COMPUTATIONAL CREATIVITY IN MUSIC: MUSIC GENERATION AND STYLE TRANSFER

ZHEJING HU

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

Department of Computing

Computational Creativity in Music: Music Generation and Style Transfer

Zhejing Hu

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

October 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Zhejing Hu (Name of student)

Abstract

Computational creativity, an interdisciplinary field blending artificial intelligence and psychology, simulates creative thought and behavior. This field has seen remarkable advancements in artistic creation in recent times, such as transferring an image captured with a smartphone into a Picasso-style painting using deep learning techniques. Studies in computational artistic creation highlight intriguing aspects of human intelligence and hold great potential for advancing true machine intelligence. Among the various subareas of artistic creation using computational models, this thesis focuses on computational creativity in music, covering an array of valuable and challenging problems. For instance, the music generation problem requires the study of abstract music structure, while music style transfer necessitates studying content and style features from music in different application scenarios. Despite current advancements, music generation and music style transfer still have room for improvement, necessitating the development of new techniques.

Thus, we propose a novel framework to improve the performance of music generation and music style transfer via computational creativity models. First, we propose a Motif-to-music Generation Model (MGM) that studies the structure of music at the motif level and generates a complete piece of music based on a few key motifs. MGM contains a motif-level repetition generator (MRG) and an outline-to-music generator (O2MG). MRG learns to generate motif-level repetitions, while O2MG learns to generate a complete piece of music based on the music's outline. MGM is trained on a new music repetition dataset (MRD), mitigating the problem that machine-composed music tends to be random and structureless. Both subjective and objective experimental results demonstrate that MGM can generate different motif-level repetitions and a complete piece of music of high quality.

Second, we introduce a novel Call-Response Generator (CRG) featuring a knowledge-enhanced mechanism that studies the structure of music at the phrase level and generates the response of the music given the call. We manually segment 909 pop songs and label 19,155 call-response pairs, design a knowledge-enhancement module to select instructive training data based on rhythm, melody, and harmony quality, and train the composition module on the call-response pairs, augmenting it with musical knowledge. Our experiments show that the proposed model suc-

cessfully generates a wide variety of engaging and appealing responses for different musical calls, enhancing the quality of machine-composed music.

Third, we introduce a novel transfer model for a new music style transfer task, therapeutic music transfer. The limited availability of suitable music pieces for different individuals has posed a challenge in the field of music therapy. We create therapeutic music according to the input music even if the data sample is limited. Given the limited nature of therapeutic music and a user's favorite music, we design a new domain adaptation algorithm that transfers the learning result for music genre classification to therapeutic music transfer. We then utilize a joint minimization technique to optimize the output music. Subjective experiments on anxiety sufferers prove that the customized therapeutic music has achieved better and stable performance in anxiety reduction.

Fourth, we introduce the User Preference Transformer (UP-Transformer) for a new music style transfer task referred to as User Preference Music Transfer (UPMT). This task helps to tailor music to a user's individual preferences, which can significantly enhance musical diversity and improve mental health. The UP-Transformer is a combined model that utilizes deep learning for training and relies on prior knowledge of just a single piece of music favored by the user for inference. We improve the Transformer-based model by introducing a new loss function, the favorite-aware loss function, which takes into account the user's favorite music. During the inference phase, we perform UPMT using the musical knowledge extracted from the user's favorite piece. To address the problem of evaluating melodic similarity in music style transfer, we propose a new metric called pattern similarity (PS) to assess the similarity between two pieces of music. Both objective and subjective evaluations demonstrate that the music transferred using our proposed method outperforms traditional methods in terms of musicality, similarity, and user preferences.

Publications Arising from the Thesis

1. **Zhejing Hu**, Yan Liu, Gong Chen, Sheng-hua Zhong, and Aiwei Zhang. "Make Your Favorite Music Curative: Music Style Transfer for Anxiety Reduction." ACM Multimedia, 2020.

2. **Zhejing Hu**, Yan Liu, Gong Chen, and Yongxu Liu. "Can Machines Generate Personalized Music A Hybrid Favorite-aware Method for User Preference Music Transfer." IEEE Transactions on Multimedia, 2022.

3. **Zhejing Hu**, Xiao Ma, Yan Liu, Gong Chen, and Yongxu Liu. "The Beauty of Repetition in Machine Composition Scenarios." ACM Multimedia, 2022.

4. **Zhejing Hu**, Xiao Ma, Yan Liu, Gong Chen, Yongxu Liu, and Roger B. Dannenberg, "The Beauty of Repetition: an Algorithmic Composition Model with Motif-level Repetition Generator and Outline-to-music Generator." IEEE Transactions on Multimedia, 2023.

5. **Zhejing Hu**, Gong Chen, Yan Liu, Xiao Ma, Nianhong Guan and Xiaoying Wang. "Make a Song Curative: A Spatio-temporal Therapeutic Music Transfer Model for Anxiety Reduction." Experts Systems and Applications, 2023.

6. Gong Chen, **Zhejing Hu**, Nianhong Guan, and Xiaoying Wang. "Finding Therapeutic Music for Anxiety Using Scoring Model." International Journal of Intelligent Systems, 2021.

7. Yongxu Liu, Yan Liu, Bruce X.B. Yu, Shenghua Zhong, and **Zhejing Hu**. "Noise-robust Oversampling for Mixed-type and Multi-class Imbalanced Data Classification." Pattern Recognition, 2023.

8. **Zhejing Hu**, Yan Liu, Gong Chen, Xiao Ma, and Shenghua Zhong. "Responding to the Call: Exploring Machine Composition Using a Knowledge-Enhanced Model." Submitted to AAAI, 2024.

Acknowledgements

I would like to express my deepest gratitude to my advisor Dr. Yan LIU, for her continuous support, patience, guidance, and encouragement throughout my doctoral studies. Her expertise, invaluable insights, and unwavering commitment to my academic and personal development have been instrumental in shaping my research and career.

Furthermore, I would like to thank my colleagues, Dr. Gong CHEN, Dr. Bruce X.B. YU, Dr. Yongxu LIU, Miss Xiang ZHANG, Mr. Zhi ZHANG, and Mr. Xiao MA for their support, encouragement, and inspiration throughout my doctoral studies. I have had the privilege of working with some of the most brilliant and dedicated researchers in my field, and I am grateful for the opportunities and experiences that we have shared.

I am grateful to The Hong Kong Polytechnic University for providing me with the resources, infrastructure, and opportunities to pursue my academic and research goals. I also acknowledge the financial support I have received that has enabled me to carry out my research.

Finally, I would like to express my gratitude to my family for their unwavering love, support, and encouragement throughout my academic journey. Their sacrifices, understanding, and support have been instrumental in enabling me to pursue my academic and personal goals.

This work would not have been possible without the support of these individuals and organizations, and for that, I am truly grateful.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Creativity is a driving force of human development in science and culture. Among types of creative output, works of art such as paintings and music are often the most enigmatic. Artistic creativity is highly valued across diverse cultures. Even though artwork is not essential for survival, it has existed since before ancient civilizations and has been developed over thousands of years. Researchers from various fields, such as artificial intelligence and multimedia computing, have sought to simulate artistic creation using computational models in order to approximate human intelligence. Studying computational creativity is also beneficial for understanding motivations and uncovering the secrets of human intelligence.

Among the subareas of artistic creation using computational models, this thesis focuses on computational creativity in music because it encompasses many worthwhile yet complex problems. For example, compared to painting, music is more abstract. Music generation requires a model that can learn the abstract structure of music. Music style transfer necessitates a model that can distinguish style features from content features in different application scenarios, which is a nuanced task in music. Although music has long been produced using computational approaches, current performance leaves room for improvement. The question of how to build a model that can truly understand the process of music creation has been a captivating but elusive goal. This thesis proposes a novel framework to tackle various problems in music generation and music style transfer using computational creativity models. In this chapter, we discuss the motivation behind the proposed framework, describe the structure of the proposed computational modeling framework, and outline the organization of the thesis.

1.1 Motivation

Within computational creativity in music, music generation (Chapter 2.1) and music style transfer (Chapter 2.2) are two main tasks. Music generation aims to create a new piece of music from scratch or from some prompts, while music style transfer aims to transform an existing piece of music from style A to style B without losing the content of the music. Despite great efforts devoted to these tasks over past decades, the solutions for generating and transferring music could be improved.

There exists a hierarchical structure of music representation, with musical scores at the top, audio sounds at the bottom, and performance control in the middle. Consequently, the process of creating music can also be divided into three stages [5], as shown in Figure 1.1: the first stage involves composers creating musical scores; the second stage involves performers interpreting scores; and the third stage involves the audio rendering of the performance for the listeners. These stages correspond to three levels of music generation: score generation, performance generation, and audio generation [6]. All three stages play crucial roles in artistic expression concerning music generation.

This thesis concentrates on the first stage of music generation, namely, score generation, also referred to as algorithmic composition or machine composition. This stage deals with Musical instrument digital interface (MIDI) data, including pitch, chord, and tonality, as well as rich structural information such as motifs and phrases. To be heard, the music generated through score generation will be converted into audio file formats, such as MP3 or WAV.

Recent research has shown that exploring the hierarchical structure of music can lead to the creation of music that is structured and rhythmic, similar to human-composed work [7–10]. However, existing models have not adequately addressed the issue of repetition and call-and-response in music, two essential hierarchical elements in human-composed music, and thus are unable to control the musical trend effectively.

In terms of music style transfer, the concept was first proposed in the computer vision domain, with input transferred from one style to another [11]. Unlike image style transfer, the boundary between content and style in music is highly dynamic, and different objectives relate to distinct style transfer problems. Current style transfer work mainly focuses on genre transfer.

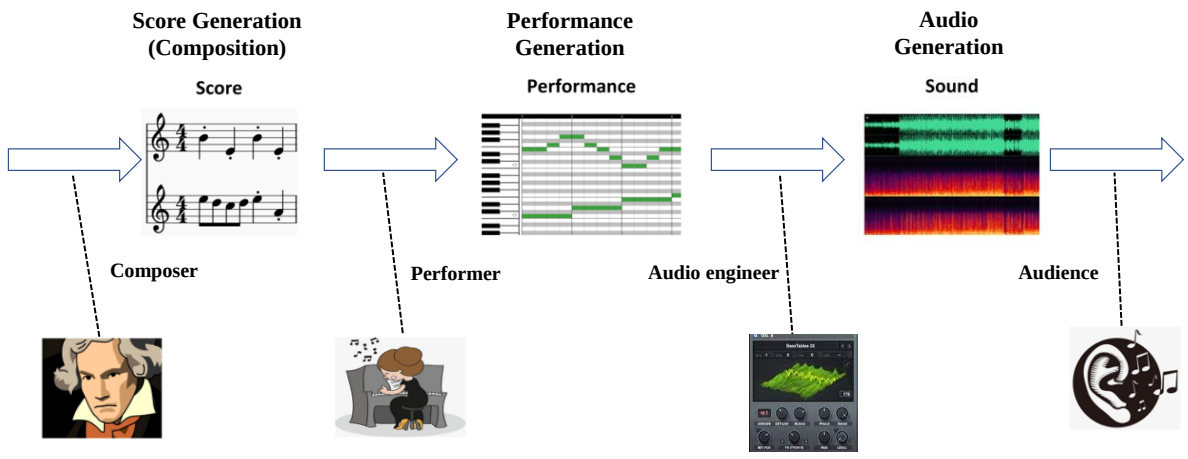


Figure 1.1: The process of music generation.

However, current music style transfer methods are not directly applicable to more applications because more aspects of composition style transfer need to be studied, and data insufficiency remains a significant challenge in real-world scenarios, such as music therapy and music personalization.

In this thesis, we tackle the crucial and underlying problems of limited composition elements and data inadequacy in the field of music style transfer. We have undertaken two studies to address these issues. The details of the challenges and the solutions we have proposed will be explored in subsequent chapters.

1.2 Proposed Framework

In this thesis, we present four key works in the fields of music generation and music style transfer.

For music generation, we propose two works focused on motif-level and phrase-level elements. We begin by generating motif-level music repetition, which is the most fundamental unit in a music piece, and propose the creation of complete music from motifs: 1) "The Beauty of Repetition: An Algorithmic Composition Model with a Motif-level Repetition Generator and Outline-to-music Generator". Then, we study the phrase-level structure of music by exploring

call-and-response skills in music composition and propose one work: 2) “Responding to the Call: Exploring Phrase-level Algorithmic Composition Using a Knowledge-Enhanced Model”.

For music style transfer, we concentrate on the application of style transfer in different scenarios suffering from the problem of data insufficiency, especially in therapeutic music transfer and personalized music transfer. For therapeutic music transfer, we propose one work: 3) “A Domain Adaptation Method for Therapeutic Music Transfer”. For personalized music transfer, we propose one work: 4) “A Hybrid Favorite-aware Method for Personalized Music Transfer”.

1.2.1 The Beauty of Repetition: a Motif-level Algorithmic Composition Model

Within the realm of music creation, the role of recurring elements is crucial, enhancing the overall allure of the compositions. However, this element is often underserved in computational music composition studies. This research endeavors to bridge this void, introducing an innovative technique for the construction and integration of motif-level repetitions into musical pieces. We employ a blend of empirical and theoretical knowledge-driven methodologies. Our Motif-to-Music Generation Model (MGM) is a synergistic integration of the Motif-Level Repetition Generator (MRG) and Outline-to-Music Generator (O2MG), refined through training on the specially compiled music repetition dataset (MRD). In the MRG component, the Transformer encoder is attuned to the nuances of musical notes’ representations sourced from MRD, while a specialized learner, attuned to repetition dynamics grounded in musical theory, is also incorporated. On the other hand, O2MG employs an unprecedented outline-to-music educational paradigm, dedicated to discerning the intricate connections between motif-level repetitive elements and the ensuing musical creations. MRD is a rich repository featuring 584,329 motif repetition instances and an additional 3,545 outline-music sequences, extracted from the genre of pop piano compositions. Experimental outcomes underscore MGM’s prowess in crafting varied and aesthetically pleasing repetitive elements tailored to specific motifs [12].

1.2.2 Responding to the Call: Exploring Phrase-level Algorithmic Composition Using a Knowledge-Enhanced Model

Call-and-response is a musical technique that enriches the artistic quality of music by crafting coherent musical ideas, reflecting the back-and-forth nature of human dialogue with distinct musical characteristics. While this technique plays an essential role in numerous musical works, it remains underexplored in machine learning-based composition. To improve the artistry of machine-composed music, we propose a novel Call-Response Generator (CRG) featuring a knowledge-enhanced mechanism. First, we manually segment 909 pop songs and label 19,155 call-response pairs. Second, we design a knowledge-enhancement module to select instructive training data based on the quality of rhythm, melody, and harmony. Third, we train the composition module on the call-response pairs, augmenting it with musical knowledge. Our experiments demonstrate that the proposed model successfully generates a wide variety of engaging and appealing responses for different musical calls, enhancing the quality of machine-composed music.

1.2.3 A Domain Adaptation Method for Therapeutic Music Transfer

The use of music therapy has been established to alleviate anxiety through clinical trials. However, the limited availability of suitable music pieces for different individuals has become a challenge in the field of music therapy. To overcome this limitation, we present a new approach for personalizing therapeutic music using a user's favorite song. Our method involves extracting music features using a music feature extraction network and utilizing a domain adaptation algorithm to transfer learning results from music genre classification to therapeutic music transfer. Three Convolutional Neural Networks are employed to minimize the loss functions and generate custom therapeutic music with just one input of the user's preferred song. Our experiments on individuals with anxiety have shown that the personalized therapeutic music provides more effective and consistent results in reducing anxiety [13].

1.2.4 A Hybrid Favorite-aware Method for User Preference Music Transfer

A new and understudied problem in music style transfer called User Preference Music Transfer (UPMT) is explored. The goal of UPMT is to tailor music to a user's individual preferences, which can greatly enhance musical diversity and improve mental health. Most music style transfer methods depend on data-driven techniques, but building a comprehensive training dataset is often challenging because users can rarely provide enough of their favorite songs. To overcome this obstacle, a novel hybrid approach called User Preference Transformer (UP-Transformer) is introduced. The UP-Transformer model first fine-tuned the Transformer-XL model based on a newly designed favorite-aware loss function. Then, a two-step transfer process is proposed to perform UPMT based on the patterns extracted from the user's favorite music in the inference phase. Moreover, a new concept of pattern similarity (PS) is introduced to evaluate melodic similarity in music style transfer and is found to be consistent with qualitative similarity scores through statistical testing. Results of user testing show that the transferred music produced by the UP-Transformer model outperforms traditional methods in musicality, similarity, and user preferences [14].

1.3 Organization

The organization of this thesis is illustrated in Figure 1.2. We review related works in music generation and music style transfer in Chapter 2. The content of Chapters 3 and 4 corresponds to music generation, while Chapters 5 and 6 correspond to music style transfer. Chapter 3 presents a novel Motif-to-Music Generation Model (MGM) that can generate a complete piece of music based on one motif or several motifs. Chapter 4 presents a novel Call-and-Response Generator (CRG) that can generate a response based on the call. In Chapter 5, we propose a novel domain adaptation method for therapeutic music transfer. In Chapter 6, we study a new problem, user preference music transfer, and propose a novel User Preference Transformer (UP-Transformer) that can transfer a piece of music to fit user preferences. The thesis concludes with future work in Chapter 7.

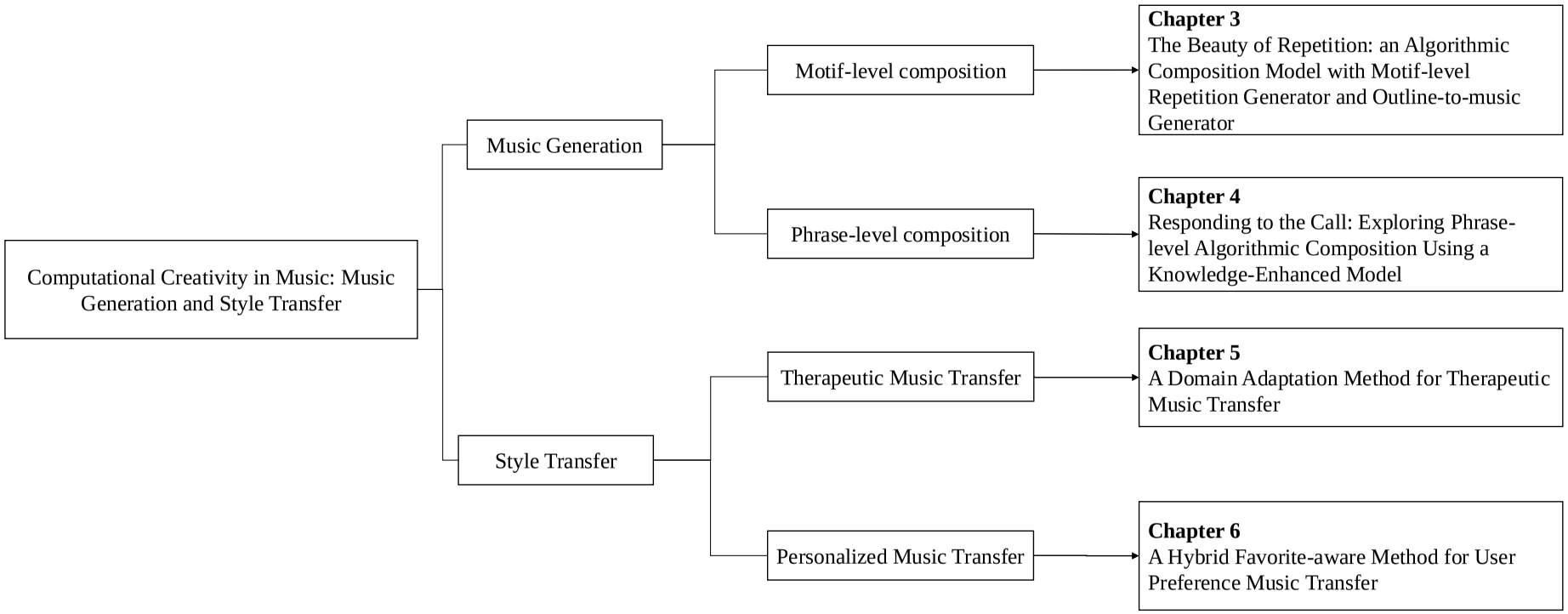


Figure 1.2: Framework of this thesis.

Chapter 2

Literature Review

The proposed framework in computational creativity in music can be divided into music generation and music style transfer. In this chapter, we describe some representative methods in these two tasks.

2.1 Music Generation

The Illiac Suite by Hiller and Isaacson is a well-known early work in modern music generation, which utilized a Markov chain to generate notes in a string quartet, and then filtered the generated material based on desired properties using rules [15]. Music theory has since been incorporated into computational model design to improve the structure, melody, and harmony of generated music [16]. Additionally, programming languages specific to composition have been developed based on musical rules [17].

Advances in machine learning have spawned a number of generative models in music. Boulanger-Lewandowski *et al.* proposed a model called RNN-RBM that uses deep learning to learn the rules governing harmony and rhythm in polyphonic music scores [18]. Since then, more advanced generative models such as VAEs, GANs, and Transformers have been proposed to generate diverse musical content in new ways [19]. Dong *et al.* proposed Muse-GAN to generate multi-track music, which is considered to be the first multi-track music generation model [20]. MusicVAE was proposed to capture polyphonic music's long-term structure and compose music [21]. Morgen proposed a C-RNN-GAN that works on a classical music col-

lection [22]. To enhance model performance on long sequences, some researchers explored a Transformer-based architecture for music generation [23, 24]. The initial model that employed the Transformer architecture in music generation, called Music Transformer [23], demonstrated the capability of a Transformer model to generate coherent and lengthy polyphonic piano music with suitable variations and repetitions. Following this, several Transformer-based models have been proposed to generate music, such as those in [24–31]. These models can learn musical features automatically, without the need for manual rule definition, and can produce a wide variety of music pieces.

Recently, researchers realized that inserting music knowledge into the deep learning model helps to generate music since composition is a skill that requires learning from music theory and music examples. Therefore, a new trend is to propose a hybrid model that incorporates music knowledge. For example, Shih *et al.* proposed a Theme-Transformer that introduces the idea of "music theme" into the model so the model can learn the theme of the music [32].

2.2 Music Style Transfer

In music style transfer, researchers generally focus on transferring the genre and rhythm of a piece of music, as these are the most easily recognizable elements of style for humans. The main challenge in current music style transfer research is the lack of aligned data, causing most studies to adopt unsupervised learning frameworks.

For genre transfer, Brunner *et al.* employed a MIDI-VAE network to modify the genre of a musical piece by automatically adjusting pitch, dynamics, and instrumentation, marking the first instance of successful unaligned style transfer in complete musical compositions [33]. Later, they used CycleGAN to transfer between two genres based on note pitches, which was the first application of GANs in symbolic music transfer [34]. Lu *et al.* used the melody as a condition and transferred the accompaniment of the music using WaveNet [35]. This allowed them to transfer an arbitrary piece of homophonic music to the styles of Bach's 4-part chorales or Jazz. Wang *et al.* proposed a two-GAN architecture that facilitates information transfer between styles A and B in both directions [36]. One GAN is used to perform style transfer from A to B, while the other GAN is used to transfer information from B to A. To address the issue of lacking

aligned data, Cífka *et al.* designed a synthetic data generation scheme to generate aligned data, and created a one-shot style transfer method using an encoder-decoder neural network [37]. However, this approach requires parallel training data for supervised learning in the training phase and the quality of the synthetic data is uncertain, which could impact the style transfer performance.

In terms of rhythm transfer, researchers focus on changing the duration of notes in a piece of music. Yang *et al.* introduced an autoencoder-based approach to disentangle pitch and rhythm features into a latent representation for transferring 2-bar music segments, which served as a foundation for their subsequent work on rhythm style transfer [38]. Building on this, they developed an explicitly-constrained conditional variational autoencoder (EC2-VAE) to disentangle the pitch and rhythm representation of 8-beat music clips based on the chords, followed by concatenating the rhythm features and pitch latent vector to reconstruct a new melody [39]. However, modifying the rhythm of a melody can have an impact on the pitch contour, which may lead to complex behaviors in the latent space that are difficult to interpret.

Chapter 3

The Beauty of Repetition: a Motif-level Algorithmic Composition Model

3.1 Background and Motivation

Repetition, which is the act of repeating or the occurrence of something being repeated, is a common aspect in daily life. It can be seen in cellular division, the rotation of the Earth, and activities such as swinging and composing poetry. It also affects people's daily routines through the cycles of the seasons and variations in tides, among other things.

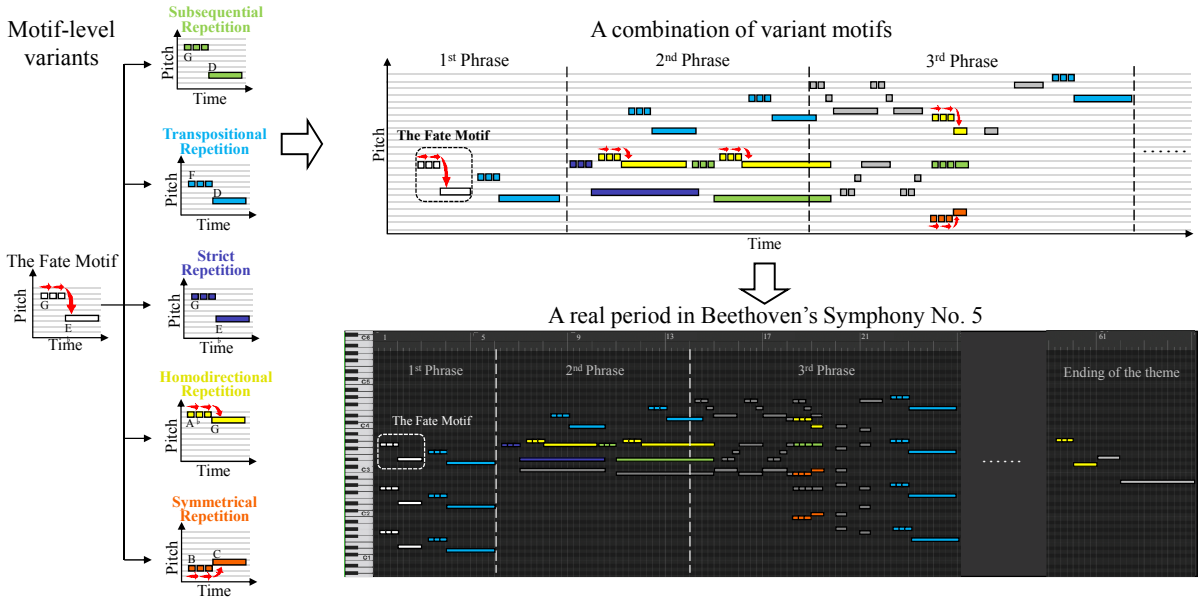


Figure 3.1: An example of Beethoven's Fifth Symphony.

In the realm of music, the art of repetition and its diverse forms conjure a mesmerizing musical panorama, delivering captivating auditory experiences. We initiate our journey into this elemental compositional technique with an examination of Beethoven's iconic Fifth Symphony (Fig. 3.1). The inaugural movement is characterized by the renowned "fate motif": a quartet of pronounced, compelling notes "G-G-G-Eflat", orchestrated in a short-short-short-long sequence, embodying the ominous echo of "fate knocking at the door". This motif then undergoes a transpositional repetition (TrR), descending a step, imbuing the auditory space with an eerie prelude of uncertainty and apprehension. Unexpectedly, Beethoven transitions into a comparatively serene melody in the following phrase, eschewing the preceding sinister tonality, yet without fully embracing an optimistic or jubilant character. The expansion of the theme is executed through the incorporation of two groups of three ascending fate motif repetitions, followed by a strict repetition (StR), dual homodirectional repetitions (HoR), a subsequential repetition (SuR), and a pair of TrRs, articulated in a more buoyant tonal quality. These succinct, imitative sequences weave seamlessly, propelled by a consistent rhythmic pattern, culminating in a harmonious, fluid melody. The third segment of the composition commences with a series of motifs, alternating in downward and upward progressions, reminiscent of a musical dialogue of call and response. This alternating pattern amplifies the anticipation, ascending towards the apex of the primary theme. A triumvirate of fate motif repetitions, encompassing a HoR, a SuR, and a symmetric repetition (SyR), converge in unison, peaking in a dramatic crescendo before transitioning into a sustained elevated note. The phrase concludes with another TrR, plunging the mood back into a restrained, somber atmosphere.

Repetition is widely utilized in music across different genres, cultures, and time periods to create an impactful effect. However, the use of repetition patterns in machine composition is still a relatively unexplored area [40]. William Carlos Williams stated that repeating the theme is a fundamental principle of music and that repetition increases the pace [40]. Steve Reich, a composer known for his repetition-focused music, showcased this concept in his work "The Desert Music" [41]. Repetition in music does not simply involve repeating identical elements, but rather involves creating new variations on echoed [42]. Thus, repetition patterns in music are not just a matter of musical composition, but also invite the listener to actively participate.

When it comes to the category of repetition in music, it is always accompanied with diversity and complication. First, different criteria can be used to classify repetition (e.g., pitch values, rhythm, and harmony) [43, 44]. Second, it can be divided by level—the low-level (note and motif) and high-level(phrase and period). To avoid overlap due to different criteria, we focus on repetition in terms of pitch value. In addition, as the most basic analyzable and meaningful element of a musical composition, motifs create dynamic coherence through repetition, variation, and combination at different scales [45]. In this chapter, we start from motif-level repetitions and try to generate multi-phrase music from motifs (Table 3.1).

The diversity and complication of repetition in music further results in two challenges. First, the scarcity of repetition-related data precludes investigation of repetition types. No public dataset is available to specify repetition types; it is accordingly difficult to further analyze music repetition. Second, it is still challenging to produce and combine explicit repetitions to make pleasant music. Currently, some work employs memory modules in deep learning models, such as Transformer and long short-term memory (LSTM), to generate music [23, 24, 30, 46–48]. Others use statistical and rule-based methods to construct long-term repetitive structures [7, 49–52]. However, these models cannot be controlled to generate explicit repetition types since they lack the domain knowledge of music repetition types. It is hard to understand and follow the structure and direction of the generated music. In other words, current techniques in machine composition cannot recognize and generate different repetitions, which further leads to the problem of random direction and structureless.

To overcome these challenges, we propose an innovative Model for Motif-to-Music Generation (MGM) that generates music through a three-stage process (motif, outline, music) while incorporating repetitive motifs into the music generation process (see Fig. 3.2 (a)). The motif stage represents a central motif, or a few key motifs, from the music. The outline stage serves as a composer's sketch, organizing thoughts and ideas when creating the music. The outline can be seen as a roadmap composed of a sequence of events, aiding in the construction of the overall musical structure. Each event consists of key motifs and their variations found within the corresponding musical phrase. The music stage ultimately represents a multi-phrase composition. To train MGM, we have developed a comprehensive system that includes two novel

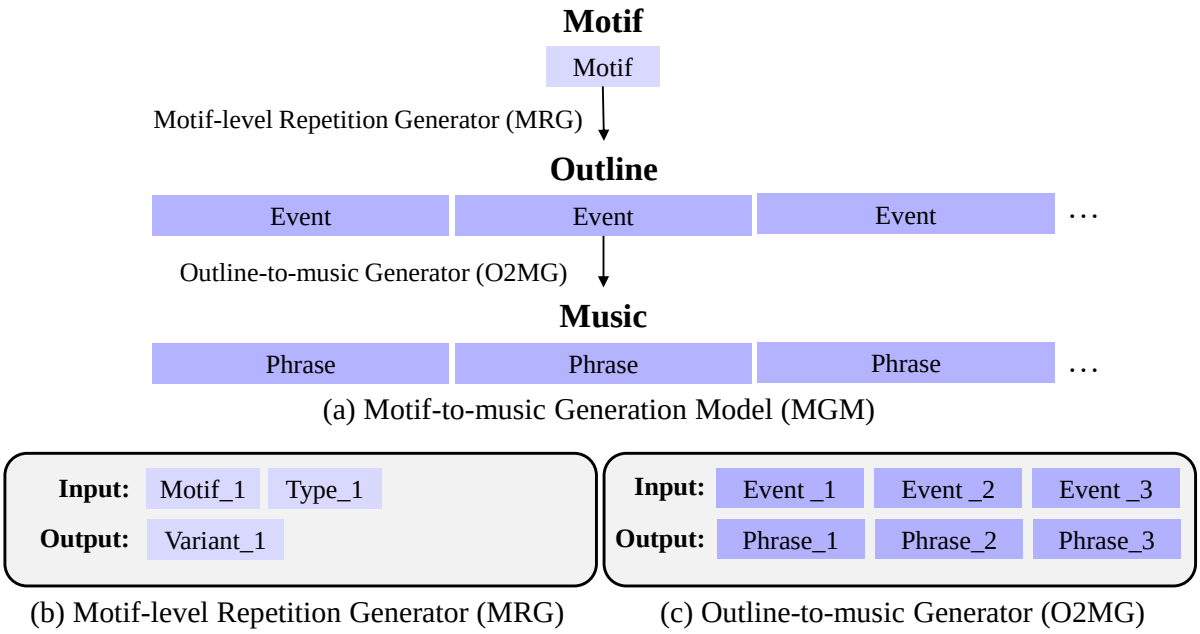


Figure 3.2: Description of Motif-to-music Generation Model (MGM). MGM has three stages: motif, outline, and music.

generators: the Motif-Level Repetition Generator (MRG) and the Outline-to-Music Generator (O2MG). Additionally, we have created a new dataset. The MRG enables our proposed MGM model to accurately generate motif-level repetitions by learning distinctive features of different repetitions. Given an original motif and a specific repetition type, the MRG generates a variation of the motif (Fig. 3.2 (b)). The O2MG empowers our model to create music in a structured manner by constructing an outline and understanding the relationships among motif variations. By utilizing a music outline comprising sequential events, with each event incorporating a set of repetitive motifs from the corresponding musical phrase, the O2MG generates a multi-phrase composition that aligns with and complements the outline (Fig. 3.2 (c)). Our newly developed dataset, the Music Repetition Dataset (MRD), is divided into two sections. The first section involves motifs paired with repetition labels, while the second section consists of outline-music sequences.

