*mathematics*

MDPI

*Article*

# MRBERT: Pre-Training of Melody and Rhythm for Automatic Music Generation

**Shuyu Li [1] and Yunsick Sung [2,*]**

[1] Department of Multimedia Engineering, Graduate School, Dongguk University–Seoul, Seoul 04620, Republic of Korea
[2] Department of Multimedia Engineering, Dongguk University–Seoul, Seoul 04620, Republic of Korea
* Correspondence: sung@dongguk.edu; Tel.: +82-2-2260-3338

**Abstract:** Deep learning technology has been extensively studied for its potential in music, notably for creative music generation research. Traditional music generation approaches based on recurrent neural networks cannot provide satisfactory long-distance dependencies. These approaches are typically designed for specific tasks, such as melody and chord generation, and cannot generate diverse music simultaneously. Pre-training is used in natural language processing to accomplish various tasks and overcome the limitation of long-distance dependencies. However, pre-training is not yet widely used in automatic music generation. Because of the differences in the attributes of language and music, traditional pre-trained models utilized in language modeling cannot be directly applied to music fields. This paper proposes a pre-trained model, MRBERT, for multitask-based music generation to learn melody and rhythm representation. The pre-trained model can be applied to music generation applications such as web-based music composers that includes the functions of melody and rhythm generation, modification, completion, and chord matching after being fine-tuned. The results of ablation experiments performed on the proposed model revealed that under the evaluation metrics of HITS@k, the pre-trained MRBERT considerably improved the performance of the generation tasks by 0.09–13.10% and 0.02–7.37%, compared to the usage of RNNs and the original BERT, respectively.

**Keywords:** automatic music generation; generative pre-training; embedding; representation learning

**MSC:** 68T99



**Citation:** Li, S.; Sung, Y. MRBERT: Pre-Training of Melody and Rhythm for Automatic Music Generation. *Mathematics* **2023**, *11*, 798. https://doi.org/10.3390/math11040798

Academic Editor: Ioannis G. Tsoulos

Received: 26 December 2022
Revised: 11 January 2023
Accepted: 1 February 2023
Published: 4 February 2023



## 1. Introduction

In the past decade, artificial intelligence has made breakthroughs due to the introduction of deep learning, which allows the use of various artificial intelligence models in different fields. Representation learning has been in the spotlight because it significantly reduces the amount of data required to train a model through semi-supervised and self-supervised learning, and, more importantly, it overcomes the limitations of traditional supervised learning that requires annotated training data. Representation learning has achieved excellent results in computer vision [1], natural language processing [2], and music generation [3,4].

Deep learning-based music technology has been extensively studied for its potential in music. This includes music generation [3,4], music classification [5,6], melody recognition [7,8], and music evaluation [9,10]. These functions rely on learning and summarizing knowledge from music corpus, rather than obtaining it from music theory. Among them, music generation research is notable because it involves performing a creative task. Music generation tasks can be categorized into three categories, namely autoregressive [11], conditional [12], and sequence-to-sequence (Seq2Seq) generation [13]. In autoregressive generation, the current value is predicted based on the information from previous values. For music, each predicted note becomes a consideration when predicting the following

notes, and a piece of music can be generated by looping this process. In conditional generation, contextual information is used to predict the missing value. When predicting the missing values in random positions of music, contextual information from both left and right directions should be considered. Thus, music completion can be realized. In Seq2Seq generation, a novel sequence based on the given sequence is generated. Seq2Seq generation involves two processes: understanding the given sequence and then generating a new sequence subsequently using the understood content. Seq2Seq generation can be applied in music to generate matching chords based on a given melody.

The above-mentioned traditional music generation models are typically designed to accomplish only one of the aforementioned three categories and cannot be generalized to other tasks. Inspired by natural language modeling, music generation requires a model that can be applied to multitasking without requiring large training resources [2]. Bidirectional encoder representations from transformers (BERT) [14] is a language representation model in natural language modeling that is used to pre-train deep directional representations from unlabeled text by jointly conditioning on both left and right contextual information in all layers. The pre-trained model can be fine-tuned with only an additional output layer to create state-of-the-art models for numerous tasks without substantial task-specific architecture modifications. Therefore, this paper will also focus on the application of representation models in music generation.

Compared to traditional music generation models, pre-trained model-based automatic music generation models exhibit several advantages. First, pre-trained models can learn better representations of music than traditional music generation models. Traditional music generation models utilize PianoRoll [15] as the representation, which is similar to one-hot encoding. Therefore, PianoRoll exhibits the same sparse matrix problem as one-hot encoding, and contextual information is ignored. However, music in the pre-trained model is mapped into n-dimensional spaces, which is a non-sparse representation by considering the contextual information from two directions [14]. Second, pre-trained models can handle long-distance dependencies. Traditional models [16–18] of music generation typically utilize recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) and gate recurrent unit (GRU), to generate music because of their ability to memorize temporal information. However, RNNs exhibit vanishing gradients caused by backpropagation through time (BPTT) and cannot handle long-distance dependences. Although LSTM and GRU alleviate the long-distance dependency problem by adding memory cells and gates, their effect is limited because of BPTT [19]. BERT, based on the multihead attention mechanism, can link long-distance notes and consider global features [20]. Finally, pre-trained models can process data in parallel, whereas RNN-like models run recurrently, which not only causes vanishing gradients but also wastes computing resources. Because the transformers in BERT run in parallel mode, all tokens in the sequence are embedded into them without waiting for the data of the previous time step to be processed [20]. However, applying traditional natural language pre-trained models directly for music representation learning cannot provide the desired results. The problem is that there is no concept of rhythm in natural language, but the rhythm is as important as the melody in music. Therefore, an approach for learning musical representation that takes into account both the melody and rhythm is needed for use in music generation.

