

The Beauty of Repetition in Machine Composition Scenarios

Zhejiang Hu
The Hong Kong Polytechnic
University
Hong Kong SAR, China
cszhu@comp.polyu.edu.hk

Xiao Ma
The Hong Kong Polytechnic
University
Hong Kong SAR, China
xiao1ma@comp.polyu.edu.hk

Yan Liu*
The Hong Kong Polytechnic
University
Hong Kong SAR, China
csyliu@comp.polyu.edu.hk

Gong Chen
The Hong Kong Polytechnic
University
Hong Kong SAR, China
csgchen@comp.polyu.edu.hk

Yongxu Liu
The Hong Kong Polytechnic
University
Hong Kong SAR, China
csyxl@comp.polyu.edu.hk

ABSTRACT

Repetition, a basic form of artistic creation, appears in most musical works and delivers enthralling aesthetic experiences. However, repetition remains underexplored in terms of automatic music composition. As an initial effort in repetition modelling, this paper focuses on generating motif-level repetitions via domain knowledge-based and example-based learning techniques. A novel repetition transformer (R-Transformer) that combines a Transformer encoder and a repetition-aware learner is trained on a new repetition dataset with 584,329 samples from different categories of motif repetition. The Transformer encoder learns the representation among music notes from the repetition dataset; the novel repetition-aware learner exploits repetitions' unique characteristics based on music theory. Experiments show that, with any given motif, R-Transformer can generate a large number of variable and beautiful repetitions. With ingenious fusion of these high-quality pieces, the musicality and appeal of machine-composed music have been greatly improved.

CCS CONCEPTS

• Applied computing → Sound and music computing.

KEYWORDS

Automatic music composition; symbolic music generation; repetition

ACM Reference Format:

Zhejiang Hu, Xiao Ma, Yan Liu, Gong Chen, and Yongxu Liu. 2022. The Beauty of Repetition in Machine Composition Scenarios. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*, October 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3503161.3548130>

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

MM '22, October 10–14, 2022, Lisboa, Portugal.

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9203-7/22/10...\$15.00

<https://doi.org/10.1145/3503161.3548130>

1 INTRODUCTION

Repetition, an act or an instance of repeating or being repeated, is ubiquitous in everyday life. It manifests in DNA sequencing; the Earth's rotation; and activities such as swinging, exercising, and writing poems. It further influences people's daily lives through the changing of seasons, tidal variation, and so forth.

In music world, repetition and its variants generate musical phantasmagoria with thrilling effects. Let's begin our exploration of this simple composition skill from the appreciation of a well-known classical work, Beethoven's Fifth Symphony (Figure 1). An eighth note at the pitch of "G" with its two repetitions rings out, breaking the stillness, delivering a strong and firm signal. Then, a half note at the pitch of "E-flat" ends the motif succinctly that leaves the listener with a sense of breathlessness, dread and anticipation for the music to come. The "G"s and the "E-flat" make up a descending Major 3rd interval and this is the famous fate motif, which delivers a feeling of "fate knocking at the door". Then, a transpositional repetition (TrR) of the fate motif that is the same motif repeats in a lower tone makes the atmosphere more solemn and dignified. In the second phrase, a strict repetition (StR) of the fate motif appears again with the same pitch value but longer "E-flat", leading to a smooth expansion of spectacular views. A homodirectional repetition (HoR) of triple "A-flat" and "G" triggers a precarious feeling. A subsequential repetition (SuR) "G-G-G-D" occurs with a long "D" note, forms a similar but lower "platform" like the strict repetition of the fate motif, embodying a groaning of the spirit. In addition, a transpositional repetition reaches a higher pitch, making the atmosphere more intense and enormous. In the third phrase, after a few alternating downward and upward notes, the appearance of a symmetric repetition (SyR) offers an upward movement, making a contrast that is distinct compared with other motifs. Three variants, a homodirectional repetition, a subsequential repetition, and a symmetric repetition, ring out together, forming a sense of contraction, bringing the piece to a tranquil state, symbolizing a truce, and heralding the arrival of a greater storm.

Music from every genre, culture, and period employs repetition for effect, yet these patterns have rarely been studied in machine composition [26]. One of music's most repetition-obsessed composers, Steve Reich, exemplified this phenomenon in "The Desert Music" [21]. Repetition in music does not simply entail identical elements but rather echoing elements in a new way [10]. In other

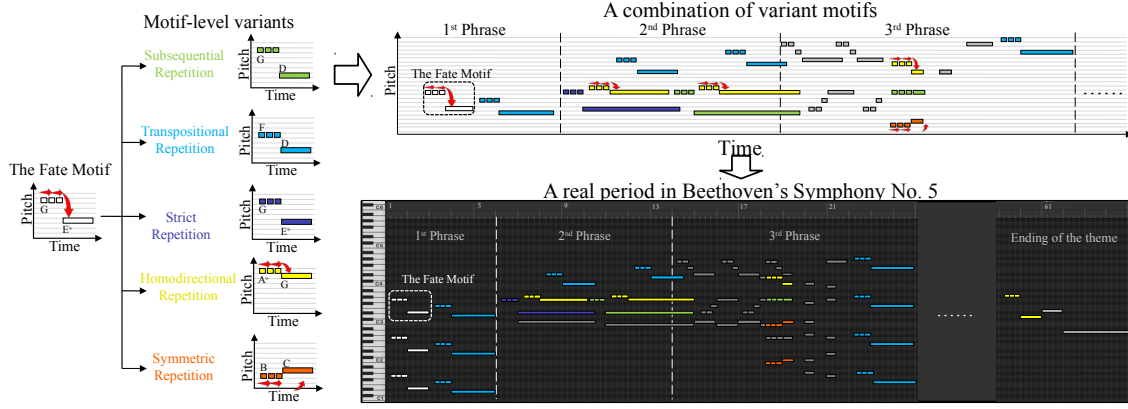


Figure 1: Beethoven's Fifth Symphony (simplified). The first four-note is the most recognizable “fate motif”.

words, patterns of repetition are more than composition skills—they invite listeners to participate.

