auditory modality, i.e. mean and standard deviations of the amplitudes and the temporal distance between successive peaks, to investigate evocations of rubbing and scratching using an actuator attached to a pen. Then we conducted a perceptual test in which subjects were asked to explore the surface of a graphic tablet and to determine (on a continuous cursor) whether the sensation evoked scraping or rubbing. The experimental protocol is described in the next section.

### 3 Perceptual Evaluations

Sixteen subjects evaluated 96 evoked continuous interactions induced by auditory and tactile stimulations. They wore anti-noise headphones when evaluating the tactile stimuli. During the tactile evaluations, they were asked to hold a pen equipped with the actuator and to explore a surface of a graphic tablet. After the exploration they evaluated the evocation on a continuous one-dimensional scale between the (French) words “gratter” (scratch) and “frotter” (rub). While the preliminary results confirmed previous findings related to the impact distribution as the most influent parameter on the evocation of rubbing and scratching in the auditory domain [3], this parameter did not turn out to have a significant influence in the tactile domain. On the other hand, amplitude variations tended to be more important in the tactile domain and had a significant influence on the perceived action. Scratching evocations were associated with strong amplitudes while the weakest amplitudes were associated with rubbing.

### 4 Audio-tactile Synthesizer

The current study suggests that perceptual invariants differ in the case of auditory and tactile perception. In the case of simulations of continuous friction interactions, temporal variations are essential in the auditory domain while amplitude variations seem to play a greater role in the tactile domain. The proposed demo consists of a multimodal synthesizer calibrated according to the previous perceptual results that enables participants to explore auditory and tactile signal invariants and to combine the evocations with auditory evocations of objects (see Fig. 2). The user is invited to wear headphones (for auditory stimulations) and to hold a pen equipped with the actuator (for tactile stimulations) coupled with a tablet. A computer displays a graphical interface on which the user can choose the type of interactions (rubbing or scratching) in a continuous way.



Fig. 2. Set up of the synthesizer device

**Acknowledgements** This work was partly financed by the French National Research Agency (ANR) in the case of the France Relance program (C. Bernard) and the COMMUTE ANR-22-CE33-0009 project and the by Institute of Language, Communication and the Brain (ILCB)/Center of Excellence on Brain and Language (BLRI) Grant Nos. ANR-16-CONV-0002 (ILCB) and ANR-11-LABX-0036 (BLRI), the Excellence Initiative of Aix-Marseille University (AMIDEX). We would like to thank Raphaël Vancheri for his precious contribution to the perceptive evaluations.

## References

1. Aramaki, M., Besson, M., Kronland-Martinet, R., Ystad, S. Controlling the Perceived Material in an Impact Sound Synthesizer. *IEEE Transactions on Audio, Speech, and Language Processing* 19(2):301–314. (2012)
2. Bernard, C., Monnoyer, J., Wiertelowski, M., Ystad, S. Rhythm perception is shared between audio and haptics. *Scientific Reports*, Nature Publishing Group, 2022, 12, 10.1038/s41598-022-08152-w (2022)
3. Conan S., Thoret E., Aramaki M., Derrien O., Gondre C., Kronland-Martinet R., Ystad S. An Intuitive Synthesizer of Continuous-Interaction Sounds: Rubbing, Scratching, and Rolling. *Computer Music Journal*, vol. 38(4), pp. 24-37. (2014)
4. Gibson, J. J. *The Ecological Approach to Visual Perception*. Boston, Massachusetts: Houghton Mifflin. (1979)
5. Guest, S., Catmur, C., Lloyd, D., Spence, C.. Audiotactile interactions in roughness perception. *Exp Brain Res.* 146(2):161-71. doi: 10.1007/s00221-002-1164-z. PMID: 12195518. (2002)
6. Jousmäki, V. and Hari, R.. Parchment-skin illusion: sound-biased touch. *Curr Biol.* Mar 12;8(6):R190. doi: 10.1016/s0960-9822(98)70120-4. PMID: 9512426. (1998)
7. Lederman, S.J.. Auditory texture perception. 8(1):93-103. doi: 10.1068/p080093. PMID: 432084. (1979)
8. McGurk, H., MacDonald, J. Hearing lips and seeing voices, *Nature*, vol. 264, n° 5588, p. 746–748 (1976)
9. Poirot, S., Bilbao, S., Aramaki, M., Ystad, S. and Kronland-Martinet, R., A Perceptually Evaluated Signal Model: Collisions Between a Vibrating Object and an Obstacle. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2338-2350, doi: 10.1109/TASLP.2023.3284515 (2023).
10. Rocchesso, D., Monache, S., Papetti, S., Multisensory texture exploration at the tip of the pen. *International Journal of Human-Computer Studies* (2016)
11. Romano, Kuchenbecker. Creating Realistic Virtual Textures from Contact Acceleration Data. *IEEE Trans Haptics.* 2012 Apr-Jun;5(2):109-19. doi: 10.1109/TOH.2011.38. PMID: 26964067. (2012)
12. Thoret E., Aramaki M., Bringoux L., Ystad S., Kronland-Martinet R. Seeing circles and drawing ellipses : when sound biases reproduction of visual motion. *PloS One*, 11(4) :e0154475, doi.org/10.1371/ journal.pone.0154475 (2016)
13. Van den Doel, K., Kry, P., Pai, D., FOLEYAUTOMATIC : Physically-based Sound Effects for Interactive Simulation and Animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 537–544 (2001)

## From jSymbolic 2 to 3: More Musical Features

Cory McKay<sup>1</sup>

<sup>1</sup> Marianopolis College and the CIRMMT  
cory.mckay@mail.mcgill.ca

**Abstract.** This demo will provide participants with the opportunity to experiment with the jSymbolic software, which extracts a broad range of statistical features from digital scores in formats such as MIDI. Participants will be able to use and compare both the current 2.2 release version and the pre-release jSymbolic 3. Past research using jSymbolic in diverse areas of computational musicology and music information retrieval (MIR) will be discussed, involving machine learning and statistical analysis. Participants will be encouraged to engage in dialogue on how jSymbolic might be incorporated into their own research, and on ideas for new features that could be added to the jSymbolic catalogue that would benefit their work. Research focusing on symbolic data as well as multimodal investigations will both be emphasized. jSymbolic is entirely open source.

**Keywords:** Symbolic music; Features; Computational musicology; MIR.

### 1 jSymbolic and its Motivation

Three essential advantages of applying computational methodologies to musicology are: 1) the ability to directly consider and compare corpora consisting of hundreds or thousands of pieces of music, more than would be feasible using manual techniques; 2) movement towards “objectively” analyzing music in ways that filter out at least some of the biases we are all subject to when manually analyzing music; and 3) the creation of opportunities to explore music in novel ways that can reveal musically meaningful insights in areas we might not have thought to consider using traditional techniques.

This demo presents the jSymbolic software, whose primary purpose is to help music researchers and scholars benefit from these advantages. It automatically extracts features (characteristic pieces of information) from digital scores encoded in symbolic file formats such as MIDI or MEI. Each feature describes a clearly defined characteristic of music that can be consistently extracted and compared, as a single number or as a vector of associated values. For example, a “range” feature could be defined as the number of semitones separating the lowest and highest pitches in the music being analyzed. Features extracted from multiple pieces can be aggregated into categories of interest (e.g., composers, genres, regions, etc.), which can themselves be compared collectively. Although jSymbolic allows features to be extracted over smaller windows of



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



set length, typically features are extracted from pieces or major sections (e.g., mass movements) in their entirety, so as to more broadly characterize them.

Once extracted by jSymbolic, features can be used in a variety of ways, such as for training classifiers using supervised machine learning, or for exploratory clustering via unsupervised learning. For example, a classifier could be trained on jSymbolic features to model the compositional styles of various Renaissance composers, and this classifier could then be used to help identify probable authorship of unattributed or controversially attributed works. This approach, often carried out using relatively simple algorithms like support vector machines, is particularly appropriate in situations where there are relatively few extant training exemplars, as is often the case with early music, since deep learning alternatives (which tend to in effect learn their own features from raw musical data) can have too many parameters to perform well in such circumstances.

Another important advantage of engineered features like jSymbolic's is that they are largely musically interpretable. This enables the application of statistical techniques, such as information gain analysis or feature selection (e.g., with genetic algorithms), to determine which features most meaningfully separate different pieces or groups of music; this can be more musicologically important than actual classifications themselves. So, to continue the Renaissance sample use case, one might use jSymbolic features to gain insight into what specifically statistically differentiates the styles of composers like Josquin, de la Rue and Ockeghem, in *musically meaningful* terms.

Feature values can also be examined directly by domain experts if desired. They can be saved as CSV files (or in specialized machine learning-oriented formats), which can then be imported into spreadsheet or data analysis software for study or visualization.

Although much of the jSymbolic research to date has focused on either Western early music or popular music, jSymbolic can also be applied to many other musics, or combined with other types of data (e.g., audio) in multimodal research. All the jSymbolic features can be extracted from any kind of music that can be encoded as MIDI, which permits complex rhythmic structures and pitches outside the Western chromatic scale.

## **2 Previous Work**

jSymbolic was first released in 2006 [1], was included in the multimodal jMIR music research suite in 2010 [2] and the current release version (2.2) was published in in 2018 [3]. jSymbolic has been used as a core part of many published research projects, in areas including popular music genres [2], investigating the origins of Renaissance genres [4], regional style [5], compositional style [6] and multimodal analysis [7].

Of course, jSymbolic is not the only research platform available for computational musicology or symbolic MIR. However, its focus specifically on standardized high-level summary features means it has quite different use cases from other excellent software like Humdrum [8], Music21 [9], pretty\_midi [10], CRIM [11] and MIDI Toolbox [12], which emphasize tasks like retrieving specified instances of local events, or visualizing or manipulating user-specified elements. The alternatives are less suited to the macro statistical analysis that jSymbolic specializes in. Notably, Music21 can extract features, but mostly just a subset of ported-over jSymbolic 1.2 features.

### 3 Details About jSymbolic 2.2

The current release version (2.2) of jSymbolic extracts 246 unique features, which total to 1497 feature values when feature vectors are expanded. This feature catalogue is designed to be diverse, so that it is applicable to as many characteristics of as many types of music as possible. The jSymbolic features can be divided into these groups:

- **Pitch Statistics:** How common are various pitches and pitch classes relative to one another? How are they distributed and how much do they vary?
- **Melodic Intervals:** What melodic intervals are present? How much melodic variation is there? What can be observed from melodic contour measurements?
- **Chords and Vertical Intervals:** What vertical intervals are present? What types of chords do they represent? What kinds of harmonic movement are present?
- **Rhythm:** Information associated with note attacks, durations and rests, measured in ways that are both dependent and independent of tempo. Information on meter and rhythmic variability, including rubato.
- **Instrumentation:** Which instruments are present, and which are emphasized relative to others? Both pitched and non-pitched instruments are considered.
- **Texture:** How many independent voices are there and how do they interact (e.g., parallel vs. contrary motion)? What is the relative importance of voices?
- **Dynamics:** How loud are notes and what kinds of variations in dynamics occur?

jSymbolic has a graphical user interface (Fig. 1) as well as a command line interface and a Java API for those wishing to use jSymbolic via scripting or to integrate it into their own software. There is also a detailed manual, which includes individual feature explanations, and a tutorial with worked examples.

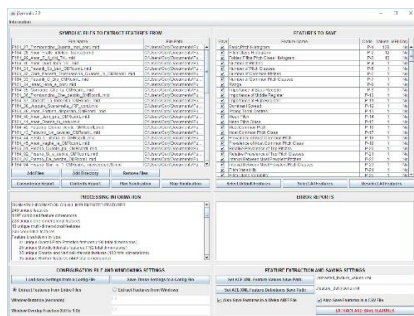


Fig. 1. The jSymbolic 2.2 graphical user interface.

In addition to being a ready-to-use application, jSymbolic is also intended as a platform for building new features, including ones of increasing sophistication built upon existing features. New features are added as plug-ins, and jSymbolic automatically schedules extraction to resolve feature dependencies. Many of the features added since jSymbolic was first released have resulted from consultation and collaboration with musicologists, theorists and MIR researchers, and it is hoped that this will continue.

## 4 Towards jSymbolic 3

The upcoming jSymbolic 3 is currently undergoing final testing and improvement before release. In addition to many miscellaneous usability improvements, it has a substantially expanded feature catalogue of 533 unique features and 2040 feature values in total. Of particular interest, these include a new *n-gram* group of features, which extract information from aggregated sequences of musical events, thus giving jSymbolic more insight into local context. Features are extracted from three types of n-grams: melodic interval, vertical interval and rhythmic.