Table 3.1: Different repetition types based on pitch.

Level	Scale	No.	Repetition Name	Definition
Low-level	Note	1	Same note	Two notes are the same.
		2	Same pitch class	Two notes are in the same pitch class.
	Motif	3	Strict repetition ((StR)	Two motifs are the same.
		4	Transpositional repetition (TrR)	Two motifs have the same relationships between pitches but with a different tone.
		5	Subsequential repetition (SuR)	Two motifs are not identical or transpositional, but share a common subsequence.
		6	Homodirectional repetition (HoR)	Two motifs are not SuR, but have homodirectional repetition development[1].
		7	Symmetric repetition (SyR)	Two motifs are not SuR, but have similar symmetric repetition development.
High-level	Phrase	8	Similar melody	Two phrases have similar melody.
	Period	9	Structural repetition	Two periods have similar structure.

We summarize the contributions of this paper as follows:

1. We propose MGM, a 3-stage generative model of music (motif, outline, music). This model generates motif-level repetitions and incorporates repetitive motifs into the algorithmic composition process that imitates the composition process of a human composer.

2. We propose two novel generators, MRG and O2MG. MRG can generate motif-level repetition variants precisely based on designated repetition type. O2MG can achieve outline-to-music generation based on key motifs and the outline structure.

3. We present a new music repetition dataset named MRD. 584,329 samples are provided with the labels of motif-level repetitions for repetition variant generation. 3,545 outline-music sequences are provided for outline-to-music generation.

4. The learned composition skills on pop music dataset also demonstrate interesting results on "fate motif" from the classical music. Objective results and subjective evaluation on professional users show that the proposed model help to generate various of repetitions and improve the overall quality of the machine composed music obviously.

3.2 Related Work

[1]Development: the direction of the pitch sequence from one note to the next.

Unlike rule-based music generation models [50], Monte-Carlo tree search-based models [53], or CNN-based models [20, 54, 55], Transformer-based models [23, 56] have become popular as the backbone of generative models in music. The "self-attention" mechanism in Transformer-based models enables them to have a longer memory and learn the repetitive structure of music. Several Transformer-based models have been proposed for symbolic music generation [24, 26–31]. However, these models often lose a sense of direction and the generated music can be random and lack clear patterns [57, 58].

To address this issue, researchers have started to analyze the hierarchical structure of music [7–10]. For example, MusicFrameworks [48] uses a Transform and LSTM-based model to create a melody guided by a long-term repetitive structure. PopMNet [47] has a melody generation network and a structured generation network based on the structural representation and chord progression of input music. MELONS [44] uses a graph representation and a multi-step generation method with Transformer-based models for music structure generation and structure-conditional melody generation. Theme Transformer [32] generates themes by producing them multiple times in the output music to follow the thematic information. These models have been designed based on multiple levels of music structure. However, they have not studied music repetition and have not explored incorporating explicit repetition types into the algorithmic composition process. Therefore, they cannot control repetition and music trend, and generating repetitive patterns and rhythmically understandable music remains a challenge.

3.3 Proposed Method

Fig. 3.3 provides the general framework of the proposed work. Mathematically, in repetition variant generation, the goal is to generate a new motif $\hat{\mathbf{x}} \in \hat{\mathbf{X}}$ given an input motif after tokenization $\mathbf{x} \in \mathbf{X}$ and a repetition type $\mathbf{y} \in \mathbf{Y}$, where $\mathbf{X}, \hat{\mathbf{X}} \subseteq \mathbb{R}^{L \times K}$ is the input space and $\mathbf{Y} \subseteq \mathbb{R}^{M}$ is the label space. L is the length of the input, K is the number of music attributes such as note pitch, note duration, type and etc., and M is the number of repetition types.

In outline-to-music generation, the goal is to first construct an augmented outline-music sequence $\mathbf{x}_A \in \mathbf{X}_A$ for training, then generate a new piece of music $\mathbf{x}_{\text{new}} \in \mathbf{X}_{\text{new}}$ given the outline of the new music or without any condition, where $\mathbf{X}_A, \mathbf{X}_{\text{new}} \subseteq \mathbb{R}^{L_A \times K}$ and L_A is the

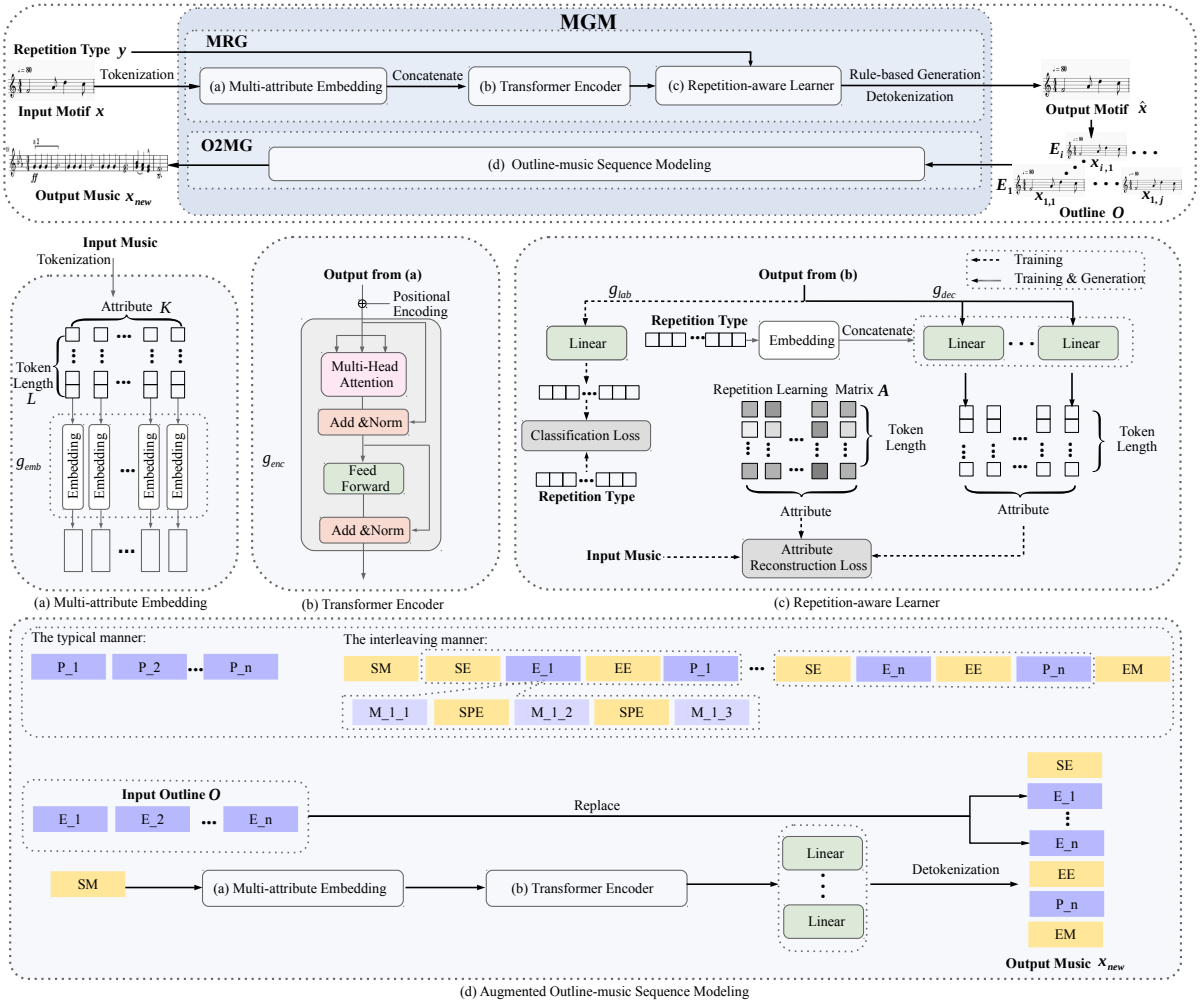


Figure 3.3: The general framework of Motif-to-music Generation Model (MGM), which contains Motif Repetition Generator (MRG) and Outline-to-Music Generator (O2MG). SM: <|startofmusic|>, SE: <|startofevent|>, E: an event, EE: <|endofevent|>, P: a phrase, M: a key motif, SPE: <|separationofevent|>, EM: <|endofmusic|>.

length of the augmented outline-music sequence.

3.3.1 Motif-Level Repetition Generator

3.3.1.1 Multi-Attribute Embedding

We incorporate a multi-attribute embedding module to embed various musical attributes, such as note duration and pitch, into separate dimensions. The embedding space is defined as $\mathbf{H} \subseteq \mathbb{R}^{L \times H_1}$, where H_1 represents the number of embedding dimensions. The multi-attribute-embedding function $g_{\text{emb}} : \mathbf{X} \rightarrow \mathbf{H}$ maps the musical motif $\mathbf{x}$ to the output. The result of the multi-attribute embedding, obtained through concatenation, is expressed as $[\text{Embed}_1(\mathbf{x}_{:,1}), \cdots, \text{Embed}_K(\mathbf{x}_{:,K})]$,

where $\text{Embed}_k(\cdot)$ is the embedding layer for the k-th attribute and $[\cdot,\cdot]$ denotes concatenation.

3.3.1.2 Transformer Encoder Architecture

We utilize the well-established Transformer encoder architecture [56], which has been frequently utilized in music generation [30, 44]. The architecture is illustrated in Fig. 3.3 (b) and follows the standard Transformer encoder structure [56]. The feature space, $\mathbf{F}$, is defined as $\mathbf{F} \subseteq \mathbb{R}^{L \times H_2}$, where H_2 represents the feature dimensions. An encoder/feature mapping, $g_{\text{enc}} : \mathbf{H} \to \mathbf{F}$, transfers the input sequence of tokens $g_{\text{emb}}(\mathbf{x}) + \mathbf{e}\text{pos}$, where $\mathbf{e}\text{pos} \in \mathbb{R}^{L \times H_1}$ is a positional embedding vector, to the encoded features. The output of the Transformer encoder, g_{enc}, serves as the aggregated representation for the entire input sequence and employs standard self-attention [56] and layer normalization [59] in the multi-head-attention module and following the multi-head-attention and linear projection modules.

3.3.1.3 Repetition-aware Learner

We introduce a new module called the repetition-aware learner to train the model and generate variations of motifs. This generator uses a combined training strategy of reconstruction and classification to produce different types of repetition.

Let $f_c : \mathbf{X} \to \mathbf{Y}$ represent the labeling process and $f_r : \mathbf{X} \to \hat{\mathbf{X}}$ represent the data reconstruction process. The decoder $g_{\text{dec}} : \mathbf{F} \to \hat{\mathbf{X}}$ transforms the encoded features into reconstructed data, and the feature-labeling function $g_{\text{lab}} : \mathbf{F} \to \mathbf{Y}$ maps the encoded features to labels. Both the labeling process f_c and the reconstruction process f_r can be decomposed as follows:

$$f_c(\mathbf{x}) = (g_{\text{lab}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}), \tag{3.1}$$

$$f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y}), \tag{3.2}$$

where $\circ$ is the composition of two functions. In Equation 3.1, the multi-attribute embedding module (Fig. 3.3 (a)) is represented by g_{emb}. The Transformer encoder (Fig. 3.3 (b)) is represented by g_{enc}. The label prediction module (Fig. 3.3 (c)) is a linear projection and is represented by g_{lab}. In Equation 3.2, g_{enc} is concatenated with the target repetition label $\mathbf{y}$ and fed into the

decoder g_{dec}, which is composed of K linear projection layers to predict the K attributes of the output music (Fig. 3.3 (c)).

The feature-labeling function g_{lab} predicts the repetition type $\mathbf{y}$ using the output of g_{enc}. The linear decoder g_{dec} uses the output of g_{enc} concatenated with the target repetition label to reconstruct the input music $\mathbf{x}$. To train the model, we first define a repetition learning matrix that allows us to directly control the training process based on the repetition type definition. Then, we propose a reconstruction and classification cooperation learning strategy.

Repetition learning matrix: We aim to capture the distinct features of different repetition types and motifs by considering multi-aspect attributes. To do this, we define a repetition learning matrix, $\mathbf{A}$, with dimensions $L \times K$, where L is the sequence length and K is the number of attributes considered.

The element $a_{l,k}$ of the repetition learning matrix $\mathbf{A}$ can be computed as:

$$a_{l,k} = \gamma \cdot (1 + \omega_{l,k}), \tag{3.3}$$

where γ is an attribute importance hyper-parameter that controls the importance of each attribute. $0 \leq \omega \leq 1$ is determined by the appearance frequency of each element in the music and can be calculated as

$$\omega_{l,k} = \frac{Counter(x_{l,k}, \mathbf{x}_{:,k})}{L}, \tag{3.4}$$

where $Counter(\cdot)$ is a function that counts the occurrence number of each element in each attribute.

The repetition-aware learning process is illustrated in Fig. 3.3 (c). This multi-aspect attribute learning takes into account two key considerations. Firstly, different attributes have different representations across different repetition types. For example, two StR motifs will have the same number of notes and pitch values, while two SuR motifs might have different note counts and different note positions, durations. Therefore, if k represents pitch, position, and duration in Equation 3.3, then γ will be assigned higher importance in SuR. Secondly, the representation of different motifs varies within the same repetition type. That is, the distribution of each attribute is distinct among motifs. For example, the pitches "C4-E4-D4-G4-F4" and "C4-E4-F4-C4-C4"

have different representations. If k represents pitch, then based on Equations 3.3 and 3.4, $a_{1,k}$, $a_{4,k}$, and $a_{5,k}$ will be larger for the latter case, as "C4" appears more frequently than other pitch values.

Learning algorithm: Consider a set of labeled input music samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where each $\mathbf{y}_i \in \{0,1\}^M$ is a one-hot vector with M classes and N is the number of music samples. Let $\theta_c = \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}$ and $\theta_r = \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}$ denote the parameters of the label learning pipeline f_c and the data reconstruction pipeline f_r. θ_{emb} and θ_{enc} are shared parameters for the embedding g_{emb} and feature mapping g_{enc}. Let ℓ_c and ℓ_r be the classification and reconstruction loss, respectively. Based on Equations 3.1 and 3.2, the empirical classification loss is first defined as follows.

$$\mathcal{L}_c(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}) := \sum_{i=1}^{N} \ell_c(f_c(\mathbf{x}_i; \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}), \mathbf{y}_i). \tag{3.5}$$

Typically, the empirical classification loss, ℓ_c, is expressed as a cross-entropy loss, which is the sum of the log of the predicted probability of the true class for each of the M classes, $\sum_{m=1}^{M} y_k log[f_c(\mathbf{x})]_m$ (recall that $f_c(\mathbf{x})$ is the softmax output). The attribute reconstruction loss is defined as follows:

$$\mathcal{L}_r(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}) := \sum_{i=1}^{N} \ell_r(f_r(\mathbf{x}_i, \mathbf{y}_i; \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}), \mathbf{x}_i). \tag{3.6}$$

ℓ_r is of the form squared loss combined with repetition learning matrix:

$$\ell_r = ||\mathbf{A} \otimes (\hat{\mathbf{x}} - f_r(\mathbf{x}, \mathbf{y}))||_2^2, \tag{3.7}$$

where $\otimes$ is the element-wise product of two matrices.

The goal of our proposed method is to minimize the total loss $\mathcal{L}$, which is expressed as follows:

$$\min_{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{lab}}} \lambda \mathcal{L}_c(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}) + (1-\lambda)\mathcal{L}_r(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}), \tag{3.8}$$

where $0 \leq \lambda \leq 1$ is a hyper-parameter controlling the trade-off between classification and reconstruction. The objective is a convex combination of supervised and unsupervised loss

functions.

The objective in Equation 3.8 can be achieved by minimizing $\mathcal{L}_c$ and $\mathcal{L}_r$. The process is stopped when the average total loss stabilizes.

Algorithm 1 MRG in the training & generation phase.

Training phase:
Input: Music data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$
Parameter: Learning rate α, trade-off parameter λ
Output: Optimal parameters $\theta^* = \{\theta^*_{\text{emb}}, \theta^*_{\text{enc}}, \theta^*_{\text{lab}}, \theta^*_{\text{dec}}\}$

Initialize parameters $\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}, \theta_{\text{dec}}$;
while not converge **do**
  for each batch of size n **do**
    Forward $f_c(\mathbf{x}) = (g_{\text{lab}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x})$;
    Forward $f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y})$;
    Calculate repetition learning matrix **A** based on Equations 3.3 and 3.4;
    Update $\theta : \theta \leftarrow \theta - \alpha \bigtriangledown_\theta \mathcal{L}^n(\theta)$;
  end for
end while

Generation phase:
Input: Music data **x** and designated repetition type **y**
Parameter: $\theta^*_{\text{emb}}, \theta^*_{\text{enc}}, \theta^*_{\text{dec}}$
Output: Output music $\hat{\mathbf{x}}$

if **y** is StR **then**
  if k is pitch **then**
    $\hat{x}_{l,k} = x_{l,k}$;
  end if
else if **y** is TrR **then**
  if k is pitch **then**
    $\hat{x}_{l,k} = x_{l,k} + t$;
  end if
else
  $\hat{\mathbf{x}} = f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y})$;
end if

3.3.1.4 Rule-based generation

Only the reconstruction pipeline of the repetition-aware learner is utilized to produce the output, which is expressed as $\hat{\mathbf{x}} = f_r(\mathbf{x}, \mathbf{y})$. The input music and the designated repetition type are represented by **x** and **y**, respectively. We also adjust the output according to the characteristics of different repetitions, as defined in music theory. For instance, in a StR (Strict Repetition), the

pitch is fixed to be the same as the input motif, so the pitch values in the output are set equal to those in the input. If pitch is represented by k, then $\hat{x}_{l,k} = x_{l,k}$. In a TrR (Transpositional Repetition), the pitch difference is kept constant, so the pitch values are adjusted according to a transposition value $t = \{\cdots, -2, -1, 1, 2, \cdots\}$ after the first pitch value is predicted. If pitch is represented by k, then $\hat{x}_{l,k} = x_{l,k} + t$. In SuR (Subsequential Repetition), HoR (homodirectional Repetition), and SyR (Symmetric Repetition), all music attributes are generated based on what the model has learned, rather than being fixed or adjusted. The entire MRG learning algorithm, including both the training phase and the generation phase, is outlined in Algorithm 1.

3.3.2 Outline-to-music Generator

3.3.2.1 Outline-music Sequence Representation

We consider key motifs and their variants as the outline of the music and use special type tokens as delimiters [60] to connect outline events with music phrases. Consequently, each outline-music leads to an augmented sequence for music modeling. We build an augmented outline-music sequence $\mathbf{x}_A$ as follows:

1. Key motif selection: For each phrase in the music that contains K motifs, we select k motifs in the phrase as key motifs, where $0 < k < K$. Key motifs should be repetitive motifs that follow the definition of Table 3.1.

2. Outline construction: The outline consists of several events. For each event in the outline, i.e., a set of key motifs from a music phrase, we build an event sequence by prepending a "<|startofevent|>" type token to initiate the event, inserting a "<|sepofevent|>" type token to separate key motifs, and appending a "<|endofevent|>" type token to stop the event.

3. Outline-music combination: For a list of event sequences in the outline and phrases in the music, we join them in an interleaving manner. The architecture and example of augmented outline-music sequence are shown in Fig. 3.3 (d). The typical music sequence only contains sequential note information in the form of phrases, while the augmented outline-music sequence contains key motifs in the event as the outline before each music phrase. In addition, the music will be initiated with a "<|startofmusic|>" type token at the

beginning of the first event sequence and will be end with a "<|endofmusic|>" type token at the end of the last phrase.

The augmented outline-music sequence has two advantages. First, it strengthens the sequential and paired correspondences between outline events and phrases; second, it improves flexibility and controllability of generation since the outline events of following phrases can be modified during generation.

3.3.2.2 Training and Generation

The core of this model is formulated as unsupervised distribution estimation of a sequence of tokens, which is a sequence modeling task (Fig. 3.3 (d)). The training and generation of O2MG is summarized in Algorithm 2. At the training phase, the model performs music token modeling on the augmented outline-music sequences. Given augmented outline-music sequences $\{\mathbf{x}_{A,i}\}_{i=1}^{N_A}$, where N_A is the number of augmented outline-music samples. Let ϕ denote the parameters of the generator f_{O2MG}, which includes a embedding, an encoder and a decoder module. Let ℓ_{O2MG} be the loss of outline-to-music generation. The objective in O2MG is to minimize the O2MG loss $\mathcal{L}_{\text{O2MG}}$ and is formulated as follows:

$$\min_{\phi} \sum_{i=1}^{N_A} \ell_{\text{O2MG}}(f_{\text{O2MG}}(\mathbf{x}_{A,i}; \phi), \mathbf{x}_{A,i}), \tag{3.9}$$

where ℓ_{O2MG} is typically the negative log likelihood. The proposed method reserves the generation capabilities of the pre-trained model and meanwhile grasps the conditional relationship from outline to music naturally.

The music is sequentially decoded following the same format of augmented outline-music sequences. We first define an outline of the target music $\mathbf{O} = \{E_1, E_2, \cdots, E_i\} = \{\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \cdots, \mathbf{x}_{i,j}\}$, where E_i is the i-th event and j is the number of key motifs in the i-th event. $\mathbf{x}_{i,j}$ can be generated from MRG or provided by the user. O2MG endows the ability to generate succeeding event sequences including the outline and the music since event sequences are inherently modeled as part of the music. Users have the flexibility to modify and extend those events before generating succeeding phrases. For example, users can let the model generate the outline and

the music if no outline is given. Users can also change a generated outline event sequence to another designated event sequence in O starting from "<|startofevent|>" to "<|endofevent|>". Then the model will generate music phrases based on the replaced outline event. One example will be demonstrated in experiments to show the level of human supervision.

3.4 Music Repetition Dataset

In order to carry out our task, we need two types of datasets: sets of motif-level repetitions with repetition-type labels and paired human-composed outline-music. However, such datasets are not readily available as most music datasets only contain complete music pieces without annotating their hierarchical structure. To begin with, we chose two commonly used pop piano datasets, the Pop Piano Dataset (PPD) from [30] and Pop909 from [1]. This is because the structure of pop music is relatively simple for analysis and these datasets include all the repetition types listed in Table 3.1. PPD and Pop909 consist of 1,748 and 909 pieces of pop piano music obtained from the internet. In PPD, each song has been converted into a symbolic sequence through transcription, synchronization, quantization, and analysis, and all note information is included in a single track (such as the melody and accompaniment). In Pop909, each arrangement is stored in three separate MIDI tracks: MELODY, BRIDGE, and PIANO (accompaniment). Following [32], we omitted the BRIDGE track and trained the model on MELODY and PIANO separately to learn the melody and accompaniment, as the quality of the original tracks does not significantly decrease without the BRIDGE track and its purpose falls between melody and accompaniment.

3.4.1 Motif-level Repetition Extraction

We divide the musical piece into repeating patterns and assess the pitch of these patterns. The reason for focusing on pitch and development is that they are frequently used in music composition and are easily recognizable by listeners [61]. If two patterns meet the criteria outlined in Table 3.1, then they are considered to be repetitions, otherwise, they are not.

[3]Note that a key motif is usually a sequence of notes rather than a single note, so the actual replacement process has more steps.

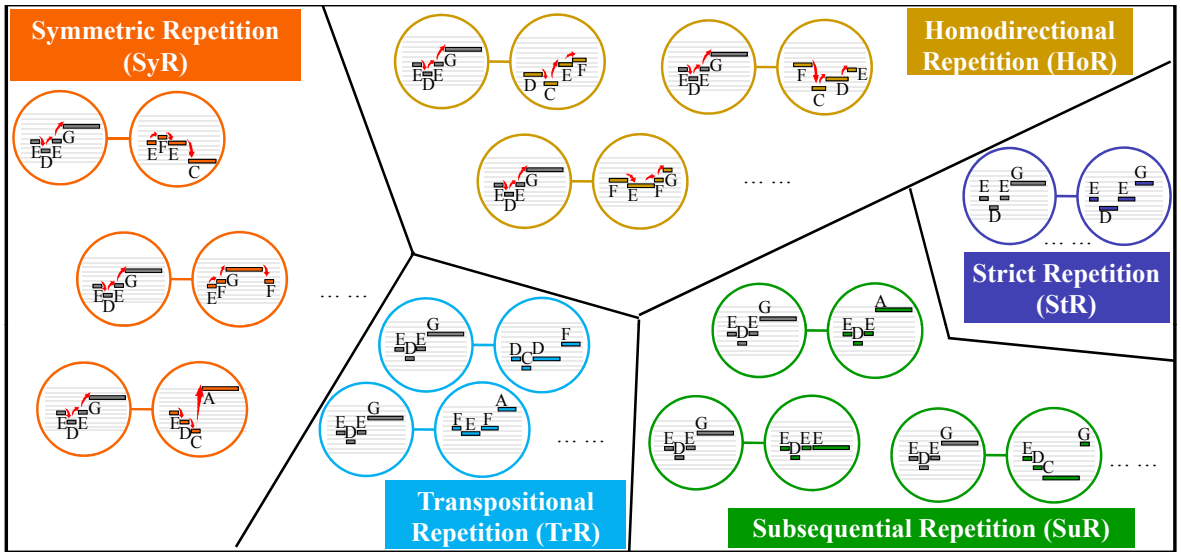


Figure 3.4: Five different types of repetition in Music Repetition Datase (MRD).

Table 3.2: Motif-level repetition dataset statistics.

Dataset	StR	TrR	SuR	HoR	SyR
Train	21807 (3.88%)	27852 (4.95%)	73774 (13.11%)	152060 (27.02%)	287070 (51.03%)
Test	1395 (6.41%)	1635 (7.51%)	3331 (15.31%)	5038 (23.15%)	10367 (47.63%)

In SyR, two motifs that display vertical, horizontal, or rotational symmetry create a symmetrical repetition. In SuR, HoR, and SyR, the similarity is set at 75 %. Motifs that fulfill both HoR and SyR (e.g., "C5-D5-E5" and "C5-E5-G5") are excluded as this could result in overlap between repetition types and is not commonly found in the dataset (less than 1 %). However, this type of repetition represents a fascinating subject for further investigation. Fig. 3.4 showcases examples of the five repetition types in MRD. These repetition types are exclusive based on their definition and each motif is assigned to one specific repetition type. StR has the smallest size since its definition is the strictest, requiring all note pitches to be identical. On the other hand, SyR has the largest size as it considers the progression of the music and encompasses three symmetrical scenarios. The longest motif sequence is used as the default length to preserve information from the motif. Out of the total motifs, 50 songs were randomly chosen for testing, while the remaining motifs were used to train MRG. Table 3.2 presents the basic statistics of the repetition types.

3.4.2 Outline Extraction

In the absence of outlines annotated by humans, we've devised a novel methodology to distill outlines from a musical composition. Mirroring the motif-level repetition extraction approach, the initial step involves the identification of all motif-level repetitions contained within a musical piece. We establish the extent of a phrase K to comprise 8 motifs. Given that repetitive motifs are interspersed throughout various segments of the music, certain phrases may encapsulate a multitude of repetitions, while others host fewer. We institute a range for the count of repetitive motifs within a single phrase k, setting the bounds at a minimum of 1 and a maximum of 4. This ensures the extraction of at least a single motif, even in scenarios where a phrase is devoid of repetitive elements, and caps the extraction at four motifs when repetitions are abundant.

In essence, the auto-extraction process is engineered to pinpoint pivotal motifs and delineate the progression and evolution of music within a composition encompassing multiple phrases. Table 3.3 offers an in-depth examination of the dataset's statistical attributes.

Table 3.3: Statistics of outline-to-music in two dataset.

Dataset	Music Number	Music Average Length	Music Average Phrases	Event Average Motifs	Motif Average Length
Melody_train (Pop909)	849	1247.88	10.88	3.26	12.08
Melody_test (Pop909)	50	1225.48	10.68	3.34	12.13
Accom_train (Pop909)	849	2796.60	10.88	3.79	22.27
Accom_train (Pop909)	50	2830.44	10.68	3.84	22.87
Combine_train (PPD)	1697	3238.99	13.44	3.74	20.51
Combine_test (PPD)	50	3053.60	13.26	3.78	19.51

3.5 Experiments

To validate the effectiveness of MGM, we conduct experiments to answer the following questions:

Q1: How to evaluate the quality of the generated repetition in music pieces, especially from motif-level and song-level?

Q2: Can MGM generate motif-level repetitions precisely given a motif and a designated repetition type?

Q3: Can MGM improve the overall quality of generated music, especially on the structure? How effective are the outline-to-music mechanism?

3.5.1 Experimental Setup

3.5.1.1 Implementation Details

According to [30], the Transformer encoder module includes 6 self-attention layers with 4 heads and 256 hidden states, and the inner layer size of the feed-forward part is 2048. The symbolic music is transformed into a sequence of words, known as compound words, using a pre-defined music attribute vocabulary [30]. The music is described using six attributes: tempo, position, type, pitch, duration, and velocity. The sizes of the embeddings for these seven attributes and the label are 128, 32, 512, 128, 128, 32, respectively. The embedding sizes are selected based on the vocabulary size of each token type, and the tokens that describe the same object are concatenated and linearly projected to the same size as the corresponding module's hidden state. The Adam optimizer [62] is used with a learning rate of 10^{-4} for both the classification and reconstruction models. During training, dropout with a rate of 0.1 is also applied. In the motif-level repetition generator, the repetition importance factors are determined as follows: the pitch attribute has a γ value of 4 for all repetition types since it plays a more significant role in repetition. For SuR, HoR, and SyR, the γ values of position, duration, and velocity are 2, since they also contribute to the representation of the notes. For the remaining attributes, γ is set to 1. The value of λ in Equation 3.8 is 0.5.

3.5.1.2 Model Comparison

For comparison purposes, five comparative models are utilized: two versions, CP-C and CP-NC, from the state-of-the-art music generation model [30]. CP-C generates music by utilizing a single motif as a conditioning factor while CP-NC generates music without any conditions. PopMNet [47], Theme [32], and MELONS [44] generate music by analyzing the musical structure. PopMNet uses a combination of CNN and RNN models, Theme uses Transformer-based models, and MELONS combines Transformer- and graph-based models. Online music samples from these models are used in objective and subjective experiments, as the source code for

PopMNet and MELONS is not accessible online.

To assess the effectiveness of MGM's components (MRG and O2MG), two variants of MGM-MRG, MGM-MRG-V and MGM-MRG-R, are designed. MGM-MRG-V is a basic MRG that doesn't incorporate the repetition learning matrix or rule-based generation. MGM-MRG-R is an MRG that doesn't use rule-based generation, but employs the repetition learning matrix. MGM-MRG-RR, on the other hand, utilizes both the repetition learning matrix and rule-based generation. Additionally, one variant of MGM-O2MG, MGM-O2MG-V, is created. MGM-O2MG-V is a straightforward O2MG that generates both the outline and complete music, relying solely on its own generated event sequence once the first key motif is given.

3.5.2 Evaluation Metrics of MGM

We explore to develop one objective metric to evaluate the motif-level repetition accuracy and conduct one subjective experiment to evaluate the music structure and quality.

The objective evaluation involves the generation of motif-level results from each model, using motif conditions drawn from 100 test samples. In this evaluation, 1-bar polyphonic piano music is generated based on the original motif. The performance of each model is measured using the matching rate, $\mathcal{M}$, which is calculated as the ratio of the number of matched motifs to the total number of motifs. A match is determined when two motifs meet the criteria outlined in Table 3.1.

In the subjective evaluation, the subjects need to first answer two questions that relate to the overall performance of the music:

• Overall quality (Q): How is the overall quality of the generated music compared to human compositions?

• Groove (G): Does the music sound fluent and pause suitably? How is the groove of the music?

Then, the subjects are asked to evaluate the quality of music structure:

• Coherence (C): How natural and coherent are the transitions and the developments between the contiguous phrases?

• Repetition (R): How strong is the motif repetition and its variants influence the music?

• Integrity (I): Does the music own the complete section structure, such as intro, verse, chorus, bridge, and outro? How clear are the boundaries between the sections?

3.5.3 Model Accuracy

3.5.3.1 Motif-level Repetition Generator (MRG) Accuracy

Table 3.4: Objective results: matching rate.

Models	StR ↑	TrR ↑	SuR ↑	HoR ↑	SyR ↑
CP-C [30]	0.00±0.00	0.04±0.08	0.21±0.41	0.68±0.47	0.71±0.46
CP-NC [30]	0.01±0.07	0.11±0.31	0.36±0.48	0.62±0.49	0.64±0.48
PopMNet [47]	0.00±0.00	0.00±0.00	0.10±0.30	0.60±0.49	0.70±0.45
Theme [32]	0.00±0.00	0.00±0.00	0.48±0.18	0.47±0.25	0.47±0.25
MELONS [44]	0.20±0.40	0.20±0.40	0.40±0.49	0.70±0.46	0.70±0.46
MGM-MRG-V	0.54±0.49	0.79±0.41	0.74±0.44	0.85±0.36	0.74±0.42
MGM-MRG-R	0.84±0.36	0.96±0.18	**0.75±0.43**	0.95±0.17	0.75±0.44
MGM-MRG-RR	**1.00±0.00**	**1.00±0.00**	**0.75±0.43**	**0.97±0.17**	**0.75±0.43**

Table 3.4 shows the performance of the three variants of the proposed model in terms of matching rate for different repetition types. The results indicate that the proposed model performs well in generating various repetition types.