In this paper, a modification of BERT, namely MRBERT, is proposed for the pre-training of the melody and rhythm for fine-tuning music generation. In pre-training, the melody and rhythm are embedded separately. For exchanging the information of the melody and rhythm, semi-cross attention instead of merging, as performed in traditional methods, is used, which prevents features loss. In fine-tuning, the following three generation tasks are designed: autoregressive, conditional, and Seq2Seq. Thus, a pre-trained model is fine-tuned with the output layers corresponding to the three types of generation tasks to realize multitask music generation.

The contributions of this paper are as follows: (1) A novel generative pre-trained model based on melody and rhythm, namely MRBERT, is proposed for multitask music

generation, including autoregressive and conditional generation, as well as Seq2Seq generation. (2) In pre-training for representation learning, the melody and rhythm are considered separately, based on the assumption that they have strong dependencies on themselves and weak dependencies between each other. Experimental results have also shown that this assumption is reasonable and can be widely applied to related research. (3) The proposed MRBERT with three generation tasks allows users to generate melodies and rhythms from scratch through interaction with the user, or to modify or complete existing melodies and rhythms, or even to generate matching chords based on existing melodies and rhythms.

## 2. Related Work

This section describes BERT [14] first as a well-known representation learning model and then two music representation learning studies, MusicBERT [21] and MidiBERT [22], based on BERT are introduced.

BERT is a language representation model that is designed to learn deep bidirectional representations from unlabeled text. It did this by conditioning on both the left and right context in all layers of the model. BERT is able to achieve state-of-the-art results on a wide range of natural language processing tasks, including question answering and language inference, by being fine-tuned with only one additional output layer. It has been shown to perform particularly well on a number of benchmarks, including the GLUE benchmark, the MultiNLI dataset, and the SQuAD question answering dataset. The main contribution of BERT is that it proves the importance of bidirectional pre-training for representation learning. Unlike previous language modeling approaches that used a unidirectional language model for pre-training [2] and used a shallow concatenation of independently trained left-to-right and right-to-left language modeling (LM) [23], BERT used a masked language model (MLM) to enable pre-trained deep bidirectional representations.

Due to BERT's success in natural language processing tasks, researchers have started to apply representation learning to music data. Two representative studies in this area are MusicBERT and MidiBERT.

MusicBERT is a large-scale pre-trained model for music understanding and consists of large symbolic music corpus containing more than 1 million pieces of music and songs. MusicBERT designed several mechanisms, including OctupleMIDI encoding and a bar-level masking strategy, to enhance the pre-training of symbolic music data. Furthermore, four music understanding-based tasks were designed, two of which were generation tasks, melody completion and accompaniment suggestion; the other two were classification tasks, genre and style classification.

MidiBERT used a smaller corpus than MusicBERT and focused on piano music. For the token representation, it used the beat-based revamped MIDI-derived events [24] token representation and borrowed Compound words [25] representation to reduce the length of the token sequences. Furthermore, MidiBERT established a benchmark for symbolic music understanding, including not only note-level tasks, melody extraction, and velocity prediction but also sequence-level tasks, composer classification, and emotion classification.

Unlike these two studies, the proposed MRBERT model is a pre-trained model that can be used for music generation tasks. In the MRBERT, a music corpus called OpenEWLD [26], which is a leadsheet-based corpus that contains the necessary information for music generation, such as the melody, rhythm, and chords, is used. The MRBERT differs from other models in that melody and rhythm are divided into separate token sequences. Additionally, the embedding layer of the traditional BERT and the attention layer in its transformer are modified to better fit the pre-training of the melody and rhythm. Finally, the MRBERT was designed to differentiate from the prediction and classification tasks of traditional methods by using three generation tasks, which are used to evaluate the performance of the pre-trained model for music generation.

## 3. Automatic Music Generation Based on MRBERT

In this paper, the MRBERT is proposed to learn the representations of the melody and rhythm for automatic music generation. First, the token representation is described. The structure and the pre-training of the MRBERT is explained and, finally, the strategies of fine-tuning are described.