Repetition can be categorized along multiple dimensions since it is diverse, complicated, and includes different composition forms. First, different criteria can be used to classify repetition (e.g., pitch values, rhythm, and harmony) [6, 37]. To avoid overlap due to different criteria, we focus on the in-depth deconstruction of musical repetition in terms of pitch value (Table 1). Second, repetition can be divided by level—the note level, motif level, phrase level, and period level. As the most basic analyzable and meaningful element of a musical composition, motifs create dynamic coherence through repetition, variation, and combination at different scales [25]. Therefore, motif-level repetitions are examined in this paper.

The diversity and complication of repetition in music further results in two challenges. First, the scarcity of repetition-related data precludes investigation of different repetition types. No public dataset is available to specify repetition types; it is accordingly difficult to further analyze music repetition. Second, how to produce and combine explicit repetitions to make pleasant music is still challenging. Currently, some work uses rule-based and statistical methods to construct long-term repetitive structures [3, 5, 8, 12, 29]. Others employ memory modules in deep learning models, such as long short-term memory (LSTM) and Transformer, to generate music [4, 13–15, 22, 33]. However, these models cannot 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.

To surmount these obstacles, we first present a new dataset, Music Repetition Dataset (MRD), with repetition labels based on these definitions. We also develop a novel model called Repetition Transformer (R-Transformer) that can be controlled to generate designated music repetition given a music piece. This model combines music knowledge and a Transformer encoder. The domain knowledge-based mechanism allows the model to explicitly follow music theory while the transformer encoder mechanism enables the model to learn the representation of notes. In this model, a novel repetition-aware learner is designed to generate different repetitions based on their unique characteristics learned from a Transformer encoder. Based on multi-aspect attribute representation in

music repetition, parameters in repetition-aware learner can be controlled to exploit unique characteristics of different repetitions. To better produce different repetition types, a reconstruction and classification cooperation method is proposed to train R-Transformer. In generation phase, a rule-based generation module is applied to generate the designated repetition given an repetition type, which helps users control the process of composing different repetition types. They can thus create music more precisely.

The rest of this paper is organized as follows. In Section 2, we review related work on music repetition generation. In Section 3, we describe the proposed method in detail. Sections 4 and 5 discuss the implementation details and experimental results. We close with our conclusions and directions for future work.

2 RELATED WORK

Different from rule-based [5] or convolutional neural network (CNN)-based generation models [2, 7, 35], Transformer [31] has been widely adopted as the backbone generative model: the “self-attention” mechanism can provide a much longer memory so the model can learn the repetitive structure of music. Music Transformer [14] was the first Transformer-based model in symbolic music generation, showing that a Transformer model can generate coherent minute-long polyphonic piano music with reasonable repetitions and variance. Other Transformer-based models have since been proposed to generate music [9, 13, 15, 17, 24, 28, 36]. These models can learn musical features without hand-crafted rules and produce diverse music pieces. However, without studying music repetition and analyzing the music structure, the model tends to lose a specific sense of direction [11] and the generated music tends to be random without rhythmic patterns [34]. Overall, machine learning-based models continue to struggle to adhere to certain key musical ideas in composition and to generate repetitive rhythmic patterns.

Scholars have sought to analyze the hierarchical structure of music to generate repetitive music patterns [3, 16, 23, 32]. Recently, MusicFrameworks [4] was proposed to use a Transform and LSTM-based model to create a full-length melody guided by a long-term repetitive structure. PopMNet [33] involves a structured generation network and a melody generation network based on the structural

Table 1: Types of repeated patterns in terms of pitch.

Scale	No.	Repetition Name	Definition
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) ¹	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 ² .
	7	Symmetric repetition (SyR)	Two motifs are not SuR, but have similar symmetric repetition development.
Phrase	8	Similar melody	Two phrases have similar melody.
Period	9	Structural repetition	Two periods have similar structure.

representation and chord progression of input music. MELONS [37] entails a multi-step generation method with Transformer-based models and graph representation for music structure generation and structure-conditional melody generation. Theme Transformer [30] achieves theme-based conditioning by producing it multiple times in the generation result so that the output music follows the thematic material. These models were constructed based on the structure of music from multiple levels (e.g., the motif level and phrase level). Generating music based on the structure of the music can relieve the problem that generated music might lose a specific sense of direction. Yet none of these models studies music repetition modeling and they cannot generate explicit repetition types. Without modeling music repetition, users cannot control repetition and music trend. Generating repetitive patterns and understandable music rhythmically is still a challenge in these models.

3 METHOD

In this section, we introduce our R-Transformer architecture and elaborate on classification and reconstruction cooperation learning to train R-Transformer from scratch. Figure 2 provides an overview of the architecture. We feed the input music and its repetition type into the model, where the raw input is first tokenized and then embedded to multiple attribute-embedding vectors followed by a Transformer encoder and a repetition-aware learner. There are three major modules in R-Transformer: 1) a multi-attribute-embedding layer; 2) a single backbone Transformer encoder applied to all music attributes; and 3) a repetition-aware learner to learn each attribute separately. A reconstruction and classification cooperation training strategy is proposed in this paper to generate different repetition types.