During this demo, participants will be given the first public hands-on look at jSymbolic 3, and will be able to compare it with jSymbolic 2.2.

jSymbolic, its code and documentation are all available at <http://jmir.sourceforge.net> (version 2.2) and at <https://github.com/DDMAL/jSymbolic2/> (version 3 development).

## References

1. McKay, C., Fujinaga, I.: jSymbolic: A feature extractor for MIDI files. In: Proceedings of the International Computer Music Conference, pp. 302–305. Miami (2006).
2. McKay, C.: Automatic music classification with jMIR. Ph.D. Dissertation. McGill University, Canada (2010).
3. McKay, C., Cumming, J., Fujinaga, I.: jSymbolic 2.2: Extracting features from symbolic music for use in musicological and MIR research. In: Proceedings of the International Society for Music Information Retrieval Conference, pp. 348–354. Paris (2018).
4. Cumming, J., McKay, C.: Using corpus studies to find the origins of the madrigal. In: Proceedings of the Future Directions of Music Cognition International Conference, pp. 38–42. Online (2021).
5. Cuenca, M. E., McKay, C.: Exploring musical style in the anonymous and doubtfully attributed mass movements of the Coimbra manuscripts: A statistical and machine learning approach. *Journal of New Music Research* 50(3), 199–219 (2021).
6. Rodríguez-García, E., McKay, C.: Composer attribution of Renaissance motets: A case study using statistical features and machine learning. In: *The Anatomy of Iberian Polyphony around 1500*, eds. Rodríguez-García, E., d’Alvarenga, J. P., 401–438. Edition Reichenberger, Kassel (2021).
7. Vatolkin, I., McKay, C.: Multi-objective investigation of six feature source types for multi-modal music classification. *Transactions of the International Society for Music Information Retrieval* 5(1), 1–19 (2022).
8. Huron, D.: Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. *Computer Music Journal* 26(2), 11–26 (2002).
9. Cuthbert, M. S., Ariza, C., Friedland, L.: Feature extraction and machine learning on symbolic music using the music21 toolkit. In: Proceedings of the International Society for Music Information Retrieval Conference, pp. 387–392. Miami (2011).
10. Raffel, C., Ellis, D. P. W.: Intuitive analysis, creation and manipulation of MIDI data with pretty\_midi. In: *International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*. Taipei (2014).
11. CRIM Intervals Search Tools, <https://crimintervals.streamlit.app/>, last accessed 2023/07/31.
12. Eerola, T., Toivianen, P.: MIR in Matlab: The MIDI Toolbox. In: *Proceedings of the International Conference on Music Information Retrieval*, pp. 22–27. Barcelona (2004).

## Comparing vocoders for automatic vocal tuning

D. H. Molina Villota<sup>1</sup> and C. D’Alessandro<sup>1</sup> \*

Institut Jean Le Rond d’Alembert  
Equipe Lutheries-Acoustique-Musique  
Sorbonne Université - Centre National de la Recherche Scientifique  
Paris, France  
daniel.molina.villota@sorbonne-universite.fr

**Abstract.** We present a compendium of sounds and analyses that support a comprehensive approach to the musical use of the vocoder in automatic vocal tuning correction. Vocoder design has primarily focused on refining the vocoder as a realistic vocal transformer. However, its application within modern music emphasizes its unique sonic identity, adding distinctive coloration to the performer’s voice. In this demo, we propose a benchmark that encompasses the vocoder’s key elements. The vocoder is considered and analyzed as an audio effect playing an important role in vocal composition, in an approach similar to the study of musical instruments.

**Keywords:** Vocoder Benchmark Voice Transformation

### 1 Introduction

The term “vocoder” [1] has two meanings: it can either refer to (i) a software device for transparent voice coding, transmission and natural transformation, or to (ii) a musical device for cross-synthesis and pitch flattening. In this paper, we address the first definition, keeping in mind that this technology may also be used in musical applications, in particular for auto-tuning.

The aim of this work is to establish a parametric benchmark that will facilitate technical discussion of the vocoder, particularly in the case of automatic vocal tuning and audio distortion. In establishing such a benchmark, one should be wary of judging vocoders based on the same criteria as natural voice, whose sound description is extremely challenging [3]. In this demo, we present an audio and graphics repository that supports our benchmark, which can help define the vocoder identity.

### 2 The Benchmark

Currently, there are no studies that merge musicological and technical approaches to describe the vocoder as a vocal coloring instrument. Acoustically, the vocoder can be

\* This Research is funded by National Research Agency: Analysis and Transformation of Singing Style ANR19CE380001 & GEsture and PEducation of inTOnation ANR19CE280018



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

seen as just one more of the many parts that compose the vocal apparatus. The vocoder has its own characteristics and identity which are inherent to its technique. We propose a benchmark that precisely frames the unique characteristics of the vocoder as a vocal coloring instrument. The modern music repertoire evidences two main uses: the distortion due to the technique itself (re-synthesizing with the original F0) and the re-pitching technique (like Autotune).

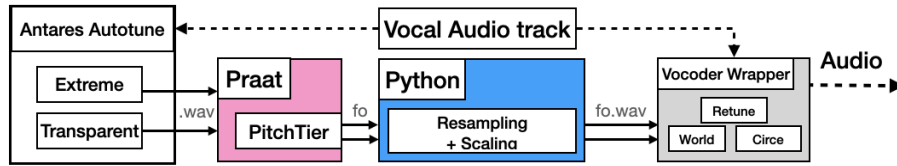
*Methodology:* We started with a sample sound which was passed through the Antares autotune software. We framed the two main use cases (presets): one with extreme correction that merely quantizes pitch, and another “transparent” preset that modifies neither pitch nor any other characteristic. The resulting audio files were analyzed with Praat and shaped with Python, generating an f0.wav file as shown in Figure 1. This file, along with the original sound file, was then processed through various vocoders to obtain the sounds with **extreme correction** and the desired **transparent** modification. The samples used come from previous studies at our lab. They can be heard in an online library along with the vocoded tracks(<https://on.soundcloud.com/1d7mx>).

We have used the following vocoders: **Circe** is based on deep learning [4]. The encoder generates a latent code for selected features, and the decoder transforms it back for a given f0 using a bottleneck technique [5]. **Retune** [7] uses frequency and time domain methods such as the Reduced Heisenberg Uncertainty Transform and the Cross-Frequency Phase Coupling . It is used in ZTX, MAX, Digital Performer, and MOTU. **Autotune Antares** (Abbreviated as ATA) [6] serves as an intonation corrector. It is the most commonly used vocoder in contemporary music. **World** [8] is a vocoder based on a custom spectral representation that generates high-quality audio and fast processing . The benchmark descriptors proposal is summarized in table .

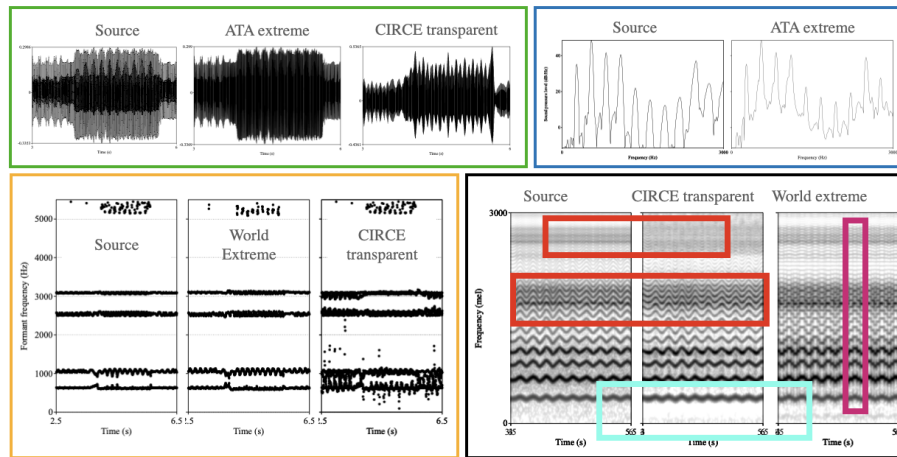
## 2.1 Descriptors of the benchmark

In this section, we summarize some examples of the benchmark. First, we can identify some descriptors independently of the preset used (transparency or extreme retuning). **Latency** is the first appreciable descriptor: retune has the largest latency and ATA the smallest latency. In addition, vocoding involves changes in spectrum, formants and f0-spreading. For those, the transparent preset allows to test the technique alone, avoiding the f0-jumps collateral effect. If the spectrum and signal shape remain unchanged, the vocoder can be considered “**distortion-free**”; ATA and World exhibit this characteristic. Regarding **formants**, World tends to **deepen** them and Circe/retune to **distort** them. Although Circe is known for performing constant transposition well: it generates a **tremolo aligned to vibrato** when using the transparent preset, we also include this effect as descriptor. Concerning harmony, vocoders can present increasing **harmonic differences** (World) or **residual noise** (Retune); we include these changes as descriptors as well. As discussed later, they also appear with the extreme retuning preset.

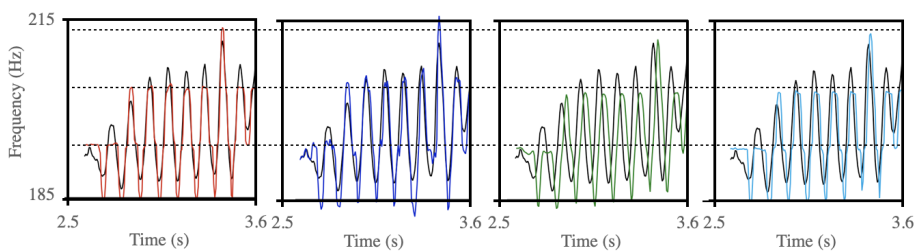
The extreme retuning preset also involves latency, changes in signal shape, spectrum and formants. ATA and World show good **preservation of the signal shape** despite the pitch jumps. The extreme retuning preset causes discrete pitch steps; the transitory parts generate spectral changes which manifest as vertical lines on the spectrogram. Those are related to local **f0-spreading** (or f0-loss), which deteriorates pitch perception and vocoder realism on a global scale. On the other hand, f0-spreading adds a particular



**Fig. 1.** Flow diagram for the methodology for vocoding with two presets: transparent and extreme retuning ( $f_0$  discrete curve).



**Fig. 2.** Green block (Signal Shape): Changes are observed for 2 vocoders. Autotune extreme correction case shows minimal changes while Circe transparent case exhibits significant shape variations. Yellow block (Formants): World shows notable deepening in formant variation and CIRCE exhibits substantial formant alterations. Blue block: (spectral slices):  $f_0$ -spreading at a given time for original audio and ATA extreme retuning. Black block (spectral changes): In the CIRCE re-synthesis case, upper harmonics appear spread (shown in red), while lower harmonic content seems more prominent in relation to noise (shown in sky blue). In the World retuning case, vertical lines (purple) correspond spectral content spreading at each  $f_0$ -steps. The audio sample used for all the examples is “real3maleintervals.wav”.



**Fig. 3.** F0-Path for extreme retuning using (left to right): Autotune, CIRCE, Retune and World. Autotune and World reach exact pitch values more accurately than the others. Retune presents a bigger latency than the other ones.

color due the transient (inherent to the technique) and it contributes to the unique timbre of each vocoder. Each vocoding technique affects harmonics and timbre differently, giving rise to the **harmonic coloration and amplification** descriptors. Circe and Retune are visible examples that alter the harmonic content. Similarly, we observe the **inharmonic coloration** descriptor, which involves residual noise in the low and high-frequency regions of the spectrum. It is notably present in the retune extreme retuning case. Inharmonic coloration affects the presence of noise notably around silences. A summary of the parameters can be seen in Figure 2 and Table 1.

Table 1. Benchmark

Sound Parameter	Latency	Bypass Or resynth Transparency	Formant Deepening	Formant Distortion	Signal Shape Changing	Tremolo Aligned to Vibrato	F0-Spreading	Upper-Harmonics Modification	Sub-Harmonics Modification	In-harmonic Adding and Residual Noise
Autotune	X	X								
Circe	X		X	X	X	X	X	X	X	X
World		X								
Retune	X		X	X	X			X		X

### 3 Discussion

Vocoders can introduce changes in timbre properties, like coloration (filter-like action) or discrete pitch variation, while preserving articulation and prosodic content. Our demo provides an audio and visual comparison of the auditory changes introduced by the use of various vocoders. This comparison has been carried out in a systematic way, yielding the benchmark summarized in table 1. Such a benchmark could serve as basis to develop a shared language for technicians and musicians to describe a vocoder's identity.