Compared to other music generation models, the proposed model demonstrates superior performance in generating repetitions. This is due to the model being specifically designed for different repetition generation. Furthermore, HoR and SyR have higher matching rates compared to StR, TrR, and SuR in other models, which is attributed to the fact that StR, TrR, and SuR have more strict definitions and tend to learn what is easily learned or appears the most, as there is no supervisory signal for repetitions. This highlights the limitations of current music generation models.

The results show that MGM-MRG-R performs more robustly than MGM-MRG-V, as the repetition learning matrix informs the model which attributes are more important for a specific

repetition type during training. A comparison of MGM-MRG-RR and MGM-MRG-R demonstrates that StR and TrR have a matching rate of 1. This is expected, as these repetitions follow explicit rules based on their definitions, and therefore, implementing rules to generate pitch would lead to a perfect match.

3.5.3.2 Outline-to-music Generator (02MG) Accuracy

We evaluate the reconstruction quality of O2MG on different datasets and report the average value across all test samples in Table 3.5. We measure the difference between reconstruction and ground-truth in the sequence for our O2MG and compared it with the original CP-NC [30]. As shown in Table 3.5, with an outline-music structure, O2MG can reconstruct music sequence more accurately than the original CP-NC.

Table 3.5: O2MG reconstruction accuracy.

Model	RMSE on different datasets ↓		
	PPD	Pop909_melody	Pop909_accom
CP-NC [30]	3.06	5.34	5.27
MGM-O2MG	**2.65**	**3.93**	**2.96**

3.5.4 Music Quality by Human Listeners

Table 3.6: Subjective results of models.

Model	Quality		Groove		Coherence		Repetition		Integrity	
	score ↑	P-value ↑	score ↑	P-value ↑	score ↑	P-value ↑	score ↑	P-value ↑	score ↑	P-value ↑
CP-C [30]	2.90±1.00	<0.001	3.18±0.84	<0.001	2.90±0.80	<0.001	2.33±0.78	<0.001	2.65±1.00	<0.001
CP-NC [30]	3.35±0.77	<0.001	2.82±0.63	<0.001	3.14±0.82	<0.001	2.47±1.03	<0.001	3.04±0.89	<0.001
PopMNet [47]	3.11±0.79	<0.001	3.01±0.87	<0.001	3.08±0.85	<0.001	2.68±1.14	<0.001	2.54±0.81	<0.001
Theme [32]	3.47±0.74	<0.001	3.68±0.70	<0.001	3.26±0.97	<0.001	3.10±0.80	<0.001	3.10±0.80	<0.001
MELONS [44]	3.54±0.71	<0.001	3.40±0.66	<0.001	3.35±1.07	<0.001	3.01±0.94	<0.001	2.97±0.80	<0.001
MGM-O2MG-V	3.72±0.80	<0.001	3.68 ±0.83	<0.001	3.35±1.07	<0.001	**4.01±0.89**	**0.422**	3.11±0.83	<0.001
MGM-O2MG	**4.04±0.92**	**0.187**	**3.74 ±0.83**	**<0.001**	**3.54±0.83**	**0.011**	3.85±0.73	0.139	**3.53±0.73**	**<0.001**

To validate that MGM can generate pleasant music with a few repetitive patterns and their variants, we further design a subjective experiment on social media. A total of 72 subjects who had

musical training and were familiar with music theory were recruited.

In the study, participants listened to eight sets of musical pieces, with seven sets being generated by models and one set being composed by human musicians. The subjects randomly chose the pieces to listen to, and none of them reported having heard the original motif prior to the experiment. After listening to each piece of music, each participant was asked to rate the musical piece on a 5-point scale (ranging from 1 to 5), with a higher score indicating a better piece of music.

Table 3.6 presents statistical data on opinion scores and a paired t-test comparing machine-composed and human-composed music. The evaluation criteria include overall quality, groove, coherence, repetition, and integrity. When comparing the proposed model, MGM-O2MG-V, to other models such as MGM-O2MG, it is observed that the proposed model outperforms in terms of music quality and structure. This finding confirms the significance of utilizing repetitions in creating high-quality music and establishing a clear direction. Additionally, when comparing MGM-O2MG-V to CP-C and CP-NC, the proposed model shows a superior performance across all aspects, suggesting that the new generation mechanism, outline-to-music generation, improves the overall music generation process. Moreover, in a comparison between MGM-O2MG and MGM-O2MG-V, it is evident that the user-provided outline influences the structure of the generated music. This is supported by O2MG achieving a higher integrity score. On the other hand, MGM-O2MG-V tends to generate more repetitive patterns similar to the input motif, leading to a higher repetition score.

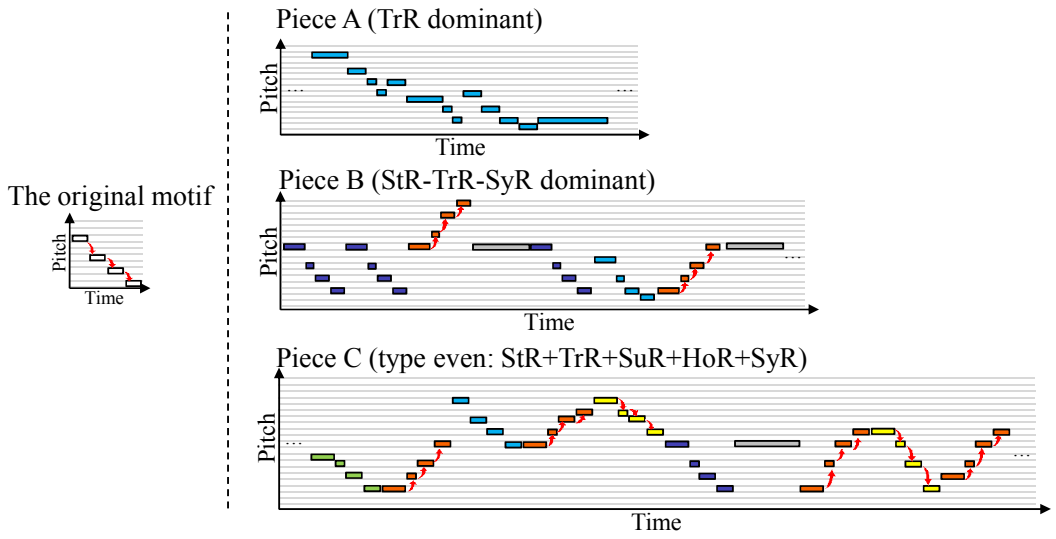
T-test is used to evaluate whether differences exist between two variables for the same subject. If $p<0.001$, then there is a statistically significant difference between two variables. MGM-O2MG performs similarly compared to human-composed music in terms of quality and repetition since there is no statistically significant difference ($p=0.187$ and $p=0.139$), verifying our model's authenticity. These strengths likely emerged for two reasons. First, by combining different repetition types, the results of the proposed model sound vivid and structural because the entire music piece is developed from a few motifs. Second, a well designed repetition is confirmed to be an important factor that makes the music pleasant. In addition, there is a significant difference of all aspects when comparing results generated by other models and orig-

inal human-composed music ($p<0.001$), indicating a better music quality and structure of the proposed model. However, there is still a gap between machine-composed music and human-composed music in terms of groove, coherence, and integrity. We will exert more effort on these directions in the future.

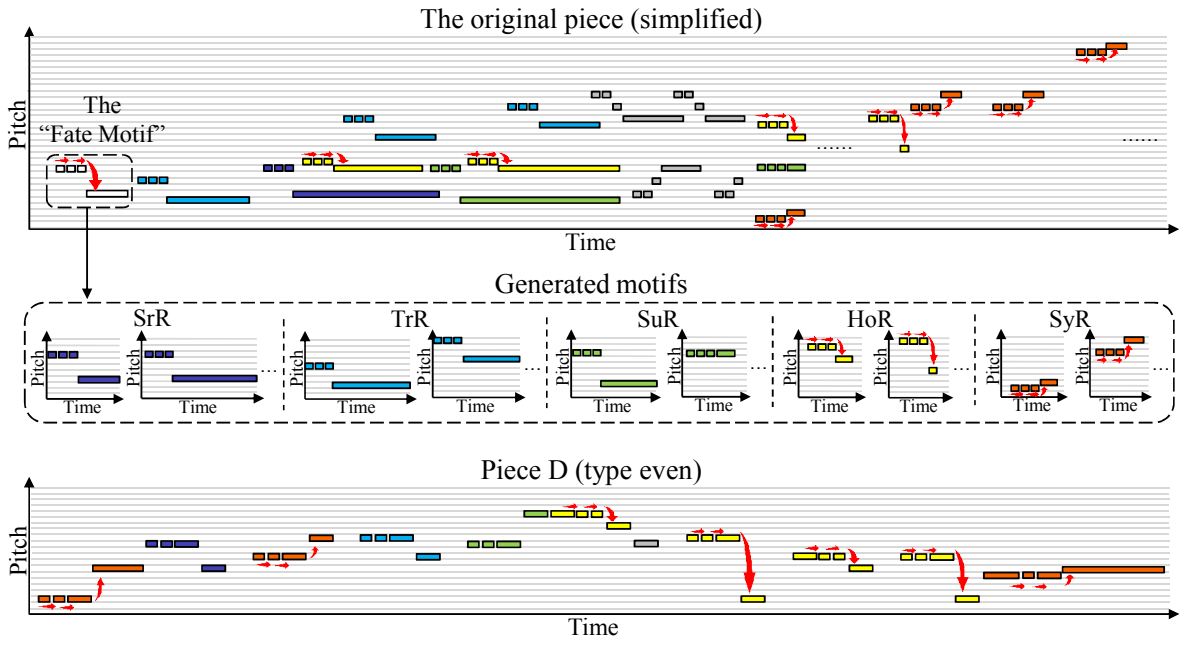
3.5.5 Analysis of Motif-level Examples

Two examples are presented in Fig. 3.5. The first example showcases a four-note motif descending in pitch (G#-F-D#-C#) in Piece A. This piece is characterized by TrR, which provides a sense of "pacing" or "jumping". Although the motifs have fixed intervals, TrR expands the pitch range and introduces an oscillation at different pitch levels. Piece B, on the other hand, incorporates StR-TrR-SyR, creating melodic and emotional waves. While StR reinforces the original pattern, SyR adds variation in melodic direction and richer intervals. Piece C is a more intricate and coherent version that incorporates all five repetition types. SuR slightly alters the original motif, adding both familiarity and novelty, while HoR introduces flexible intervals with the same direction, resulting in unrestrained and diverse music. This exquisite combination of repetitions effectively conveys complex emotions.

The second example focuses on the "fate motif" from Beethoven's Fifth Symphony. The model successfully generates various variants of this motif. The middle row of Fig. 3.5 (b) displays some of these variants, which appear in the first movement and set the tone for the symphony. By selecting and combining these generated variant motifs, it becomes possible to produce a piece similar to the original work, evoking similar emotions. Additionally, these variants can be used to create new, unique compositions. Piece D, in particular, is a type-even version based on the "fate motif". It differs from Beethoven's work and exudes a lighthearted and relaxed feel. The adjacent motifs with opposite movements create a sense of "question and answer", and the music flows smoothly without abrupt changes in direction. This further highlights the richness of repetitions.



(a)

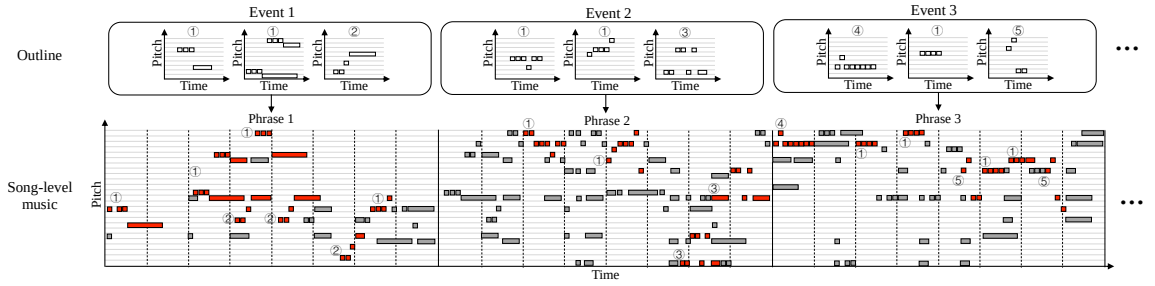


(b)

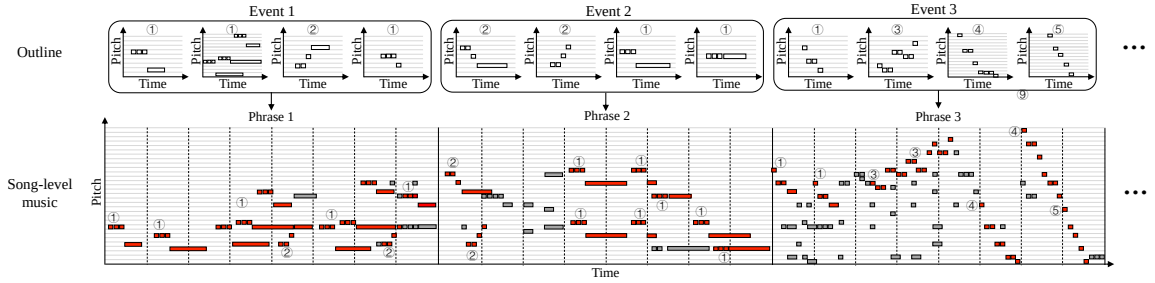
Figure 3.5: Motif-level generation results. Different colors represent different repetition types. Purple is strict repetition (StR). Blue is transpositional repetition (TrR). Yellow is subsequential repetition (SuR). Green is homodirectional repetition (HoR). Orange is symmetric repetition (SyR). Gray is ornamentation note.

3.5.6 End-to-end Generation and Fine-grained Control

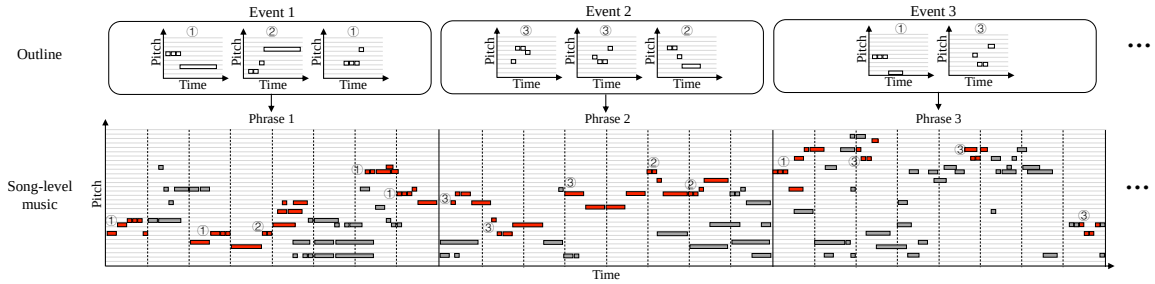
The 3-stage modeling approach used in MGM allows users to customize the output to their desired level of control. For instance, users can choose to let the model guide the generation process by providing a single motif, or they can make modifications at specific stages of the



(a) End-to-end generation: O2MG generates the outline and music based on the "fate motif".



(b) Fine-grained control: MRG first generates variants of five motifs (① - ⑤), then O2MG generates music that is similar to the Beethoven's Fifth Symphony based on the outline.



(c) Fine-grained control: MRG first generates variants of three motifs (① -③), then O2MG generates music that is very different from the Beethoven's Fifth Symphony based on the outline.

Figure 3.6: Song-level generation results. Numbers mean repetitive variants are generated from the same motif. The red color represents patterns that are similar to the outline.

hierarchy.

In Fig. 3.6 (a), the results of a fully generated end-to-end generation are presented. By inputting the "fate motif", the model generates subsequent key motifs and automatically constructs an outline for song-level music generation. The generated piece showcases simple variations of the "fate motif" in all three phrases, all of which are solely generated by the model. Despite the varied melodic development, the piece lacks a prominent motif, creating a mood of wavering, hesitation, and instability.

Fig. 3.6 (b) and (c) exhibit examples of fine-grained control, wherein an end user can manipulate the music's outline. By providing an informative and well-designed outline, the model can generate a new piece that is similar to or distinct from the original music. In Fig. 3.6 (b), the user incorporates an explicit motif outline inspired by Beethoven's Fifth Symphony. Many

of the key features found in Beethoven's Fifth Symphony can be identified, although the local articulation and melodic development may not be entirely consistent. This results in a streamlined and exquisite adaptation of the original song. In Fig. 3.6 (c), the user designs a new outline based on three motifs from Beethoven's Fifth Symphony. The MRG first generates repetition variants of these three motifs, which are then combined to form an outline for input into O2MG. In this example, each phrase showcases a unique and delicate melody, despite the presence of motifs similar to the original work. The three phrases progressively develop, with a strong and powerful first phrase, a slow and heavy second phrase, and an upbeat and lively third phrase.

3.6 Summary

To the best of our knowledge, this is the first thorough examination of repetition in machine composition. We introduce a new MGM system capable of generating structured music using a few motifs and their repetitive variations. Furthermore, we present two generators, MRG and O2MG, that are based on the transformer encoder and a repetition-aware learner and outline-music generation structure respectively. MRG can generate a large number of motif-level repetitions of various types, while O2MG is able to create a piece of structured music based on an outline. We also construct a dataset known as MRD, which contains 562,563 training data and 21,766 test data with labels for motif-level repetitions, as well as 3,545 outline-music sequences for outline-to-music generation.

Algorithm 2 O2MG in the training & generation phase.

Training phase:
Input: Augmented outline-music sequence data $\{\mathbf{x}_{A,i}\}_{i=1}^{N_A}$
Output: Optimal parameters ϕ^*

Initialize parameter ϕ;
while not converge **do**
  for each batch of size n **do**
    Do a forward pass $f_{\text{O2MG}}(\mathbf{x}_A)$;
    Update $\phi : \phi \leftarrow \phi - \phi \bigtriangledown_\phi \mathcal{L}^n_{\text{O2MG}}(\phi)$;
  end for
end while

Generation phase:
Input: an outline $\mathbf{O} = \{\mathbf{x}_{1,1}, \mathbf{x}_{1,2}, \cdots, \mathbf{x}_{i,j}\}$
Parameter: ϕ^*
Output: Output music $\mathbf{x}_{\text{new}}$

$t = 0, i = 1, j = 1$;
$Type(x_{\text{new},t}) =$ <|startofmusic|>;
while $Type(x_{\text{new},t}) \neq$ <|endofmusic|> **do**
  $t = t + 1$;
  $x_{\text{new},t} = f_{\text{O2MG}}(x_{\text{new},<t})$;
  if $O \neq \emptyset$ **then**
    if $Type(x_{\text{new},t-1}) \in \{$<|startofevent|>, <|sepofevent|>$\}$ **then**
      $x_{\text{new},t}$ will be replaced by $\mathbf{x}_{i,j}$ from O; [3]
      if $Type(x_{\text{new},t-1}) =$ <|sepofevent|> **then**
        $j = j + 1$;
      end if
    end if
    if $Type(x_{\text{new},t-1}) =$ <|endofevent|> **then**
      $i = i + 1, j = 1$;
    end if
  end if
end while

Chapter 4

Responding to the Call: Exploring Phrase-level Algorithmic Composition Using a Knowledge-Enhanced Model

4.1 Background and Motivation

Call-and-response, also known as antecedent and consequent phrases, is a prevalent technique in musicology, where one musical phrase serves as the "call" and is "answered" by a related yet distinct musical phrase [63, 64]. This technique is often employed in improvisation and choral settings, where singers exchange musical phrases to emphasize real-time interaction, spontaneity, or traditional call-and-response dynamics [65, 66]. With its broad applications in various composition works, our goal is to explore a more general composition scenario with the aim of enhancing the artistic quality of the composed pieces.

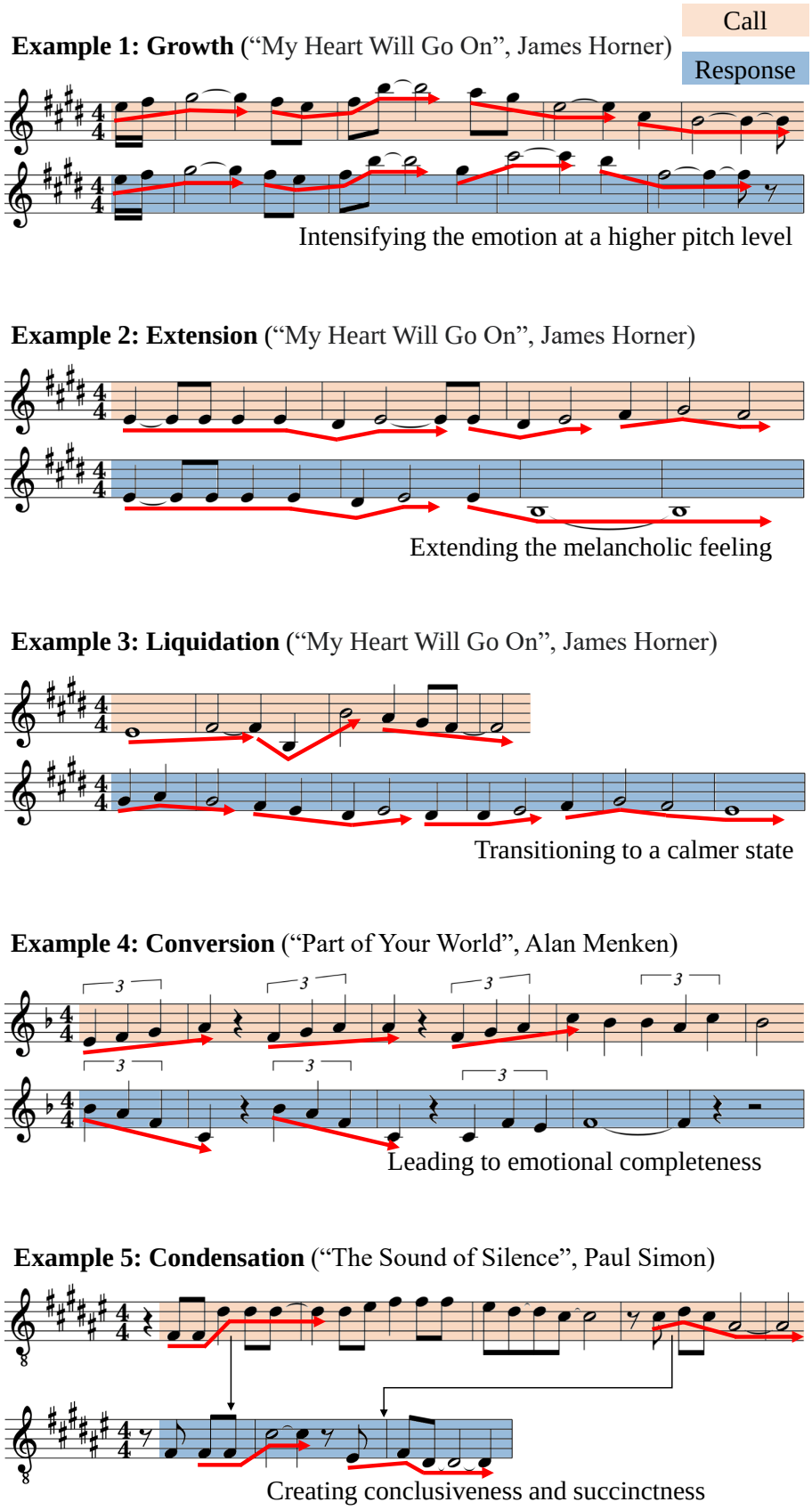


Figure 4.1: Examples of call-and-response in human-composed music, highlighting its role in enhancing artistic quality. Audio files are available in supplementary materials.

In composition, the call-and-response technique can introduce a variety of effects to music, collectively enhancing its artistic quality [67]. To immerse ourselves in the artistic depth that call-and-response infuses, let's revel in the timeless beauty of "My Heart Will Go On", composed by James Horner (Figure 4.1). The song unfolds with a growth call-and-response that nurtures the emerging emotion, offering the listener a resonant and evolving response. This response follows the call's melodic contour and rhythm, intensifying the emotion through a more expressive melody featuring a higher pitch and ending point (Example 1). Next, the composer artfully employs a call-and-response with an extension effect, captivating listeners with a dynamic experience that unveils a melancholic and yearning feeling, all the while preserving coherence and connection to the original call, crafting a spellbinding narrative (Example 2). In the wake of this, a call-and-response with a liquidation effect emerges, where the response gently reduces or simplifies the call's musical idea, delicately and steadily diluting and streamlining the concept as it transitions from the climax of emotion to a calmer state (Example 3). The composer masterfully employs call-and-response techniques to convey the song's central theme, rendering it unforgettable and deeply moving.

Indeed, call-and-response can be employed to express various effects in music. For instance, the response can be a conversion of the call, deftly transforming it from featuring three similar upward-moving motifs and a higher pitch level, invoking aspiration, to a response with two downward motifs that lower the pitches and lead to emotional completeness (Example 4). Finally, condensation distills a musical idea, yielding a response that is shorter or more concise than the call, effectively evoking a sense of conclusiveness and succinctness (Example 5).

This technique facilitates a reciprocal exchange within the music, as the "call" introduces a musical idea and the "response" follows up with a connected concept, echoing the back-and-forth nature of human dialogue. As a fundamental compositional technique to enhance music's artistic quality, call-and-response has been widely employed in human-composed music but remains underrepresented in machine-composed music, even when generated by the most advanced music generation products [2–4], as shown in Figure 4.2.[1] Music as an art form is appreciated not only for its correctness but also for its beauty and widely accepted artistic quality.

[1]Products such as MusicLM are prompt-based audio generation products; however, we include them in our comparison to highlight the gap between human-composed and machine-composed music.

Therefore, it is crucial to elevate machine composition beyond simply adhering to music theory and to achieve a greater level of artistry.

To further improve the artistry of machine-composed music, we develop a new approach through the investigation of the call-and-response technique, providing a new perspective and striving to transition machine-composed music from "correct" to "art". We introduce the Call-Response Dataset (CRD), consisting of 19,155 annotated call-response pairs in music, and simultaneously develop comprehensive objective evaluation metrics encompassing rhythm, melody, and harmony, which facilitate research and analysis of this musical technique. Furthermore, we propose a novel Call-Response Generator (CRG) featuring a knowledge-enhanced mechanism that generates appropriate responses for a given call. This model combines a music knowledge-enhanced module with an encoder-decoder architecture, where the former ensures compliance with music theory, and the latter supports learning the relationship between music notes. Finally, we implement a knowledge incorporation mechanism within the model to select and incorporate information from various sources, enhancing the model's capability to produce musically rich responses. Experimental results indicate that our proposed model effectively generates appealing and musically coherent responses to different musical calls, enriching and enhancing the musical experience provided by machine-generated compositions.

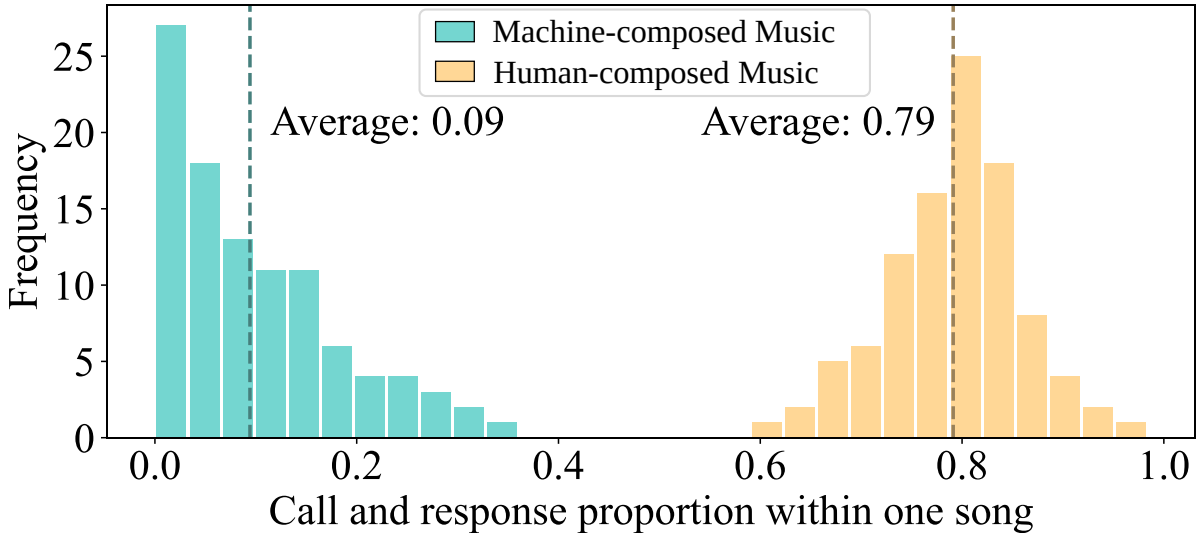


Figure 4.2: Comparative Analysis of Call-and-Response Proportions in Human-Composed and Machine-Composed Music: A Study of 100 Pop909 Songs [1] and 100 Machine-Composed Songs from [2–4].

4.2 Related Work

In pursuit of enabling machines to compose high-quality music, researchers in the machine composition domain have adopted two primary methods. First, learning-based methods have been developed, relying on large data samples to generate music [6, 68]. Prominent examples include MidiNet [54], MuseGAN [20], Music Transformer [23], and Pop Music Transformer [24], which employ various models such as CNN, GAN, and Transformer to learn musical features. Although these methods can generate creative and minute-long music based on patterns from training data, they often result in music that lacks central musical ideas, particularly when capturing essential compositional techniques [69]. One solution could be to further increase the size of the music data, but this would require more well-structured data and a significant investment in time and resources for collection and pre-processing. Even with a larger data sample, learning-based methods do not have specific mechanisms to guarantee the capture of desired compositional techniques, leading to unpredictable costs [58].

Recognizing that music is an art requiring both rule-based and learning-based approaches, researchers have designed a combination of learning-based and rule-based methods by incorporating music theory [8, 25, 70, 71]. This approach reduces the reliance on vast data samples and closely emulates the human process of studying composition, which involves learning music theory and listening to numerous music pieces. Many studies have demonstrated that combining learning-based and rule-based methods can enhance the quality of machine composition [48, 72, 73]. Some typical knowledge-enhanced machine composition methods involve incorporating knowledge such as genres [74], music themes [32], melody structures [47], harmonic features [31], and internal graph structure [44] into machine learning models for composition. Recent studies have explored attention mechanisms for improved structure and coherence [28, 30, 75], as well as reinforcement learning algorithms for generating music with specific characteristics like motif and harmony [76].

While these innovative approaches have advanced machine composition by integrating various aspects of music knowledge, they fall short in addressing call-and-response generation. Call-and-response embodies a conversation-like form, with the response immediately following the call and complementing it in rhythm, melody, or harmony, enhancing musical coherence

and unity between phrases [77]. The technique's uniqueness necessitates a specific approach that can understand and model the nuanced connections between these musical elements. Consequently, research specifically targeting call-and-response generation is needed to bridge the gap between current methods and the effective generation of music using this essential compositional technique.

4.3 Call-Response Dataset

4.3.1 Music Collection

In this study, our primary objective is to investigate the call-and-response technique in music composition and develop an effective model for generating music responses based on input music calls. To achieve this goal, we selected the Pop909 dataset [1], a comprehensive collection of 909 pop piano music pieces, as the foundation for our research. We chose this dataset for several reasons. First, pop music frequently employs call-and-response, making it an ideal choice for studying this technique. Second, this dataset provides well-structured music with note timing and chord information. By manually extracting and annotating call-and-response pairs from this dataset, we aim to gain a deeper understanding of this technique, paving the way for more advanced and targeted music generation models that can effectively incorporate call-and-response.

4.3.2 Annotation Process

To accurately extract call-response pairs, we enlisted two annotators with musical training backgrounds to manually identify calls and responses.

4.3.2.1 Music Pre-processing

During this step, each music piece is evaluated to determine whether it is complete and contains call-and-response pairs. Music pieces that pass this step are further annotated.

4.3.2.2 Annotation Collection

In this step, an annotators is asked to label all possible call-response pairs within the music. The annotator extracts and saves the final pairs in separate MIDI files. All pairs preserve the information from the original dataset, which consists of three tracks: melody (lead melody), piano (primary accompaniment arrangement), and bridge (secondary melodies or lead instrument arrangements).

4.3.2.3 Quality Control

To ensure the accuracy and consistency of annotations, we implemented a quality control process. A second annotator was asked to independently review the call-and-response pairs and check for errors or inconsistencies. Any disagreements were then discussed and resolved. The final call-and-response pairs were compiled and used for subsequent experiments.

4.3.3 Dataset Exploration

4.3.3.1 Objective Evaluation Metrics

We evaluate the quality of calls and responses based on their rhythm, melody, and harmony by examining their similarity. For this purpose, we use dynamic time warping (DTW) distance and maximum common subsequence (MCS). The DTW measures the similarity between two sequences of different lengths, while the MCS measures common patterns between two sequences, even if they are not numerical. These metrics have been widely used in music composition for similarity comparisons [12, 72, 78, 79].