Mathematically, given an input music repetition after tokenization $\mathbf{x} \in X$ and a repetition type $\mathbf{y} \in Y$, where $X \subseteq \mathbb{R}^{L \times K}$ is the input space and $Y \subseteq \mathbb{R}^M$ is the label space. L is the length of the input, K is the number of music attributes such as note pitch, note duration, and etc., and M is the number of repetition types. The goal is to generate a new piece of music $\hat{\mathbf{x}} \in \hat{X}$, where $\hat{X} \subseteq \mathbb{R}^{L \times K}$. Let $f_c : X \rightarrow Y$ be the label learning pipeline and $f_r : X \rightarrow \hat{X}$ be

the data reconstruction pipeline of R-Transformer. We define an embedding space $H \subseteq \mathbb{R}^{L \times H_1}$ and a feature space $F \subseteq \mathbb{R}^{L \times H_2}$, where H_1 and H_2 are embedding and feature dimensions. We also define four additional functions: 1) a multi-attribute-embedding mapping $g_{\text{emb}} : X \rightarrow H$; 2) an encoder/feature mapping $g_{\text{enc}} : H \rightarrow F$; 3) a decoder $g_{\text{dec}} : F \rightarrow \hat{X}$; 4) a feature-labeling function $g_{\text{lab}} : F \rightarrow Y$. The label learning pipeline f_c and the reconstruction pipeline f_r can be decomposed such that

$$f_c(\mathbf{x}) = (g_{\text{lab}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}), \quad (1)$$

$$f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y}), \quad (2)$$

where \circ is the composition of two functions. In Equation 1, the embedding g_{emb} is a multi-attribute embedding module (Figure 2 (a)). The encoder g_{enc} is a Transformer encoder (Figure 2 (b)). The labeling module g_{lab} consists of a linear projection (Figure 2 (c)). In Equation 2, the output of g_{enc} is concatenated with a target repetition label \mathbf{y} and fed into g_{dec} , which consists of K linear projection layers to predict K attributes of the output music (Figure 2 (c)).

3.1 Multi-Attribute Embedding

In this step, a multi-attribute embedding module is implemented to embed multi-aspect attributes of the music separately, which contain music information such as note pitch, note duration, and so on. The input music \mathbf{x} is embedded and the output of multi-attribute embedding g_{emb} after concatenation is $[\text{Embed}_1(\mathbf{x}_{:,1}), \dots, \text{Embed}_K(\mathbf{x}_{:,K})]$, where $\text{Embed}_k(\cdot)$ is the embedding layer of k -th attribute and $[\cdot, \cdot]$ is the action of concatenation.

3.2 Transformer Encoder Architecture

We adopt the most established Transformer encoder architecture [31], which has been frequently applied in music generation as of late [13, 37]. We follow the standard Transformer encoder architecture [31] and illustrate it in Figure 2 (b). The sequence of input tokens to the Transformer encoder is $g_{\text{emb}}(\mathbf{x}) + \mathbf{e}_{\text{pos}}$, where $\mathbf{e}_{\text{pos}} \in \mathbb{R}^{L \times H_1}$ is a positional embedding vector. The output in the Transformer encoder is used as the aggregated representation for the entire input sequence. This output will later be used for classification (g_{lab}) and reconstruction (g_{dec}). We employ a standard self-attention as the multi-head-attention module with GeLU [31]

¹Both chromatic and diatonic transposition are considered. Chromatic is scalar transposition within the chromatic scale, implying that every pitch in a collection of notes is shifted by the same number of semitones. Diatonic transposition is scalar transposition within a diatonic scale (a standard scale under some tonality indicated by a certain standard key signature).

²Development: the direction of the pitch sequence from one note to the next.

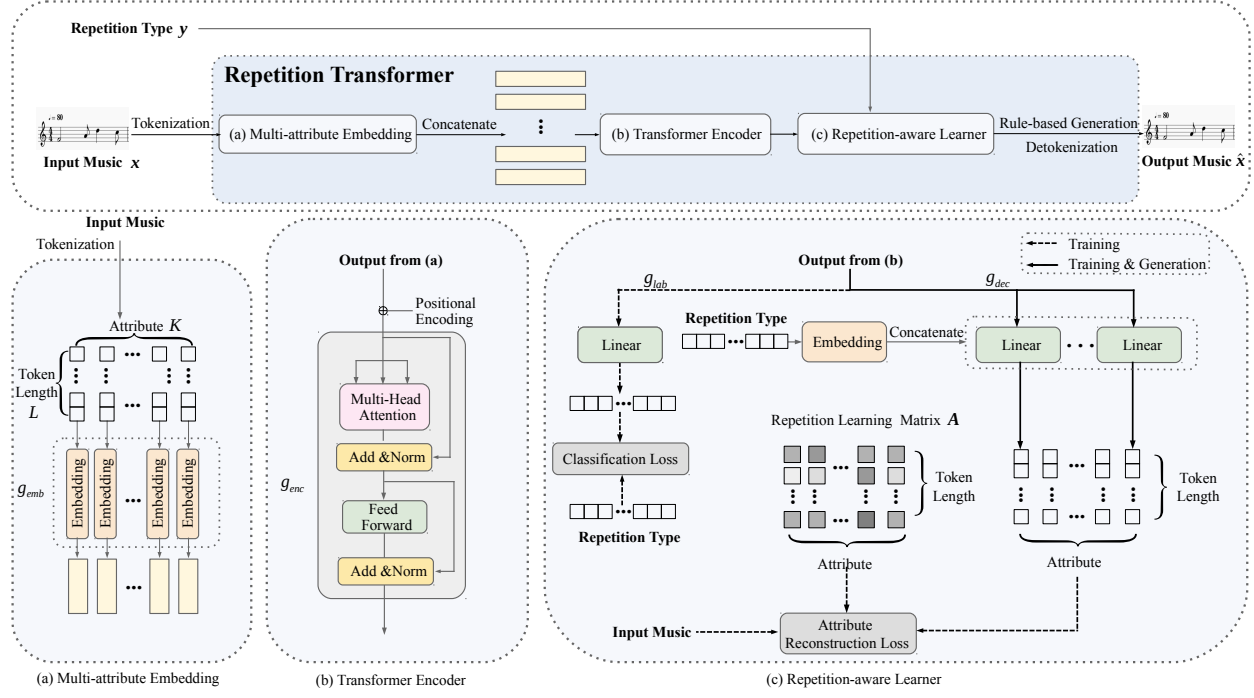


Figure 2: A general framework for R-Transformer. R-Transformer linearly embeds multi-aspect attributes and feeds them into a Transformer encoder. We define a repetition-aware learner that can account for the contributions of different attributes. We employ one classification and one reconstruction loss to train the model.

as the activation in the linear projection layer. We also use layer normalization [1] after the multi-head-attention and linear projection modules.