### References

1. Dolson, M.: The phase vocoder:A tutorial. In: Comput. Music J. vol 10 no.4, pp. 14-27 (1986)
2. Lanchantin, P. et al.: Vivos Voco: A survey of recent research on voice transformation at IRCAM. In: Int. Conf. on Digit. Audio Effects, pp.277-285. Paris, France (2011)
3. Castellengo, M.: Perception(s) de la voix chantée. In: La Voix Chantée entre Sciences et Pratiques (N. Henrich),pp. 35-64. De Boeck. Paris, France (2014)
4. Roebel, A. and Bous F.: Neural Vocoding for Singing and Speaking Voices with the Multi-Band Excited WaveNet. In: Information 13(3) 103, pp 1-29 (2022)
5. Bous, F and Roebel.: A. A Bottleneck Auto-Encoder for F0 Transformations on Speech and Singing Voice. In: Information 13(3) 102, pp 1-19 (2022)
6. Hildebrand, H.: Pitch detection and intonation correction apparatus and method. Auburn Audio Technologies, Auburn, AL, USA Patent US5973252A, G10H-007/00, pp 10-18 (1992)
7. Bernsee, S. and Gökdag, D.: Methods for extending freq transforms to resolve feats in the spatio-temporal dom. Zynaptiq GmbH. Hannover(DE). Patent EP3271736B1, pp 1-51 (2016)
8. Morise, M. et al.: WORLD: A Vocoder-Based High-Quality Speech Synthesis Sys. for Real-Time Applications. In: IEICE Transactions on Inf. and Sys., E99.D (7), pp 1877-1884, (2016)

## Music recognition, encoding, and transcription (MuRET) online tool demonstration

David Rizo<sup>1,2</sup>, Jorge Calvo-Zaragoza<sup>1</sup>, Juan C. Martínez-Sevilla<sup>1</sup>, Adrián Roselló<sup>1</sup>, and  
Eliseo Fuentes-Martínez<sup>1</sup> \*

<sup>1</sup> Universidad de Alicante

<sup>2</sup> Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana (ISEA.CV)  
drizo@dlsi.ua.es

**Abstract.** Most of the musical heritage is only available as physical documents. Their mere availability as scanned images does not enable tasks such as indexing or editing unless they are transcribed into a structured digital format. Many transcription processes have been traditionally performed following a fully manual workflow. At most, it has received some technological support in particular stages, like optical music recognition (OMR), or transcription to modern notation with music edition applications. A new online tool named MuRET has been recently developed, which covers all transcription phases, from the manuscript image to an digital score. MuRET is designed as a machine-learning based research tool, allowing different processing approaches to be used, and producing both the expected transcribed contents in standard encodings and data for research activities. The objective of the demonstration is to showcase it for an efficient transcription process and provide guidelines on how to get the most out of it.

### 1 Description of the demonstration

MuRET is a research oriented optical music recognition tool (OMR) based on a series of machine learning techniques, mainly deep neural networks, that has been recently ported to be an online application [4] from the original desktop application proposal.

The demonstration will focus on showing all the possibilities that MuRET [4] offers and the process required to convert a series of input images into a digital score in MEI format. MuRET being a research tool, a discussion will be held on possible extensions of interest to the community. Specifically, the demo will consist of the following items:

1. Collection handling (Fig. 1a) in order to organize large corpora.
2. Section and images management (Fig. 1b), used to group, correctly ordering the images to be transcribed, and setup the correct nature of the document (parts based, incipits book, etc.).

\* This work has been supported by the Spanish Ministerio de Ciencia e Innovación through project MultiScore (No. PID2020-118447RA-I00), supported by UE FEDER funds.



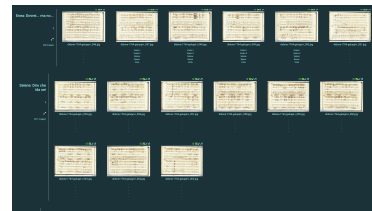
This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



3. Document analysis (Fig. 2) to show how the system has detected the different regions, namely, staves, title, and lyrics, and how user can edit them.
4. Part linking (Fig. 3) in order to let the system identify which instrument belongs each image or crop of the image.
5. Region contents recognition (Fig. 4) where the machine learning models identify the sequence of symbols contained in each region using different approaches depending on the content type, lyrics or music, or the granularity and interaction strategy. In this step, the recognized symbols are just graphical representations (denoted as *agnostic* representation in [2]) without musical meaning.
6. Music encoding of individual staves (Fig. 5) to obtain an actual music encoding of the agnostic representation obtained in the previous step. We will introduce the extension of the formats *\*\*kern* and *\*\*mens* used to accommodate layout information besides the musical content itself. The possibility of transliterate early notations into modern ones will be also explored.
7. Scoring up and exporting (Fig. 6) as the final step in a transcription project, where user can obtain a whole score from the different spread parts, and export a MEI file, either as a whole MEI file or divided into parts including facsimile information to be used by other tools such as MP Editor [3].
8. Offline model training and uploading to allow the user to use his/her already tagged collections to create new fine-tuned models.
9. User action logs analysis from interaction data to obtain the actual transcription times and study the real improvement of new models and approaches.



(a) User collections



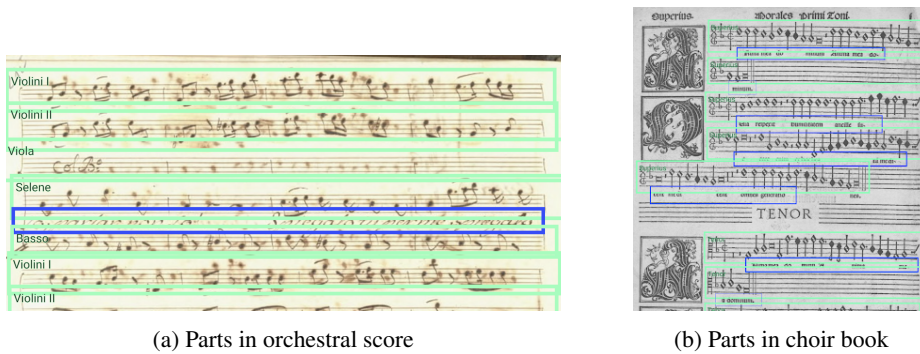
(b) Two sections of a document of a complete opera shown at left column

Fig. 1: Work organization

At the end of the tutorial, attendees should understand the operation of MuRET and how systems based on machine learning can be interactively improved. Also, we hope that attendees will perceive how the use of MuRET, even without being able to guarantee absolute accuracy, significantly decreases the temporal cost of transcription compared to a completely manual process [1].



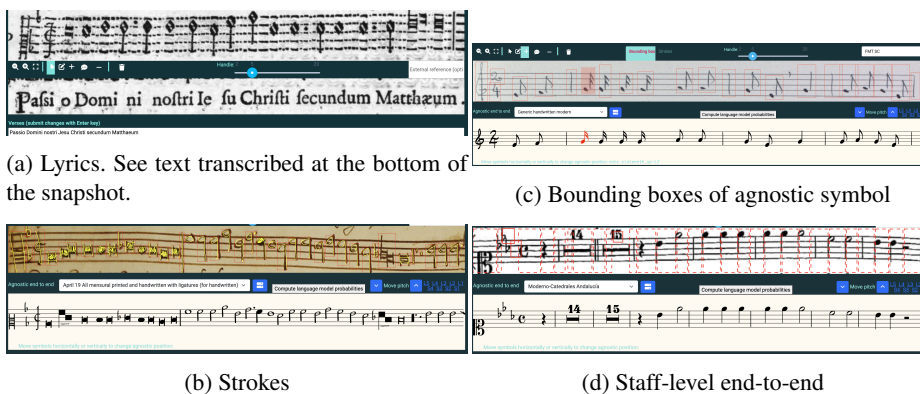
Fig. 2: Document analysis screen excerpt. In this example, only the staves and lyrics regions are segmented. The snapshot shows controls to rotate, manually or automatically, the image, and two possible classifiers to perform the operation automatically. The current catalog of region types shown at the left of the image can be easily modified.



(a) Parts in orchestral score

(b) Parts in choir book

Fig. 3: Different parts and arrangements. All regions must be attributed to a part.



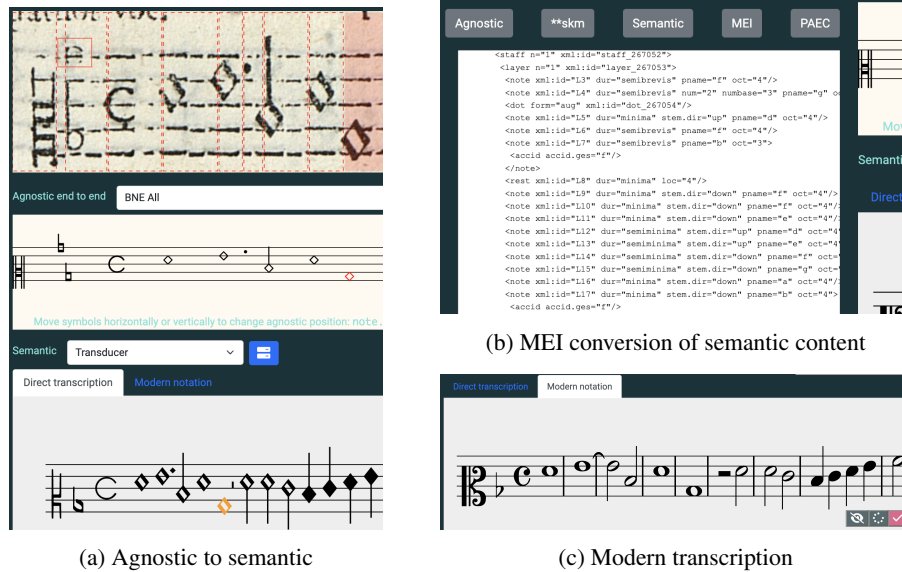
(a) Lyrics. See text transcribed at the bottom of the snapshot.

(c) Bounding boxes of agnostic symbol

(b) Strokes

(d) Staff-level end-to-end

Fig. 4: Transcription of regions.

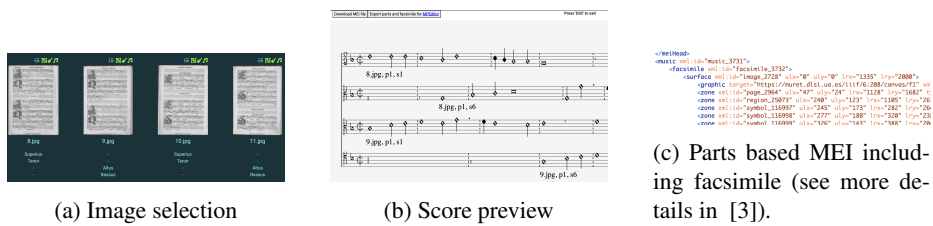


(a) Agnostic to semantic

(b) MEI conversion of semantic content

(c) Modern transcription

Fig. 5: Semantic contents recognized from the image.



(a) Image selection

(b) Score preview

(c) Parts based MEI including facsimile (see more details in [3]).

Fig. 6: Previsualizing and exporting

## References

1. M. Alfaro-Contreras, D. Rizo, J.M. Iñesta, and J. Calvo-Zaragoza. OMR-assisted transcription: a case study with early prints. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 35–41, Online, November 2021. ISMIR.
2. J. Calvo-Zaragoza and D. Rizo. Camera-PrIMuS: Neural end-to-end optical music recognition on realistic monophonic scores. In *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, 2018*, pages 248–255, 2018.
3. K. Desmond, L. Pugin, J. Regimbal, D. Rizo, C. Sapp, and M. E. Thomae. Encoding polyphony from medieval manuscripts notated in mensural notation. In *Music Encoding Conference Proceedings 2021*, page 197–219. Humanities Commons, May 2022.
4. D. Rizo, J. Calvo-Zaragoza, J.C. Martínez-Sevilla, A. Roselló, and E. Fuentes-Martínez. Design of a music recognition, encoding, and transcription online tool. In *16th International Symposium on Computer Music Multidisciplinary Research, Tokyo*, (accepted) 2023.

# Microtonal Music Dataset v1

Tatsunori Hirai<sup>1</sup>, Lamo Nagasaka<sup>1</sup>, and Takuya Kato<sup>1,2</sup> \*

<sup>1</sup> Komazawa University

<sup>2</sup> ExaWizards Inc.

thirai@komazawa-u.ac.jp

**Abstract.** In this paper, we propose a microtonal music dataset, comprising musical compositions that utilize microtones, tones with intervals that are more refined than those found in the 12 equal temperament. As part of the Microtonal music dataset v1, we present 100 manually created microtonal music pieces, along with their characteristics and statistical information. Furthermore, we will discuss the potential for future music information processing research that can be realized using the microtonal music dataset.

**Keywords:** Microtonal music; microtone; dataset

## 1 Introduction

The recent advancements in generative AI technology are progressing at an astonishing speed, and the distinctions between human-composed music and AI-generated music are becoming increasingly blurred. As a result, we are reaching a point where music can be generated with a single click, reducing the need for human composition, especially for music for trivial purposes.