For rhythm and melody evaluation, we use DTW to measure quality. Rhythm is determined by two different attributes: note position and note duration. We measure the DTW distance of note position and note duration between the call and the ground-truth response, with the results denoted as DTW_P and DTW_D. Next, we calculate the DTW distance of note position and note duration between the input call and generated response, denoted as $\hat{DTW}_P$ and $\hat{DTW}_D$.

Rhythm Quality (RQ) is then computed as a normalized metric ranging from 0 to 1:

$$RQ = 1 - \frac{1}{2} \times (\frac{|DTW_P - D\hat{T}W_P|}{max(DTW_P, D\hat{T}W_P)} + \frac{|DTW_D - D\hat{T}W_D|}{max(DTW_D, D\hat{T}W_D)}) \tag{4.1}$$

If the generated response differs significantly from the ground-truth response, RQ approaches 0; otherwise, it is closer to 1. A similar process is applied to measure Melody Quality (MQ) using the DTW metric:

$$MQ = 1 - \frac{|DTW_M - D\hat{T}W_M|}{max(DTW_M, D\hat{T}W_M)} \tag{4.2}$$

For harmony evaluation, we measure the maximum common subsequence of chord progressions between two music pieces, as chord progression is a fundamental element of harmony. This method identifies repetitive patterns in chord progressions. Harmony Quality (HQ) is calculated as:

$$HQ = 1 - \frac{|MCS_H - M\hat{C}S_H|}{max(MCS_H, M\hat{C}S_H)} \tag{4.3}$$

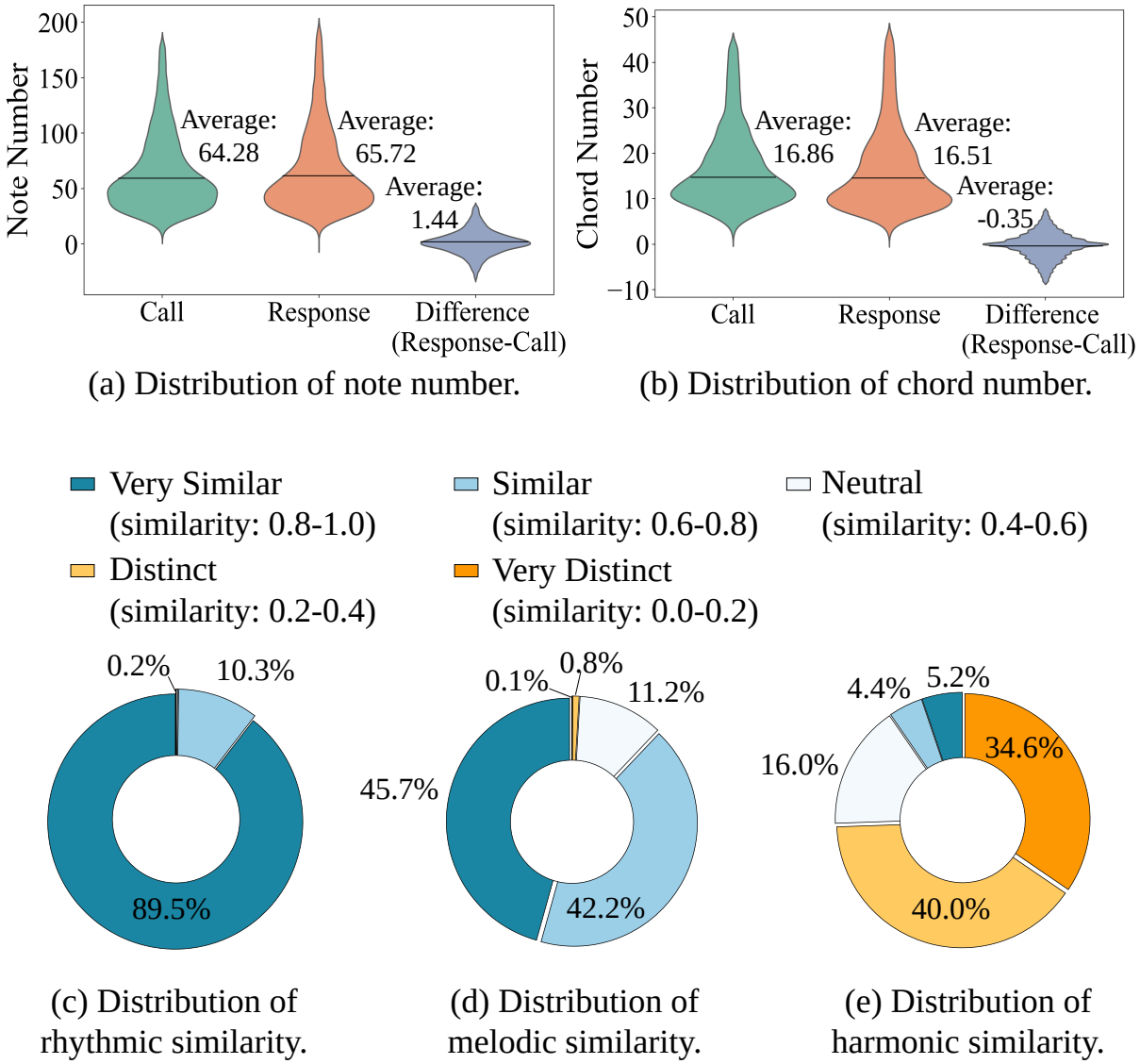
In addition, we use N-gram to measure the diversity of the generated responses. The N-gram measures the average number of times each n-gram appears in the generated music. This is a metric widely used for assessing the diversity of text generated by language models [80] and has been adopted in music composition [31]. We define the n-gram diversity of the pitch sequence within the generated response D_n as:

$$D_n = \frac{|T|}{|G| - n + 1}, \tag{4.4}$$

where $|T|$ is the number of unique n-grams in G, and $|G| - n + 1$ is the number of total n-grams that can be formed from G.

4.3.3.2 Dataset Statistics

We extracted a total of 19,155 call-response pairs, totaling over 108 hours. Our CRD is the first to annotate the call-and-response technique in machine composition. Figure 4.3 (a) and (b)



(a) Distribution of note number.

(b) Distribution of chord number.

(c) Distribution of rhythmic similarity.

(d) Distribution of melodic similarity.

(e) Distribution of harmonic similarity.

Figure 4.3: Call-Response Dataset (CRD) statistics. (a-b) Distributions of note and chord numbers. (c-e) Distributions of similarity between call-response pairs in terms of rhythm, melody, and harmony.

illustrate the distribution of note and chord numbers in calls, responses, and call-response differences. Most music pieces feature approximately 65 notes and 16 chords. The length difference is centered around 0, which suggests that the lengths of call-and-response are similar in most cases. Additionally, we demonstrate the rhythm, melody, and harmony similarity between each call-response pair in Figure 4.3 (c-e). For rhythm and melody, most call-response pairs exhibit similar rhythm and melody patterns. For harmony, we find that most call-response pairs have distinct harmony patterns.

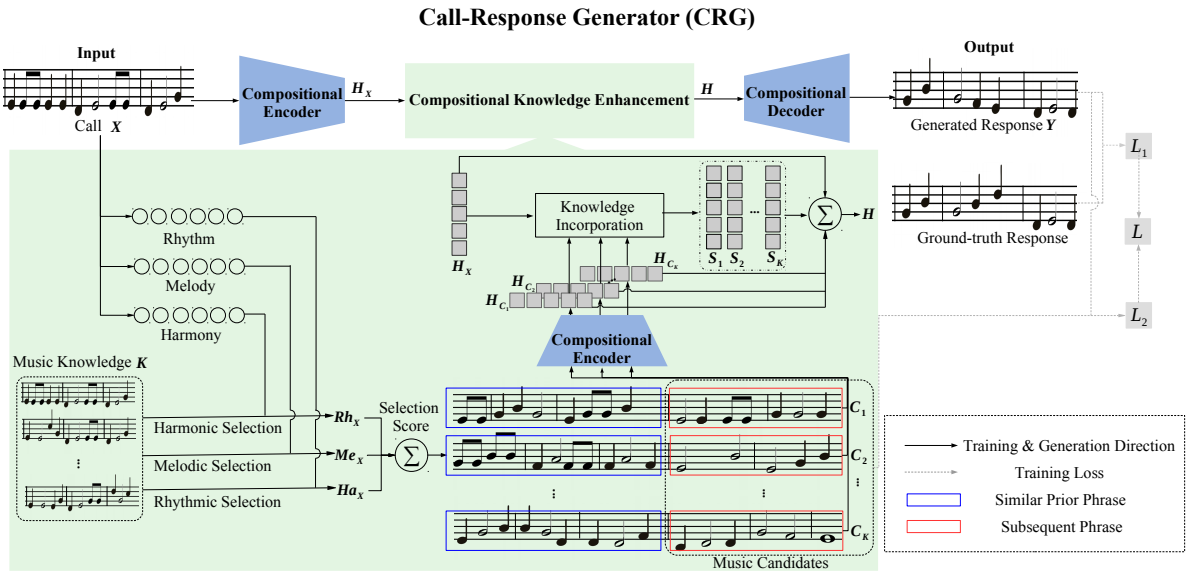


Figure 4.4: The framework for Call-Response Generator (CRG). CRG takes a call as input and utilizes knowledge as a reference to generate high-quality responses.

4.4 Proposed Method

4.4.1 Overview

In this section, we introduce the Call-Response Generator (CRG), a model that features a knowledge-enhanced mechanism for call-and-response generation, as illustrated in Figure 4.4. The model takes a music piece as input, referred to as the music call, and generates a music response that complements the input. Additionally, the model utilizes information from music knowledge as a reference to produce improved responses.

CRG consists of two main modules: a compositional encoder and decoder module, and a compositional knowledge enhancement module. The compositional encoder and decoder architecture learns the hidden representation of the input and generates the output music. The compositional knowledge enhancement module first identifies potential music candidates based on specific attributes from the input call and incorporates information from music candidates into the model, assisting the compositional decoder in generating different responses.

Mathematically, let $\mathbf{X}$ be a two-dimensional matrix of size $L_1 \times A$, where L_1 represents the length of the music call sequence and A denotes the number of attributes. Each token of $\mathbf{X}$ is a natural number, so $\mathbf{X} \in \mathbb{N}^{L_1 \times A}$. We construct music knowledge $\mathcal{K}$ and design a model M_θ that generates response $\mathbf{Y} = M_\theta(\mathbf{X}, \mathcal{K})$. The response $\mathbf{Y} \in \mathbb{N}^{L_2 \times A}$ has a length L_2 corresponding to

the input.

4.4.2 Compositional Encoder and Decoder

The compositional encoder and decoder together constitute a sequence-to-sequence model, which has proven effective in constructing end-to-end music generation systems [23]. The encoder maps a variable-length input music sequence (call) into a fixed-length vector representation, while the decoder translates the vector representation into a variable-length output music sequence (response).

From a probabilistic standpoint, the encoder-decoder framework learns the conditional distribution of a variable-length sequence given another variable-length sequence:

$$P(\mathbf{Y}|\mathbf{X}) = P(y_1, \ldots, y_{L_2}|x_1, \ldots, x_{L_1}) = \prod_{t=1}^{L_2} p(y_t|\mathbf{X}, y_1, \ldots, y_{t-1}). \tag{4.5}$$

The encoder learns to encode a variable-length sequence into a fixed-length vector representation. We denote this process as:

$$\mathbf{H}_\mathbf{X} = \text{ENCODER}(E(x_1), E(x_2), \ldots, E(x_{L_1})), \tag{4.6}$$

where ENCODER$(\cdot)$ is the encoder, $E(\cdot)$ is the embedding layer, and $\mathbf{H}_\mathbf{X} \in \mathbb{R}^{L_1 \times H}$ is a sequence of hidden states $\{\mathbf{h}_i\}_{i=1}^{L_1}$ mapped from the input sequence. We use x_i for simplicity, but the actual process involves concatenating all hidden features of attributes at position i since music has more than one attributes.

The decoder aims to decode a given fixed-length vector representation into a variable-length sequence. Specifically, the decoder generates an output sequence one token at a time. At each step, the model is auto-regressive, taking the previously generated tokens as additional input when generating the next token. The decoding function is formally represented as:

$$\mathbf{s}_t = \text{DECODER}(\mathbf{s}_{t-1}, E(y_{t-1})), \tag{4.7}$$

$$p(y_t|y_{t-1}, y_{t-2}, \ldots, y_1) = \text{FC}(\mathbf{s}_t), \tag{4.8}$$

where $\mathbf{s}_t$ is the hidden state at time t, DECODER$(\cdot)$ is the decoder, and FC$(\cdot)$ is a fully connected layer that transforms the hidden states into the probability of y_t. It should be noted that the initial hidden state $\mathbf{s}_0$ corresponds to the hidden representation of the input $\mathbf{H_X}$.

The compositional encoder and decoder can be trained as follows:

$$\mathcal{L}_0(\theta) = -logp_\theta(\mathbf{Y}|\mathbf{X}) = -\sum_{t=1}^{L_2} log(p_\theta(y_t|y_{<t}, \mathbf{X})). \tag{4.9}$$

4.4.3 Compositional Knowledge Enhancement

Incorporating knowledge into a model can be a complex task due to the vast amount of information that needs to be processed. To effectively extract valuable information from knowledge, we must develop a method for filtering and identifying relevant data. In the context of music theory, the phenomenon of musical call-and-response requires intelligibility, which leads to specific correlations between the call-and-response concerning rhythm, melody, and harmony [67, 81].

We propose referring to similar compositions found in knowledge $\mathcal{K}$ and making the assumption that if a prior phrase is similar to the input call, then the subsequent music phrase will be useful for generating the desired response. Therefore, we establish three selection rules based on music theory, considering rhythm, melody, and harmony perspectives [67, 81]:

1. Rhythmic selection: Music phrases should possess a rhythmic pattern similar to the original input call.

2. Harmonic selection: Music phrases should display a chord progression similar to the original input call.

3. Melodic selection: Music phrases should showcase a melodic pattern similar to the original input call.

We extract the rhythm information of the input call and compare it with the rhythm information of music in knowledge $\mathcal{K}$. The comparison is executed by calculating the similarity between the rhythm of the input call and all phrases from $\mathcal{K}$. We denote the similarity after

rhythmic selection as $Rh_{\mathbf{X}} \in \mathbb{R}^N$, where N is the number of music pieces in $\mathcal{K}$. Similarly, we can extract the melody and harmony of the input call $\mathbf{X}$ and calculate the similarity, denoting them as $Me_{\mathbf{X}} \in \mathbb{R}^N$ and $Ha_{\mathbf{X}} \in \mathbb{R}^N$. In addition, we define the selection score as the summation of rhythm, melody, and harmony:

$$\text{Selection Score} = Rh_{\mathbf{X}} + Me_{\mathbf{X}} + Ha_{\mathbf{X}} \tag{4.10}$$

Next, we sort all music in knowledge in descending order based on their selection score. We select the top K elements from knowledge and construct a music candidate list from their subsequent phrases $\mathbf{C} = \{\mathbf{C}_1,\ \mathbf{C}_2, ..., \mathbf{C}_K\}$, where $\mathbf{C}_k \in \mathcal{K}$.

For K music candidates, the hidden representation after the compositional encoder can be denoted as $\{\mathbf{H}_{\mathbf{C}_1}, \mathbf{H}_{\mathbf{C}_2}, \ldots, \mathbf{H}_{\mathbf{C}_K}\}$, following Equation 4.6. Given the input hidden representation $\mathbf{H}_{\mathbf{X}}$ and hidden representations from candidates, the knowledge incorporation module calculates the similarity between music candidates and the input music:

$$\mathbf{S}_k = \frac{FC(\mathbf{H}_{\mathbf{X}}) \cdot FC(\mathbf{H}_{\mathbf{C}_k})^T}{||FC(\mathbf{H}_{\mathbf{X}})|| * ||FC(\mathbf{H}_{\mathbf{C}_k})||^T}, \tag{4.11}$$

where $\cdot$ represents the dot product along the dimension of hidden space H, $*$ indicates element-wise multiplication, and $||\cdot||$ denotes the L2 norm.

Then, it incorporates the knowledge from music candidates into the input based on similarity. The final hidden representation fed into the decoder at time $t = 0$ is as follows:

$$\mathbf{H} = \mathbf{H}_{\mathbf{X}} + \sum_{k=1}^{K} \mathbf{H}_{\mathbf{C}_k} * (\mathbf{S}_k \cdot \mathbf{1}^\top), \tag{4.12}$$

where $\mathbf{1}$ represents a vector of ones with length H. Finally, $\mathbf{H}$ is fed into the compositional decoder to generate the output response $\mathbf{Y}$.

4.4.4 Knowledge-enhanced Response Generation

We first design two different objectives to train the model. Objective 1 is designed as a knowledge-enhanced objective, where the model is trained on $(\{\mathbf{C}_1, \ldots, \mathbf{C}_K, \mathbf{X}\}, \mathbf{Y})$ training examples. The

objective function is:

$$\mathcal{L}_1(\theta) = -logp_\theta(\mathbf{Y}|\mathbf{X}, \mathcal{K}) = -\sum_{t=1}^{L_2} log(p_\theta(y_t|y_{<t}, \mathbf{X}, \{\mathbf{C}_1, \ldots, \mathbf{C}_K\})), \tag{4.13}$$

where $\mathcal{K}$ represents knowledge, and $\mathbf{C}_i$ can be selected through knowledge selection.

Objective 2 is designed as an autoencoder-based knowledge-enhanced objective, trained on $(\{\mathbf{C}_1, \ldots, \mathbf{C}_K, \mathbf{X}\}, \mathbf{C}_i)$ training examples. The goal is to predict music candidates instead of the ground-truth response. The objective function is:

$$\begin{aligned} \mathcal{L}_2(\theta) &= -\frac{1}{K} * \sum_{i=1}^{K} logp_\theta(\mathbf{C}_i|\mathbf{X}, \mathcal{K}) \\ &= -\frac{1}{K} * \sum_{i=1}^{K}\sum_{t=1}^{L_2} log(p_\theta(c_{i,t}|c_{i,<t}, \mathbf{X}, \{\mathbf{C}_1, \ldots, \mathbf{C}_K\})). \end{aligned} \tag{4.14}$$

The model is optimized based on Objective 1 and 2 on call-response pairs. The goal can be expressed as follows:

$$\mathcal{L} = \lambda\mathcal{L}_1 + (1 - \lambda)\mathcal{L}_2, \tag{4.15}$$

where λ is a hyper-parameter.

During the generation phase, the response is generated in a recurrent manner. The input call X is fed into the model, and CRG first identifies K music candidates. Subsequently, it generates each token sequentially until the <EOM> (end of music) token is produced. In order to convert the output probability $p_\theta(y_t|y_{<t}, \mathbf{X}, \mathbf{C}_1, \ldots, \mathbf{C}_K)$ into the actual token, we employ a stochastic temperature-controlled sampling method [30]. By leveraging the music candidates, CRG can generate responses that exhibit similar rhythmic, melodic, and harmonic patterns to the input call, resulting in a more musically satisfying output. In addition, this generation process allows the model to create a range of responses while maintaining coherence with the input call.

4.5 Experiments

4.5.1 Implementation Details

In our experiment, we convert symbolic music into a sequence of compound words using a predefined music attribute vocabulary [30]. We represent musical information through seven attributes: tempo, chord, position, type, pitch, duration, and velocity. Each sequence is set to a length of 256 tokens. For knowledge selection, we employ DTW and MCS to calculate the similarity between input call and knowledge, and normalize DTW between 0 and 1 by dividing the maximum possible DTW value for each pair of sequences. We use the Pop1K7 dataset [30] as music knowledge. During training, we set the hyper-parameters λ_1 and λ_2 to $\frac{1}{2}$. Additionally, we reserve call-response pairs from 40 songs for validation and testing purposes. This experimental setup enables us to evaluate the model's performance and its ability to generate high-quality musical responses.

Table 4.1: Model comparison: results of the objective and subjective evaluations. † indicates statistically significant improvement over Music-Transformer.

	Objective						Subjective			
Model	RQ ↑	MQ ↑	HQ ↑	1-gram ↑	2-gram ↑	3-gram ↑	Quality ↑	Groove ↑	Coherence ↑	Diversity ↑
Music-Transformer [23]	0.30	0.26	0.52	0.26	0.66	0.87	2.82	2.99	2.19	2.28
CP-Transformer [30]	0.29	0.29	0.61	0.19	0.62	0.87	2.61	2.33	2.09	2.09
HAT [31]	0.41	0.29	0.57	0.24	0.65	0.87	2.79	3.08	1.90	2.66
CRG	**0.59**†	**0.57**†	**0.72**†	**0.31**†	**0.77**†	**0.92**†	**3.62**†	**3.76**†	**3.48**†	**3.91**†

Table 4.2: Ablation study: results of the objective and subjective evaluations. † indicates statistically significant improvement over CRG-Base.

	Objective						Subjective			
Model	RQ ↑	MQ ↑	HQ ↑	1-gram ↑	2-gram ↑	3-gram ↑	Quality ↑	Groove ↑	Coherence ↑	Diversity ↑
CRG-Base	0.44	0.29	0.57	0.24	0.66	0.87	3.01	2.75	2.90	2.75
CRG-K	0.48	0.57	0.62	0.26	0.73	0.91	**3.68**	3.59	3.42	2.99
CRG-KM1	0.57	0.55	0.71	0.30	0.75	0.91	3.58	3.60	3.45	2.96
CRG-KM2	0.38	0.54	0.50	0.21	0.65	0.87	3.31	2.89	3.39	3.76
CRG-KM12	**0.59**†	**0.57**†	**0.72**†	**0.31**†	**0.77**†	**0.92**†	3.62†	**3.76**†	**3.48**†	**3.91**†

4.5.2 Subjective Metrics and Participants

In addition to the objective metrics outlined in Section 3.3, we also carry out a subjective evaluation on social media to ensure that the generated responses are enjoyable and coherent with their corresponding calls. For each call, we design several sets of responses for different models and one set of human-composed responses. In each set, we randomly select three versions of responses for the diversity test of the response. Each set is played randomly and the user can play the call an unlimited number of times during the experiment for easy comparison. In each rating session, a listener first enters their music background information and then provides ratings for the four response sets. All responses in each set are generated from the same call. For human-composed music, we select three different versions of the response in the same music. For each set, the listener answers: whether they heard the songs before the survey (yes or no); and then evaluates the music in terms of quality, groove, coherence, and diversity [12, 31]:

- Overall Quality (Q): Please rate the overall quality of the response on a scale of 1 (lowest) to 5 (highest).
- Groove (G): Please rate the fluency of the response and the appropriateness of pauses on a scale of 1 (lowest) to 5 (highest).
- Coherence (C): Please rate the coherence of the call-and-response connection on a scale of 1 (lowest) to 5 (highest).
- Diversity (D): Please rate the diversity of the different responses on a scale of 1 (lowest) to 5 (highest).

We collected 114 listener reports by distributing a survey on social media. After removing invalid answers that didn't meet our criteria (such as responses submitted too quickly, no response given, hearing the music before the test, and so on), we were left with 89 valid samples. Among the 89 participants, 34 were men and 55 were women; 18 were under age 20, 24 were between 20 and 30, 34 were between 31 and 40, 8 were between 41 and 50, and 5 were older than 50. In addition, 19 had musical training for more than 5 years and were familiar with music

theory; the others had no or less than 5 years of musical training. All participants were Chinese, given that the POP909 dataset mainly consists of Chinese pop songs, and listeners with a stronger familiarity with this genre are expected to provide more dependable and discerning evaluations. All participants in the subjective experiment were informed of the study's purpose and provided their consent for the use of their data in our research.

4.5.3 Model Comparison

We evaluate the effectiveness of our proposed model by comparing it with several machine composition models. While there are no existing models that specifically target call-and-response in music composition, these three methods represent various approaches in the field of machine composition that address similar tasks or share some common goals.

- **Music Transformer**: This model employs the Transformer settings [23] and was the first Transformer-based model developed for composition.
- **CP-Transformer**: Using the Transformer encoder and a linear decoder on music compound word representation [30], this model represents a state-of-the-art learning-based method for composition.
- **HAT**: Based on the settings in [31], this model is a state-of-the-art knowledge-enhanced method for pop music generation that incorporates music harmonic knowledge.

Table 4.1 presents a comparison of our proposed model's performance with that of other models in both objective and subjective experiments, highlighting the improvements achieved in terms of quality and diversity. The objective experimental results demonstrate that our model, by concentrating on crucial aspects such as rhythm, melody, and harmony, is able to generate responses with enhanced musical quality. Additionally, the diversity of the generated responses is increased, contributing to a more varied listening experience.

As for the subjective experimental results, the music generated by our proposed model surpasses state-of-the-art models in all four aspects, emphasizing the effectiveness of our approach. Moreover, the coherence score also confirms that other models struggle to generate suitable re-

sponses that echo the call, thus limiting the development of artistic quality in machine-composed music.

4.5.4 Ablation Analysis

We have designed five ablation variants of the CRG model to better understand the contributions of the proposed approach. These variants include:

- **CRG-Base**: This model adopts the Transformer encoder-decoder settings and is directly trained on call-response pairs using L_0 without the knowledge-enhanced mechanism.
- **CRG-K**: This is our proposed model with the knowledge-enhanced mechanism, directly trained on Objective 1, aiming to evaluate the influence of the knowledge-enhanced mechanism.
- **CRG-KM1**: Another proposed model variant that integrates the knowledge-enhanced mechanism, trained based on L_0, and further trained on Objective 1.
- **CRG-KM2**: Another proposed model variant that integrates the knowledge-enhanced mechanism and is further trained on Objective 2.
- **CRG-KM12**: Another proposed model variant that integrates the knowledge-enhanced mechanism and is further trained on both Objectives 1 and 2.

Table 4.2 demonstrates that the knowledge-enhanced mechanism and multiple training objectives improve the proposed model's performance. The CRG-KM12 variant, which incorporates the mechanism and trains on both objectives, achieves the best scores in objective and subjective evaluations. This highlights the importance of combining music knowledge with multiple training objectives to generate high-quality, diverse, and engaging call-response music.

4.5.5 Knowledge Analysis

4.5.5.1 Knowledge Type

Different types of knowledge can impact model performance, as they provide varying information to the model. In this experiment, we examine three distinct types of knowledge:

- Random knowledge: Knowledge candidates are randomly chosen from available knowledge.
- Distinct knowledge: Knowledge candidates possess composition skills that are distinct from those of the input call.
- Similar knowledge: Knowledge candidates have composition skills that closely resemble those of the input call.

Table 4.3 and 4.4 demonstrate that all types of knowledge contribute to improved performance since knowledge is derived from music samples, which consistently adhere to music theory and provide information that aids composition. However, knowledge candidates with similar composition attributes yield greater enhancements in model performance compared to the other two knowledge types. This observation aligns with music theory, which suggests that specific correlations exist between calls and responses.

Table 4.3: Objective evaluation on different knowledge types: music quality.

Type	Music Quality		
	RQ ↑	MQ ↑	HQ ↑
Random	0.55±0.19	0.54±0.39	0.71±0.22
Distinct	0.56±0.19	0.52±0.39	0.71±0.22
Similar	**0.57±0.20**	**0.57±0.39**	**0.72±0.22**

4.5.5.2 Knowledge Candidate Number

The diversity of the call is influenced by the number of candidates. Figure 4.5 displays the music quality and diversity as the number of knowledge increases. A trade-off between music diversity and music quality is observed when increasing the number of knowledge candidates.

Table 4.4: Objective evaluation on different knowledge types: music diversity.

Type	Music Diversity		
	1-gram ↑	2-gram ↑	3-gram ↑
Random	0.29±0.11	0.72±0.12	0.89±0.08
Distinct	0.30±0.10	0.73±0.11	0.90±0.07
Similar	**0.31±0.11**	**0.76±0.11**	**0.92±0.06**

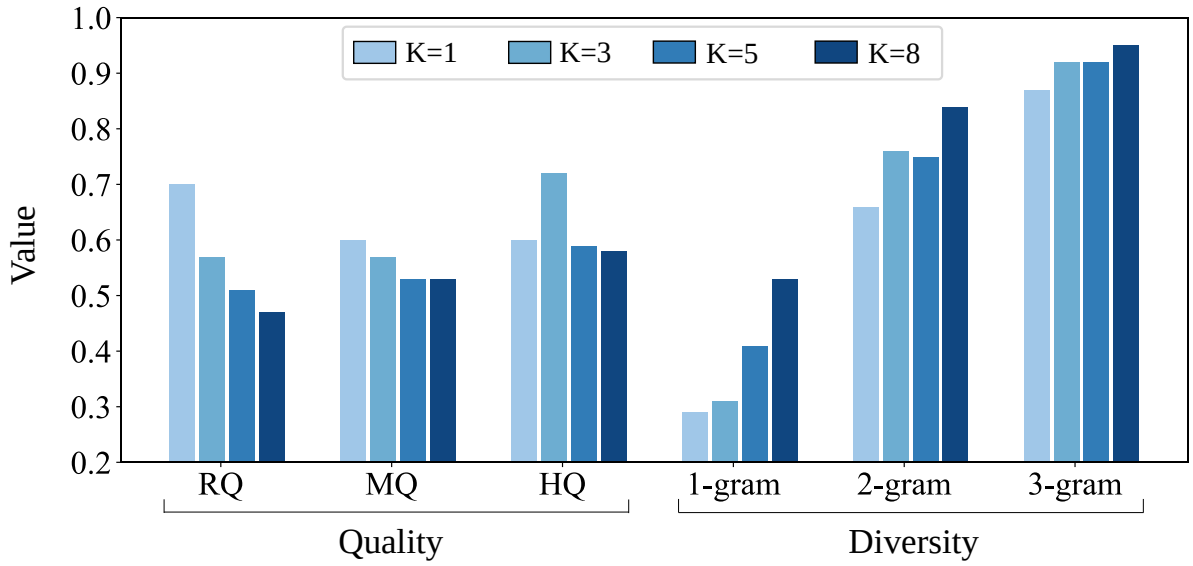


Figure 4.5: Objective evaluation on different knowledge candidate numbers: music quality and diversity.

4.5.6 Comparison with Human-Composed Music

We also conducted a comparison between human-composed music and machine-composed music from CRG, as shown in Figure 4.6. Although the results suggest that human-composed music received higher ratings than the proposed model in terms of quality, groove, and coherence, the subjects perceived greater diversity in the music generated by CRG. Figure 4.7 demonstrates the adaptability of CRG in generating a variety of outcomes tailored to a specific call. Unlike human-composed music, which often maintains few consistent call-and-response effects to preserve the main idea of the song, CRG can produce diverse effects based on the call, making it a valuable tool for arrangement or style transfer. Furthermore, the CRG's ability to generate diverse responses can potentially inspire and complement human creativity during the composition process. By offering different call-and-response effects, the CRG serves as a valuable resource for composers seeking fresh ideas or novel perspectives while crafting their musical

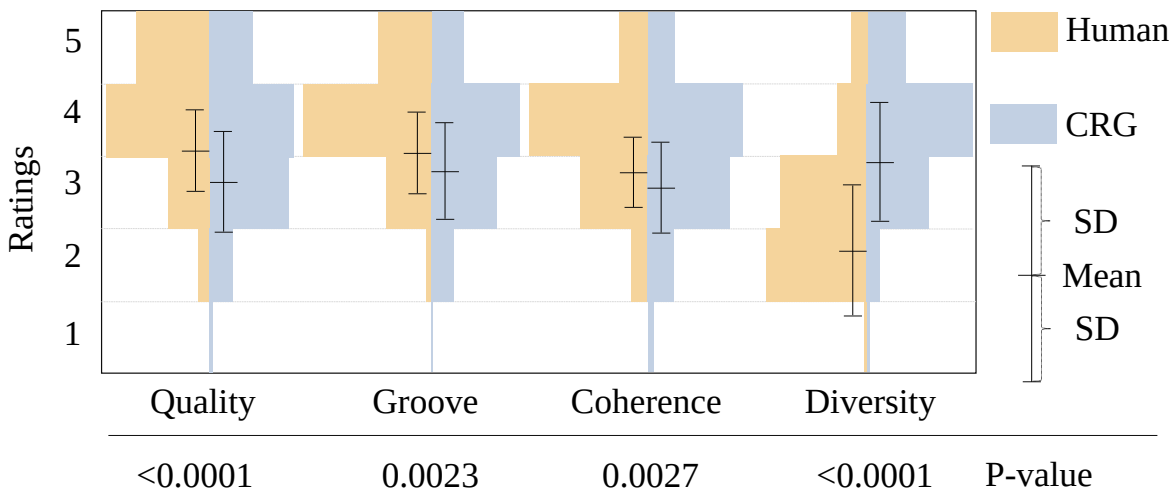


Figure 4.6: Subjective evaluation between human-composed music and machine-composed music (Call-Response Generator).

pieces.

4.6 Summary

To the best of our knowledge, this is the first work that applies the call-and-response technique, an essential composition method, to learning-based algorithms. We contribute a new pop music dataset consisting of 19,155 call-response pairs derived from 909 songs. In addition, we define three objective evaluation metrics to assess the quality of the music. Based on this dataset, we propose a novel Call-Response Generator (CRG) with a knowledge-enhanced mechanism that incorporates relevant music theory by providing more instructive training data, in addition to the ground-truth call-response labels. Compared to existing learning-based models, our proposed techniques have significantly improved the quality of machine-composed music in terms of rhythm, melody, and harmony.