3.3 Repetition-aware Learner

In this step, we propose a novel repetition-aware learner module to train the model and generate music. The output of g_{enc} is fed into a feature-labeling function g_{lab} to predict the repetition type y . g_{enc} is also fed into a linear decoder g_{dec} concatenated with the target repetition label to reconstruct the input music x .

Specifically, we first define a repetition learning matrix that enables us to directly control the training process following the definition of repetition types. Then, a reconstruction and classification cooperation learning strategy is proposed to train the model.

Repetition learning matrix: It is known that different repetition types have different characteristics; and different motifs have different representations. To learn these differences, we study multi-aspect attributes by first defining a repetition learning matrix $A \in \mathbb{R}^{L \times K}$.

The element $a_{l,k}$ in repetition learning matrix A can be calculated as

$$a_{l,k} = \gamma \cdot (1 + \omega_{l,k}), \quad (3)$$

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

$$\omega_{l,k} = \frac{Counter(x_{l,k}, \mathbf{x}_{:,k})}{L}, \quad (4)$$

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

The process of repetition-aware learning is depicted in Figure 2 (c). The main intuition behind the multi-aspect attribute learning is from two aspects. First, as described in Table 1, different attributes represent differently in different types. Although pitch is the main attribute that determines repetition types, other attributes such as note duration, note position and note velocity also contribute differently. For example, strict repetition (StR) have same number of notes and pitch values are identical, while subsequential repetition (SuR) of two motifs might have different number of notes and different position, duration and velocity of notes. Therefore, in SuR, γ in Equation 3 will be given a higher attribute importance if k is pitch, position, duration and velocity. Second, the representation of different motifs varies within the same repetition type. In other words, the distribution of each attribute is totally different among motifs. For example, in pitch dimension, “C3-D3-E3-F3-G3” and “C3-D3-C3-E3-C3” have different representations. If k is pitch, then $a_{1,k}$, $a_{3,k}$ and $a_{5,k}$ will be larger based on Equations 3 and 4 in the latter case, since “C3” appears more frequently than other pitch values.

Learning algorithm: Given labeled input music samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{y}_i \in \{0, 1\}^M$ is a one-hot vector with M classes and N is the number of music samples. Let $\theta_c = \{\theta_{emb}, \theta_{enc}, \theta_{lab}\}$ and $\theta_r = \{\theta_{emb}, \theta_{enc}, \theta_{dec}\}$ 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 1 and 2, we first define the empirical classification loss as

$$\mathcal{L}_c(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}) := \sum_{i=1}^N \ell_c(f_c(\mathbf{x}_i; \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}), \mathbf{y}_i). \quad (5)$$

Typically, ℓ_c is of the form cross-entropy loss $\sum_{m=1}^M y_k \log[f_c(\mathbf{x})]_m$ (recall that $f_c(\mathbf{x})$ is the softmax output). In addition, the attribute reconstruction loss is defined as

$$\mathcal{L}_r(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}) := \sum_{i=1}^N \ell_r(f_r(\mathbf{x}_i, \mathbf{y}_i; \{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}), \hat{\mathbf{x}}_i). \quad (6)$$

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

$$\ell_r = \|\mathbf{A} \otimes (\hat{\mathbf{x}} - f_r(\mathbf{x}, \mathbf{y}))\|_2^2, \quad (7)$$

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

The final objective of our proposed method to minimize the total loss \mathcal{L} and is formulated as follows:

$$\min_{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}, \theta_{\text{lab}}} \lambda \mathcal{L}_c(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}\}) + (1-\lambda) \mathcal{L}_r(\{\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{dec}}\}), \quad (8)$$

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 8 can be achieved by minimizing \mathcal{L}_c and \mathcal{L}_r using stochastic gradient descent. The stopping criterion for the algorithm is determined by monitoring the average total loss during training – the process is stopped when the average total loss stabilizes.

Rule-based generation: In the generation phase, only the reconstruction pipeline in the repetition-aware learner will be activated to generate the output, that is, $\hat{\mathbf{x}} = f_r(\mathbf{x}, \mathbf{y})$ where \mathbf{x} and \mathbf{y} are the input music and the designated repetition type. In addition, we also update the results by following the definition of different repetitions based on music theory. For instance, in a StR, the pitch value is assumed to be identical to the input motif, so we fix the pitch value. Specifically, if k represents pitch, then $\hat{x}_{l,k} = x_{l,k}$. In a TrR, the difference of pitch value is assumed to be fixed, so we follow the rule to generate pitch value after the first pitch value is predicted. Specifically, if k represents pitch, then $\hat{x}_{l,k} = x_{l,k} + t$, where $t = \{\dots, -2, -1, 1, 2, \dots\}$ is a transposition value. In SuR, HoR and SyR, the representation of different notes is learned by the model, so all music attributes are generated based on the example-based mechanism. Furthermore, user can generate multiple repetition types by giving the model multiple labels. For example, the model can generate four motifs sequentially if four labels are given. The training and generation phase of R-Transformer learning algorithm is summarized in Algorithm 1. In terms of StR and TrR, the proposed model utilizes rule-based generation to ensure the pitch attribute is identical to the input. In addition, example-based mechanism allows the proposed model to learn representations in terms of other attributes from music examples, which makes the output music diverse to hear.