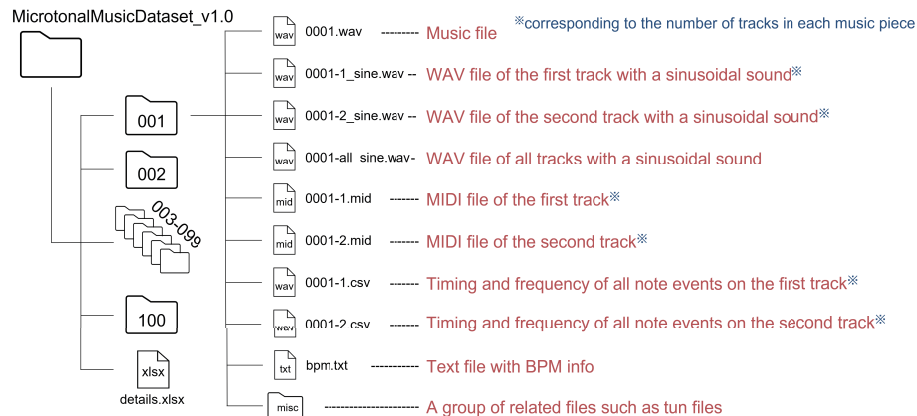
Microtonal music is cited as one of the musical expressions that necessitates tools for supporting expression. Microtonal music refers to music that uses microtones, pitches that do not conform to the 12 equal temperament, which is difficult to perform with many traditional instruments. Composing microtonal music is challenging even for people with experience in composing conventional music. We believe that expanding human expressive capabilities through AI assistance, especially for music that are difficult to perform or compose within current frameworks, can contribute to the development of musical culture. Therefore, in this study, we propose a microtonal music dataset to accelerate research on the technology capable of handling microtonal music.

If technology capable of handling microtones is realized, it could enable support such as redesigning the piano roll according to the temperament inferred from the microtones input by the user[1], or providing accompaniment to the microtone melodies composed by the user. These tasks are currently challenging even for humans, and assistance through technologies such as AI is effective.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS KAKENHI Grant Number JP23K17023.



**Fig. 1.** Directory structure of the dataset.

In this study, as a first step towards realizing technology capable of handling such microtones, we propose a dataset composed of 100 manually composed microtonal music pieces.

## 2 Related Work

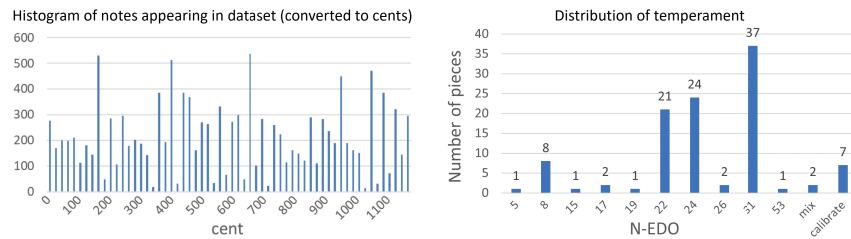
The RWC Music Dataset, consisting of 315 songs and 50 types of instrumental sounds, has significantly impacted music information processing research. This dataset avoids copyright issues in research, and continues to be influential in the field.

The JSB Chorales Dataset, which digitizes 382 four-part chorales composed by Bach, has been utilized in many studies on music generation technologies[3], [4]. On the other hand, no dataset related to microtonal music has been proposed so far.

## 3 Design of Microtonal Music Dataset v1

### 3.1 Structures of the Dataset

The structure of the Microtonal Music Dataset directory is illustrated in Fig 1. The data for each piece is stored in folders with IDs 001 to 100. Each folder contains the music file, MIDI data for each track that composes the piece, wav data written with sine waves for those tracks, a CSV file recording the frequency, onset, and offset of each note for every track, and a text file recording the BPM data of the piece. Since MIDI data cannot record the exact frequency of microtones, the frequency information is included in the CSV file. The misc folder includes items such as tuning files (.tun) loaded into the software synthesizer for playing microtones. The file details.xlsx consolidates various information, including the statistical data of the pieces comprising the dataset and the temperament information.



**Fig. 2.** Histogram of notes in cent (left) and distribution of temperament in the dataset (right).

### 3.2 Dataset Creation Method

The pieces comprising the dataset were produced using the DAW software Studio One v5.5.2 by one of the authors. To create pieces including microtones in Studio One, we used the software synthesizer Vital, which can play microtones by loading tuning files, and SimpleMicrotonalSynth, which allows for a variety of selectable microtonal tuning options. Each piece was created either by inputting one microtone at a time or by using a MIDI controller for real-time input.

The tuning files for loading into Vital were created using a web page called ScaleWorkShop. We also used microtones expressed by tuning the synthesizer in cent units.

## 4 Statistic of the Dataset

Here we describe the characteristics of the dataset. Piece lengths average 22.7 seconds, ranging from 6.0 to 74.0 seconds, with an average BPM of 130.4, between 80 and 180.

Fig. 2 (left) illustrates the histogram of notes appearing in the dataset, converted into cents. Here, we set 261.626Hz as the 0-cent reference point and, by utilizing octave equivalence, we convert all notes to frequencies within the same octave before calculating their values in cents. In fig. 2 (left), bins at multiples of 100 represent the notes in 12 equal temperament. The fact that these bins do not show particularly high values indicates that no specific 12 equal temperament notes are being used extensively. Conversely, the infrequent use of sounds in certain frequency bands, such as those around 340 cents and 1040 cents, is intriguing. Despite being a microtonal music dataset, many pieces also include tones from 12 equal temperament. Notably, in 24 equal temperament, half the tones align with the 12 equal temperament. In the pieces, the proportion of 12 equal temperament notes ranged from 0% to 67.1%, with an average of 21.4% across the dataset. Considering the familiarity of the music, a certain degree of use of the 12 equal temperament notes is allowed.

In the dataset, 525 distinct pitches appear, including 474 types of microtones and 51 tones of the 12 equal temperament. These microtones could be candidates for pitches in microtonal music generation models using this dataset.

This dataset includes microtonal music based on N-equal temperament (N-EDO) other than 12, music created by uniformly shifting tunings from specific temperament by X cents (calibrate), and their combinations (mix). Fig. 2 (right) shows the distribution of temperament that make up this dataset. As a temperament that is easy for

the composer to create music, scales that felt harmonically familiar were frequently adopted, particularly pieces with the 22, 24, and 31 equal temperaments. In the future, we plan to enhance the diversity of temperaments.

## 5 Potential Uses of the Dataset

The primary envisioned use of this dataset is for machine learning models. In deep music generation models based on 12 equal temperament, the model is constructed on the assumption that the input and output data are in 12 equal temperament. When extending this to microtonal music, it is anticipated that simply changing the input and output layers would not be sufficient. By utilizing this dataset, it becomes possible to further conduct research into models that can handle microtones.

Additionally, it can be utilized as test data to further generalize conventional music recognition techniques. In music analysis, concepts that presuppose 12 equal temperament, such as chromagrams, are sometimes used; however, these cannot be applied to microtonal music. We expect this dataset to be valuable for developing how conventional techniques, such as pitch recognition and chord recognition, can be generalized to microtones.

## 6 Conclusion

In this paper, we introduced a dataset titled Microtonal Music Dataset v1, consisting of 100 short pieces of music that include microtones. By advancing research based on this dataset, we believe that current music information processing techniques can be extended to include microtones, and ultimately, this could lead to the application of generative AI technology to enhance human musical expression.

In the current dataset, 100 pieces of microtonal music were created, but because the range of sounds that microtones encompass is diverse, there are plans to increase both the number of pieces and the diversity of the music in upcoming versions such as version 2 and beyond. Specifically, in order to enable the conversion of music in 12 equal temperament into microtonal music, we would like to increase the data of microtonal pieces that are paired with 12 equal temperament music. By doing so, we believe it will be possible to microtonalize existing pieces in the 12 equal temperament and significantly increase the size of the dataset. Additionally, by exploring methods of data augmentation, we plan to develop this dataset into a resource that is adequately applicable to deep learning techniques, which are indispensable for large-scale data.

## References

1. Hirai, T.: Redesigning a Piano Roll: A Melody Input Interface That Can Play Microtones with an Arbitrary Number of Keys. SMC, pp.1–7 (2022)
2. Goto, M.: Development of the RWC music database. ICA, Vol. 1, pp.553–556 (2004)
3. Allan, M., and Williams, C.: Harmonising chorales by probabilistic inference. NIPS 17 (2004)
4. Boulanger-Lewandowski, N., et al.: Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. ICML, pp.1881–1888 (2012)



# Lighting Control based on Colors Associated with Lyrics at Bar Positions

Shoyu Shinjo<sup>1</sup> and Aiko Uemura<sup>1</sup> \*

Nihon University  
cish19085@g.nihon-u.ac.jp  
uemura.aiko@nihon-u.ac.jp

**Abstract.** This study proposes a control method for changing light to a suitable color according to the timing of a bar position in synchronization with the music. The aim is to provide users with more realistic experiences when they are enjoying online live performances at home by changing light colors to match the music. Conventional methods switch the light for each word, and there are some variations associated with words within lyrics. Therefore, the proposed method increases the variations of the color image scale and the colors associated with the words to match the lyrics and song information. Moreover, our system is designed to change the lighting color at the timing of each bar position based on beat estimation from the song.

**Keywords:** lighting control, lyric, color image scale

## 1 Introduction

Online live performances have been increasing as part of the new life styles that emerged during the COVID-19 pandemic. However, watching live-streaming performances at home tends to be less present than watching in person due to insufficient lighting effects, venue size, and sound volume. As a result, participants only have partial enjoyment of their experiences.

The aim of our study was to consider a method that could easily create lighting effects suitable for songs without specialized knowledge. The previous study [1] developed lighting control on the stage, and the system incorporated 300 words that corresponded to a color image scale [2]. This color image scale was developed psychologically to define the common senses of color images and facilitated the classification and correlation of images of words within lyrics. We apply this idea to the PHILIPS Hue Go portable accent light [3], which can be used in the home. However, it should be noted that the timing of the lighting transitions in the original scheme did not match the song.

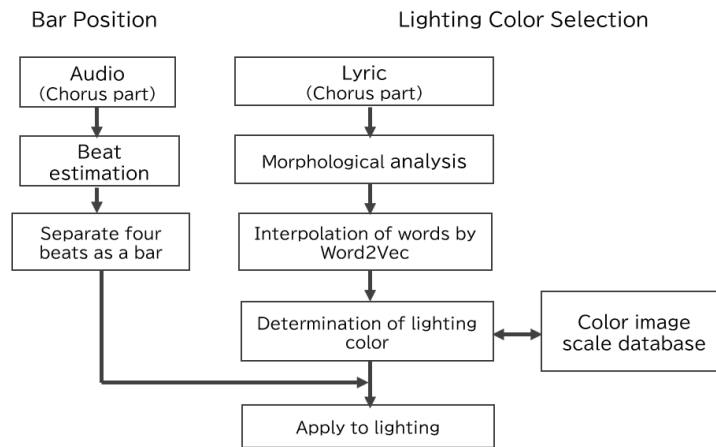
---

\* This work was supported by JSPS KAKENHI Grant Numbers 20K19947 and 22H03711.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).





**Fig. 1.** Overall view

In this study, a lighting control method was developed that instigates light transitions at appropriate timing using the colors associated with words within lyrics. This is achieved by changing the lighting color at bar positions containing each word. In addition, we increase the number of image scale color variations to 1,317 words, allowing the method to suggest light colors that match the impression of lyrics more effectively.

## 2 Methods

Figure 1 provides an overview of the proposed method, in which there are two main processes: estimating bar positions from acoustic signals and selecting colors from a color image scale that match each word in the lyrics. It should be noted that we only focused on the first chorus part of songs.

### 2.1 Bar Position

Bar positions are often helpful in providing hints for the locations of structure boundaries and turning points within the music. Thus, we extract bar positions to switch the lighting color according to appropriate timings in the music. We estimated the time of each beat [4] and calculated the bar position time assuming a time signature of 4/4.

### 2.2 Lighting Color Selection

We search for the appropriate image scale color for each word in the lyrics according to the following procedure.

1. Extract nouns and adjectives from lyrics using MeCab (Yet Another Part-of-Speech and Morphological Analyzer) [5]. For the model dictionary, we used mecab-ipadic-NEologd [6], which is robust for new words and proper expressions.



**Fig. 2.** Example of lighting “*CHE.R.RY* (Artist: YUI)” (upper left: koi / love, upper right: hoshi / star, lower left: cherry, lower right: message.)