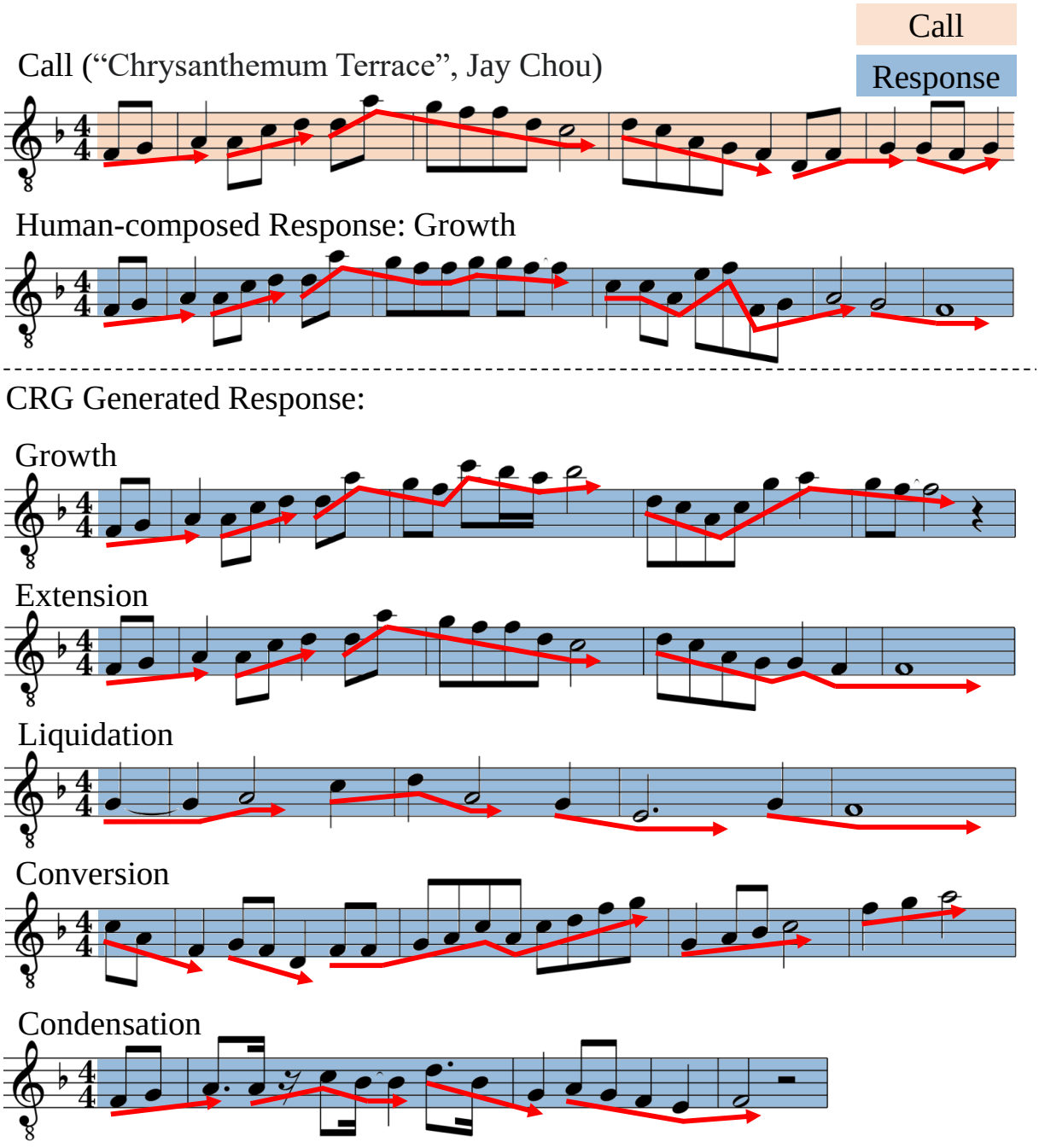


Figure 4.7: Demonstration of responses generated by Call-Response Generator (CRG) and those composed by humans. Audio files are available in supplementary materials.

Chapter 5

A Domain Adaptation Method for Therapeutic Music Transfer

5.1 Background and Motivation

Anxiety is becoming one of the most common mental health problems in the world, affecting nearly 300 million people [82]. Individuals with anxiety display various symptoms, such as trouble concentrating and restlessness [83], which influence their everyday lives. This condition has also grown into a social phenomenon that cannot be ignored; anxiety has only continued to escalate due to the pandemic [84]. Music therapy offers a useful means of anxiety reduction [85, 86] with negligible side effects. As the fundamental of music therapy, available therapeutic music cannot satisfy demand. For example, people from different social and cultural backgrounds have diverse music preferences and distinct feelings about the same types of music. It is challenging for therapists to find music that both meets subjects' requirements and is suitable for anxiety reduction. In addition, the time-limited nature of therapy renders music selection difficult and further impedes the performance of music therapy.

Given the growing demand for therapeutic music, music style transfer represents one way to generate a larger volume of therapeutic music for music therapy sessions within a short time. Different from music generation tasks [87] that generate music from random noises, music style transfer enables the transfer of one piece of music to new styles while preserving the content of

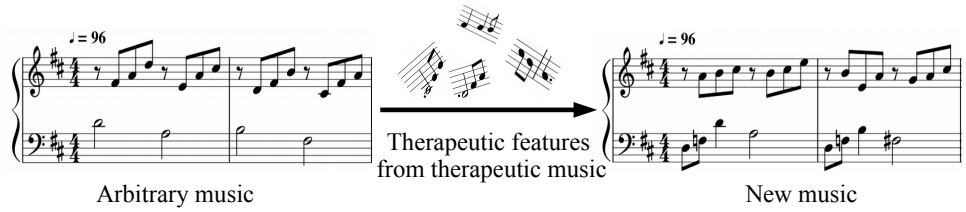


Figure 5.1: A demonstration of therapeutic music transfer for therapeutic music.

the piece. Similar to "style features" in different music styles, we consider existing therapeutic music as a music style and they share certain "therapeutic features" that can reduce patients' anxiety. We use style transfer techniques to learn these features and make a piece therapeutic. By applying music style transfer techniques, the new music learns "therapeutic features" while maintaining the content of the original music. A schematic diagram is provided in Figure 5.1.

The challenge of generating new therapeutic music through music style transfer lies in two main obstacles. First, the style transfer methods demand an extensive amount of training data, which is limited for both user-favorite music and therapeutic music. Only a handful of songs have been proven to reduce anxiety, leaving the impact of other songs yet to be determined [88]. In a clinical setting, it is impractical and time-consuming for therapists to test each song on different patients. Second, defining therapeutic style is difficult. The selection and validation of therapeutic music is based on years of clinical experience and encompasses a wide range of genres and arrangements, making it difficult to pinpoint the specific features that make a piece of music therapeutic. This lack of consensus on what constitutes therapeutic style makes it inappropriate to apply existing music style transfer methods to generate therapeutic music.

To tackle these difficulties, we have created a new approach, known as the Therapeutic Music Transfer Model, which can transfer the user's preferred music into a new piece that has therapeutic properties. Our model includes a unique feature extraction network and optimization method. The feature extraction network, based on a convolutional neural network, identifies main and secondary features of music from a musical genre dataset. The main features of music include elements such as pitch, note length, chord progressions, repeating patterns, etc., which are crucial to understanding the music. Secondary features, on the other hand, are derived from the main features and reflect relationships between different pitches, chords, and repeating patterns, adding stylistic information to the input. The network's objective is to mini-

mize the difference in secondary features within the same style and maximize the difference in secondary features between different styles. Using the feature extraction network, we have developed a joint optimization method to maintain the critical information of the user's preferred music, considered as main features, and to learn therapeutic information from therapeutic music, represented in secondary features. With just two pieces of music input - one user's favorite and one therapeutic - the model outputs a new piece of music that retains the melody of the user's preferred music while also having therapeutic properties. Furthermore, our model significantly speeds up the learning process.

To the best of our knowledge, this is the first application of music style transfer in the field of music therapy. Our model demonstrates its effectiveness and feasibility through both objective and subjective experiments and provides a convenient solution to the limitations of limited favorite and therapeutic songs in clinical practice.

5.2 Related Work

5.2.1 Therapeutic Music

Therapeutic music refers to music intended to assuage physical, emotional, or mental concerns and is often used in music therapy. The relationship between music, emotion and mental health has been studied for decades [89–91]. However, the development of music therapy has made it more challenging for available pieces to satisfy diverse subjects. Academic researchers have used different types of music to reduce anxiety, such as subjects' favorite music [92–94], natural sounds[95, 96], and classical music [97–99]. Few papers have indicated which music was used in experiments. The music in some experiments is too personalized to be used with subjects from diverse backgrounds. Therapeutic music has also been adopted for different subjects in clinical trials; however, few books have described the music applied in music therapy, which means therapeutic music is limited and the target patients are restricted. It is challenging to collect a therapeutic music dataset, not to mention a large dataset for deep learning training. As an exception, in [88], the author elaborated on music played in different music therapy stages and scenarios. We, therefore, gathered the music mentioned in [88] from open online sources.

Based on music used in clinical trials, the proposed model seeks to transfer more music to be therapeutic for subjects from an array of social and cultural backgrounds.

5.2.2 Music Style Transfer

Style transfer was first studied and proposed by Gatys *et al.* , who transferred one image to a different image style by controlling hidden features learned via convolutional neural networks (CNNs)[11]. Scholars later introduced the idea of style transfer to the music domain. Styles of music tend to be more abstract than those of images because music can be appreciated from different aspects. Three general types of music style transfer currently exist: timbre style transfer, performance style transfer, and composition style transfer [100].

Timbre style transfer transfers the timbre of music while maintaining the original piece's melody, and expression. This method is usually applied to audio data such as waveforms, spectrograms of waveforms, or acoustic features extracted from audio data [101–105]. Performance style transfer aims to transfer performance from one style to another while making the music more realistic [106, 107]. This type of transfer has not been well studied. In one attempt, Choi *et al.* presented a Transformer autoencoder to combine global representation with temporal embeddings to transfer music performance and melody [108]. Composition style transfer, which is the goal of the present paper, involves transferring distinguishable characteristics of input music (e.g., note pitch, note duration, and note position) while meaningfully preserving certain characteristics [33, 34, 37–39, 109, 110]. Some studies in this regard have involved audio data. For example, Mor *et al.* proposed a wavenet autoencoder–based model that can transfer music across musical instruments, genres, and styles [110]. Many studies on this form of transfer have featured symbolic music data. Brunner *et al.* applied MIDI-VAE and the CycleGan network for music genre transfer while taking symbolic music data as a matrix [33, 34]. Yang *et al.* separated the pitch and rhythm features of music and recombined them to transfer a piece using autoencoders [38, 39]. However, these methods require a large dataset of the same styles to train a data-driven model. Cífka *et al.* devised a one-shot style transfer method using an encoder–decoder neural network [37]. Yet this approach requires the generation of parallel training data for supervised learning in the training phase, and the quality of synthetic data is hard to guaran-

tee. Therefore, these methods are not directly applicable to therapeutic music since therapeutic music is limited and there is no parallel data to supervise the learning process.

Music transfer in therapeutic music is a new field worth exploring in terms of timbre, performance, and composition. Hu *et al.* proposed a method regarding transfer from a composition perspective. They first proposed a feature extraction network to extract main and secondary features representing the music's content and therapeutic characteristics and optimized the output music by minimizing the main and secondary feature differences between output music, therapeutic music, and users' favorite music [13]. However, this model requires an optimization problem to be solved for each transfer; the approach is thus inefficient. In addition, the authors did not consider long-term features, which are important in music. We focus on composition style transfer in this chapter. We specifically propose a well-trained model that considers spatial and short- and long-term temporal musical features and can transfer music without optimizing the model at each instance.

5.3 Proposed Method

The proposed therapeutic transfer model has a clear structure, as shown in Figure 5.2. The model takes two pieces of music as inputs - one chosen by the user and the other chosen from a list of therapeutic music. The output is a transferred piece of music that combines the user's favorite melody with therapeutic elements that can help reduce anxiety. The model is comprised of two parts - a pre-trained feature extraction network that uses convolutional neural networks (CNNs) and a joint optimization process for music style transfer.

To address the limited training data for user-favorite music, the feature extraction network is trained on a large music genre dataset using CNNs. This is because CNNs can maintain the second-order tensor of the pitch-temporal space during feature extraction, and also because they can abstract features at different levels, which can help to capture both the global and local structure of the music. The network is then used in the music style transfer process. The transferred music is initialized with the user's favorite music, and the features of different hidden layers are extracted. The model aims to minimize the difference between the generated music and both the user's favorite music (at the middle-level hidden features) and the therapeutic music

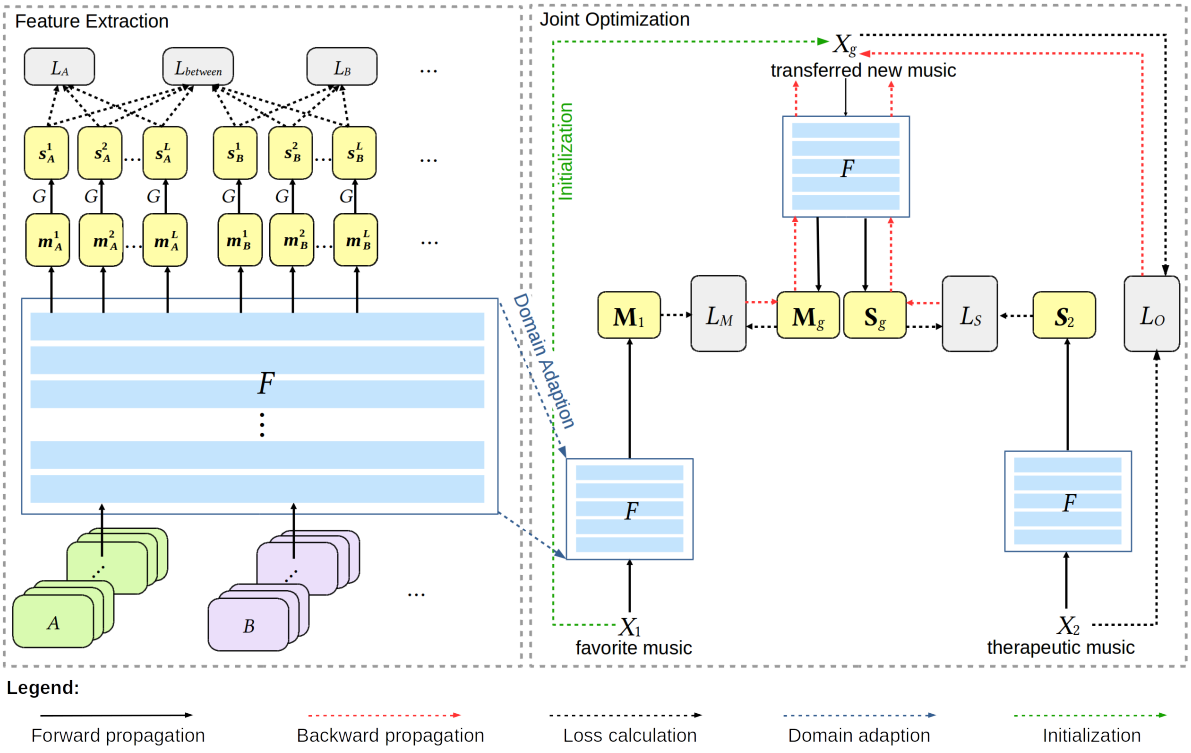


Figure 5.2: The proposed therapeutic music transfer model. Left: Feature Extraction, A and B represent two different music genres. Right: Joint Optimization. The model takes X_1 and X_2 as input, the feature extraction network F learns music features, the transferred music X_g is initialized as X_1.

(at the high-level hidden features and original features), by revising the generated music through iterative optimization.

5.3.1 Music Representation

Before discussing the proposed model at length, music representation is described in this subsection. We study symbolic music data in the MIDI format in this chapter. MIDI is a popular format used to represent valuable information in music problems [20, 33, 34, 54, 111]. Under this representation method, data can be constructed as a matrix containing pitch information, time information, and track (instrument) information. For simplicity, We merge all music tracks into one and play with piano, such that each piece of music x can be considered a 2-D matrix and represented as $x \in \{0, 1\}^{P \times B}$. The values of 1 and 0 in each entry respectively denote whether the note is played. "Note on" indicates that a note is played, whereas "note off" indicates that the note ends. The value P is the number of possible pitch values; B indicates beats, represented as a time variable to measure the music length x. Additionally, notes that have higher pitch values

are usually considered as the melody of the music while lower pitches are the accompaniment of the music.

5.3.2 Feature Extraction Network

Before diving into the architecture of the proposed model, three key statements are made. The idea to pre-train a music feature extraction network is derived from image style transfer: researchers typically rely on a well-trained feature extraction network (e.g., VGG-16 [112]) to extract content and style features. If the network is well-trained on an image classification dataset, then it can be used to extract hidden features of different image styles in image style transfer tasks.

Statement 1 A feature extraction network can be used for different types of music, even if it has been trained on a limited number of styles.

Statement 2 Secondary features contain stylistic information.

Statement 3 Therapeutic music shares common therapeutic properties and secondary features play a significant role in determining these therapeutic properties.

In light of these statements, we first establish a feature extraction network F that can learn hidden features. For this purpose, we use convolutional neural networks to extract features, and the architecture of the proposed feature extraction network is illustrated on the left side of Figure 5.2. The network consists of ten convolutional layers, and based on Statement 1, it is trained on music from various genres, with two genres - genre A and B - used for illustration. The output of the network contains different levels of hidden features learned by the convolutional layers, which are referred to as main features $\mathbf{M}$. These main features represent local information of the input, such as pitch value, note length, chord, and so on. Features from early layers are indistinct from the input, while features from higher layers represent the "deep" structure of the input [113]. To capture these differences, it is important to include features from different layers. Specifically, let $f^l(\cdot)$ be the activation functions of the feature extraction network F at layer l. The main feature of each layer $\mathbf{m}^l$ can be defined as $\mathbf{m}^l = f^l(\cdot)$ and $\mathbf{m}^l \in \mathbb{R}^{N^l \times H^l \times W^l}$ where

N^l indicates the number of feature maps at layer l, and H^l and W^l are the height and width of feature maps at layer l. In addition, $\mathbf{M} = [\mathbf{m}^1, \mathbf{m}^2, \cdots, \mathbf{m}^L]$, where L is the total number of levels of hidden features included in the main features $\mathbf{M}$, and the subscript is ignored when not specifying either the A or B domains.

We also introduce secondary features $\mathbf{S} = [\mathbf{s}^1, \mathbf{s}^2, \cdots, \mathbf{s}^L]$ to represent relationships among main features. $\mathbf{s}^l$ represents the secondary feature at layer l and is defined as the Gram matrix $G^l(\cdot)$ of $\mathbf{m}^l$. The elements can be calculated as:

$$G^l(\cdot)_{ii'} = \frac{1}{N^l H^l W^l} \sum_{h=1}^{H^l} \sum_{w=1}^{W^l} f^l(\cdot)_{h,w,i} f^l(\cdot)_{h,w,i'}, \tag{5.1}$$

where ii' is the coordinate of Gram matrix. Given that the main features of music contain information such as pitch and chord length, it is logical to believe that the relationship between these features plays a role in defining a music style. As a result, based on the second statement, music of the same style should have similar secondary features. In other words, the differences in secondary features within the same style should be minimized. The loss within style A is calculated as:

$$L_A = \sum_{l=1}^{L} \sum_{i,j=1}^{N_A} ||\mathbf{s}_{Ai}^l - \mathbf{s}_{Aj}^l||_F^2, \tag{5.2}$$

where N_A is the total number of music samples in style A and $|| \cdot ||_F$ is the Frobenius norm. Similarly, the loss within style B is calculated as:

$$L_B = \sum_{l=1}^{L} \sum_{i,j=1}^{N_B} ||\mathbf{s}_{Bi}^l - \mathbf{s}_{Bj}^l||_F^2, \tag{5.3}$$

where N_B is the total number of music samples in style B. The total loss within the style is then:

$$L_{within} = \alpha L_A + (1 - \alpha) L_B, \tag{5.4}$$

where α is the weight factor for two different styles.

Additionally, the difference in secondary features $\mathbf{S}$ between different styles should be maximized, as the stylistic information from different styles should be distinctive. The distance

between styles at layer l is defined as:

$$d^l_{AB} = ||Center^l_A - Center^l_B||^2_F, \tag{5.5}$$

where $Center^l A$ and $Center^l B$ represent the centers of secondary features A and B at layer l, respectively. According to [114], $Center^l A$ and $Center^l B$ are fixed as the sample mean of the initial network outputs to prevent all features from collapsing to 0 during training.

The within-style distance at layer l is defined as $d^l Ai = ||\mathbf{s}^l Ai - Center^l A||F^2$ and $d^l Bj = ||\mathbf{s}^l Bj - Center^l B||F^2$. Samples with larger within-style distances compared to $d^l AB$ will be penalized. Specifically, if dAi^l or d^l_{Bj} is smaller than or equal to d^l_{AB}, then there will be no penalty for the difference between styles. Otherwise, the secondary features will be penalized. A safety distance d_s is also added to the equation to make sure the secondary feature at each layer is distinguishable. Therefore, the between style loss is:

$$\begin{aligned} L_{between} = \sum_{l=1}^{L} (& \frac{1}{N_A} \sum_{i=1}^{N_A} \max(0, d^l_{Ai} - d^l_{AB} + d_s) + \\ & \frac{1}{N_B} \sum_{j=1}^{N_B} \max(0,\ d^l_{Bj} - d^l_{AB} + d_s)). \end{aligned} \tag{5.6}$$

The total loss for the feature extraction network is

$$L_f = \beta L_{within} + (1 - \beta) L_{between}, \tag{5.7}$$

where β is the weight factor for within and between loss.

5.3.3 Joint Optimization Method

The structure of the joint optimization method is similar to the one presented in [11], as illustrated on the right side of Figure 5.2. According to Statement 3, we believe that there are hidden relationships between secondary features and therapeutic information. Thus, the feature extraction network is utilized to extract both main and secondary features. The parameters of the feature extraction network F are fixed, and its purpose is to train the generated music ($\mathbf{X}_g$),

which is also the output of the proposed model. Given two pieces of music, $\mathbf{X}_1$ representing the user's favorite music and $\mathbf{X}_2$ representing the therapeutic music, $\mathbf{X}_g$ is designed to maintain the main information from $\mathbf{X}_1$ while learning therapeutic information from $\mathbf{X}_2$.

Initially, $\mathbf{X}_g$ is set to equal $\mathbf{X}_1$. Then, $\mathbf{X}_1$, $\mathbf{X}_g$, and $\mathbf{X}_2$ are fed into the pre-trained feature extraction network F. The main features of $\mathbf{X}_1$ and $\mathbf{X}_g$ are learned and represented as $\mathbf{M}_1$ and $\mathbf{M}_g$, respectively. Since $\mathbf{X}_g$ is intended to preserve the majority of information from $\mathbf{X}_1$, the main feature loss between $\mathbf{M}_g$ and $\mathbf{M}_1$ should be minimized. This is calculated as:

$$L_M = \sum_{l=1}^{L} \frac{1}{N^l H^l W^l} ||\mathbf{m}_\mathbf{g}^l - \mathbf{m}_1^l||_F^2. \tag{5.8}$$

In accordance with Statement 3, secondary features (**S**) are believed to reveal deeper music information, including therapeutic information. Thus, in order for $\mathbf{X}_g$ to be therapeutic, we assume that $\mathbf{S}_g$ should be similar to $\mathbf{S}_2$, where $\mathbf{S}_g$ and $\mathbf{S}_2$ are the secondary features of $\mathbf{X}_g$ and $\mathbf{X}_2$, respectively. The secondary feature loss between $\mathbf{S}_g$ and $\mathbf{S}_2$ should also be minimized. A weight factor is introduced at each layer to account for the fact that feature values at different layers have varying magnitudes, leading to large variations in the contribution to the total loss. The weight factor at each layer l is defined as:

$$\omega^l = -\log(\frac{\bar{\mathbf{s}}^l}{\sum_{l=1}^{L} \bar{\mathbf{s}}^l}), \tag{5.9}$$

where $\bar{\mathbf{s}}^l$ is the mean value of the secondary feature at layer l. Then, the secondary feature loss is defined as:

$$L_S = \sum_{l=1}^{L} \omega^l ||\mathbf{s}_g^l - \mathbf{s}_1^l||_F^2. \tag{5.10}$$

Additionally, in musical transfer, especially in therapeutic transfer, certain musical elements such as sudden changes in pitch, irrational chords, or repetitive patterns can negatively impact the music. Our experiments have shown that adding an additional term L_O to the transferring process can improve the outcome. This term, defined in Equation 5.11 as

$$L_O = \frac{1}{P \times B} ||\mathbf{X_g} - \mathbf{X_2}||_F^2, \tag{5.11}$$

This term not only measures the difference between the therapeutic and generated music but also helps in learning global therapeutic features. However, if L_{origin} becomes too dominant during the training process, it may cause the transfer to become trivial and result in the generated music being too similar to the user's favorite music. As such, L_{origin} is used to guide the training direction, but does not have a significant impact on learning local features. The final loss function, as seen in Equation 5.12,

$$L_t = \begin{cases} \lambda_1 L_M + \lambda_2 L_S, & if \text{ iteration } \mod e \neq 0, \\ \lambda_3 L_O, & if \text{ iteration } \mod e = 0, \end{cases} \tag{5.12}$$

where λ_1 and λ_2, λ_3 are weight factors for the main, secondary and origin loss. The value of e, which represents the training epoch, is determined during the experiment. This means that the value of $\mathbf{X}_g$ is primarily updated based on L_{main} and $L_{secondary}$, but at specific epochs, it is updated based on L_{origin}.

5.4 Experiments

In this section, we provide information on the datasets and implementation details. We then present results from three experiments. The first experiment demonstrates that our model is capable of capturing stylistic information through secondary features. In the second experiment, we demonstrate the music transfer process by showing an example and comparing the training time of our model to other existing style transfer models. Finally, we conducted a subjective experiment to evaluate the therapeutic effectiveness of the generated music samples.

5.4.1 Datasets & Implementation Details

5.4.1.1 Datasets

In order to train the network for feature extraction, we utilized the dataset collected by [33]. This dataset is well-labeled with different music genres and includes 477 Jazz songs, 554 Classic songs, and 659 Pop songs. To train the network, we selected Classic and Jazz as the input styles. To evaluate the performance of the network, we tested it on seven different music styles: folk,

Table 5.1: Architecture of feature extraction network

Input: (batch × 256×128×1)					
layer	kernel	stride	channel	batch norm	activation
2×conv	3*3	1*1	64	True	Leaky ReLu
2×conv	3*3	1*1	128	True	Leaky ReLu
3×conv	3*3	1*1	256	True	Leaky ReLu
3×conv	3*3	1*1	512	True	Leaky ReLu
Output:[(batch × 256×128×64), (batch × 128×64×128) (batch × 64×32×256), (batch × 32×16×512)]					

pop, metal, blues, classic, jazz, and country. 100 pieces of music were randomly selected from each style and were collected from an online source[1]. To compare the results with therapeutic music, we also collected ten pieces of therapeutic music adopted in [88] (Appendix I). In the joint optimization method, the therapeutic music is selected from the list and the user's preferred music is provided by the subjects.

5.4.1.2 Implementation Details

Pre-processing Method The pre-processing method we follow is similar to the one described in [34]. We use two python packages, pretty_midi [115] and pypianoroll [20], to convert all MIDI files to piano-rolls. We choose a sampling rate of 16 time steps per bar, as it is commonly used in literature [33, 54], which results in the smallest note being a 16-th note. We select 16 consecutive bars for each phrase and retain all possible notes between 0 and 127, resulting in a pitch range of 128. If the number of tracks in a piece of music is less than or equal to 4, we merge all the tracks into a single one. If there are more than 4 tracks, we choose the one with the most note information. We also remove velocity information, which makes it easier to learn, and the drum track as it often has poor quality when played by another instrument. The final structure of each musical phrase is $256 * 128 * 1$.

Architecture Parameters In this study, the feature extraction network consists of 10 convolutional layers, as outlined in Table 5.1. To balance the information extracted and computational efficiency, feature maps from layers 2, 4, 7, and 10 are used to represent different levels of in-

[1]https://www.reddit.com/r/WeAreTheMusicMakers/comments/3ajwe4/the_largest_midi_collection_on_the_internet/

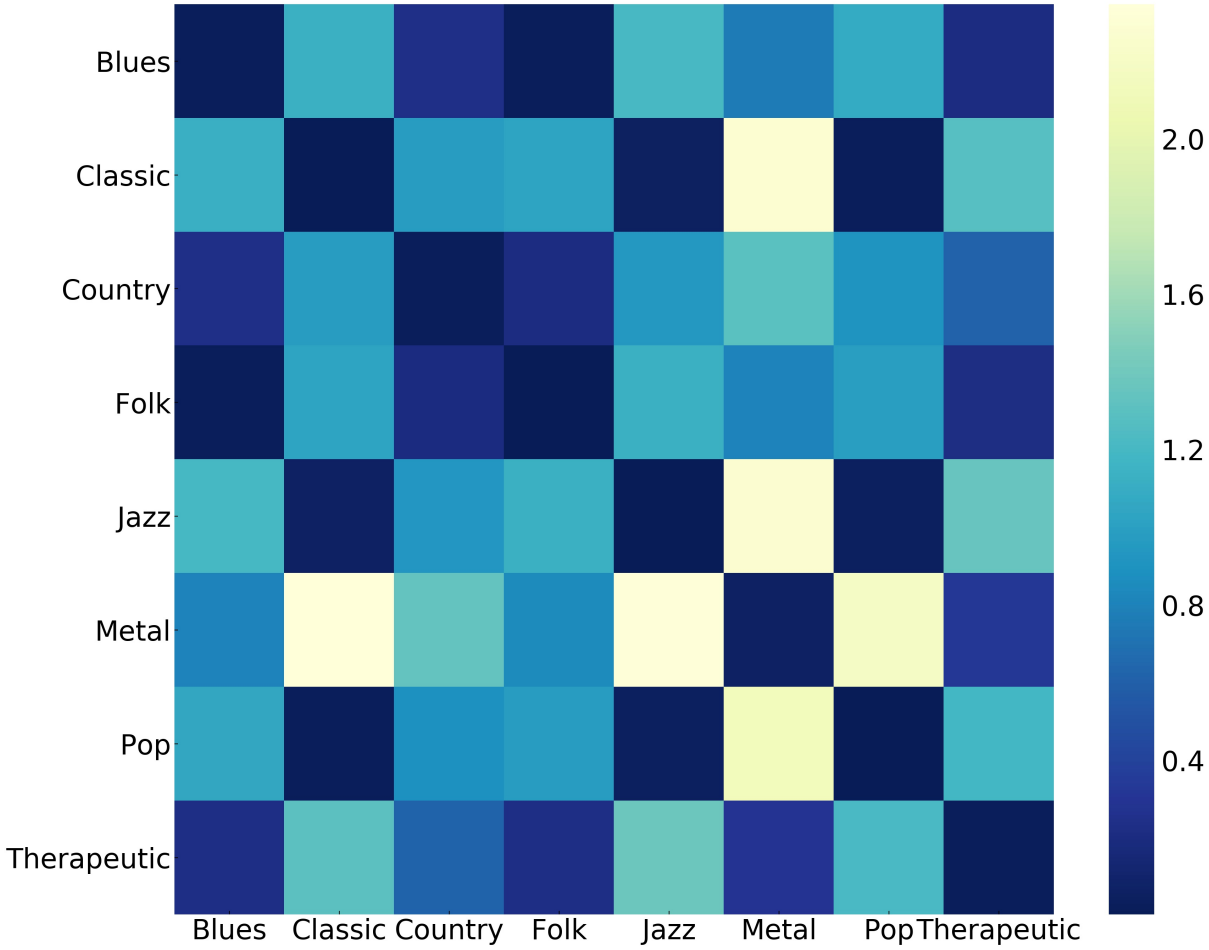


Figure 5.3: Heatmap of secondary feature difference.

formation. Techniques such as batch normalization [116] and leaky ReLU [117] are applied to stabilize the training process. The network is optimized using the Adam optimizer [118] with an initial learning rate of 0.001. The parameters α and β are set to 0.5 and 0.5 respectively, while the safety distance d_s is set to 0.01. The feature extraction network is trained for a maximum of 100 epochs. In the therapeutic transfer network, the values for λ_1, λ_2, λ_3, and e are set to 1, 100, 100, and 5 respectively. The therapeutic transfer network is also trained for 100 epochs.

5.4.2 Secondary Feature Difference

In this section, the differences between the secondary features of seven musical styles and therapeutic music are calculated. The formula for calculating the difference between two styles, A and B, is defined as $D_{AB} = \sum_{l=1}^{4} \sum_{i=1}^{N} ||\mathbf{S}^l Ai - \bar{\mathbf{S}}^l B||F^2$, where $\bar{\mathbf{S}}^l B$ represents the average value of the secondary feature B at layer l. The results of the analysis are presented in the form

of a heat map in Figure 5.3. The heat map indicates that the differences between secondary features within the same style are the smallest, while the differences between different styles are higher. This supports the first statement that the feature extraction network is capable of learning hidden representations even when the music style is new to the model. Additionally, the heat map confirms the second statement that the secondary features contain stylistic information. Some music styles, such as folk and blues, are not distinguishable, while metal music can be easily categorized, which is in line with previous findings in the literature ([119]). In the case of therapeutic music, the difference is not pronounced, as different musical styles can be therapeutic, especially those with gentle and relaxing melodies ([88]).