Algorithm 1 R-Transformer 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 $\theta^* = \{\theta_{\text{emb}}^*, \theta_{\text{enc}}^*, \theta_{\text{lab}}^*, \theta_{\text{dec}}^*\}$

- 1: Initialize parameters $\theta_{\text{emb}}, \theta_{\text{enc}}, \theta_{\text{lab}}, \theta_{\text{dec}}$
- 2: **while** not converge **do**
- 3: **for each** batch of size n **do**
- 4: Do a forward pass $f_c(\mathbf{x}) = (g_{\text{lab}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x})$;
- 5: Do a forward pass $f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y})$;
- 6: Calculate repetition learning matrix \mathbf{A} based on Equations 3 and 4;
- 7: Update $\theta : \theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}^n(\theta)$;
- 8: **end for**
- 9: **end while**

Generation phase:

Input: Music data \mathbf{x} and designated repetition type \mathbf{y}

Parameter: $\theta_{\text{emb}}^*, \theta_{\text{enc}}^*, \theta_{\text{dec}}^*$

Output: Output music $\hat{\mathbf{x}}$

- 1: **if** \mathbf{y} is StR **then**
 - 2: **if** k is pitch **then**
 - 3: $\hat{x}_{l,k} = x_{l,k}$;
 - 4: **end if**
 - 5: **else if** \mathbf{y} is TrR **then**
 - 6: **if** k is pitch **then**
 - 7: $\hat{x}_{l,k} = x_{l,k} + t$;
 - 8: **end if**
 - 9: **else**
 - 10: $\hat{\mathbf{x}} = f_r(\mathbf{x}, \mathbf{y}) = (g_{\text{dec}} \circ g_{\text{enc}} \circ g_{\text{emb}})(\mathbf{x}, \mathbf{y})$;
 - 11: **end if**
-

4 EXPERIMENTAL CONFIGURATIONS

We train our model using the Pop piano dataset from [13] because the structures in pop music are relatively simple. In addition, this dataset covers all repetition types shown in Table 1. This dataset contains 1,748 pieces of pop piano from the Internet. All songs are in 4/4 time signature, and each song is converted into a symbolic sequence following transcription, synchronization, quantization, and analysis.

4.1 Music Repetition Dataset

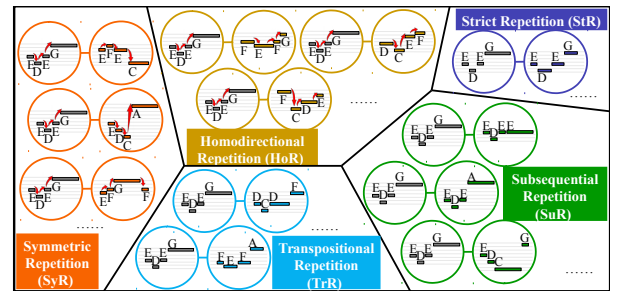


Figure 3: Examples of five repetition types in MRD.

We further partition the music piece into motifs. After partition, we compare the pitch value of all motifs. Pitch and development are used since they are both commonly used in music creation, and can be easily recognized and understood by the audience [27]. If two motifs satisfy the definition in Table 1, then it is a repetition; otherwise, it is not a repetition. In StR, the pitch value of all notes should be identical. For other repetition types, the melody of the note sequence is compared if the music contains both melody and accompaniment.

Table 2: Percentage of different repetition types in training and test dataset of MRD.

Dataset	Repetition	Numbers	Percentages	Avg Length (tokens)
Training	StR	21807	3.88%	64.93
	TrR	27852	4.95%	62.94
	SuR	73774	13.11%	73.74
	HoR	152060	27.02%	59.68
	SyR	287070	51.03%	64.62
Test	StR	1395	6.41%	61.22
	TrR	1635	7.51 %	60.81
	SuR	3331	15.31%	69.71
	HoR	5038	23.15%	64.68
	SyR	10367	47.63%	68.97

In SyR, the development of two motifs exhibiting vertical, horizontal or rotational symmetry constitutes a symmetric repetition. Inversion is an extreme case in horizontal symmetry where ascending developments are made to descend by the same degree [19]. For example, if the original motif is “E4-D4-E4-G4” (Down-Up-Up), then horizontal symmetry “E4-F4-E4-C4” (Up-Down-Down), vertical symmetry “E4-F4-G4-F4” (Up-Up-Down) and rotational symmetry “E4-D4-C4-A4” (Down-Down-Up) are symmetric repetition. In addition, in SuR, HoR, and SyR, we set the similarity of two pitch sequences or development sequences as 75 %. We omit motifs that satisfy both HoR and SyR (e.g., “C4-D4-E4” and “C4-E4-G4”), as this situation might lead to overlap between repetition types and is not prevalent in this dataset (less than 1%). However, this type of repetition presents an intriguing topic for further analysis. Figure 3 illustrates some examples of five repetition types in MRD. First, they are exclusive based on the definition in Table 1. Each motif is categorized into one repetition type. Second, the size of five repetition types is different. StR has the smallest size since the definition of StR is the most strict that all note pitches should be identical. SyR has the largest size since it measures the development of the music and contains three symmetric scenarios.

To retain most information from the motifs, we set the longest motif sequence in the dataset as the default length (i.e. 120 tokens). If the sequence is shorter than the max length, then we pad zeros to the sequence. In addition, we randomly hold out motifs from 100 songs for testing and use the remaining motifs to train R-Transformer. The basic statistics of five repetition types appear in Tables 2.

4.2 Implementation Details

Following the backbone model of [13], we also choose the linear Transformer [18] for sequence modeling. The Transformer encoder

module consists of 6 self-attention layers with 4 heads and 256 hidden states. The inner layer size of the feed-forward part is set to 2048. The symbolic music is converted to a sequence of compound words drawn from a pre-defined music attribute vocabulary [13]. Musical information is explained based on seven attributes, tempo, chord, position, type, pitch, duration, and velocity. The embedding sizes of these seven attributes and label are 128, 256, 64, 32, 512, 128, 128, and 32, respectively. The embedding sizes are chosen based on the vocabulary sizes of different token types, and embedded tokens describing the same object are concatenated together and linearly projected to the same size of the corresponding module’s hidden state. We use the Adam optimizer [20] with a learning rate of 10^{-4} for the reconstruction and classification model. We apply dropout with a rate of 0.1 during training. In our model, repetition importance factors are designed as follows: in all types, γ of pitch is 4 since pitch contributes more to the repetition type. In SuR, HoR and SyR, γ of position, duration and velocity is 2 since these contributes also determines the representation of notes. For other attributes, γ is set as 1. λ in Equation 8 is 0.5.