**Table 1.** Time [s] to switch lighting color “*CHE.R.RY* (Artist: YUI).”

Word Japanese / English	Previous	Proposals
Koi / Love		
Hoshi / Star	7.428	
Yoru / Night	1.005	6.594
Negai / Wish	0.814	
Cherry/ Cherry	2.673	
Yubisaki / Fingertip	1.144	4.389
Kimi / You	1.162	
Message / Message	0.060	2.206

2. Obtain embedding vectors for each word using Word2Vec [7] and the pre-trained Japanese Wikipedia entity vectors [8]. Each word is then complemented based on the word in the color image scale database using highest cosine similarity.
3. Search for the bars in which each word appears from the lyrics information separated into bars.
4. Select the word with the highest cosine similarity once every two bars.
5. Set the image scale color as the lighting color based on each selected word.

### 2.3 Apply to lighting

We used the Philips Hue API to set the lighting colors and start timing to Hue Light using RGB values based on the color image scale, and the start time of the bar in which the word appears. To represent the lighting color in the XYZ color space, we converted the RGB values to xy color space.

## 3 Demonstration

We conducted simulations of lighting effects based on the proposed method. Figure 2 displays an example using “*CHE.R.RY* (Artist: YUI)”<sup>1</sup>, and Table 1 presents an example of words and switching times in the song “*CHE.R.RY*.”

The proposed method selected four of eight words for the lighting color. Each word appeared in bars 1, 4, 6, and 7. In the proposed method, the word with the highest cosine similarity was “cherry,” and its similarity was 1.0. Conversely, the lowest similarities were 0.471 for both “message” and “arigatou (thank you).” In the simulation based on the method of the previous study, all eight words were used as the lighting color. The highest similarity between words was 0.466 for the words “kimi (you)” and “ureshii (happy).” In contrast, the lowest similarity was 0.256 between “hoshi (star)” and

<sup>1</sup> Our demonstration movies are available at [https://scrapbox.io/uemaiklab/Lighting\\_Control\\_Demo](https://scrapbox.io/uemaiklab/Lighting_Control_Demo)

“mabushii (dazzling).” The increment of the word variation in the image color scale also increased the variety of lighting colors. We assumed that we had enhanced the harmony between the music impression and the colors associated with the lyrics.

Table 1 displays the time corresponding to each word and the time required for each method to switch to the next light color. Table 1 indicates that the longest and shortest times for the proposed method were 6.594 and 2.206 s, respectively. In contrast, the longest and the shortest switching times in the previous study was 7.428 and 0.606 s, respectively. In the previous study, there were five locations where the time until the color switched was 1 s or less. This indicated that frequent switching of lighting colors occurred. We consider that the proposed method improved the temporal harmony because the colors were only switched for each bar.

We also found that words whose meanings were the exact opposite of each other when using Word2Vec were sometimes candidates as the most similar words. For example, “kanashimi (sadness)” was complemented with “yorokobi (happiness).” This could be because the model was trained to assume that words appearing in the same context have similar meanings.

## 4 Conclusions

We developed a lighting control method using colors associated with lyrics through word embedding and the color image scale. Furthermore, we improved the lighting timing by transitioning at bar positions instead of at each word. Ultimately, our proposed light system illuminates with appropriate colors and timing.

## References

1. Kanno, M., Fukuhara, Y.: Automatic stage illumination control system by impression of the lyrics and music tune In: 2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter), pp. 219–224, (2022).
2. Kobayashi, S.: The aim and method of the color image scale. *Color Research and Application*, 6(2), 93–107 (1981).
3. PHILIPS Hue Go portable accent light: <https://www.philips-hue.com/en-us/p/hue-white-and-color-ambiance-go-portable-accent-light/7602031U7>.
4. Ellis, D. P. W.: Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1), 51–60 (2007).
5. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 230–237, (2004).
6. Toshinori, S.: Neologism dictionary based on the language resources on the web for mecab. <https://github.com/neologd/mecab-ipadic-neologd> (2015).
7. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.
8. Suzuki, M., Matsuda, K., Sekine, S., Okazaki, N., Inui, K.: A joint neural model for fine-grained named entity classification of Wikipedia articles. *IEICE Transactions on Information and Systems*, E101.D(1), 73–81 (2018).

# Melody Changing Interfaces for Melodic Morphing

Masatoshi Hamanaka<sup>1</sup>

RIKEN `masatoshi.hamanaka@riken.jp`

**Abstract.** We have developed several applications based on the Generative Theory of Tonal Music utilizing the melodic morphing method. Since multiple melodies generated by the morphing method have similar musical structures, the global structure of the melodies does not change when a portion of one melody time axis is replaced by another. When developing the apps, we used dial-based and grid-based interfaces for switching melodies. In this paper, we present the results of a comparison of the two interfaces conducted with 30 users.

**Keywords:** Melody switching interface, Melodic morphing method, Generative Theory of Tonal Music (GTTM), Dial-type interface, Grid-type interface

## 1 Introduction

We have developed several applications using a melodic morphing method based on the Generative Theory of Tonal Music (GTTM) [1, 2]. In the GTTM, a time-span tree is a binary tree in which each branch is connected to each note (Fig. 1). The branches of a time-span tree are connected closer to the root than those connected to structurally important notes.

The main advantage of time-span trees is that they can be used to reduce notes. Specifically, reduced melodies can be extracted by cutting a time-span tree with a horizontal line and omitting the notes connected below the line. In melody reduction with GTTM, these notes are essentially absorbed by structurally more important ones.

We previously proposed a melody-morphing method that applies this reduction (Fig. 2) to generate a melody that is structurally intermediate between two input melodies[3, 4]. This is done by combining two melodies after executing the reduction on their respective time-span trees.

Since multiple melodies generated by the morphing method have similar musical structures, the global structure of the melodies does not change when a portion of one melody time axis is replaced by another. When developing apps with the melodic morphing method, we used dial-based and grid-based interfaces for switching melodies. In this work, we present the results of a comparison of the two interfaces conducted with 30 users.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

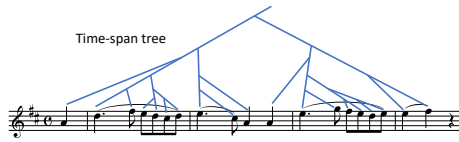


Fig. 1. Time-span tree.

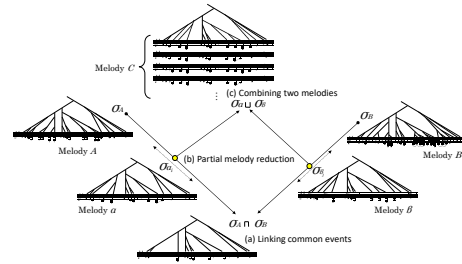


Fig. 2. Melodic morphing method.

## 2 Applications for Melodic Morphing Method

**ShakeGuitar** The ShakeGuitar (Fig. 3(a)) and ShakeGuitarHD (Fig. 3(b)) apps change the morphing level according to the speed at which the iPhone or iPad is shaken [5]. For the morphing input, we utilize the basic melody of “The Other Day I Met a Bear” and the melody of a guitar solo played with the same chord progression. The unique feature here is that not only interpolation of the two melodies but also extrapolation is performed [6]. With the extrapolation, we can generate an intense guitar solo with more notes than the original. When the iPhone is held stationary, a basic melody is played, and the faster the iPhone is shaken, the more intense the melody becomes.

ShakeGuitar and ShakeGuitarHD both feature a grid mode with time on the vertical axis and morphing level on the horizontal axis. The morphing level can be changed by touching the grid. In ShakeGuitarHD, the guitar is animated to swing up and down, and the width of the swing changes according to the morphing level. The morphing level also changes depending on how fast you swipe your finger up and down on the swinging guitar.

**Melody Slot Machine** We developed the Melody Slot Machine, a research demonstration device, to promote the melodic morphing method. With this application, the performer’s movements can be viewed on a Pepper’s ghost display (Fig. 4(a)). Melody segments are displayed on a dial, and the melody to be played can be switched by rotating the dial (Fig. 4(b)). We exhibited the Melody Slot Machine at an international conference shortly after it was developed, but the COVID-19 pandemic made it difficult to conduct further demonstrations in person [7–9]. We therefore adapted the Melody Slot Machine for the iPhone so that people could experience it simply by downloading the app.

**Melody Slot Machine for iPhone** Figure 5 shows a screenshot of the Melody Slot Machine iPhone app [10]. The horizontal axis is time, and each dial displays a melody segment in musical notation (Fig. 5(a)). By swiping up and down on each dial, you can switch between the segments. Due to the limited screen size of the iPhone, only four melody segments can be viewed simultaneously, and the currently playing segment can be viewed by automatically scrolling left as the musical piece progresses (Fig. 5(b)).

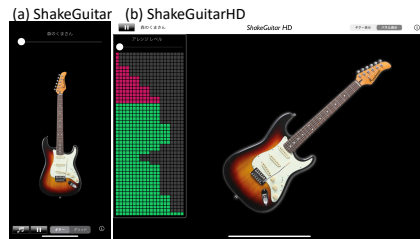


Fig. 3. Screenshot of ShakeGuitar.

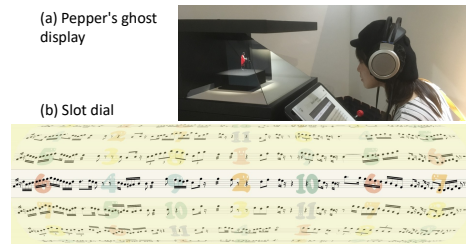


Fig. 4. Melody Slot Machine.

The dial changes for the entire musical piece are displayed in a grid at the bottom of the screen, corresponding to the numbers written on the dials. Swiping up from the bottom of the grid display brings up the full-screen grid (Fig. 5(c)), and users can touch it to change the selected grid. The change in the grid is linked to the dial, and the melody is played reflecting the change. Swiping down terminates the full-screen grid, and the dial appears again. When the iPhone is shaken up and down, each dial is shuffled to generate a new combination of melodies (Fig. 5(d)).

**Melody Slot MachineHD** Figure 6(a) shows a screenshot of Melody Slot MachineHD, in which the symbols represent changes in melody variations [11]. For example, the musical note symbol means that the same variation will continue, and the cherry symbol indicates that the variations will change one after another.

Pressing the mode-switch buttons on the left and right of the screen displays the grid tile screen, and you can check and change the variations in the entire song (Fig. 6(b)). If you use the grid to change the combination of variations, the symbols on the slot screen will also change accordingly.

The performer screen is displayed by pressing the mode-switch buttons or holding the iPad vertically (Fig. 6(c)). This display shows a performer playing new combinations of melodies determined by the slots or grids. Short interpolation video clips of the performer generated by AI are sandwiched into recorded videos of an actual performer, so the performer moves seamlessly.

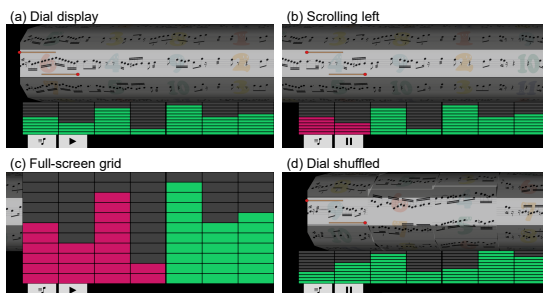


Fig. 5. Melody Slot Machine iPhone app.

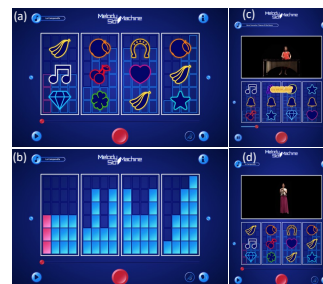


Fig. 6. Melody Slot MachineHD.

### 3 Experimental Results

We launched the Melody Slot Machine iPhone app in May 2021 and Melody Slot MachineHD in March 2022. As of July 2023, they have downloaded 657 times. After launching the app ten times, users are presented with a message inviting them to complete a questionnaire regarding its usability. The following is a portion of the questionnaire.

Q1: Which was easier to operate, the dial screen or the grid screen?

Q2: Which was more enjoyable to operate, the dial screen or the grid screen?

Thirty responses were received, 20 people said that the dial type was easier to operate and more enjoyable than the grid type.

### 4 Conclusion

In this work, we compared several interfaces that change the melody of applications using a melodic morphing method based on the Generative Theory of Tonal Music. Our findings showed that more people found the dial-type interface easier and more enjoyable to operate than the grid-based one.

In the app version of the Melody Slot Machine, the dials on the iPhone were musical notations and on the iPad they were symbols, but we plan to make it possible to switch between the two types of dials on both devices.