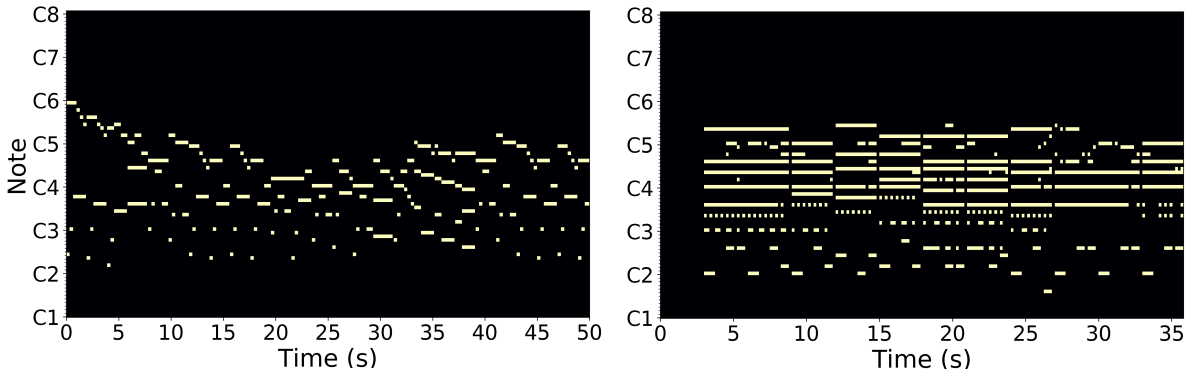
5.4.3 Therapeutic Transfer Model

5.4.3.1 One transferred example

The findings from Figure 5.4 which displays piano-rolls of one piece of randomly selected music, therapeutic music, and transferred music are noteworthy. Firstly, the pitch values of the transferred music range from C2 to C6, which aligns with the pitch range of the randomly selected sample, resulting in no sudden pitch changes in the transferred music. Secondly, the transferred music retains most musical information from the user's favorite music, including the positioning of notes, repeated patterns, chords, and more, leading to a similar musical structure as the user's favorite music. Lastly, the note length in the transferred music is different from both the user's favorite music and therapeutic music. While user's favorite music has mostly long notes, therapeutic music has more frequent short notes. The transferred music, however, has a higher number of short notes and fewer long notes, indicating a shift in the musical style.

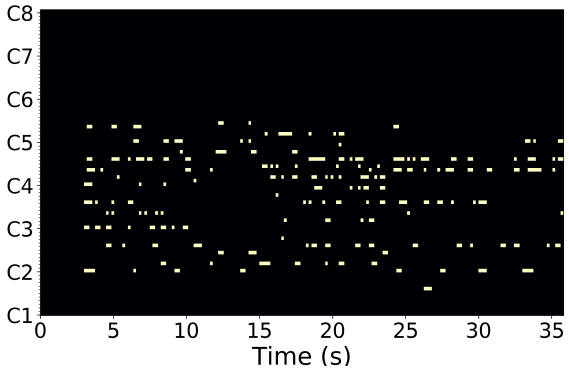
5.4.3.2 Model Comparison

Figure 5.5 compares the training time of four different models, namely UNIT [120], MUNIT [121], CycleGAN [122], and the proposed model, for music style transfer. The results indicate that the proposed model can transfer the user's favorite music to therapeutic music in a reasonable amount of time compared to the other models, which require a longer time to train. This is due to the fact that the proposed model is designed to train only the inputs, resulting in a



(a) Therapeutic music: "Moon River"

(b) Random selected music: "Absence of the Heart"



(c) Transferred music

Figure 5.4: A transferred example.

smaller model size, whereas the other models are trained on larger music datasets and require more parameters to be trained. Additionally, the proposed model doesn't require a large dataset for training, as it only needs two music samples.

5.4.4 Music Therapy Experiment

In order to test the practical applications of the proposed method, we conducted a four-day study on receptive music therapy (MT). Receptive MT is a type of therapy that uses music listening to improve mood, reduce stress, and alleviate anxiety, and it is considered passive, easily implementable, and cost-effective compared to other types of music therapy, such as active MT. It can be used with a variety of different patient populations and across a wide range of physical and psychological conditions and has been widely used for many years.

In the study, we did not compare our model to other existing music style transfer models.

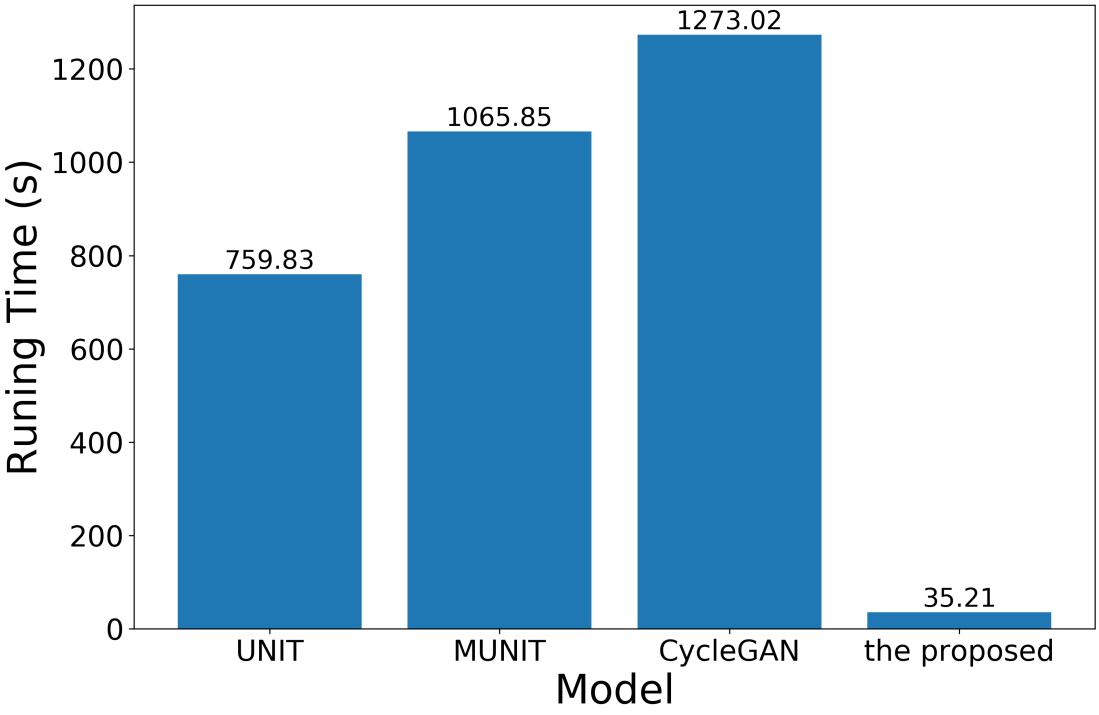


Figure 5.5: Model comparison in terms of transfer time.

This is because these models were not designed to work with only two training samples and as a result, their performance in this scenario would be much worse than in their original applications, potentially resulting in unpleasant and anxiety-inducing music. Instead, we compared the performances of the new transferred songs with the original favorite and therapeutic songs to validate the effectiveness of the proposed method.

5.4.4.1 Paradigm design

Subjects Twenty subjects (8 male, 12 female) with mild anxiety experience for the last month were recruited online. None of the subjects reported neurological or hearing dysfunctions.

Music Selection For the study, three types of music were used: favorite music, therapeutic music, and transferred music. Participants filled out a questionnaire to indicate their favorite songs and a total of eighteen favorite songs were collected, two of which were repeated by different participants and were referred to as "repeated songs". The therapeutic music was recommended by Grocke, Wigram, and Kildea [88] and had been shown to have positive therapeutic

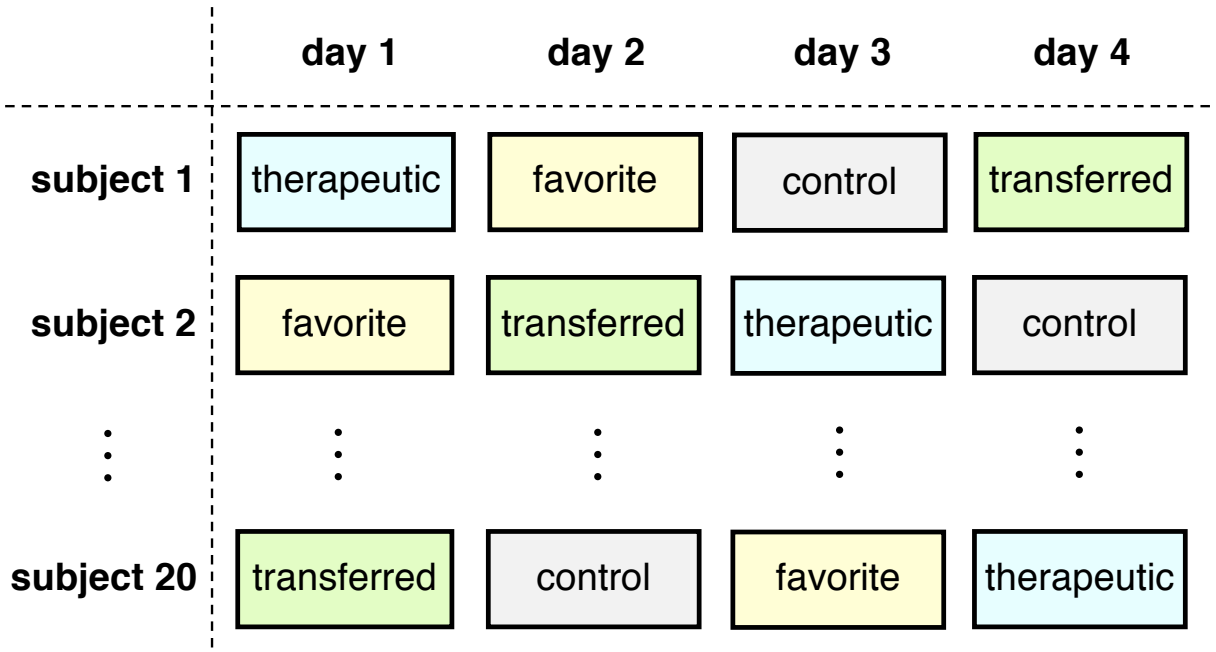


Figure 5.6: The procedure of the experiment.

effects in clinical practice at an anonymous hospital (Appendix I). The favorite songs were then transferred to randomly selected therapeutic songs, with different therapeutic songs being used for different participants with repeated songs.

Measurement of Anxiety To assess the participants' levels of anxiety, the State-Trait Anxiety Inventory (STAI, Form Y version) was used. The questionnaire consisted of two parts: the State Anxiety Inventory (STAI-S) and the Trait Anxiety Inventory (STAI-T), each with 20 items to evaluate anxiety levels [123]. Questions included statements such as "I am tense; I am worried" and "I feel calm; I feel secure". Participants rated each item on a scale from 1 to 4, with some items requiring higher scores to indicate greater anxiety and others requiring higher scores to indicate lower anxiety. The scores for Form Y of the STAI range from a minimum of 20 to a maximum of 80 for both STAI-S and STAI-T and are typically classified as "no or low anxiety" (20-37), "moderate anxiety" (38-44), or "high anxiety" (45-80) [124]. For the study, only the STAI-S was used, which determines the participants' current feelings.

Experiment Procedure The experiment lasted four days and each day involved one trial (Figure 5.6). The four trials were as follows: 1) the "control trial" in which nothing was played; 2) the

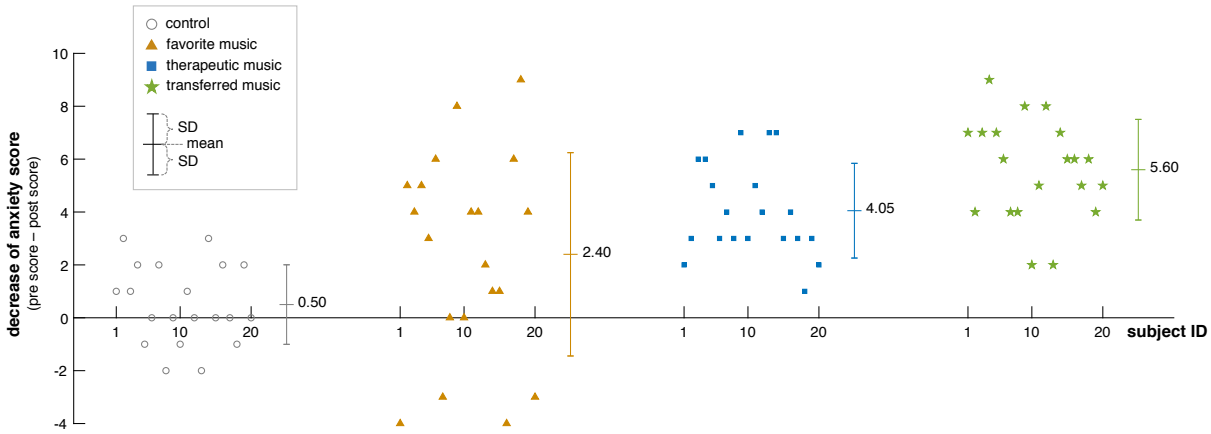


Figure 5.7: Subjective experiment results.

Table 5.2: Paired t-test results between pre- and post-STAI scores among four groups.

Condition	Control	Favorite	Therapeutic	Transferred
***p*-value**	0.154	1.16×10^{-2}	4.39×10^{-9}	5.36×10^{-11}

"favorite trial", in which the participant's favorite song was played; 3) the "therapeutic trial", in which a randomly selected therapeutic song was played; and 4) the "transferred trial", in which the song transferred from the participant's favorite song was played. Participants participated in the four trials in random order over the course of four days.

Each participant was seated in a quiet room with a comfortable chair and the lighting was adjusted to a comfortable level. After receiving a brief explanation of the experiment, they completed the STAI-S questionnaire and then listened to the music through earphones for 15 minutes while a recorded breathing exercise was played alongside the music (Appendix II). The breathing exercise was based on a clinical example used at the Royal Melbourne Hospital [88]. After listening, the participants were asked to give their opinion on the music and to fill out the STAI-S questionnaire once again to obtain post-listening results.

5.4.4.2 Results

To evaluate the efficacy of music therapy, pre- and post-listening anxiety scores are compared using a paired t-test. The results, presented in Table 5.2, reveal a significant reduction in anxiety from the pre- to post-listening period for the three types of music: favorite songs ($p < 0.05$), therapeutic songs ($p < 0.05$), and transferred songs ($p < 0.05$). There was no significant difference

in the control group. This outcome supports the anxiety-relieving effect of the three types of music, and the effectiveness of favorite and therapeutic music is in line with previous research studies [125, 126].

The performance of different approaches is illustrated in Figure 5.7, which compares the decrease in anxiety scores from pre- to post-listening periods, where larger decreases represent better therapeutic outcomes. The results are divided into four groups and each group's results are presented as scatter plots showing the decrease in anxiety scores for each individual subject and the mean and standard deviation of the 20 subjects using error bars.

The first group of results, marked by grey circles, represents the control trial where no music was played and the subjects rested for 5 minutes. There was no significant decrease in anxiety scores in the control trial, except a slight shift (0.50 on average), which may be attributed to the anxiety-reducing effect of silent rest.

The second group, marked by yellow triangles, represents the anxiety decrease of the subjects when listening to their favorite music. The average anxiety level was reduced by 2.40 points, and some subjects showed great improvements. However, a proportion of subjects reported increased anxiety. The diverse genres and emotions of the favorite songs may result in strong emotional responses, which may sometimes not be positive in the short term. For instance, one subject reported a rock and sad song, "Norwegian Forest", which may have worsened their anxiety. This observation highlights the potential risk of relying solely on users' music preferences.

The third group, marked by blue squares, represents the results of therapeutic music, which resulted in a large decrease in anxiety scores of 4.05 points, and a smaller standard deviation of 1.79 points compared to the favorite music group. This suggests the effectiveness of professionally selected therapeutic music and its robustness to different subjects. Although the three pieces of therapeutic music produced similar anxiety-reducing effects, they were diverse in many aspects, such as genre, composer, and era, and had different keys. This makes it difficult to determine a clear and universal criterion for therapeutic music selection, which highlights the importance of expert experience and the involvement of professional therapists.

The transferred music group showed the best results, marked by green stars in Figure 5.7,

with the largest mean decrease in anxiety scores of 5.60 points and a relatively small standard deviation of 1.90 points. Even the worst cases among the 20 subjects showed a decrease of 2 points in anxiety scores. The effectiveness of transferred music can be explained as follows. To begin, it is essential to understand the significance of favorite songs to the participants. Research has shown that musical preference is linked to one's musical engagement [127], and favorite songs have the potential to deepen musical engagement, leading to an impact on the subject's current emotional state. However, simply having a strong connection to the favorite songs may not be enough. Given the diversity of the participants' favorite songs, it is unclear how their emotional states are affected, as evidenced by the high standard deviation in experimental results (as indicated by the yellow error bar in Figure 5.7. To guide the emotion toward a relaxed state, it is necessary to transfer the participants to therapeutic songs. The therapeutic songs share the common feature of reducing anxiety, and it is reasonable to assume that transferring to these songs would imbue the favorite songs with similar properties while preserving most of their original content. As a result, the mechanism behind the effectiveness of transferred music can be described as follows: the participants' strong connection to the content from their favorite songs enhances their engagement, while the anxiety-reducing features from the therapeutic songs promote a relaxed emotional state.

5.5 Summary

To our knowledge, this is the first study on using music style transfer for regulating emotions and reducing anxiety. Our innovative domain adaptation algorithm significantly reduces the cost of customization. The use of style transferred music for anxiety therapy has demonstrated improved efficacy due to increased user engagement. Our study also introduces a new music therapy transfer model that converts the user's favorite music into a therapeutic piece. The results of our analysis and subjective experiments validate the effectiveness of the proposed model.

5.6 Appendices

5.6.1 Appendix I: Therapeutic Music List

We shows a list of therapeutic music that is mentioned in [88] and is adopted in this paper.

- Moon River (Henry Mancini)
- Watermark (Enya Patricia Brennan)
- Carmen Suite No.1, Intermezzo (George Bizet)
- Morning Mood (Edvard Grieg)
- Eine kleine Nachtmusik (Wolfgang Amadeus Mozart)
- The Swan (Saint-Saens)
- Pie Jesu (Gabriel Fauré)
- Piano Concerto No.21, K.467 - 2nd movement (Wolfgang Amadeus Mozart)
- Clarinet Concerto in A major, K.622 - 2nd movement (Wolfgang Amadeus Mozart)
- Gabriel's Message (Basque Folk Carol)

5.6.2 Appendix II: Auditory Recording of a Breathing Exercise

As shown in Fig. 5.8, an auditory recording of a breathing exercise was played simultaneously.

Please sit in the chair, keep your head and body in a straight line. Put your arms next to your body and hand palms up. Stop all movement and close your eyes. Focus on your body. Feel your body sitting in the chair. Feel your back, your arms, your feet resting, feel your whole body still and just rest.
Silence: 3 seconds
Follow my voice, focus on your breath, focus on the speed of your breath, focus on the depth of your breath, then gradually deepen your breath. Feel all organs are resting.
Silence: 3 seconds
Now, take long, slow and deep breaths. Feel your breath, feel the natural rhythm of your breath and start immersing yourself in the music, slowly let go, sink back into the chair and sink into the music.

Figure 5.8: Breathing Exercise.

Chapter 6

A Hybrid Favorite-aware Method for User Preference Music Transfer

6.1 Background and Motivation

In recent years, there has been an increase in research regarding music style transfer, a method that transforms one piece of music into another based on various levels of musical representations [100]. The importance of music style transfer lies in its ability to enhance music diversity through creative reproduction of existing pieces, as well as making music composition easier and more accessible to a wider audience [128].

However, the area of music style transfer based on user preferences, also known as user preference music transfer (UPMT), has been relatively understudied [13]. UPMT involves converting symbolic input music into a new piece that aligns with a user's preferences, determined by the features of their favorite music (Fig.6.1). UPMT has potential applications in areas such as anxiety reduction [129, 130], personality studies [131, 132], and music recommendation systems [133, 134]. For instance, in the field of music therapy, it has been found that individuals are more engaged when listening to their favorite music, thus leading to enhanced therapeutic effects [135]. However, the lack of personalization in therapeutic music restricts its impact. By customizing therapeutic music to suit a user's preferences, this technique can result in more effective anxiety treatments and reduce copyright issues.

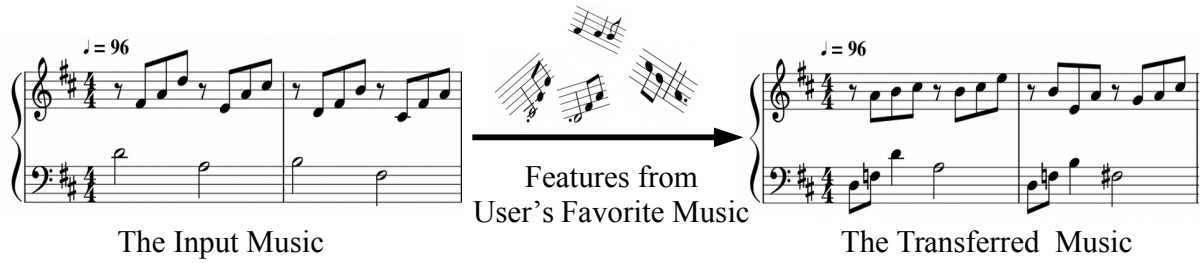


Figure 6.1: A demonstration of user preference music transfer.

Music transfer typically employs data-driven techniques such as VAE-based [33, 38] and GAN-based models [34, 103], but these methods require a large amount of data for training. However, users typically only provide a limited number of their favorite songs, making it challenging to accurately represent their musical preferences. This issue has been addressed in other fields, such as image classification and recommender systems, by introducing prior knowledge into data-driven models to improve performance and explainability.

To overcome the challenge posed by the scarcity of training data, researchers have introduced hybrid methods in areas like image classification and recommender systems that incorporate knowledge into data-driven techniques to enhance model performance and explanation [136–139]. For instance, Marino *et al.* [138] utilized prior semantic knowledge represented in knowledge graphs to improve image classification outcomes. Similarly, Donadello *et al.* [139] utilized semantic representations from a knowledge base to boost the efficiency of recommender systems. However, these methods cannot be directly applied to the domain of music due to differences in the structure of musical knowledge. Liu *et al.* [140] proposed incorporating musical knowledge, such as 18th-century counterpoint rules, to generate more training data for music, but they focused on generating more data rather than improving the model itself. Furthermore, a knowledge base is often necessary to facilitate training, but constructing such a base can be challenging in the context of UPMT as a limited number of music samples are available.

To tackle the issue of limited training data in the field of music, we introduce a hybrid approach called User Preference Transformer (UP-Transformer). The method uses only one piece of a user's favorite music as prior knowledge and employs the pre-trained Transformer-XL model as its backbone [141]. The process is divided into two phases: fine-tuning and transfer. During the fine-tuning phase, the model is adjusted using a favorite-aware loss function. In the

transfer phase, an event selection step and an event learning step are performed to produce music that aligns with the user's preferences. We evaluate the effectiveness of the UP-Transformer by introducing a new metric called pattern similarity (PS) to measure the music's style. The results of a qualitative experiment demonstrate that the output music surpasses the original in terms of musicality, similarity, and preference, suggesting that the UP-Transformer can successfully transfer music to fit a user's preferences based on just one piece of favorite music.

The following highlights our contributions:

1. We introduce a new hybrid method called UP-Transformer, which combines music knowledge based on a user's preferred music and the Transformer-XL model. UP-Transformer offers a unique approach to music transfer by modifying certain music elements in the input symbolic music, which suits different transfer tasks.

2. We devise a novel loss function that takes into account user preferences and couple it with two new transfer steps to enhance the performance of the Transformer-XL model. This method overcomes the challenge of training data-driven methods on small datasets. This method also significantly improves output quality by better suiting user preferences.

3. We present a new metric to quantify the similarity between two musical sequences in terms of melody. We conduct a statistical experiment on random music samples, showcasing the validity and utility of the new metric, which can be applied to various music-related tasks to assess melody.

6.2 Related Work

6.2.1 Transformer-based Methods

The Transformer model [56] has been successful in various generation tasks where long-range coherence is essential. Its unique hierarchical beat-bar structure and repetitive nature make it a suitable choice for music modeling. Several researchers have applied Transformer-based methods to music-related problems in recent years [23, 25, 108, 142]. One of the first applications of the Transformer to music was by Huang *et al.* [23], who proposed the Music Transformer

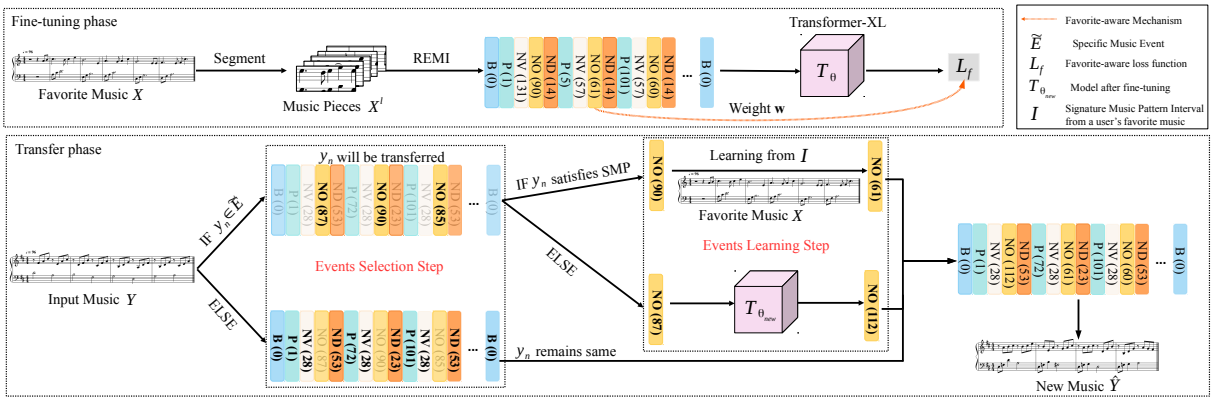


Figure 6.2: A general framework of UP-Transformer. There are two phases: a fine-tuning phase and a transfer phase. SMP: signature music pattern. In this demonstration, only *Note On* events are transferred.

model for music generation. However, their representation method using MIDI-like data missed important musical features, such as tempo changes and explicit note duration, which affected the rhythm of the output music.

Subsequent Transformer-based methods [25, 108, 142] showed improved performance. Pop music Transformer (PopMT) [24] proposed a new representation method called REMI and used the Transformer-XL model [141] to generate pop music. PopMAG [29] presented a new representation method called MuMIDI and addressed the challenge of multi-track harmony modeling. However, to the best of our knowledge, these methods are not designed for music transfer. In contrast, the proposed UP-Transformer method uses Transformer-based methods to perform music style transfer by altering and fixing certain musical events at each time step.

6.3 Proposed Method

6.3.1 Problem Formulation

In the proposed UP-Transformer framework, a user's favorite music X is used to extract prior knowledge (e.g. music event distributions and unique patterns) denoted as $K(X)$. The goal of this framework is to take an arbitrary piece of music Y and transfer it using method M with parameters ϕ to create a new piece of music $\hat{Y}$ that contains knowledge K from the user's favorite music X while preserving important information from the original music Y. This process can

be formulated as follows:

$$\hat{Y} = M(Y, K(X); \phi) \tag{6.1}$$

6.3.2 Model Overview

This model leverages musical knowledge, particularly the event distribution and SMPI found in a user's preferred music, with a pre-trained Transformer-XL forming the core. The structure of this model encompasses two stages: fine-tuning and transfer, as shown in Figure 6.2.

Each stage includes the following operations:

1. The user's favorite music X is divided into segments.

2. A recent music representation method called REMI is applied to convert symbolic music data into a music event sequence, which will be covered in Section III.C.

3. The pre-trained Transformer-XL model serves as the backbone and is fine-tuned using a novel favorite-aware loss function that takes into account the distribution of different music events in the user's favorite music, to be discussed in Section III.D.

4. During the transfer phase, two steps, event selection and event learning, will be utilized with SMPI from the user's favorite music and the fine-tuned Transformer-XL model to transfer the input music into a new piece that meets the user's preferences; this process will be described in Section III.E.

The UP-Transformer model introduces two innovations: a favorite-aware loss function that optimizes the Transformer-XL model based on the event distribution in the user's favored music, and two musical procedures at inference time that guide the transformation process using the extracted SMPI.

6.3.3 Music Representation

The REMI representation, as referenced in [24], is used for symbolic music in this section. This representation is characterized by eight distinct musical events, which are determined according to note, tempo, and chord data. Each event consists of one or more categories to reflect its

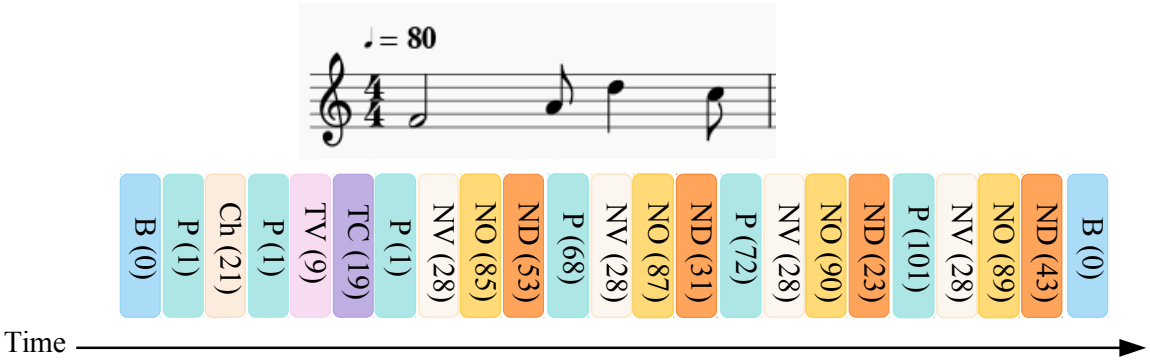


Figure 6.3: REMI representation.

variable values. For instance, the event labeled *Bar (B)* signifies the bar line on piano rolls, while a single note is divided into four distinct events: *Position (P)*, *Note Velocity (NV)*, *Note On (NO)*, and *Note Duration (ND)*. The event *P* denotes one of the possible discrete positions in a bar, where the time resolution used to represent a bar is denoted as q. *NV* is quantized to 32 levels to indicate the perceived volume of the note. The *Note On* event includes 128 classes that mark the onset of MIDI pitches, ranging from 0 to 127, and *ND* is used to specify the duration of a single note, with values varying from 1 to 64. The *Tempo* event, added at every beat, indicates local tempo variations and is denoted by *Tempo Class (TC)* and *Tempo Value (TV)* with tempo values from 30 to 209 beats per minute (BPM). *Chord (Ch)* events are also included in REMI, comprising of 12 chord roots and five chord qualities. Every *Tempo Class*, *Tempo Velocity* and *Chord* event is preceded by a *Position* event. The procedure of transforming musical data into REMI is depicted in Figure 6.3.

In comparison with the MIDI-like event matrix, REMI offers a more comprehensive representation by incorporating explicit definitions for different sequential events like tempo variations and chord data, leading to a denser representation. Furthermore, by arranging each note into four consecutive events, it allows more versatility in sequence manipulation.

6.3.4 UP-Transformer in Fine-tuning Phase

Previous studies [143] have demonstrated the effectiveness of pre-training Transformer-based models on large-scale music datasets. In line with this, we opted to use a Transformer-XL model [141] pre-trained on a pop music dataset, rather than the CNN-based style transfer tech-

Algorithm 3 UP-Transformer in fine-tuning phase.

Input: User's favorite music X, a pre-trained Transformer-XL network T_θ
Output: Optimized Transformer-XL network $T_{\theta_{new}}$

1: **while** not converge **do**
2: Calculate weight w_c with Equation 6.3;
3: Train T_θ using Equation 6.2.
4: **end while**
5: **return** $T_{\theta_{new}}$

niques used in the image domain [144, 145]. The Transformer-XL model has the ability to handle long sequences with its recurrence mechanism and revised positional encoding scheme. Additionally, to tailor the model to a user's preferred music and to encourage the learning of specific music events in cases of limited training data, we propose a novel favorite-aware loss function [24] that directly learns the music event distribution from the user's favorite music.

Each token in REMI [24] represents a class that belongs to one of the eight music events described in the previous section. In the original Transformer-XL [141], the cross-entropy loss function is used to optimize the model, treating all classes equally. However, considering the varying significance of each music event, it is appropriate to pay greater attention to events that have a bigger impact on a user's favorite music, such as note-related events. The favorite-aware loss function, calculated based on the event distribution in a user's favorite music, places more emphasis on such events. The fine-tuning procedure is detailed in Algorithm 3, and experiments have been conducted to validate this model design.

Formally, let $E = \{e_B, e_P, e_{Ch}, e_{TC}, e_{TV}, e_{NV}, e_{NO}, e_{ND}\}$ be the set of eight music events. We use e_k and c to represent an arbitrary music event and its corresponding class, respectively. Here, $e_k \in E$ is also a set that contains one or more classes and $c \in e_k$.