5 EXPERIMENT

5.1 Model Evaluation and Comparison

In both objective and subjective experiments, five relative models are adopted for comparison: **CP-C** and **CP-NC** are two versions from the state-of-the-art music generation model [13]. **CP-C** generates the new music by taking one motif as the input condition. **CP-NC** generates the music without any conditions and the matching rate is calculated based on the first generated motif. **PopMNet**, **Theme** and **MELONS** generate music by studying the repetitions and variance in the music structure. **PopMNet** uses a combination of recurrent neural network (RNN) and CNN models. **Theme** uses Transformer-based models. **MELONS** uses Transformer- and graph-based models.

In the objective evaluation, we design two additional baseline models **R-Transformer-V** and **R-Transformer-R** to validate the effect of model design. **R-Transformer-V** is a vanilla R-Transformer model that does not apply the repetition learning matrix or rule-based generation. **R-Transformer-R** is a vanilla R-Transformer model that does not apply rule-based generation but applies the repetition learning matrix. In addition, **R-Transformer-RR** is our R-Transformer model that applies both the repetition learning matrix and rule-based generation. We let each model generate motif-level results using the given motif condition retrieved from 100 test samples. Specifically, 1-bar polyphonic piano music is generated based on the original motif. We evaluate the result using the matching rate \mathcal{M} between output and input based on the definitions of different repetition types,

$$\mathcal{M} = \frac{\# \text{ of matched motifs}}{\# \text{ of total motifs}}$$

When two motifs satisfy the definition in Table 1, we consider that a match; otherwise, the motifs do not match.

It should be noted that the goal of the proposed model is different from other models since this work tries to generate music by modeling music repetitions while other models do not necessarily generate different music repetitions since they are not designed to

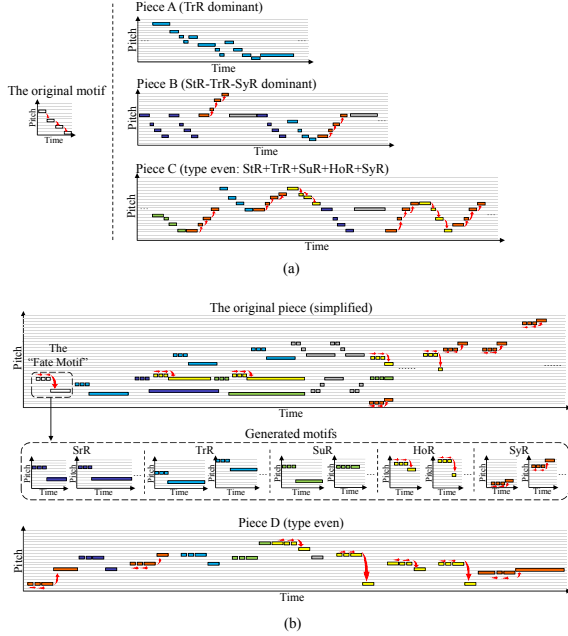


Figure 4: Examples of generated results. (a) The example is generated from the test dataset. (b) The example is generated from the “fate motif”. Purple: strict repetition (StR). Blue: transpositional repetition (TrR). Yellow: subsequential repetition (SuR). Green: homodirectional repetition (HoR). Orange: symmetric repetition (SyR). Gray: ornamentation note³. Melodies are illustrated.

learn repetitions. Thus, to have a fair comparison, in the subjective evaluation, we report the overall music quality with other models in the paper since the ultimate goal of our model and other music generation models is generating enjoyable music.

5.2 Analysis of Generated Examples

To validate the proposed model can not only generate diverse and beautiful repetitions, but also achieve pleasant and complex melodies, we demonstrate two examples that is displayed in Figure 4. In the first example, the original motif is made up of four consecutive notes (G#-F-D#-C#) with a downward melodic movement. Piece A is a TrR dominant version, which delivers a feeling of “pacing back and forth” or “jumping over and over”. Although the intervals within each motif are the same, TrR expands the pitch distribution of the melody, causing the oscillation of same pattern at different pitch levels. Piece B is a StR-TrR-SyR dominant version, which creates melodic waves and emotional ups and downs. The introduction of StR helps to strengthen the original pattern, while SyR brings a distinct melodic direction and richer intervals. Therefore, this piece is no longer a one-way repetition with the descending direction, but intersperses with a number of upward trends. Piece C is a type even version that applies all five repetition types, which is more coherent and exquisite in melodic and emotional changes. SuR offers slight changes to the original motif, bringing familiarity and freshness. HoR contains a variety of motifs with the same direction

³Ornamentation note: the addition of notes for expressive and aesthetic purposes.

of movement and shifting intervals, making the music unrestrained and diverse. Through exquisite combinations, the music expresses complex consciousness and emotions.

The second example is generated based on the “fate motif”. By comparing similarities and differences between the generated motifs and motifs in Beethoven’s Fifth Symphony, we can further verify that the rich motif variants can form diverse and beautiful musical pieces with high quality. First, the model can generate all kinds of variants of the “fate motif” that appear in the original piece. Motifs in the middle row of Figure 4 (b) mainly appear in the early stage of the first movement, which is the most recognizable part of Beethoven’s Fifth Symphony. The appearance of these motifs lays the emotional tone and presents the dominant theme for the entire first movement and even the whole symphony. Therefore, by selecting and combining the generated motifs, we can reproduce a piece of music that is similar to the original work, and even make the generated music express similar feelings. Second, Piece D is a type-even version generated from the “fate motif” that delivers distinct feelings compared to the original Beethoven’s Fifth Symphony. Overall, Piece D presents a relaxed, leisurely and humorous feeling. Specifically, some adjacent motifs with opposite directions of movement naturally form some pairs, creating a feeling of “walking together” or “asking and answering”. The development of the whole music is gentle without huge jumps and turns. All the differences are generated from one motif, but transfer the Beethoven’s Fifth Symphony from tragic and passionate to relaxed and joyful, which further confirm the richness and beauty of repetition. Listening samples can be found in supplementary materials.