### References

1. Lerdahl, F. and Jackendoff, F.: *A Generative Theory of Tonal Music*. The MIT Press, Cambridge, MA (1983).
2. Hamanaka, M., Hirata, K., and Tojo, S.: Implementing “A Generative Theory of Tonal Music”. *Journal of New Music Research*, 35(4), 249–277 (2006).
3. Hamanaka, M., Hirata, K., and Tojo, S.: Implementation of Melodic Morphing based on Generative Theory of Tonal Music. *Journal of New Music Research*, 51(1), 86–102 (2022).
4. Hamanaka, M., Hirata, K., and Tojo, S.: Melody Morphing Method based on GTTM, in *Proceedings of the 2008 International Computer Music Conference (ICMC2008)*, pp. 155–158 (2008).
5. Hamanaka, M., Yoshiya, M., and Yoshida, S.: Constructing Music Applications for Smartphones, in *Proceedings of ICMC2011*, pp. 308–311 (2011).
6. Hamanaka, M., Hirata, K., and Tojo, S.: Melody Extrapolation in GTTM Approach, in *Proceedings of ICMC2009*, pp. 89–92 (2009).
7. Hamanaka, M., Nakatsuka, T., and Morishima, S.: Melody Slot Machine, *ACM SIGGRAPH 2019 Emerging Technologies ET-245*, 2 pages (2019).
8. Masatoshi H.: Melody Slot Machine: A Controllable Holographic Virtual Performer, in *Proceedings of the 27th ACM International Conference on Multimedia (MM’19)*, pp. 2468–2477 (2019).
9. Nakatsuka, T., Hamanaka, M., and Morishima, S.: Audio-guided video interpolation via human pose features, in *Proceedings of VISIGRAPP2020*, pp. 27–35, (2020)
10. Hamanaka, M.: Melody Slot Machine on iPhone, *Submitted ACM Symposium on User Interface Software and Technology (UIST’23)*, 2 pages (2023).
11. Hamanaka, M.: Melody Slot Machine HD, *ACM SIGGRAPH 2023 AppyHour*, 2 pages (2023).

# Relative Representation of Time-Span Tree

Risa Izu<sup>1,\*</sup>, Yoshinari Takegawa<sup>1</sup> and Keiji Hirata<sup>1</sup>

Future University Hakodate  
g2123007@fun.ac.jp

**Abstract.** We propose a novel representation method of time-span tree of Generative Theory of Tonal Music (GTTM), which is suitable for deep learning using neural networks. We are interested in representing the meaning of music in a tree structure, as in natural language understanding, and employ the time-span tree of GTTM. The strengths of our method are relative tensor representation of parameter values and tree structure of variable shape and size. Our method properly reduces the number of parameter values and the amount of information describing the time-span tree structure for deep learning. That is, the same information can be expressed with fewer symbols. Through small-scale experiments, the relative representation has been shown to be promising.

**Keywords:** Generative theory of tonal music (GTTM), time-span tree, block view

## 1 Introduction

Generative theory of tonal music (GTTM) [1] which is a cognitive music theory, represents the hierarchical structure of melodies by expressing the relative importance of each note as a time-span tree. The time-span subtrees exhibit both local and global dependencies, and it is important to consider the both dependencies for a comprehensive analysis of the hierarchical structure of time span trees. Takahashi et al. [2] proposed a method in which a time-span tree is represented by the block view considered as a tensor, and Seq2Seq model with the attention mechanism captures the both local and global dependencies contained in time-span tree. However, since the block view uses absolute values for representing duration and pitch, it leads to difficulty in learning the general rules for the values and the relationships among block.

Therefore, we introduce a block view containing relative values. Specifically, we establish representations for relative vertical and horizontal positions, duration, and pitch, enabling a block view to express the relationships among subtrees. This approach is expected to reduce feature complexity, leading to improved accuracy improvement and reduced of training time.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).



## 2 Relative Representation of Time-Span Tree

Our proposed method introduces a relative block view representation, enabling a more detailed and expressive description of the hierarchical structure of melodies. Fig. 1 shows the existing block view converted to a relative representation.

Durations are represented by a combination of nine basic labels, such as quarter note, eighth note, and so on. For example, note id 1 in the 1st layer has a duration of 0.75. Converting this to a relative expression, 0.75 can be represented as the sum of 0.5 and 0.25.

The pitch class is calculated as an interval and direction of melodic change between the pitch and the parent time span that governs the pitch, that is, the block directly superior to the pitch. For example, note id 2 in the 1st layer has pitch class D and is dominated by C $\sharp$  in note id 3 in the 2nd layer. D is one interval above C $\sharp$ , and hence, the interval is 1 and the direction of melodic change is +. In some cases, melodic change may be more than one octave. At present, we assume that melodic change is within one octave (0 to 11) for such cases.

The branching information in the tree structure is represented by the sequence of left- or right-branchings from the maximum time-span position. For the depth of sequence, 0 is assigned to the initial occurrence of time-span (the maximum time-span), and + to the same time-span occurring in the subsequence. Concerning the left/right branching,  $\epsilon$  is assigned when no branching occurs, and L and R are assigned to the left- and right-branching, respectively. For example, the 4th layer is assigned [0,  $\epsilon$ ] because it has no branches and no upper layers. Furthermore, note id 1 in the 3rd layer is represented as [+ , L] because the value 1 means one-level deep from the above 4th layer and left-branching occurs.

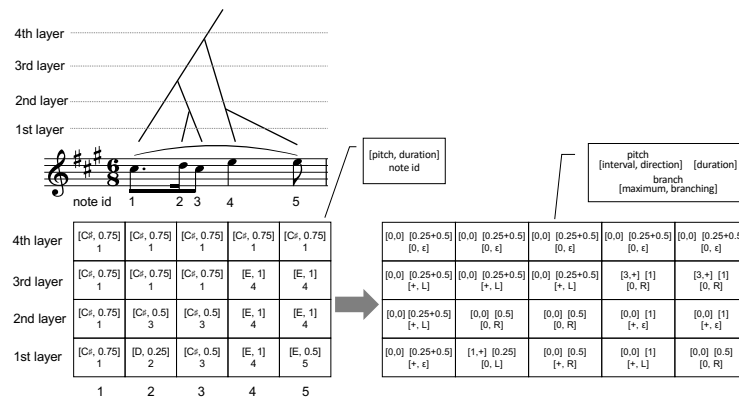


Fig. 1. Conversion of Absolute Representation of Block View to Relative One

When entering data into the model, each note information information is treated as multi-hot. Specifically, we combine a multi-hot vector indicating duration, a one-hot

vector indicating pitch interval, a one-hot vector indicating pitch direction, a one-hot vector indicating branch number, a one-hot vector indicating branch direction, a label indicating padding, a label indicating mask. Table. 1 shows details of each category.

**Table 1.** Melodic Features in Multi-Hot Vector

Category	Values or Labels	Length
Mask	mask or not	1
Padding	BOS, EOS, padding for sequences, padding for layers	4
Duration	0.125, 0.1667, 0.25, 0.3333, 0.5, 0.6667, 1.0, 2.0, 4.0	9
Pitch interval	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	12
Pitch direction	0, +, -	3
Sequence of branch	0, +	2
Left/right branching	ε, L, R	3

### 3 Experiment

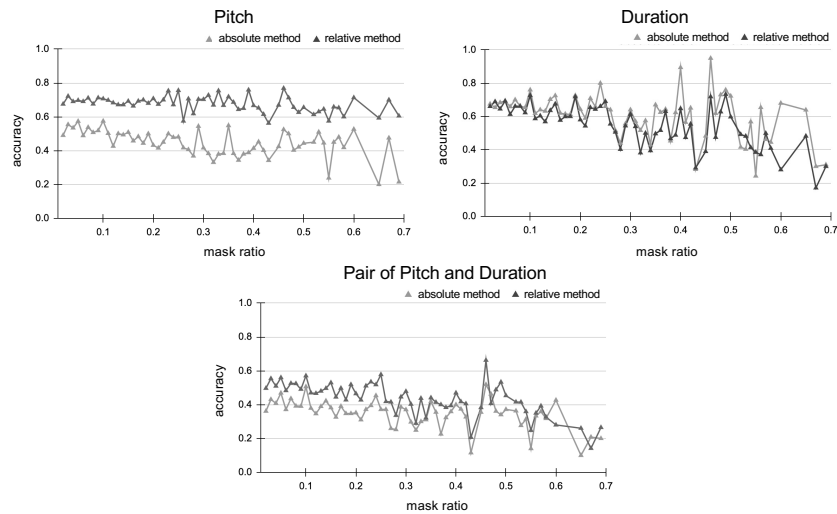
Through the fill-in-the-blank task for the block view of a time-span tree, let us validate the the proposed relative representation. In this paper, we call the representation method employed in the previous study as the absolute method [2], and the proposed method as the relative method. To evaluate how much the proposed method improves the result of the fill-in-the-blank task over the absolute one, we measure the accuracy of the following three factors: pitch, duration and a pair of pitch and duration. Since a pitch is represented by a pitch interval and the direction of melodic change in the relative method, we have the correct answer if both a pitch interval and the direction are the same.

#### 3.1 Experimental Setup

We use the GTTM database [3], with 176 songs for training data, 44 songs for validation data, and 55 songs for testing data. we crate dataset for the fill-in-the-blank task by masking each subtree. By masking, we obtain 6117 training data, 1498 validation data, and 1797 test data. Each batch contains 64 pieces of data. The embedding dimension by skip-thought is set to 300 and the size of the hidden layer of the Seq2Seq model is set to 200. We use the optimizer Adam with a learning rate of  $1.0 \times 10^{-4}$ .

#### 3.2 Results

Fig. 2 shows the results of the accuracy to masking ratios for the three factors. For pitch accuracy, the relative method exceeded accuracy of 0.60 for almost all masking ratios, and, for all masking ratios, the relative method was superior to the absolute method. For duration accuracy, the absolute method was advantageous for almost all masking ratios. Furthermore, the maximum difference of accuracy rates exceeded 20%. For a pair of pitch and duration, the relative method was equal to or better than the absolute one, and, for low masking ratios, the relative method was superior by about 10%.



**Fig. 2.** Accuracy to Masking Ratios for the Three Factors

## 4 Conclusion

We proposed the relative representation of duration, pitch, and branching information in a block view of time-span tree. The results of the fill-in-the-blank task show that the relative method is advantageous for the factors of pitch and a pair of pitch and duration.

The points to be improved in the future are as follows. To improve duration accuracy, we need to examine and refine the representation method for duration. For example, we consider the relative representation based on metrical information. Furthermore, since the current validation test is conducted on a small dataset, we need to validate on a larger dataset through data augmentation.

## References

1. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music, The MIT Press, Cambridge (1983)
2. Takahashi, R., Izu, R., Takegawa, Y., Hirata, K.: Global Prediction of Time-span Tree by Fill-in-the-blank Task, 16<sup>th</sup> International Symposium on Computer Music Multidisciplinary Research (CMMR 2023)
3. Hamanaka, M.: GTTM Database, <https://gttm.jp/gttm/ja/database/>

# Zero-Shot Music Retrieval For Japanese Manga

Megha Sharma and Yoshimasa Tsuruoka

The University of Tokyo

{meghas, yoshimasa-tsuruoka}@g.ecc.u-tokyo.ac.jp

**Abstract.** This demo proposes a novel task for curating theme music for manga (Japanese comics). One of the biggest challenges in this field is the lack of available paired data for manga and music. Hence, we employ alignment properties of pre-trained models to infer the relationship between music and manga and retrieve music given an input manga page. We call this zero-shot, as we do not train on any explicit aligned music-manga dataset. Our preliminary results show potential in the task of music retrieval from manga when fine tuned on independent manga-text and music-text pairs compared to the original AudioCLIP model.

**Keywords:** Music Retrieval, Multi-Modal, Manga, Emotion-Aware Retrieval

## 1 Introduction

Storytelling shares fictional or non-fictional accounts for knowledge, entertainment, and even branding in modern society [1]. To bridge the gap between reality and stories, artists make use of additional modalities such as illustrations, onomatopoeia, sound and movement. Comic books are an example of story telling that employs illustrations and onomatopoeia. In a country like Japan, where manga or Japanese comics hold significance in history of popular culture [2], the evolution of comic story telling has experienced waves of digitisation and animation [3]. In an effort to enhance the reading experience, such adaptations also use sound effects and music with digital comics<sup>1</sup>.

However, additional modalities can be expensive and require domain knowledge. Attempts have been made to curate background music for books using the soundtracks from the movie adaptations [4]. However, curating background music for comics remains a novel task. To our knowledge, our work remains one of the first attempts to curate music based on comics. One of our biggest challenges remains the lack of publicly available music datasets from manga or anime. Hence, we focus on extracting alignment relationships between music and manga books by training a shared embedding space of image, text and audio. In the current stage, our demo proposes a solution to retrieve music for manga based on a given page, by aligning the relationships implicitly through text. Our method is inspired from the recent success of ImageBind

<sup>1</sup> <https://www.webtoons.com/en/>



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

[6], which trained six modalities together by binding them with image embeddings. A shared embedding space is trained with independent datasets: AudioSet [7] (Music-Text) and Manga109 [8] (Manga-Text). In this paper, we report the progress of the training and results from our first iteration. We also comment on our current limitations and future work for the task. The code and some example runs are made available at <https://github.com/ms3744/Music-Manga-Retrieval>.