Given one music piece X^l with sequence length N, $X^l = \{x_1^l, x_2^l, \ldots, x_N^l\}$ where $x_n^l = c \in e_k$, then Transformer-XL [141] $T_\theta(\cdot)$ can predict the next token $\hat{x}_i^l$ conditioned on the previously generated tokens $x_{t<i}^l$, formulated as $\hat{x}_i^l = T_\theta(x_{t<i}^l)$. The favorite-aware loss L_f is defined as

$$L_f = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c * p_{i,c} log(q_{i,c}), \tag{6.2}$$

where N is the sequence length, C is the total number of classes, $p_{i,c}$ corresponds to the c^{th} ele-

ment of the one-hot encoded label of the ground truth x_i^l, and $q_{i,c}$ is the predicted category probability that observation i is of class c. w_c is the c^{th} favorite-aware weight $\mathbf{w} = \{w_1, w_2, \ldots, w_C\}$ and can be further calculated from a user's favorite music X with the following formula:

$$w_c = \begin{cases} \alpha + \dfrac{Count(c)}{\sum_{c \in \tilde{E}} Count(c)}, & if\ c \in \tilde{E}, \\ \alpha, & o.w. \end{cases} \tag{6.3}$$

where $\tilde{E} \in E$ denotes selected music events, $Count(\cdot)$ is a function that counts the appearance of an input element in the sequence X, and $\alpha > 0$ is a modifiable parameter. In this way, information that appears more frequently in the user's favorite music X will have a higher weight. For events that are not selected for transfer, the same weight will be given with a fixed value α.

6.3.5 UP-Transformer in Transfer Phase

In the UPMT's transfer phase, an initial musical sample Y of length N_y is modified into a new composition $\hat{Y}$ based on the user's favorite music X, of length N_x. To augment the model's interpretability and efficiency, two steps related to music, namely event selection and event learning, are incorporated, as detailed in Algorithm 4. Event selection only alters designated musical events $\tilde{E} \in E$ in Y, leaving other events intact. If y_n falls within $\tilde{E}$, it will be transformed and predicted as $\hat{y}_n$ through the event learning stage. If y_n is not part of $\tilde{E}$, then $\hat{y}_n = y_n$. This selective event modification assures that the newly created music aligns with the structure of the input music, retaining elements from the original Y.

For instance, if we specify $\tilde{E} = e_{NO}$, only the *NO* event in Y will be altered. All other events remain unmodified. Event selection maintains musical events that we wish to preserve and concentrates solely on a particular music event. This ensures the new composition maintains the structure of the original music and thereby conserves details from Y.

The event learning step involves directing the predicted event to align with the user's favorite music by using a distinctive musical pattern termed the Signature Music Pattern (SMP). SMPs are defined as groups of sequential notes that form a cohesive unit and act as an identifying feature that makes music distinguishable. The SMP is extracted from the chosen music events $\tilde{E}^X$ in the user's preferred music X via the SMP Extractor (refer to Algorithm 5).

Algorithm 4 UP-Transformer in transfer phase.

Input: The user's favorite music $X = \{x_1, x_2, \ldots, x_{N_x}\}$, an arbitrary piece of music $Y = \{y_1, y_2, \ldots, y_{N_y}\}$
Parameter: θ_{new}
Output: The updated sequence $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{N_y}\}$

Find all $\tilde{E}$ in X and denote as $\tilde{E}^X = \{\tilde{e}_1^X, \tilde{e}_2^X, \ldots, \tilde{e}_{z_x}^X\}$
Extract an SMP $M = \{m_1, m_2, \ldots, m_a\}$ from $\tilde{E}^X$ and denote it as music pattern interval $I = \{i_1, i_2, \ldots, i_{a-1}\}$
Let $v = 0$, flag$= 0$, $n = 1$ and $idx = 1$
while $n < N_y$ **do**
  if $y_n \in \tilde{E}$ **then**
    if flag$= 0$ **then**
      $\hat{y}_n = T_{\theta_{new}}(\hat{y}_1, \ldots, \hat{y}_{n-1})$
      if $\hat{y}_n - \hat{y}_v = i_1$ **then**
        flag=1
      end if
    else
      $\hat{y}_n = \hat{y}_v + i_{idx}$
      $idx = idx + 1$
      if $idx > a - 1$ **then**
        $idx = 1$
        flag$= 0$
      end if
    end if
    $v = n$
  else
    $\hat{y}_n = y_n$
  end if
  $n = n + 1$
end while
return $\hat{Y}$

Additionally, we compute the difference, referred to as SMPI $I = \{i_1, i_2, \ldots, i_{a-1}\}$, which is calculated as $I = \{m_2 - m_1, m_3 - m_2, \ldots, m_a - m_{a-1}\}$. For example, if the selected event $\tilde{E} = e_{NO}$ and an SMP M is $\{60, 62, 62, 64, 62, 62, 60, 68\}$, then SMPI I is represented as $\{2, 0, 2, -2, 0, -2, 8\}$. We use SMPI I instead of SMP M because music tends to have different pitch levels, but in terms of music similarity, relative differences are more representative; humans barely hear the difference if the ratio of two notes' frequencies (i.e., SMPI) is the same [146]. For instance, an SMP $M = \{72, 74, 74, 76, 74, 74, 72, 80\}$ sounds familiar to users compared with SMP $M = \{60, 62, 62, 64, 62, 62, 60, 68\}$ as I remains consistent.

The sequence for the event learning step is elaborated in Algorithm 4 and proceeds as fol-

Algorithm 5 SMP extractor

Input: $\tilde{E}^X$
Parameter: a
Output: SMP M

1: Generate a dictionary $dict$ for all sub-lists of length a in $\tilde{E}^X$.
2: **if** Repeated patterns exist **then**
3: Find all repeated patterns in $dict$ and assign them to an empty list R.
4: M = Randomly select one repeated pattern from R.
5: **else**
6: No repeated patterns, please shorten the length a.
7: **end if**
8: **return** M

lows:

1. A $flag$ is established to determine whether the event learning step will be undertaken.

2. If $flag = 0$, the event learning step will be bypassed. In this situation, the current token $\hat{y}_n$ will be predicted by the pre-trained Transformer-XL model, $T_{\theta_{new}}(\cdot)$, i.e., $\hat{y}_n = T_{\theta_{new}}(\hat{y}1, \ldots, \hat{y}_{n-1})$ (lines 6–7).

   If the difference between the current predicted token and the preceding predicted token, $\hat{y}_n - \hat{y}_v = i_1$, corresponds to the first item in the SMPI I, then $flag$ is set to 1, signifying that the event learning step will be executed in the subsequent steps (lines 8–10).

3. If $flag = 1$, the event learning step will be enacted. The subsequent tokens belonging to the selected event $\tilde{E}$ will not be predicted by the Transformer-XL model $T_{\theta_{new}}(\cdot)$; rather, the event will be updated directly based on the SMPI I until all the details from I have been infused into the new music $\hat{Y}$ (lines 11–13).

   After all the items in I have been updated to the subsequent tokens, the flag will be reset to 0 to signify that the update is complete (lines 14–17).

4. In the end, the previous predicted token is updated (line 19).

The event learning step employs the SMPI extracted from the user's favorite music and is adjusted to conform to the user's tastes, thereby enhancing the model's effectiveness.

Table 6.1: Statistics of training data.

Statistics	Mean	Std
# of Segments	49.42	41.95
# of *Note On (NO)*	1610.21	1234.69
# of *Note Velocity (NV)*	1610.21	1234.69
# of *Note Duration (ND)*	1610.21	1234.69
# of *Position (P)*	1812.73	1384.09
# of *Bar (B)*	81.16	32.63
# of *Tempo Class (TC)*	93.10	158.35
# of *Tempo Value (TV)*	1.00	0.00
# of *Chord (Ch)*	109.42	47.41

6.4 Experiments

6.4.1 Datasets

The subjects were asked to provide their favorite music, which was then collected from open sources. A total of 23 subjects participated, and 19 unique pieces of music were used for the fine-tuning phase, as some subjects had the same favorite music. These 19 pieces of music were all cut into the same length for fine-tuning. A summary of the training data is provided in Table 6.1.

For the transfer phase, 7 songs from 7 different genres (classical, jazz, pop, rock, folk, new age, and light music) were utilized as input music.

6.4.2 Implementation Details

In the fine-tuning phase, we divided one piece of a user's favorite music into segments of 128 tokens due to the limited training data. We followed the original setting of Transformer-XL from [24] and set $\alpha = 0.01$ for the favorite-aware loss function. We trained the model for 200 epochs or until the total loss was less than 0.1. In the transfer phase, we selected the track with the most note information to complete the transfer task and left the other tracks unchanged to keep the music sounding similar to the original input music. All tracks were then merged to generate new music.

6.4.3 Comparisons

To assess the effectiveness of our proposed method, we made a comparative study with four models drawn from related research works: **PopMT** [24], **PopMAG** [29], **CycleGAN** [34], and **TTM** [13]. Both PopMT and PopMAG use the Transformer-XL as their core architecture, and they have been further fine-tuned for transfer tasks, modifying specific musical events while leaving the rest constant. This was named PopMT_T and PopMAG_T respectively. CycleGAN, after its fine-tuning phase, is employed to directly adjust the input music according to the user's tastes. TTM uses a pre-trained network for feature extraction and optimizes the output music by considering the user's favorite music as the "style" and the input music as the "content".

6.4.4 Evaluation Metrics

In the field of music, there is still a lack of consensus on how to evaluate the quality of music transfer results. Inspired by [29], we adopt four evaluation metrics to compare our proposed method with other models and assess the quality of the transferred music: Pitch Class (*P*), Note (*N*), Duration (*D*), Inter-Onset Interval (*IOI*).

The similarity between two pieces of music is measured by calculating the average overlapped area (OA) of four factors. The formula for this calculation is as follows:

$$D_A = \frac{1}{N_{tracks} * N_{bars}} \sum_{i=1}^{N_{tracks}} \sum_{j=1}^{N_{bars}} OA(P_{i,j}^A, \hat{P}_{i,j}^A),$$

where OA represents the average *OA* of two distributions, $P_{i,j}^A$ denotes the distribution of feature A in the i-th bar and the j-th track in a ground truth musical piece, and $\hat{P}_{i,j}^A$ denotes that in the generated musical piece; A can be any of four factors. Note that velocity and chord were not measured, as these were evaluated in previous work and our focus was on transfer tasks where velocity and chord remained unchanged.

Although melody is a fundamental and distinctive feature used to assess music similarity, existing metrics do not evaluate melodic information. To address this gap, we introduce a new metric, PS, to assess the consecutive music pattern interval overlap between two sequences. This is described in Algorithm 6. PS measures the melodic similarity between two pieces of music if

Algorithm 6 Pattern similarity.

Input: Music event of a user's favorite music X: $\tilde{E}^X = \{\tilde{e}_1^X, \tilde{e}_2^X, \ldots, \tilde{e}_{z_X}^X\}$; music event of transferred music $\hat{Y}$: $\tilde{E}^{\hat{Y}} = \{\tilde{e}_1^{\hat{Y}}, \tilde{e}_2^{\hat{Y}}, \ldots, \tilde{e}_{z_{\hat{Y}}}^{\hat{Y}}\}$; music pattern length p
Output: Pattern similarity PS

$match = 0$
$z = 1$
calculate interval of X and $\hat{Y}$ and denote as $I^X = \{i_1^X, i_2^X, \ldots, i_{z_X-1}^X\}$ and $I^{\hat{Y}} = \{i_1^{\hat{Y}}, i_2^{\hat{Y}}, \ldots, i_{z_{\hat{Y}}-1}^{\hat{Y}}\}$
while $z < z_{\hat{Y}} - 1 - p$ **do**
  if $[i_z^{\hat{Y}}, i_{z+1}^{\hat{Y}} \ldots, i_{z+p}^{\hat{Y}}] \in I^X$ **then**
    $match = match + 1$
  end if
  $z = z + 1$
end while
$PS = \dfrac{match}{z_{\hat{Y}} - p}$
return PS

the target event is "Note On". A higher value of PS indicates that the melody of the new music is more similar to the target music, making it more familiar to users. However, if the pattern length is longer, the similarity will be lower due to the difficulty in finding a match between the two sequences.

6.4.5 Experimental Results and Analysis

6.4.5.1 Quantitative comparison

The results of the quantitative comparison can be found in Tables 6.2 and 6.3. Table 6.2 shows that UP-Transformer performs significantly better in terms of the average overlapped area with the user's favorite music, as measured by D_P, D_N, and D_D. Additionally, its performance in D_{IOI} is highly competitive, indicating that the music transferred by UP-Transformer is most similar to the user's favorite music compared to other models. Models such as PopMT_T and PopMAG_T, which are based on Transformers, show a lower level of similarity to the target music compared to the input music because they don't incorporate prior music knowledge. CycleGAN's performance is unsatisfactory due to its limited training samples. On the other hand, TTM, which uses the matrix representation method and jointly learns music features from two

Table 6.2: Objective experiment results. The ground truth: the user's favorite music.

Music	D_P	D_N	D_D	D_{IOI}
Input	0.60±.04	0.24±.05	0.34±.11	0.25±.08
PopMT_T [24]	0.58±.10	0.21±.06	0.33±.20	**0.27±.15**
PopMAG_T [29]	0.60±.12	0.23±.05	0.33±.20	0.26±.15
CycleGAN [34]	0.27±.15	0.13±.07	0.08±.06	0.04±.03
TTM [13]	0.57±.10	0.29±.12	0.27±.17	0.14±.10
UP-Transformer	**0.86±.10**	**0.70±.13**	**0.34±.11**	0.26±.08

types of input, does not specifically learn event information, including *Note On*.

Table 6.3 shows that UP-Transformer leads to the highest average overlapped area with the input music in terms of D_N, D_D, and D_{IOI} without sacrificing substantial information from the input music's *Note On* events. PopMT_T and PopMAG_T also show high scores in D_D and D_{IOI} because they only transfer *Note On* events. However, CycleGAN's performance is poor because it is difficult to train a large number of model parameters based on just two music samples.

6.4.5.2 Qualitative comparison

A qualitative evaluation was carried out to assess the transferred music by listening, which is considered the most effective way to evaluate the transferred music. A group of 23 subjects who were trained in playing musical instruments and had basic music theory knowledge were selected for the listening test. Before the experiment, each subject was asked to provide their favorite music. The experiment consisted of listening to two music samples - the original input music and their favorite music - followed by five pieces of randomly chosen music transferred by PopMT, PopMAG, CycleGAN, TTM, and UP-Transformer. The subjects were then asked to evaluate the transferred music based on the following three metrics using a 5-point scale (1 being the lowest and 5 being the highest):

- Musicality: How musically appealing do you find the music?
- Similarity: How similar is the music to your favorite music?
- Preference: How much do you like the music?

Figure 6.4 shows the results of the qualitative experiment. The summary of the results are as follows:

Table 6.3: Objective experiment results. The ground truth: the input music.

Music	D_P	D_N	D_D	D_{IOI}
PopMT_T [24]	**0.59±.15**	0.20±.07	0.85±.07	0.71±.06
PopMAG_T [29]	0.53±.10	0.21±.04	0.86±.10	0.71±.08
CycleGAN [34]	0.26±.15	0.12±.08	0.15±.14	0.05±.05
TTM [13]	0.55±.05	0.21±.05	0.25±.03	0.15±.03
UP-Transformer	0.57±.05	**0.22±.04**	**0.91±.04**	**0.78±.04**

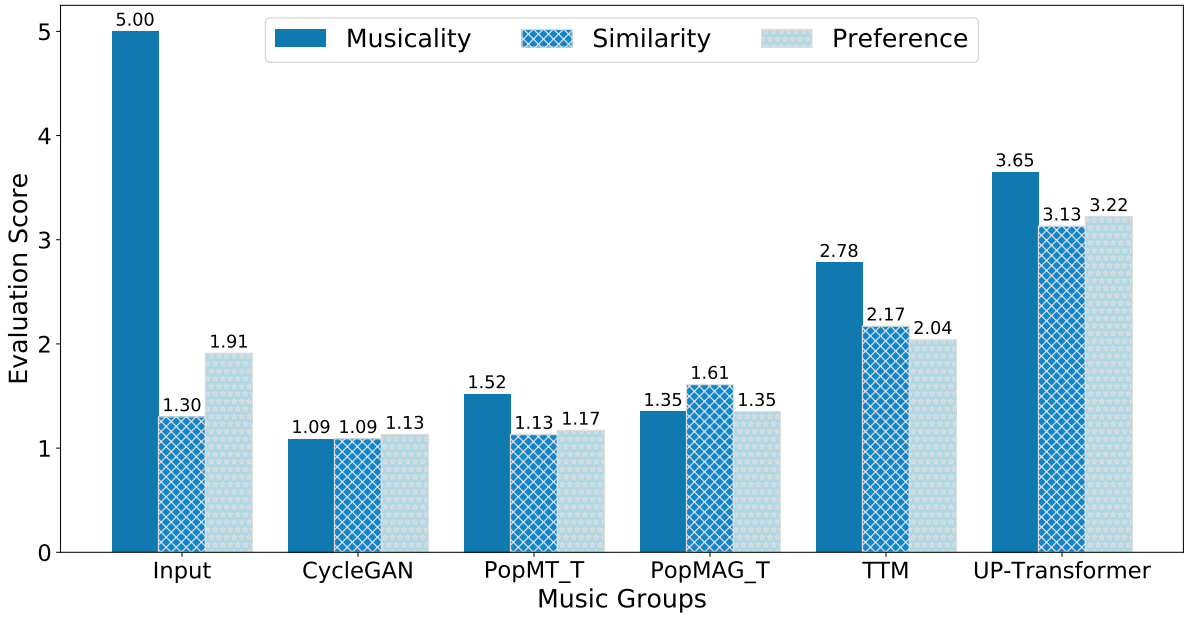


Figure 6.4: Subjective experiments result.

• In terms of musicality, UP-Transformer stands out by transferring music with the highest quality compared to other models, despite the noticeable difference between human-composed and transferred pieces.

• When it comes to similarity, UP-Transformer outperforms other models. The similarity score increases from 1.30 to 3.13, showing that music transferred by UP-Transformer is more similar to the user's favorite music than the input music. This suggests that UP-Transformer has learned some music patterns from the user's favorite music.

• The preference score also increases from 1.91 to 3.22, indicating that UP-Transformer can transfer input music to align with the user's preferences. UP-Transformer continues to display the best performance compared to other models, which should lead to increased user satisfaction with the transferred music.

6.4.6 Ablation Study

6.4.6.1 Transferring different events

The first experiment focuses on the transfer of different music events. From the mean values in Table 6.1, it's observed that the events *Note On*, *Note Duration*, and *Position* occur most frequently and substantially influence the resulting music. Thus, we assessed the transfer performance of these three events, excluding *Note Velocity* as it only affects the music's loudness, not its structure or melody.

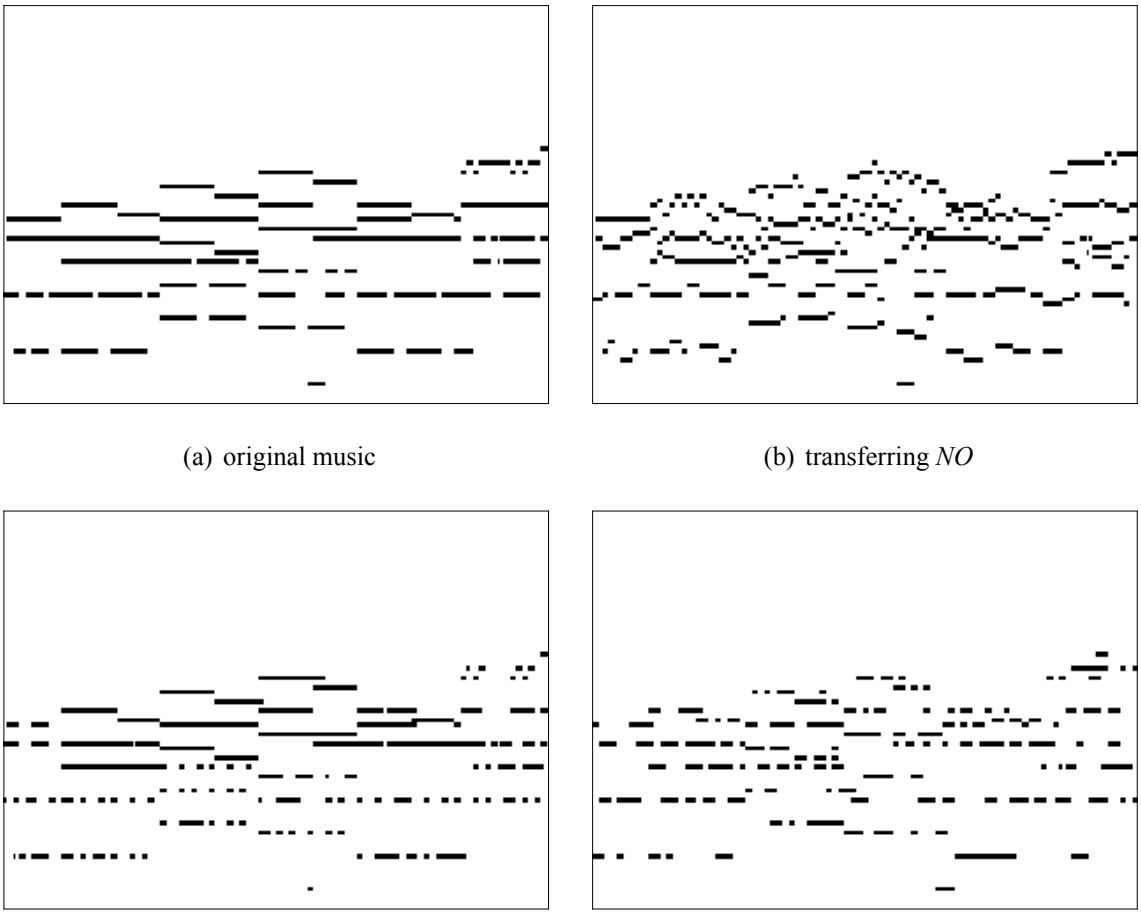
The transfer effects of "Note On", "Note Duration", and "Position" were evaluated and compared, as illustrated in Fig. 6.5. The initial sample is the original music, with the next four samples representing music post-transfer of *Note On*, *Note Duration*, and *Position*. The music changes discernibly after these transfers, with *Note On* transfer having the most profound effect on the melody. As visible in Fig. 6.5 (b), the music patterns vary considerably, leading to enhanced diversity.

The transfer results were evaluated using four metrics, D_P, D_N, D_D, and D_{IOI}, by comparing one randomly selected piece of input music for each user with their favorite music. The performance is shown in Table 6.4 for $\tilde{E} = e_P$, $\tilde{E} = e_{ND}$, and $\tilde{E} = e_{NO}$. The following observations were made when comparing all the music to the user's favorite music:

1. The differences in the four metrics were minimal when transferring *Position*. This makes sense as *Position* has no direct relationship with pitch or duration values. However, changes in *Position* can indirectly impact note duration and inter-onset interval by altering the position of notes and causing fluctuations in the results.

2. The transfer of *Note Duration* led to increased values of D_D and D_{IOI}. However, D_P and D_N remained unchanged because they measure note-on information.

3. The transfer of *Note On* events caused higher values of D_P and D_N. It's important to note that even though only *Note On* events were transferred, the pitch value can still impact the duration of consecutive notes, leading to changes in D_D and D_{IOI}. For instance, if two consecutive notes with different pitch values are transferred to have the same pitch value, the note duration and inter-onset interval will change.



(a) original music

(b) transferring *NO*

(c) transferring *ND*

(d) transferring *P*

Figure 6.5: Demonstration of one example when transferring different music events.

Table 6.4: Objective results when transferring different music events. The ground truth: the user's favorite music.

Music	D_P	D_N	D_D	D_{IOI}
Input	0.62±.12	0.24±.07	0.14±.07	0.21±.09
Transfer *P*	0.62±.12	0.24±.07	0.18±.12	0.18±.09
Transfer *ND*	0.62±.12	0.24±.07	**0.60±.16**	**0.24±.07**
Transfer *NO*	**0.75±.10**	**0.39±.04**	0.14±.07	0.22±.08

Table 6.5: Performance of backbone models. The ground truth: the user's favorite music.

Music	D_P	D_N	D_D	D_{IOI}
Input	0.62±.12	0.24±.07	0.14±.07	0.21±.09
CNN	0.29±.16	0.12±.06	0.07±.02	0.04±.02
RNN	0.59±.09	0.24±.06	0.14±.07	0.22±.08
LSTM	0.62±.10	0.25±.06	0.14±.07	0.22±.08
TF-XL	**0.75±.10**	**0.39±.04**	**0.14±.07**	**0.22±.08**

Overall, transferring *Note On* and *Note Duration* can lead to improved output music. To measure melodic similarity and further listen to the music samples, in the following experiment, we set $\tilde{E} = e_{NO}$; that is, *Note On* events were transferred.

6.4.6.2 Effects of the backbone model

To assess the effectiveness of our chosen backbone model, Transformer-XL (TF-XL), and its attention mechanism, we compared the transferred music produced by TF-XL, CNN, RNN, and LSTM. Additionally, we compared different attention mechanisms discussed in [23] based on the four metrics listed in Tables 6.5 and 6.6. Table 6.5 shows that TF-XL outperforms the other backbone models in terms of average overlapped area with the user's favorite music, as indicated by higher values of D_P and D_N. Conversely, CNN, which requires matrix representation as input, did not perform well due to the sparsity of the matrix representation and lack of focus on

Table 6.6: Performance of attention mechanisms. The ground truth: the user's favorite music.

Music	D_P	D_N	D_D	D_{IOI}
TF-XL w/ local attention	0.69±.15	0.36±.06	0.14±.07	0.22±.08
TF-XL w/ relative local attention	0.71±.06	0.36±.05	0.14±.07	0.22±.08
TF-XL w/ relative global attention	0.74±.09	0.38±.05	0.14±.07	0.22±.08
TF-XL	**0.75±.10**	**0.39±.04**	**0.14±.07**	**0.22±.08**

Table 6.7: Performance with and without implementing two steps. The ground truth: the user's favorite music.

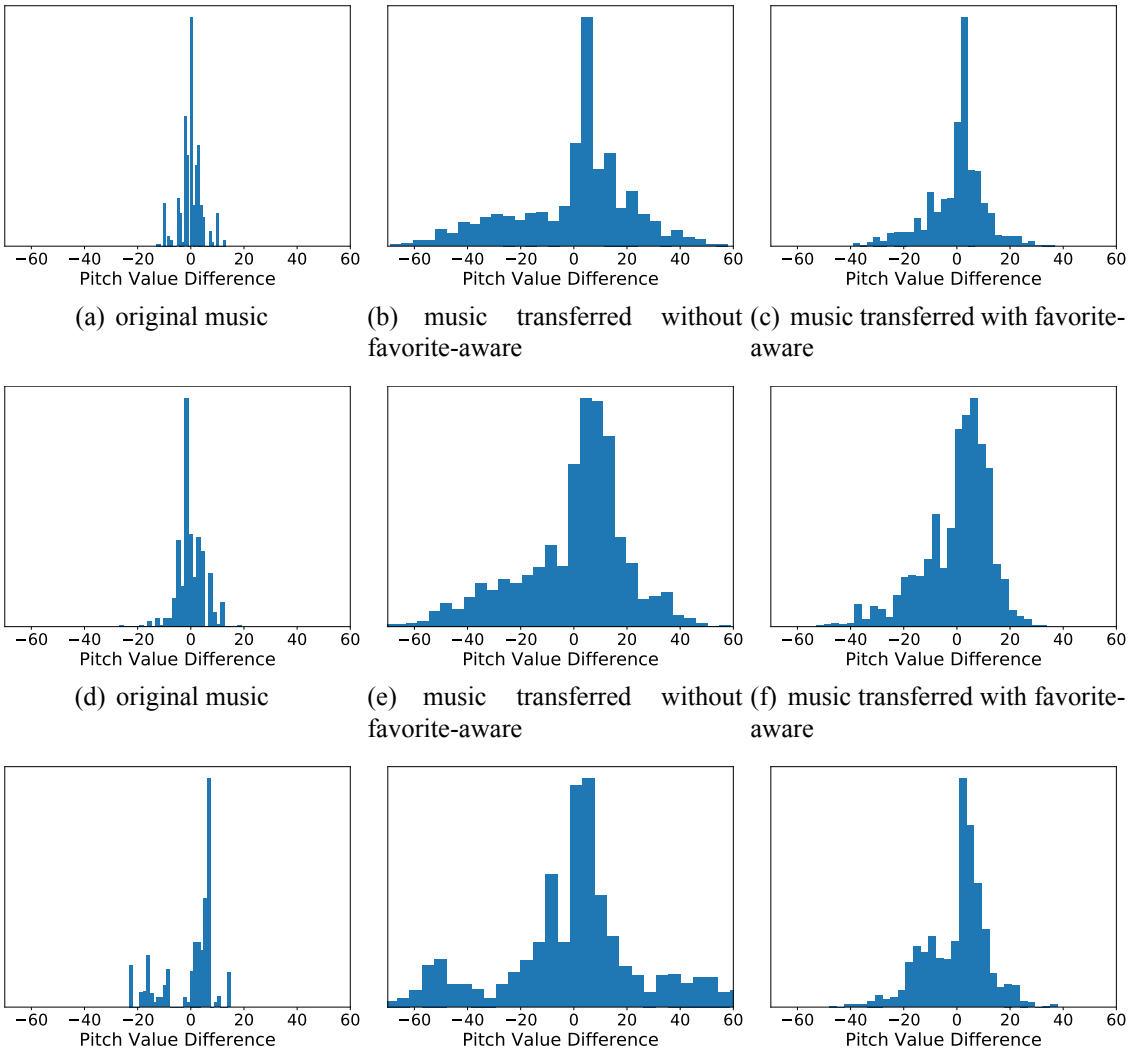
Music	D_P	D_N	D_D	D_{IOI}
Input	0.60±.04	0.24±.05	0.34±.11	0.25±.08
TF-XL w/o favorite-aware	0.62±.06	0.33±.06	0.34±.11	0.26±.08
TF-XL w/ favorite-aware	0.66±.04	0.36±.06	0.34±.11	0.25±.08
UP-Transformer	**0.86±.10**	**0.70±.13**	**0.34±.11**	**0.26±.07**

learning event information. Table 6.6 shows that the various attention mechanisms produced similar results, but the TF-XL backbone yielded slightly better results. Note that D_D and D_{IOI} remained constant across all cases, as only *Note On* events were transferred.

6.4.6.3 Effects of favorite-aware loss function

To determine the efficacy of the favorite-aware loss function, we compared the distribution of melodic intervals in the original music, music transferred without using the loss function, and music transferred using the loss function. Fig. 6.6 displays these distributions. The melodic interval is calculated as the difference in pitch values between consecutive notes and is an important factor in determining the melody of a piece of music. Our hypothesis was that after fine-tuning with the favorite-aware loss function, the transferred music would have a similar distribution of melodic intervals to the original music. As shown in Fig. 6.6, the distribution of melodic intervals in the last column is more similar to that of the original music (first column) compared to the middle column. This suggests that the favorite-aware loss function helps the model better capture melodic interval information from the user's favorite music.

The effectiveness of the favorite-aware loss function is also shown in Table 6.7, which compares four sets of music to the user's favorite music. The sets include input music, music generated without implementing the favorite-aware loss function and two steps (using only cross-entropy loss function [TF-XL without favorite-aware]), music generated without two steps (TF-XL with favorite-aware), and music generated with the favorite-aware loss function and two steps (UP-Transformer). Results are presented in terms of means and standard deviations. As indicated by the first three rows, both TF-XL without favorite-aware and TF-XL with favorite-aware are capable of producing music that is more similar to the user's favorite music, as the

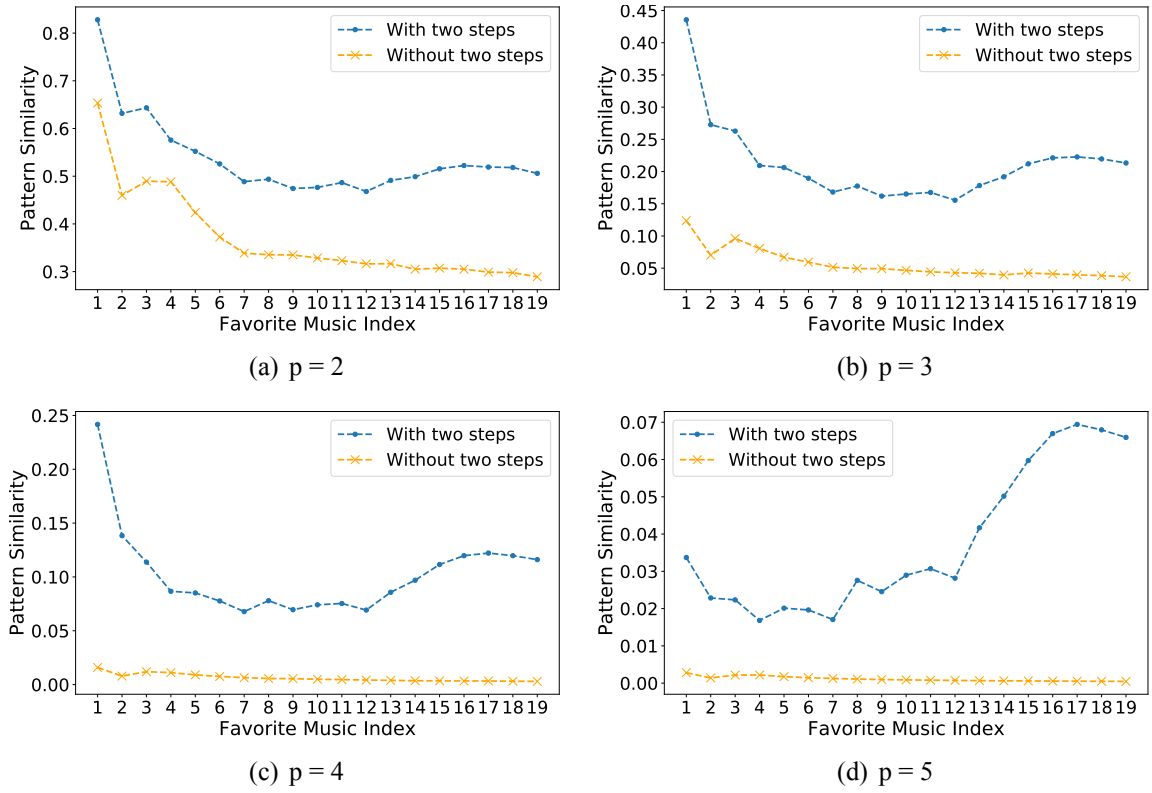


(a) original music (b) music transferred without favorite-aware (c) music transferred with favorite-aware

(d) original music (e) music transferred without favorite-aware (f) music transferred with favorite-aware

(g) original music (h) music transferred without favorite-aware (i) music transferred with favorite-aware

Figure 6.6: Three examples of distribution of melodic interval.