5.3 Objective Evaluation

Table 3 lists the matching rate of five music generation models and three variants of the proposed model when generating one bar. Overall, the proposed model produces a good performance when generating different repetition types.

Upon comparing the results of the proposed model with other music generation models in terms of repetition generation, the proposed model outperforms all other models in all repetition types since the proposed model is designed to recognize different repetitions. In addition, HoR and SyR are higher than StR, TrR and SuR in other models. The reason is that StR, TrR and SuR have a more strict definition and these models tend to learn those that appear the most or the easiest to learn since there is no supervisory signal for repetitions. This result also confirms the deficiency of current models.

It is also noted that R-Transformer-R is more robust than R-Transformer-V after applying the repetition learning matrix. It is reasonable since repetition learning matrix tells the model which attributes are more important to a specific repetition type during training. A comparison of R-Transformer-RR with R-Transformer-R reveals that StR and TrR achieve a matching rate of 1. StR and TrR follow explicit rules based on the definition; intuitively, implementing rules to generate pitch will lead to a matching rate of 1.

Table 3: Objective evaluation results in terms of the matching rate.

Models	StR	TrR	SuR	HoR	SyR
CP-C [13]	0.00± 0.00	0.04± 0.08	0.21± 0.41	0.68±0.47	0.71 ±0.46
CP-NC [13]	0.01± 0.07	0.11± 0.31	0.36± 0.48	0.62±0.49	0.64 ±0.48
PopMNet [33]	0.00± 0.00	0.00± 0.00	0.10± 0.30	0.60±0.49	0.70 ±0.45
Theme [30]	0.00± 0.00	0.00± 0.00	0.48± 0.18	0.47±0.25	0.47 ±0.25
MELONS [37]	0.20± 0.40	0.20± 0.40	0.40± 0.49	0.70±0.46	0.70 ±0.46
R-Transformer-V	0.54± 0.49	0.79 ±0.41	0.74± 0.44	0.85±0.36	0.74 ±0.42
R-Transformer-R	0.84 ±0.36	0.96 ±0.18	0.75± 0.43	0.95 ±0.17	0.75 ±0.44
R-Transformer-RR	1.00 ±0.00	1.00 ±0.00	0.75± 0.43	0.97± 0.17	0.75± 0.43

5.4 Subjective Evaluation

To validate that different repetitions can actually create pleasant music, we also design a subjective evaluation on Chinese social media. 150 subjects were randomly recruited on the internet. 27 reports were detected invalid because of some reasons such as submission too fast, no response, random answer, etc. The remaining 123 reports were used for analysis. None of the subjects has any prior knowledge of our study. Among the 123 participants, 57 were men and 66 were women; 65 were under age 25, 46 were between 25 and 50, and 12 were older than 50. In addition, 44 had musical training and were familiar with music theory (labeled “pro”); the others possessed no musical training and were labeled “non-pro”.

In the experiment, we let subjects listen to seven sets of music: six sets of music are generated by models and one by human composers, which is the original music from the test dataset based on the selected motif. In all sets, subjects are asked to listen to two pieces of music, namely an original motif and a piece comprising several motifs. In R-Transformer set, the music piece is combined with different types of repetitions generated by the model. Subjects randomly select the music in each set; none indicated having heard the original motif before the test. During the experiment, each subject is asked to rate the overall quality on a 5-point scale (from 1 to 5; the higher the better) after each piece of music.

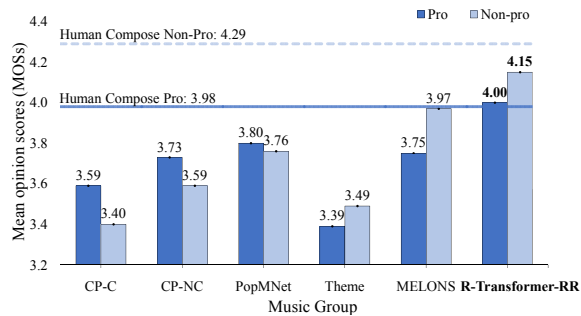
**Figure 5: Subjective evaluation in terms of quality.**

Figure 5 shows the mean opinion scores (MOSs) on the overall quality in the “Pro” and the “Non-pro” group. The results demonstrate the effectiveness of the proposed method. In terms of overall quality, the proposed model outperforms other models in both groups, indicating that the proposed model can generate pleasant music based on different repetition types. Surprisingly, the proposed model performs better than human-composed music in the

“Pro” group, verifying our model’s authenticity. These strengths likely emerged for two reasons. First, by combining different repetition types, the results of R-Transformer sound vivid and structural because the entire music piece is developed from one motif. Second, repetition is confirmed to be an important factor that makes the music pleasant.

Table 4 shows a paired t-test of all 123 subjects between the human-composed music and the machine-composed music, which is used to evaluate whether differences exist between two variables for the same subject. There is no significant difference between the music generated by the proposed model and the original human-composed music ($p=0.351$), while there is a significant difference when comparing results generated by other models and original human-composed music ($p<0.05$), indicating a better music quality of the proposed model.

Table 4: Paired t-test results between machine-composed music and human-composed music.