## 2 Implementation

At the current stage, we approach the problem as a manga-to-music retrieval task. Hence, our focus is on strengthening the embedding space between the modalities. An ESResNeXT encoder [12] is used for audio while the Res-Net and Transformer encoders from CLIP [11] are used as image and text encoders respectively. The three encoders share a multi-modal embedding space using the AudioCLIP architecture [10], which is an extension of the CLIP model with an audio encoder. We use the pre-trained weights from AudioCLIP as large-scale trained models exhibit emergent properties in modality alignment [6]. We first train the audio encoder on (audio-text) pairs, and then finetune the image and text encoders on (image-text) pairs.

**Datasets.** The model is fine-tuned on two datasets independently. The first is a subset of AudioSet [7]. We use the “Music Mood”<sup>2</sup> collection which contains music audio classified into seven classes, namely: *Happy, Funny, Sad, Tender, Exciting, Angry, and Scary*. We use the label of each audio as its corresponding text in the audio-text pairs. Due to the subjectivity of music retrieval, we focus on class-based descriptors instead of long-form descriptors as seen in MusicCap [9] to analyse the preliminary results in the current stage. We also use the Manga109 [8] dataset for manga-text pairs, which contains a collection of pages of 109 manga books from 12 genres. We use the genre of each manga as the corresponding text label for the manga-text pairs.

**Model Hyper-parameters.** The model is trained with an SGD optimizer, using a momentum of 0.9. The learning rate is 5e-5. While training on AudioSet, we trained the model for 30 epochs, used a batch size of 32, and applied audio augmentation techniques on the training set from [10]. For training on Manga109, we trained the model for 50 epochs, used a batch size of 64, and applied image normalisation techniques from CLIP [11] on the training and validation set. Since the AudioCLIP model is already pre-trained on AudioSet, we need fewer epochs to improve the performance on the music mood subset. We only saved the model with the best validation loss while training. For training the model with each dataset, we use a symmetric contrastive learning loss [13] for the (text, image) pairs with Manga109 and (text, audio) pairs with Audioset.

**Evaluation Methods.** Given the subjective nature of the task, our focus was on measuring the quantitative performance while training. We use the mean Average Precision (mAP) scores and mean accuracy scores for evaluating on validation sets of AudioSet and Manga109, respectively. For the qualitative performance, we report example retrievals based on input images and audios, and discuss our results here.

<sup>2</sup> [https://research.google.com/audioset/ontology/music\\_mood\\_1.html](https://research.google.com/audioset/ontology/music_mood_1.html)

### 3 Results and Discussion

Table 1 highlights the validation results on the two datasets, before and after fine-tuning the model. We achieve a 4% improvement on the Music-Mood subset of AudioSet after training the model further for 30 epochs. Unlike the original training in AudioCLIP [10], we train only on audio and text modalities to strengthen the relationship between music and text. We achieved high accuracy on the Manga109 dataset after training the model for 50 epochs compared to the zero-shot accuracy. The zero-shot model tends to classify images of manga panels as “fantasy” genre label. However, the pre-trained encoders from CLIP [11] fine-tune well on the manga dataset, and the model achieves over 97% accuracy on classifying input images. On the other hand, the audio-head is a much slower learner, and does not seem to fine-tune as well on the audio dataset.

**Table 1.** Classification results from the validation set. We used a train-validation split of 80-20. \*Original number of epochs for training the AudioCLIP on AudioSet [10]. \*\*Results from zero-shot classification using CLIP [11]

Dataset	Metric	Epochs	Score
AudioSet (Music Mood Subset)	mAP	60	40.8%
		30*	36.8%
Manga109	Accuracy	50	97.2%
		0**	6.9%

To understand the emergent trends of the model on the unaligned music and manga, we queried the model on 200 samples for image (manga) to music and vice versa. We then calculated the average confidence for each music mood retrieved given all genres of manga and vice versa. Our qualitative results imply some subjective emergence of relation between the moods of the music and the genre of the Manga. When querying images of genre “historical drama”, *Sad* (54% confidence) and *Tender* (45% confidence) are the most common moods of the retrieved audio files. Genres like “romantic comedy” retrieve *Happy* (37% confidence) while “suspense” retrieves *Scary* (34.4% confidence) music. However, certain genres like “horror” have very low confidence, with *Tender* music (19.6% confidence) being the strongest case. We note that the relationship music to manga retrieval is not as strong. “animal” is a common manga genre that is most likely to be retrieved when querying *Angry*, *Happy*, *Scary* and *Tender* music. We plan to continue experiments to understand the reason for this behaviour in future work.

Although there is no explicit baseline in our model, we compare our results to original AudioCLIP [10]. Interestingly, when we query any manga image, the model does not achieve confidence beyond 20%. The highest confidence is *Sad* music (15.6%) for “historical drama” manga. In general, the model has 5% confidence, and retrieved music with poor subjective compatibility, such as retrieving *Angry* music for “humour” and “romantic comedy” manga. Similarly, music to manga retrieval performs inconsistently, with 72% confidence for retrieving “romantic comedy” manga for *Angry* music. Overall, the model seems to achieve better zero-shot retrieval after training the model. This

implies the potential of the model for building a music generation system for Manga without aligned music and manga datasets.

#### 4 Future Work and Conclusion

We propose a task for retrieving music from pages of Japanese manga and demonstrate how strong multi-modal embeddings have the potential to solve the novel task through emergent properties. We plan to demonstrate this capability to a larger audience, and understand the behaviour of these emergent relationships. We view our work as preliminary to future work including, improving training efficiencies for the audio encoder, incorporating long form text such as manga dialogues and music descriptions, and conducting a qualitative test of the model. Ultimately, we plan to build a decoder for the model to generate novel music for any given Manga.

#### References

1. Fog, K., Budtz, C., Yakaboylu, B.: *Storytelling*. Springer (2005)
2. Bouissou, J. M.: Manga: A historical overview. *Manga: An anthology of global and cultural perspectives*. 17-33 (2010)
3. Wong, T. T., Igarashi, T., Xu, Y. Q., Shi, D.: The Computational manga and anime. In *SIGGRAPH Asia 2013 Courses*. 1–52 (2013)
4. Shriram, J., Tapaswi, M., Alluri, V.: Sonus Texere! Automated Dense Soundtrack Construction for Books using Movie Adaptations. In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. (2022)
5. Brenner, Robin E.: *Understanding manga and anime*. Greenwood Publishing Group (2007)
6. Girdhar, R., El-Nouby, A., Liu, Z., Singh, M., Alwala, K. V., Joulin, A., Misra, I.: Imagebind: One embedding space to bind them all. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190. (2023)
7. Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M.: Audio set: An ontology and human-labeled dataset for audio events. In: *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 776–780. IEEE (2017)
8. Fujimoto, A., Ogawa, T., Yamamoto, K., Matsui, Y., Yamasaki, T., Aizawa, K.: Manga109 dataset and creation of metadata. In: *Proceedings of the 1st international workshop on comics analysis, processing and understanding*, pp. 1–5. IEEE (2016)
9. Agostinelli, A., Denk, T.I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., Sharifi, M.: Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325* (2023)
10. Guzhov, A., Raue, F., Hees, J., Dengel, A.: Audioclip: Extending clip to image, text and audio. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980. IEEE (2022)
11. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G.: Learning transferable visual models from natural language supervision. In: *International conference on machine learning*, pp. 8748–8763. PMLR (2021)
12. Guzhov, A., Raue, F., Hees, J., Dengel, A.: Esresne (x) t-fbsp: Learning robust time-frequency transformation of audio. In: *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE (2021)
13. Oord, A.V.D., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018)

# Visualizing Musical Structure of House Music

Justin Tomoya Wulf<sup>1</sup> and Tetsuro Kitahara<sup>1</sup> \*

<sup>1</sup> College of Humanities and Sciences, Nihon University, Japan  
{wulf, kitahara}@kthrlab.jp

**Abstract.** This paper describes a simple method for visualizing the musical structure of house music. A given audio signal is separated into drums, bass, vocals, and others, and then the sound pressure of each instrument part is calculated and visualized. Using those sound pressures, the audio signal is segmented into four sections: intro, drop, break, and outro. A preliminary analysis revealed that the drums and bass have a significant impact in delineating musical structure.

**Keywords:** House Music, Musical Structure, Music Analysis, Visualization, RMS, Audio Segmentation

## 1 Introduction

House music is a type of dance music that has a different structure from that of popular music. Whereas popular music often makes a structure by changing the characteristics of the melody and chord progression, house music enhances groove by repeating the same music loops and creates movement by adding new music loops and/or removing the added music loops.

Visualizing such a structure of house music will provide an opportunity for a deeper understanding of individual pieces. Therefore, we aim to develop a system that provides useful information to composers and DJs by analyzing and visualizing various house music pieces.

There have been many methods for visualizing musical structure in musical compositions, such as those based on a self-similarity matrix[1], a greedy search algorithm[2], and pattern matching of note sequences[3]. However, a house-specific method has not yet been proposed.

This paper describes preliminary results of visualizing the musical structure of house music. A given audio signal is first separated into drums, bass, vocals, and others, and then their sound pressures are calculated. By visualizing the sound pressure of each instrument part, it enables the user to grasp the musical structure.



This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

\* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.



## 2 Proposed System

Although no clear definition exists for the designation of musical structure in house and other dance music, this paper defines the following four sections.

- Intro: Introductory section of a song
- Drop: The climax of the song
- Break: The part of the song other than the climax
- Outro: The end of the song

The drop corresponds to the 'chorus' in popular music, and the break is similarly positioned to the interlude after the chorus or verses A and B before the chorus. In house music, new sound materials called music loops are often added to give the music a more lively feel. Therefore, the musical structure as described here is expected to be highly related to sound pressure.

### 2.1 Pre-processing of sound sources

From a given audio signal, the sound source for each part (drums, bass, vocals, other) is extracted. We use Demucs for sound source separation. The sound source format is mp3 and the bit rate is 320 kbps, and the sampling frequency is 44.1 kHz.

### 2.2 Calculation of sound pressure for each part and unseparated sound source

The sound pressure is calculated as the root mean square (RMS) of the waveform of the target sound source. Specifically, the sound pressure  $S_i(t)$  at waveform  $y(t)$  is expressed by

$$S_i(t) = \sqrt{\frac{1}{T} \sum_{\tau=0}^T y(t + \tau)^2 dt}$$

where  $i$  denotes the part ( $i \in \{\text{drums, bass, vocals, others, mixed}\}$ ). The window width  $T$  for calculating RMS was set to 65000 samples, and the time resolution (time interval of  $S(t)$ ) was set to 16250 samples.

The RMS is calculated using Librosa, a Python module for music analysis. The RMS values of each part are then normalized so that the maximum value of the pre-separation source is 1 and the minimum value is 0.

### 2.3 Drawing Graphs

Using  $S_{\text{bass}}$ ,  $S_{\text{vocals}}$ ,  $S_{\text{others}}$  as well as RMS of the pre-separation source  $S_{\text{mixed}}$ , the audio signal is segmented into four sections (Intro, Drop, Break, Outro) The judgment criteria for each section are as follows.

- Intro (yellow): The section from the start of the song to the time when  $S_{\text{mixed}}$  first exceeds 0.85

- Drop (red): The interval where  $S_{\text{mixed}}$  exceeds 0.85
- Break (green): Interval when  $S_{\text{mixed}}$  is below 0.85 (excluding intro and outro)
- Outro (blue): the interval from the last time when  $S_{\text{mixed}}$  exceeds 0.85 to the end of the song

This criterion is based on the hypothesis that the drop corresponding to the chorus is basically louder than the other sections; the threshold of 0.85 was determined experimentally by testing several songs.

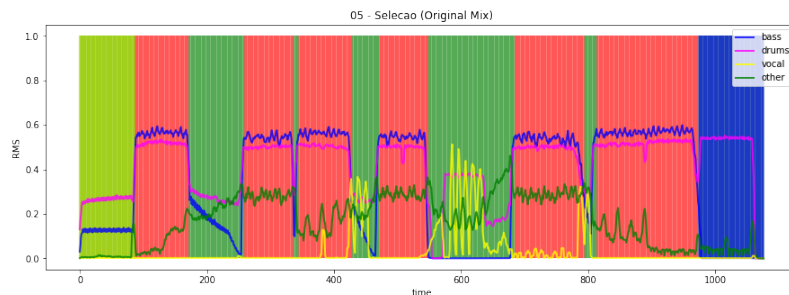
### 3 Preliminary Results

We conducted a preliminary experiment on visualizing house music using the method described in Section 2. The following songs were used.