(a) p = 2 (b) p = 3

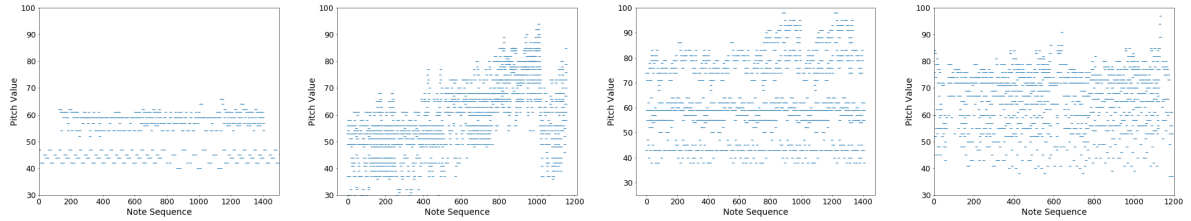
(c) p = 4 (d) p = 5

Figure 6.7: Pattern similarity (PS) results on 19 subjects in terms of different music pattern lengths.

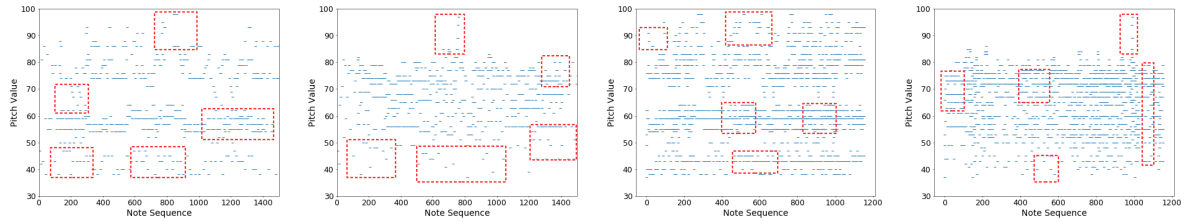
average overlap between the generated music and the user's favorite music increases in terms of D_N and D_P. This suggests that the favorite-aware loss function in TF-XL with favorite-aware allows the model to more accurately learn the musical events from the user's favorite music, resulting in generated music that is more closely aligned to the user's preferences.

6.4.6.4 Effects of two steps

To evaluate the effect of the proposed two steps in the transfer phase on the performance of the output, we measured the pattern similarity before and after these steps were applied. Figure 6.7 shows the pattern similarity between the transferred music and 19 pieces of the user's favorite music, before and after the two steps were implemented. The pattern lengths (p) ranging from 2 to 5 were taken into consideration. The results showed that smaller pattern lengths had higher pattern similarity, which is expected because shorter patterns are more likely to match the user's favorite music. Additionally, the results showed that the implementation of the two steps improved the pattern similarity in all musical samples, implying that these steps enhance the similarity between the transferred music and the user's favorite music.



(a) input music A (b) input music B (c) user's favorite music A (d) user's favorite music B



(e) transferred music A: input music A + user's favorite music A (f) transferred music B: input music A + user's favorite music B (g) transferred music C: input music B + user's favorite music A (h) transferred music D: input music B + user's favorite music B

Figure 6.8: Pattern Similarity (PS) result demonstration when music pattern length $p = 4$. Red: the pattern matches the melody in the user's favorite music.

Additionally, the results in Table 6.7 showed that the similarity in pitch value between the transferred music and the user's favorite music, as measured by D_P and D_N, significantly increased in the UP-Transformer model compared to the other models and input. However, the values of D_D and D_{IOI} did not change much, as the experiment only transferred *Note On* events. This result met the goal of incorporating knowledge from the user's favorite music (*Note On*) while preserving other information from the input music (e.g. *Note Duration*, *Position*, etc.).

Fig. 6.8 illustrates two randomly selected input examples, two examples of the user's favorite music, and four transferred samples with music pattern similarity for $p = 4$. Note duration and instrument information were omitted for simplicity. The transferred music is seen to have a similar structure to the input music, with the note position, note numbers, and total length of the note sequence remaining the same. The transferred music also contains elements of the user's favorite music's melody, as seen by the presence of red rectangles enclosing matching patterns throughout the music piece.

6.4.7 Correlation test

To prove that the PS metric accurately measures melodic similarity and is in line with human perception, we utilized the Kendall correlation to quantify the correlation between PS and the

Table 6.8: Kendall correlation test.

Music Pattern Length	Kendall correlation	P-value
$p = 2$	0.64	0.0005
$p = 3$	0.80	0.0000
$p = 4$	0.72	0.0001
$p = 5$	0.22	0.2199

subjective similarity score. The Kendall correlation is suitable for small datasets and does not make assumptions about the relationships, noise, or distributions of the data [147]. If the music has a high PS value, then it will receive a high qualitative similarity score, as PS represents melodic similarity, resulting in a high Kendall correlation. Table 6.8 shows the Kendall correlation and p-value for various music pattern lengths p. The results demonstrate a positive relationship between human judgment of music similarity and PS for $p = 2, 3, 4$, implying that people can recognize the similarities even with short pattern lengths. Although the p-value is not significant for $p = 5$, it still shows a positive relationship (Kendall correlation $= 0.22$) between PS and the qualitative similarity score.

6.5 Summary

In this study, we introduce a new task in music style transfer, known as UPMT, and propose a hybrid model, UP-Transformer, to tackle it. The model blends Transformer-XL with music knowledge based on a single piece of the user's favorite music. The implementation of the novel favorite-aware loss function during training and two inference steps during transfer enable the model to produce personalized output music for different users. Additionally, we developed a new metric to evaluate the pattern similarity (PS) between two pieces of music. The results of both quantitative and qualitative experiments demonstrate that UP-Transformer effectively transfers input music to fit users' preferences with high quality.

Chapter 7

Conclusion and Future Work

In this thesis, we study the computational creativity in music and present a compilation of our research efforts. Our focus is on two areas that play a significant role in musical creation: music generation and music style transfer. This chapter summarizes the proposed works in the thesis and highlights potential avenues for future research.

7.1 Conclusion

This thesis begins by introducing the background and motivation for computational creativity in music. We first study the composition skill of music by proposing motif-level repetition and phrase-level call-response, and then we apply music style transfer techniques to application scenarios including music therapy and music personalization. By enhancing the composition skills of machine-composed music, we aim to apply music generation techniques to more application scenarios to improve user experiences.

First, we propose a novel motif-level repetition generator and outline-to-music generator for music generation. We are the first to examine repetition in machine composition in a comprehensive manner. The study first proposes a new dataset named MRD, which was constructed based on the Pop piano dataset and includes 562,563 training data and 21,766 test data with labels for motif-level repetitions. The study then designs a novel MGM that can generate structured music based on a few motifs and their repetitive variants. It has two newly designed generators, MRG and O2MG. The MRG can generate a large amount of motif-level repetition of different

types and the O2MG can generate a piece of structured music based on the outline. Its learned composition skills on the Pop music dataset also show promising results on the "fate motif" from classical music. Evaluations by both musicians and non-professional users suggest that the proposed techniques significantly improve the quality of machine composed music.

Second, we present a novel call-and-response generator for music generation. This work stands as the first application of the call-and-response technique—an essential composition method—in learning-based algorithms. We develop a new pop music dataset of 19,155 call-response pairs from 909 songs and define three objective evaluation metrics. Our Call-Response Generator (CRG), enhanced with a knowledge-based mechanism, outperforms existing learning-based models in terms of rhythm, melody, and harmony.

Third, we propose a novel domain adaptation method to transfer a user's favorite music to be therapeutic for therapeutic music transfer. It is well-known that a person's favorite music can significantly increase their engagement in the music therapy, which has been widely adopted for anxiety reduction. However, the limited number of a user's favorite songs can pose a challenge in customizing the therapeutic music. We propose a music style transfer method to transfer the user's favorite music to be therapeutic based on one piece of therapeutic music and one piece of a user's favorite music. A new domain adaptation algorithm was designed to transfer the results from music genre classification to music personalization. The experiments on anxiety sufferers showed that the customized therapeutic music was more effective and stable in reducing anxiety.

Fourth, we propose a novel music transfer model named User Preference Transformer (UP-Transformer) for transferring diverse music to fit user preference. User preference music transfer can be applied to many applications such as music therapy where the music needs to satisfy requirements from both source domain and target domain. However, this problem remains understudied. We propose a hybrid model that combines Transformer-XL with musical knowledge based on a single piece of music chosen by the user. The unique favorite-aware loss function and two-step transfer process enables the model to personalize the input music to fit user preference. It is challenging to quantitatively measure music quality in music style transfer. Therefore, we also introduce a new metric to measure the Personalized Similarity (PS) between two pieces of music. The results of both quantitative and qualitative experiments confirm the effectiveness of

the UP-Transformer in transferring the input music to fit the user's preferences with high quality.

7.2 Future Work

Although significant progress has been made in the areas of music generation and style transfer, there is still a long way to go. Many opportunities for further exploration exist in the field. These potential areas of exploration can be broadly categorized into three aspects: music generation, therapeutic music transfer, and personalized music transfer. The following section outlines specific future works in each of these areas.

7.2.1 Music Generation

1. We plan to generate music based on higher level structures, such as period-level structure of the music. It will further improve the coherence and integrity of the music.

2. We will explore more evaluation methods to measure music in terms of structure besides subjective experiments.

3. We will further explore the diversity of repetition from the dimension of rhythm and harmony.

7.2.2 Therapeutic Music Transfer

1. We will explore more factors to investigate the relationship between therapeutic and stylistic features and to further disentangle therapeutic features.

2. We will include a psychological experiment. We intend to measure the emotional fluctuations of individuals using EEG signals. By monitoring and analyzing EEG signals during the listening test, we can study therapeutic music features in detail.

3. We plan to include lyrics in subsequent work. Lyrics are closely related to music appreciation and play important roles in music therapy. Techniques such as lyric generation [10] could be employed to make therapeutic transfer more effective and interesting.

7.2.3 Personalized Music Transfer

1. We plan to study instances where a person provides several pieces of preferred music with distinct styles.

2. We plan to investigate more underlying factors beside melody patterns in sequence and consider different music features such as lyrics and performance.

7.2.4 Pre-training for Computational Creativity in Music

Clear improvements in the accuracy, efficiency, and robustness of downstream tasks have validated the feasibility of pre-training in domains including computer vision and natural language processing (NLP). However, pre-training has not been adequately addressed in machine composition. Available pre-training techniques are not directly applicable to computer music due to the special properties of machine composition, including the characters of music data, the nature of composition tasks, and the artistic work dataset. Therefore, I plan to explore new pre-training models with the underlying algorithms to improve the machine composition from three dimensions of musicality that offers enjoyment, appeal that strengthens the engagement, and the novelty that brings refreshing sentiment.

Like the pre-training model in NLP, the pre-training model for music is based on big datasets. Although current music dataset can generate reasonable music, they are either too small or without music theory annotations. I plan to introduce more data modalities to enrich the process of creativity. Specifically, I plan to propose three new datasets. First, a composition theory corpus dataset will be constructed by labeling lower-level and higher-level music structures based on music theory. I have collected over 500,000 motif-level common music repetitions. The final dataset is expected to contain over 1,000,000 multi-level music repetition corpus that follow music theory. Second, an emotion multi-label dataset will be constructed that contains music segments with the emotion spectrum. Third, a style characteristics dataset will be developed. In the first stage, we generate a style-related music corpus by following the history of musical styles. In the second step, I plan to apply the rules of music theory to construct specific patterns that adhere to music theory but have not been used extensively in earlier compositions.

References

[1] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 38–45, 2020.

[2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[3] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[4] C. Payne, "Musenet," *OpenAI Blog*, vol. 3, 2019.

[5] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, pp. 955–967, 2020.

[6] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv preprint arXiv:2011.06801*, 2020.

[7] T. Collins and R. Laney, "Computer-generated stylistic compositions with long-term repetitive and phrasal structure," *Journal of Creative Music Systems*, vol. 1, no. 2, 2017.

[8] G. Medeot, S. Cherla, K. Kosta, M. McVicar, S. Abdallah, M. Selvi, E. Newton-Rex, and K. Webster, "Structurenet: Inducing structure in generated melodies.," in *Proceedings*

of the 19th International Society for Music Information Retrieval Conference (ISMIR), pp. 725–731, 2018.

[9] H. Jhamtani and T. Berg-Kirkpatrick, "Modeling self-repetition in music generation using generative adversarial networks," in *Machine Learning for Music Discovery Workshop, ICML*, 2019.

[10] X. Wu, Z. Du, Y. Guo, and H. Fujita, "Hierarchical attention based long short-term memory for chinese lyric generation," *Applied Intelligence*, vol. 49, no. 1, pp. 44–52, 2019.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, vol. 16, no. 12, pp. 326–326, 2015.

[12] Z. Hu, X. Ma, Y. Liu, G. Chen, and Y. Liu, "The beauty of repetition in machine composition scenarios," in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1223–1231, 2022.

[13] Z. Hu, Y. Liu, G. Chen, S.-h. Zhong, and A. Zhang, "Make your favorite music curative: Music style transfer for anxiety reduction," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1189–1197, 2020.

[14] Z. Hu, Y. Liu, G. Chen, and Y. Liu, "Can machines generate personalized music a hybrid favorite-aware method for user preference music transfer," *IEEE Transactions on Multimedia*, pp. 1–13, 2022.

[15] L. A. Hiller Jr and L. M. Isaacson, "Musical composition with a high speed digital computer," in *Audio Engineering Society Convention 9*, Audio Engineering Society, 1957.

[16] G. A. Wiggins, "Automated generation of musical harmony: what's missing," in *Proceedings of the International Joint Conference in Artificial Intelligence*, 1999.

[17] R. B. Dannenberg, "Languages for computer music," *Frontiers in Digital Humanities*, vol. 5, p. 26, 2018.

[18] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and

transcription," in *Proceedings of the 29th International Conference on Machine Learning, ICML*, 2012.

[19] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation–a survey," *arXiv preprint arXiv:1709.01620*, 2017.

[20] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 34–41, 2018.

[21] A. Roberts, J. Engel, C. Raffel, I. Simon, and C. Hawthorne, "Musicvae: Creating a palette for musical scores with machine learning," *Magenta*, 2018.

[22] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[23] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *7th International Conference on Learning Representations, ICLR*, 2019.

[24] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1180–1188, 2020.

[25] J. Jiang, G. G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer vae: A hierarchical model for structure-aware and interpretable music representation learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 516–520, IEEE, 2020.

[26] N. Zhang, "Learning adversarial transformer for symbolic music generation," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[27] A. Muhamed, L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton,

and A. J. Smola, “Symbolic music generation with transformer-gans,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 408–417, 2021.

[28] J. Ens and P. Pasquier, “Mmm: Exploring conditional multi-track music generation with the transformer,” *arXiv preprint arXiv:2008.06048*, 2020.

[29] Y. Ren, J. He, X. Tan, T. Qin, Z. Zhao, and T.-Y. Liu, “Popmag: Pop music accompaniment generation,” in *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1198–1206, 2020.

[30] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang, “Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 178–186, 2021.

[31] X. Zhang, J. Zhang, Y. Qiu, L. Wang, and J. Zhou, “Structure-enhanced pop music generation via harmony-aware learning,” in *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1204–1213, 2022.

[32] Y.-J. Shih, S.-L. Wu, F. Zalkow, M. Muller, and Y.-H. Yang, “Theme transformer: Symbolic music generation with theme-conditioned transformer,” *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[33] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, “MIDI-VAE: modeling dynamics and instrumentation of music with applications to style transfer,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 747–754, 2018.

[34] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao, “Symbolic music genre transfer with cyclegan,” in *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 786–793, IEEE, 2018.

[35] W. T. Lu, L. Su, *et al.*, “Transferring the style of homophonic music using recurrent neural networks and autoregressive model.,” in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 740–746, 2018.

[36] J. Wang, C. Jin, W. Zhao, S. Liu, and X. Lv, "An unsupervised methodology for musical style translation," in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, pp. 216–220, IEEE, 2019.

[37] O. Cífka, U. Şimşekli, and G. Richard, "Groove2groove: One-shot music style transfer with supervision from synthetic data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2638–2650, 2020.

[38] R. Yang, T. Chen, Y. Zhang, and G. Xia, "Inspecting and interacting with meaningful music representations using VAE," in *19th International Conference on New Interfaces for Musical Expression, NIME*, pp. 307–312, nime.org, 2019.

[39] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, ISMIR, 2019.

[40] J. B. Mailman, "Repetition in music: Theoretical and metatheoretical perspectives," *Psychology of Music*, vol. 35, no. 2, pp. 363–375, 2007.

[41] L. Grif, "Steve reich. the desert music," *Les Cahiers Du Grif*, vol. 41-42, 1989.

[42] R. Fink, *Repeating ourselves*. University of California Press, 2005.

[43] S. Dai, H. Zhang, and R. B. Dannenberg, "Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music," in *Proceedings of the 2020 Joint Conference on AI Music Creativity and International Workshop on Music Metacreation (CSMC+MUME)*, 2020.

[44] Y. Zou, P. Zou, Y. Zhao, K. Zhang, R. Zhang, and X. Wang, "Melons: generating melody with long-term structure using transformers and structure graph," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 191–195, IEEE, 2022.

[45] J.-J. Nattiez, *Music and discourse – Toward a semiology of music*. Princeton University Press, 1990.

[46] S. Mangal, R. Modak, and P. Joshi, "Lstm based music generation system," *arXiv preprint arXiv:1908.01080*, 2019.

[47] J. Wu, X. Liu, X. Hu, and J. Zhu, "Popmnet: Generating structured pop music melodies using neural networks," *Artificial Intelligence*, vol. 286, p. 103303, 2020.

[48] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[49] A. Elowsson and A. Friberg, "Algorithmic composition of popular music," in *The 12th international conference on music perception and cognition and the 8th triennial conference of the european society for the cognitive sciences of music*, pp. 276–285, 2012.

[50] S. Dai, X. Ma, Y. Wang, and R. B. Dannenberg, "Personalised popular music generation using imitation and structure," *Journal of New Music Research*, pp. 1–17, 2023.

[51] V. Andries and W. Schulze, "Music generation with markov models," *IEEE Multimedia*, vol. 18, no. 3, pp. 78–85, 2011.

[52] D. Herremans and E. Chew, "Morpheus: generating structured music with constrained patterns and tension," *IEEE Transactions on Affective Computing*, pp. 1–1, 2017.

[53] X. Fu, H. Deng, X. Yuan, and J. Hu, "Generating high coherence monophonic music using monte-carlo tree search," *IEEE Transactions on Multimedia*, 2022.

[54] L. Yang, S. Chou, and Y. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 324–331, 2017.

[55] G. Chen, Y. Liu, S.-h. Zhong, and X. Zhang, "Musicality-novelty generative adversarial nets for algorithmic composition," in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1607–1615, 2018.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing*

Systems 30: Annual Conference on Neural Information Processing Systems, pp. 5998–6008, 2017.

[57] C. Hernandez-Olivan and J. R. Beltran, "Music composition with deep learning: A review," *arXiv preprint arXiv:2108.12290*, 2021.

[58] S.-L. Wu and Y.-H. Yang, "The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 142–149, 2020.

[59] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[60] L. Fang, T. Zeng, C. Liu, L. Bo, W. Dong, and C. Chen, "Outline to story: Fine-grained controllable story generation from cascaded events," *arXiv preprint arXiv:2101.00822*, 2021.

[61] G. Pareyon, *On Musical Self-Similarity : Intersemiosis as synecdoche and analogy*. Gabriel Pareyon, 2011.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[63] J. T. Titon, *Worlds of music: an introduction to the music of the world's peoples*. 2016.

[64] B. Benward, *Music in Theory and Practice vol. 2*. 2018.

[65] C. Benetatos, J. VanderStel, and Z. Duan, "Bachduet: A deep learning system for human-machine counterpoint improvisation," in *20th International Conference on New Interfaces for Musical Expression, NIME*, pp. 635–640, 2020.

[66] J. Biles *et al.*, "Genjam: A genetic algorithm for generating jazz solos," in *ICMC*, vol. 94, pp. 131–137, 1994.

[67] A. Schoenberg, L. Stein, and G. Strang, *Fundamentals of musical composition*. 1967.

[68] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, *Deep learning techniques for music generation*, vol. 1. 2020.

[69] C. Hernandez-Olivan and J. R. Beltran, "Music composition with deep learning: A review," *Advances in Speech and Music Technology: Computational Aspects and Applications*, pp. 25–50, 2022.

[70] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International conference on machine learning*, pp. 4364–4373, 2018.

[71] T. Akama, "Controlling symbolic music generation based on concept learning from domain knowledge," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pp. 816–823, 2019.

[72] S. Dai, X. Ma, Y. Wang, and R. B. Dannenberg, "Personalised popular music generation using imitation and structure," *Journal of New Music Research*, pp. 1–17, 2023.

[73] P. Lu, X. Tan, B. Yu, T. Qin, S. Zhao, and T.-Y. Liu, "Meloform: Generating melody with musical form based on expert systems and neural networks," *arXiv preprint arXiv:2208.14345*, 2022.

[74] H. H. Mao, T. Shin, and G. Cottrell, "Deepj: Style-specific music generation," in *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pp. 377–382, 2018.

[75] G. Keerti, A. Vaishnavi, P. Mukherjee, A. S. Vidya, G. S. Sreenithya, and D. Nayab, "Attentional networks for music generation," *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 5179–5189, 2022.

[76] X. Guo, H. Xu, and K. Xu, "Fine-tuning music generation with reinforcement learning based on transformer," in *2022 IEEE 16th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pp. 1–5, 2022.

[77] M. Bretan, S. Oore, J. Engel, D. Eck, and L. Heck, "Deep music: Towards musical dialogue," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5081–5082, 2017.

[78] N. Collins, "Automatic composition of electroacoustic art music utilizing machine listening," *Computer Music Journal*, vol. 36, no. 3, pp. 8–23, 2012.

[79] S. Chikkamath and S. Nirmala, "Melody generation using lstm and bi-lstm network," in *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, pp. 1–6, 2021.

[80] G. H. de Rosa and J. P. Papa, "A survey on text generation using generative adversarial networks," *Pattern Recognition*, vol. 119, p. 108098, 2021.

[81] J. Kachulis, *The Songwriter's Workshop: Harmony*. 2004.

[82] World Health Organization, "Depression and other common mental disorders: global health estimates," tech. rep., World Health Organization, 2017.

[83] K. S. Kendler, A. C. Heath, N. G. Martin, and L. J. Eaves, "Symptoms of anxiety and symptoms of depression: same genes, different environments?," *Archives of general psychiatry*, vol. 44, no. 5, pp. 451–457, 1987.

[84] N. Salari, A. Hosseinian-Far, R. Jalali, A. Vaisi-Raygani, S. Rasoulpoor, M. Mohammadi, S. Rasoulpoor, and B. Khaledi-Paveh, "Prevalence of stress, anxiety, depression among the general population during the covid-19 pandemic: a systematic review and meta-analysis," *Globalization and health*, vol. 16, no. 1, pp. 1–11, 2020.

[85] W. B. Davis, K. E. Gfeller, and M. H. Thaut, *An introduction to music therapy: Theory and practice*. ERIC, 2008.

[86] H. Chiu, L. Lin, M. Kuo, H. Chiang, and C. Hsu, "Using heart rate variability analysis to assess the effect of music therapy on anxiety reduction of patients," in *Computers in Cardiology, 2003*, pp. 469–472, 2003.

[87] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," *Expert Systems with Applications*, pp. 118–190, 2022.

[88] D. Grocke and T. Wigram, *Receptive methods in music therapy: Techniques and clinical applications for music therapy clinicians, educators and students*. Jessica Kingsley Publishers, 2006.

[89] I. Daly, D. Williams, A. Malik, J. Weaver, A. Kirke, F. Hwang, E. Miranda, and S. J. Nasuto, "Personalised, multi-modal, affective state detection for hybrid brain-computer music interfacing," *IEEE Transactions on Affective Computing*, vol. 11, no. 1, pp. 111–124, 2020.

[90] J.-C. Wang, Y.-H. Yang, H.-M. Wang, and S.-K. Jeng, "Modeling the affective content of music with a gaussian mixture model," *IEEE Transactions on Affective Computing*, vol. 6, no. 1, pp. 56–68, 2015.

[91] L. Lu, D. Liu, and H.-J. Zhang, "Automatic mood detection and tracking of music audio signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 5–18, 2006.

[92] S.-M. Wang, L. Kulkarni, J. Dolev, and Z. N. Kain, "Music and preoperative anxiety: a randomized, controlled study," *Anesthesia & Analgesia*, vol. 94, no. 6, pp. 1489–1494, 2002.

[93] E. Yilmaz, S. Ozcan, M. Basar, H. Basar, E. Batislam, and M. Ferhat, "Music decreases anxiety and provides sedation in extracorporeal shock wave lithotripsy," *Urology*, vol. 61, no. 2, pp. 282–286, 2003.

[94] R. Kanehira, Y. Ito, M. Suzuki, and F. Hideo, "Enhanced relaxation effect of music therapy with vr," in *International Conference on Natural Computation*, pp. 1374–1378, 2018.

[95] K. Wang, W. Wen, and G.-Y. Liu, "The autonomic nervous mechanism of music therapy for dental anxiety," in *2016 13th International Computer Conference on Wavelet Ac-*

tive Media Technology and Information Processing (ICCWAMTIP), pp. 289–292, IEEE, 2016.

[96] R. A. Latif, "Preferred sound type for stress therapy," in *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*, pp. 1–6, IEEE, 2018.

[97] M. A. Khan, M. Chennafi, G. Li, and G. Wang, "Electroencephalogram-based comparative study of music effect on mental stress relief," in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–5, IEEE, 2018.

[98] F. Li and Y. Xiong, "Application of music therapy combined with computer biofeedback in the treatment of anxiety disorders," in *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 90–93, IEEE, 2016.

[99] M. Schedl, E. Gómez, E. S. Trent, M. Tkalčič, H. Eghbal-Zadeh, and A. Martorell, "On the interrelation between listener characteristics and the perception of emotions in classical orchestra music," *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 507–525, 2018.

[100] G. G. Xia and S. Dai, "Music style transfer: A position paper," in *Proceedings of the 6th International Workshop on Musical Metacreation*, pp. 1–6, MUME, 2018.

[101] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer," in *International Conference on Learning Representations*, 2019.

[102] A. Haque, M. Guo, and P. Verma, "Conditional end-to-end audio transforms," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association*, pp. 2295–2299, ISCA, 2018.

[103] C.-Y. Lu, M.-X. Xue, C.-C. Chang, C.-R. Lee, and L. Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1061–1068, 2019.

[104] Y.-C. Chang, W.-C. Chen, and M.-C. Hu, "Semi-supervised many-to-many music timbre transfer," in *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 442–446, 2021.

[105] Y.-J. Luo, K. Agres, and D. Herremans, "Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, (ISMIR)*, pp. 746–753, 2019.

[106] I. Malik and C. H. Ek, "Neural translation of musical style," *arXiv preprint arXiv:1708.03535*, 2017.

[107] C. Hawthorne, A. Huang, D. Ippolito, and D. Eck, "Transformer-nade for piano performances," in *NIPS 2nd Workshop on Machine Learning for Creativity and Design*, 2018.

[108] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel, "Encoding musical style with transformer autoencoders," in *International Conference on Machine Learning*, pp. 1899–1908, PMLR, 2020.

[109] O. Cífka, A. Ozerov, U. Şimşekli, and G. Richard, "Self-supervised vq-vae for one-shot music style transfer," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 96–100, 2021.

[110] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, "A universal music translation network," in *7th International Conference on Learning Representations, ICLR*, OpenReview.net, 2019.

[111] S. Mukherjee and M. Mulimani, "Composeinstyle: Music composition with and without style transfer," *Expert Systems with Applications*, vol. 191, p. 116195, 2022.

[112] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[113] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and

super-resolution," in *European conference on computer vision*, pp. 694–711, Springer, 2016.

[114] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*, pp. 4393–4402, 2018.

[115] C. Raffel and D. P. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty midi," in *15th International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*, pp. 84–93, 2014.

[116] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[117] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proceedings of the International Conference on Machine Learning*, 2013.

[118] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR*, 2014.

[119] G. Gessle and S. Åkesson, "A comparative analysis of cnn and lstm for music genre classification," 2019.

[120] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in neural information processing systems*, pp. 700–708, 2017.

[121] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 172–189, 2018.

[122] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[123] C. D. Spielberger, S. J. Sydeman, A. E. Owen, and B. J. Marsh, *Measuring anxiety and anger with the State-Trait Anxiety Inventory (STAI) and the State-Trait Anger Expression Inventory (STAXI).* Lawrence Erlbaum Associates Publishers, 1999.

[124] O. Kayikcioglu, S. Bilgin, G. Seymenoglu, and A. Deveci, "State and trait anxiety scores of patients receiving intravitreal injections," *Biomedicine hub*, vol. 2, no. 2, pp. 1–5, 2017.

[125] H.-C. Sung, A. M. Chang, and W.-L. Lee, "A preferred music listening intervention to reduce anxiety in older adults with dementia in nursing homes," *Journal of clinical nursing*, vol. 19, no. 7-8, pp. 1056–1064, 2010.

[126] S. L. Robb, R. J. Nichols, R. L. Rutan, B. L. Bishop, and J. C. Parker, "The effects of music assisted relaxation on preoperative anxiety," *Journal of music therapy*, vol. 32, no. 1, pp. 2–21, 1995.

[127] A. E. Greasley and A. M. Lamont, "Music preference in adulthood: Why do we like the music we do," in *Proceedings of the 9th international conference on music perception and cognition*, pp. 960–966, Citeseer, 2006.

[128] Y. Hung, I. Chiang, Y. Chen, and Y. Yang, "Musical composition style transfer via disentangled timbre representations," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 4697–4703, 2019.

[129] G. C. Mornhinweg, "Effects of music preference and selection on stress reduction," *Journal of Holistic Nursing*, vol. 10, no. 2, pp. 101–109, 1992.

[130] J. Bradt, C. Dileo, and N. Potvin, "Music for stress and anxiety reduction in coronary heart disease patients," *Cochrane Database of Systematic Reviews*, no. 12, 2013.

[131] D. Rawlings and V. Ciancarelli, "Music preference and the five-factor model of the neo personality inventory," *Psychology of Music*, vol. 25, no. 2, pp. 120–132, 1997.

[132] P. G. Dunn, B. de Ruyter, and D. G. Bouwhuis, "Toward a better understanding of the relation between music preference, listening behavior, and personality," *Psychology of Music*, vol. 40, no. 4, pp. 411–428, 2012.

[133] D. Bogdanov, M. Haro, F. Fuhrmann, E. Gómez, and P. Herrera, "Content-based music recommendation based on user preference examples," in *ACM conf. on recommender systems. Workshop on music recommendation and discovery (Womrad 2010)*, 2010.

[134] O. Celma, "Music recommendation," in *Music recommendation and discovery*, pp. 43–85, Springer, 2010.

[135] R. Kanehira, Y. Ito, M. Suzuki, and F. Hideo, "Enhanced relaxation effect of music therapy with vr," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 1374–1378, IEEE, 2018.

[136] I. Donadello, L. Serafini, and A. S. d'Avila Garcez, "Logic tensor networks for semantic image interpretation," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 1596–1602, 2017.

[137] A. S. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning," *FLAP*, vol. 6, no. 4, pp. 611–632, 2019.

[138] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[139] I. Donadello, *Semantic image interpretation-integration of numerical data and logical knowledge for cognitive vision*. PhD thesis, University of Trento, 2018.

[140] A. Liu, A. Fang, G. Hadjeres, P. Seetharaman, *et al.*, "Incorporating music knowledge in continual dataset augmentation for music generation," *arXiv preprint arXiv:2006.13331*, 2020.

[141] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, 2019.

[142] J. Park, K. Choi, S. Jeon, D. Kim, and J. Park, "A bi-directional transformer for musical chord recognition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 620–627, 2019.

[143] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. J. McAuley, "Lakhnes: Improving multi-instrumental music generation with cross-domain pre-training," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, (ISMIR)*, pp. 685–692, 2019.

[144] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5880–5888, 2019.

[145] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "One-shot object detection with co-attention and co-excitation," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019.

[146] L. B. Meyer, *Style and music: Theory, history, and ideology*. University of Chicago Press, 1996.

[147] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 362–373, 2011.