Music Group	Mean	Std	T-test
CP-C [13]	3.47	1.15	$t=-6.01, p<0.001$
CP-NC [13]	3.64	0.96	$t=-4.74, p<0.001$
PopMNet [33]	3.77	0.95	$t=-4.41, p<0.001$
Theme [30]	3.46	0.91	$t=-7.37, p<0.001$
MELONS [37]	3.89	1.85	$t=-3.43, p=0.001$
The proposed	4.10	0.95	$t=-0.94, p=0.351$
Human-composed	4.18	0.84	-

6 CONCLUSION

To the best of our knowledge, this is the first comprehensive study of repetition in machine composition. We construct a dataset named MRD based on the Pop piano dataset [13]. 562,563 training data and 21,766 test data are provided with the labels of motif level repetitions. A novel repetition generator is designed based on the transformer encoder and a repetition-aware learner. The proposed R-Transformer can generate a large amount of motif-level repetition of different types. Moreover, the learned composition skills on Pop music dataset also demonstrate very interesting results on fate motif from the classical music. Subjective evaluation on both musicians and non-professional users show that the proposed techniques help to improve the quality of the machine composed music obviously.

7 ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their constructive comments. The work is supported by The Hong Kong Polytechnic University for the project: Artificial Intelligence and Robotics (AIR) Group: Artificial Intelligence Art under Grant No.: P0033738.

REFERENCES

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).
- [2] Gong Chen, Yan Liu, Sheng-hua Zhong, and Xiang Zhang. 2018. Musicality-novelty generative adversarial nets for algorithmic composition. In *Proceedings of the 26th ACM international conference on Multimedia*. 1607–1615.
- [3] Tom Collins and Robin Laney. 2017. Computer-generated stylistic compositions with long-term repetitive and phrasal structure. *Journal of Creative Music Systems* (2017).
- [4] Shuqi Dai, Zeyu Jin, Celso Gomes, and Roger B Dannenberg. 2021. Controllable deep melody generation via hierarchical music structure representation. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*.
- [5] Shuqi Dai, Xichu Ma, Ye Wang, and Roger B Dannenberg. 2021. Personalized popular music generation using imitation and structure. *arXiv preprint arXiv:2105.04709* (2021).
- [6] Shuqi Dai, Huan Zhang, and Roger B Dannenberg. 2020. Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. *arXiv preprint arXiv:2010.07518* (2020).
- [7] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [8] Anders Eklöwsson and Anders Friberg. 2012. 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*. 276–285.
- [9] Jeff Ens and Philippe Pasquier. 2020. Mmm: Exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048* (2020).
- [10] Robert Fink. 2005. *Repeating ourselves*. University of California Press.
- [11] Carlos Hernandez-Olivan and Jose R Beltran. 2021. Music Composition with Deep Learning: A Review. *arXiv preprint arXiv:2108.12290* (2021).
- [12] Dorien Herremans and Elaine Chew. 2017. MorpheuS: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing* 10, 4 (2017), 510–523.
- [13] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. 2021. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 178–186.
- [14] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinulescu, and Douglas Eck. 2018. Music transformer. *arXiv preprint arXiv:1809.04281* (2018).
- [15] Yu-Siang Huang and Yi-Hsuan Yang. 2020. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1180–1188.
- [16] Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2019. Modeling self-repetition in music generation using generative adversarial networks. In *Machine Learning for Music Discovery Workshop, ICML*.
- [17] Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. 2020. 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)*. 516–520.
- [18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *The 37th International Conference on Machine Learning Conference*. 5156–5165.
- [19] Davorin Kempf. 1996. What is symmetry in music? *International Review of the Aesthetics and Sociology of Music* (1996), 155–165.
- [20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [21] Gerhard R Koch. 1984. Reich's 'The Desert Music'. *Tempo* 149 (1984), 44–46.
- [22] Sanidhya Mangal, Rahul Modak, and Poorva Joshi. 2019. Lstm based music generation system. *arXiv preprint arXiv:1908.01080* (2019).
- [23] Gabriele Medea, Srikanth Cherla, Katerina Kosta, Matt McVicar, Samer Abdallah, Marco Selvi, Ed Newton-Rex, and Kevin Webster. 2018. StructureNet: Inducing Structure in Generated Melodies. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*. 725–731.
- [24] Aashiq Muhamed, Liang Li, Xingjian Shi, Suri Yaddanapudi, Wayne Chi, Dylan Jackson, Rahul Suresh, Zachary C Lipton, and Alexander J Smola. 2021. Symbolic music generation with Transformer-GANs. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 408–417.
- [25] Jean-Jacques Nattiez. 1990. *Music and discourse: Toward a semiology of music*. Princeton University Press.
- [26] Adam Ockelford. 2017. *Repetition in music: Theoretical and metatheoretical perspectives*. Routledge.
- [27] Gabriel Pareyon. 2011. *On Musical Self-Similarity: Intersemiosis as synecdoche and analogy*. Gabriel Pareyon.

- [28] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Popmag: Pop music accompaniment generation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1198–1206.
- [29] Walter Schulze and Brink Van Der Merwe. 2011. Music generation with Markov models. *IEEE MultiMedia* 18, 03 (2011), 78–85.
- [30] Yi-Jen Shih, Shih-Lun Wu, Frank Zalkow, Meinard Muller, and Yi-Hsuan Yang. 2022. Theme Transformer: Symbolic Music Generation with Theme-Conditioned Transformer. *IEEE Transactions on Multimedia* (2022), 1–1.
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* (2017).
- [32] Jian Wu, Changran Hu, Yulong Wang, Xiaolin Hu, and Jun Zhu. 2019. A hierarchical recurrent neural network for symbolic melody generation. *IEEE transactions on cybernetics* (2019), 2749–2757.
- [33] Jian Wu, Xiaoguang Liu, Xiaolin Hu, and Jun Zhu. 2020. PopMNet: Generating structured pop music melodies using neural networks. *Artificial Intelligence* 286 (2020), 103303.
- [34] Shih-Lun Wu and Yi-Hsuan Yang. 2020. The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures. *arXiv preprint arXiv:2008.01307* (2020).
- [35] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. 2017. MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. 324–331.
- [36] Ning Zhang. 2020. Learning adversarial transformer for symbolic music generation. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
- [37] Yi Zou, Pei Zou, Yi Zhao, Kaixiang Zhang, Ran Zhang, and Xiaorui Wang. 2022. 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)*. IEEE, 191–195.