**Piece 1** Seleccion - Mark Knight, Shovell

**Piece 2** Phoenix - Daft Punk

The results are shown in Figs. 1 and 2.



**Fig. 1.** Seleccion - Mark Knight, Shovell

For Piece 1 (Fig. 1), we can see that the four sections are appropriately divided and that some instrument parts enter and/or leave at the boundaries of the sections. For example, the sound pressure of the drums and bass increases when switching from the intro to the first drop. In the middle of the piece, the drums and bass leave as soon as the drop is over, the vocalist enters, and in the next drop, the vocalist leaves again and the drums and bass enter. This piece has a typical structure of house music, so the visualization is generally functioning.

On the other hand, Fig. 2 shows that almost all sections were segmented into the drop. This is because the drum sound continues with high sound pressure and it makes the sound pressure of the mixed (pre-separation) source very high in almost all sections. This implies that more sophisticated criteria will be needed to improve the judgment of the musical structure.

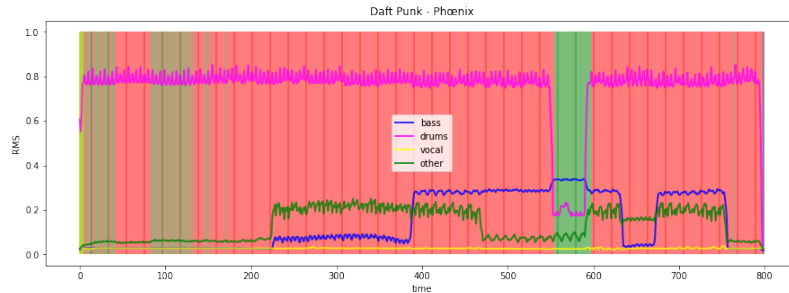


Fig. 2. Phoenix - Daft Punk

In those two pieces, drums and bass are always played during the drop section, indicating that these two parts have a significant influence on the sound pressure. Otherwise, the bass does not sound in most of the break sections, indicating that the bass tends to leave at transitions from the drop to the break. From those observations, we can consider that the drums and bass play an important role in house music to make movement from one section to another section.

## 4 Conclusion

In this paper, we attempted to visualize the music structure of house music by drawing the sound pressure of each instrument and by coloring sections judged with the sound pressure of the pre-separation sources.

The preliminary experiments revealed some influences of each part on the musical structure. In particular, drums and bass play a role in transitions of sections (e.g. entering and leaving the drop) in house music.

In the future, we plan to visualize richer information, such as detailed drum patterns obtained with a drum transcription technique. In addition, we have to make more sophisticated criteria to decide section boundaries, for example, using a technique for detecting repetitive patterns.

## References

1. Jonathan Foote: Visualizing Music and Audio using Self-Similarity, Proceedings of the Seventh ACM International Conference on Multimedia (Part 1), pp. 77–80 (1999).
2. Jouni Paulus, and Anssi Klapuri: Music Structure Analysis Using a Probabilistic Fitness Measure and a Greedy Search Algorithm, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 17, Issue: 6, pp. 1159–1170 (2009).
3. Aki Hayashi, Takayuki Itoh, and Masaki Matsubara: Colorscore - Visualization and Condensation of Structure of Classical Music -, 2011 15th International Conference on Information Visualisation, pp. 420-425 (2011).

## **Music**

### **Acts for Hacks**

*Αταραξία (Ataraxia) (Vasilis Agiomyrgianakis and Haruka Hirayama)*

### **La Solitudine Delle Moltitudini (The Solitude of the Multitudes)**

*Marco Buongiorno Nardelli, Alice Grishchenko, Gabor Kitzinger and A.Laszlo Barabasi*

### **Construction in ENSO**

*Ryo Ikeshiro*

### **Thee Doug Van Nort Electro-Acoustic Orchestra**

*Doug Van Nort*

# **Author Index**

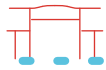
- Abiki, Kaito, 680  
Alessandrini, Patricia, 700  
Alshamrani, Hani, 667  
Amerotti, Marco, 277  
Aoyagi, Saizo, 680  
Aramaki, Mitsuko, 748  
Armstrong, Kit, 744  
Arzamendia Lopez, Arturo Alejandro, 732  
Baba, Kanako, 78  
Badineni, Amulya, 379  
Baker, Tom, 411  
Barthet, Mathieu, 42, 110, 122, 740  
Basica, Constantin, 700  
Batliner, Anton, 587  
Bell, Bryan Jacob, 599  
Beller, Grégory, 301  
Benford, Steve, 277  
Bernard, Corentin, 748  
Bernardes, Gilberto, 512  
Bizzarri, Matteo, 619  
Buongiorno Nardelli, Marco, 474  
C. Moss, Fabian, 450  
Caetano, Marcelo, 716  
Cai, Le, 667  
Calvo-Zaragoza, Jorge, 18, 760  
Cardoso, F. Amílcar, 221  
Carr, CJ, 110, 122  
Chang, Yung-Chuan, 442  
Chattopadhyay, Swarup, 257  
Chen, Ke, 411  
Choi, Eunjin, 186  
Climent, Ricardo, 411  
Collins, Tolly, 740  
Curcio, Gabrielle, 379  
D'Alessandro, Christophe, 704, 756  
Daquin, Mathieu, 631  
De Filippi, Eleonora, 54  
Diamond, Danny, 631  
Dziwis, Damian, 289  
Eipert, Tim, 450  
Fang, Gengfa, 667  
Fazekas, George, 86  
Fazekas, György, 134, 265  
Ferguson, Sam, 667  
Foscarin, Francesco, 209  
Fournier-S'niehotta, Raphaël, 631  
Frisk, Henrik, 419  
Fuentes-Martínez, Eliseo, 18, 760  
Garriga, Adan, 54  
Glette, Kyrre, 347  
Gold, Omer, 555  
Görne, Thomas, 301  
Goto, Madoka, 536  
Goto, Masataka, 30, 655  
Graf, Max, 42  
Hadjakos, Aristotelis, 335  
Hajdu, Georg, 301  
Hamanaka, Masatoshi, 611, 692, 772  
Hamasaki, Masahiro, 655  
Hattori, Akira, 680  
Hayashi, Ryusei, 643  
Hijikata, Yoshinori, 655  
Hirai, Tatsunori, 158, 367, 680, 764  
Hirata, Keiji, 66, 575, 611, 776  
Hirawata, So, 524, 736  
Honda, Ken, 680  
Hori, Gen, 403  
Horibe, Takanori, 728  
Horita, Tatsuya, 395, 684  
Hoy, Rory, 323  
Huang, Jing-Heng, 744  
Huang, Ji-Xuan, 744

- Huang, Yu-Fen, 245  
Huerta, Juan M., 98  
Hugar, Nitin, 567  
Hung, Tzu-Ching, 744  
Hyrkas, Jeremy, 500  
Iino, Nami, 146, 692  
Ikeda, Yusuke, 359  
Ito, Akinori, 732  
Izu, Risa, 575, 776  
Jacquemard, Florent, 209  
Jeong, Dasaem, 186  
Julio, Christofer, 233  
Kagawa, Rina, 146  
Kalonaris, Stefano, 555  
Karystinaios, Emmanouil, 209  
Kato, Takuya, 764  
Kawada, Taku, 395, 684  
Kawahara, Mizuki, 720  
Kim, Halla, 484  
Kim, Hyerin, 186  
Kim, Hyon, 197  
Kishi, Mayuko, 655  
Kitahara, Tetsuro, 78, 178, 547, 643, 688, 708, 720, 724, 784  
Kitaya, Koki, 78  
Koguchi, Junya, 170  
Kouzai, Tomoo, 178, 720  
Kronland-Martinet, Richard, 716, 748  
Kumar, Adarsh, 110  
Lee, Feng-Hsu, 233  
Lee, Jason Yin Hei, 599  
Lei, Qinying, 567  
Li, Shengchen, 86  
Li, Yuqiang, 86  
Lindell, Rikard, 419  
Liu, Bo, 98  
Liu, Yi-Wen, 442, 744  
Locqueville, Gregoire, 704  
Loth, Jackson, 122  
Lucas, Pedro, 347  
Lucas, Thomas, 704  
Manzolli, Jônatas, 221  
Martínez-Sevilla, Juan C., 18, 760  
Martins, Pedro, 221  
Masuda, Taro, 78  
Matsubara, Masaki, 66, 146  
Matsumura, Yuya, 78  
Matsuura, Mio, 66  
McCloskey, Matthew, 379  
McDermott, James, 631  
McGrath, Kevin, 379  
McKay, Cory, 752  
Mibayashi, Ryota, 30  
Mikami, Koji, 732  
Misra, Chandan, 257  
Miura, Hiroya, 692  
Miyaguchi, Sora, 359  
Molina Villota, D. H., 704, 756  
Morise, Masanori, 170, 728  
Nabeoka, Takuto, 387  
Nagahama, Toru, 395, 684  
Nagasaka, Lamo, 764  
Nakamura, Eita, 387, 450, 462  
Nakano, Tomoyasu, 30, 655  
Nakashika, Toru, 6, 403  
Nam, Juhan, 186  
Nandi, Arijit, 54  
Neves, João, 221  
Nishimura, Takuichi, 692  
Nowakowski, Matthias, 335  
Numao, Masayuki, 524  
Odaira, Tsuyoshi, 78  
Ohshima, Hiroaki, 30  
Okabe, Daisuke, 524, 736  
Okuta, Masaki, 708  
Osaka, Naotoshi, 359

- Oshita, Sai, 547  
Otani, Noriko, 524, 736  
Ottolin, Thomas, 567  
Papamichail, Dimitris, 379 Papamichail, Georgios, 379  
Parada-Cabaleiro, Emilia, 430, 587  
Park, Dongju, 492  
Park, Juyong, 484, 492  
Pereda Baños, Alexandre, 54  
Poirot, Samuel, 748  
Poudrier, Ève, 599  
Puspitasari, Mastuti, 403  
Rela, M. Zenha, 221  
Rigaux, Philippe, 631  
Rizo, David, 18, 760  
Roselló, Adrián, 18, 760  
Row, Eleanor, 265  
Sá Pinto, António, 512  
Sagayama, Shigeki, 3, 6, 403  
Sakai, Masahiko, 209, 536  
Sakai, Shunsuke, 688, 724  
Sankaranarayanan, Raghavasimhan, 567  
Sapp, Craig Stuart, 599  
Sarmiento, Pedro, 110, 122  
Sasaki, Ayane, 66  
Sasaki, Momoka, 655  
Schedl, Markus, 430, 587  
Schmele, Timothy, 54  
Schmitt, Maximilian, 587  
Schuller, Björn, 587  
Segawa, Hinata, 688, 724  
Seiça, Mariana, 221  
Sello, Jacob, 301  
Serra, Xavier, 197  
Sharma, Megha, 780  
Shinjo, Shoyu, 768  
Stadler, Antonia, 430  
Stone, Peter, 98  
Sturm, Bob L. T., 277  
Su, Li, 233, 245  
Su, Tung-Cheng, 442  
Takahashi, Riku, 575  
Takahashi, Takuya, 6, 403  
Takeda, Hideaki, 146, 692  
Takegawa, Yoshinari, 66, 575, 776  
Takezawa, Ayari, 78  
Tang, Jingjing, 134, 265  
Tojo, Satoshi, 209, 536, 611  
Tomiyama, Yoshitaka, 78  
Tsukuda, Kosetsu, 30  
Tsuruoka, Yoshimasa, 780  
Uemura, Aiko, 768  
Van Nort, Doug, 323  
Vear, Craig, 277  
Verma, Prateek, 700  
Watanabe, Kento, 30  
Weinberg, Gil, 567  
Widmer, Gerhard, 209  
Wiggins, Geraint, 134  
Wulf, Justin Tomoya, 784  
Xie, Dekun, 110  
Yamaguchi, Yasumasa, 395, 684  
Yamamoto, Takehiro, 30  
Yeo, Woon Seung, 696  
Yoon, Ji Won, 696  
Yoshii, Kazuyoshi, 387  
Ystad, Sølvi, 748  
Zalles Ballivian, Gabriel, 712  
Zhu, Tiange, 631  
Zukowski, Zack, 110, 122







Distinguished Sponsor:

**SPECIAL INTEREST GROUP ON MUSIC AND COMPUTER (SIGMUS), IPSJ**

Gold Sponsor:

**Piano Teachers' National Association of Japan (PTNA)**

Silver Sponsor:

**Yamaha Corporation**

Commercial Sponsor:

**CRYPTON FUTURE MEDIA, INC.**