### *3.1. Token Representation*

The melody, rhythm, and chords are extracted from OpenEWLD [26] music corpus for pre-training and fine-tuning. The OpenEWLD music corpus consists of songs in the leadsheet, as displayed in Figure 1A. In Figure 1B, the leadsheet is converted from MusicXML to events through Python library music21. Figure 1C reveals that events include Instruments, Keys, Timesignatures, Measures, ChordSymbols, and Notes, where only information related to the melody, rhythm, and chords are extracted. For example, "G4(2/4)" indicates that the pitch of the note is G in the fourth octave, and the duration of the note is 2/4. The next step is to separate the melody and rhythm sequences, as displayed in Figure 1D. The chord sequences are extracted from ChordSymbols to prepare for the Seq2Seq generation task in the fine-tuning, as presented in Figure 1E. For example, "C" represents the chord that continues with the melody until the next chord occurs.

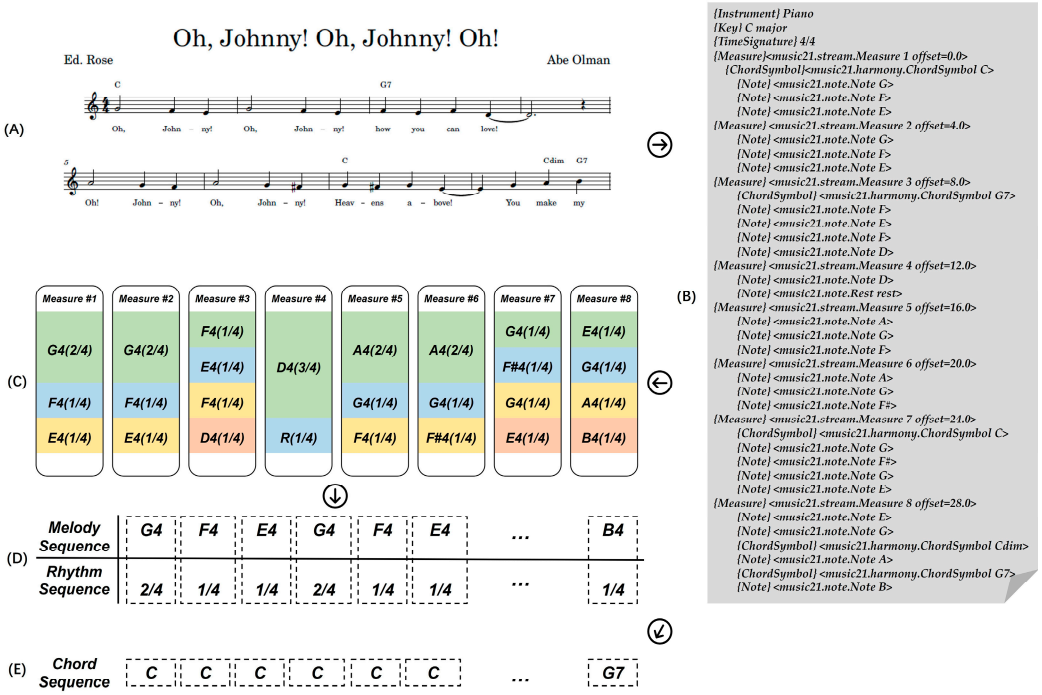


**Figure 1.** Pipeline of token representation. (**A**) A example leadsheet in music corpus; (**B**) The events converted from MusicXML; (**C**) Extracted events related to the melody, rhythm and chords; (**D**) Generated melody sequence and rhythm sequence; (**E**) Generated chord sequence.

### *3.2. Pre-Training of MRBERT*

The MRBERT is a pre-trained model for the learning representations of the melody and rhythm. As displayed in Figure 2, the melody ($m_1$, $m_2$, __, ..., $m_n$) and rhythm ($r_1$, $r_2$, __, ..., $r_n$) sequences are input to the embedding layers, where the "__" represents the random masked tokens. The tokens of the melody sequences and rhythm sequences are embedded by the corresponding token embedding layer. The position embedding layer, which is shared by the melody and rhythm, adds the position feature on them. Through the embedding layers, the melody embedding $e^M$ and the rhythm embedding $e^R$ are obtained. Next, $e^M$ and $e^R$ are input to the corresponding transformer, which exchanges information through semi-cross attention. Semi-cross attention is proposed to realize the information

exchange between the melody and rhythm. As presented in formula (1), the cross query of $e^M$ is obtained from the dot-production of the query of the melody $q^M$ with the activated query of the rhythm $q^R$ by using softmax. The use of the key $k^M$ and value $v^M$ is similar to that of the self-attention. For the rhythm, the query of the melody $q^M$ is required for calculating the cross query of $e^R$. Finally, the melody hidden states $h^M$ and rhythm hidden states $h^R$ output by the transformers are passed through the melody prediction layer and rhythm prediction layer to predict the masked melody $m'$ and rhythm $r'$.

$$\begin{gathered} Semi\ Cross\ Attention^M = softmax\left(\frac{q^M \cdot (softmax(q^R))k^{MT}}{\sqrt{d_k}}\right)v^M \\ and \\ Semi\ Cross\ Attention^R = softmax\left(\frac{q^R \cdot (softmax(q^M))k^{RT}}{\sqrt{d_k}}\right)v^R \end{gathered} \tag{1}$$

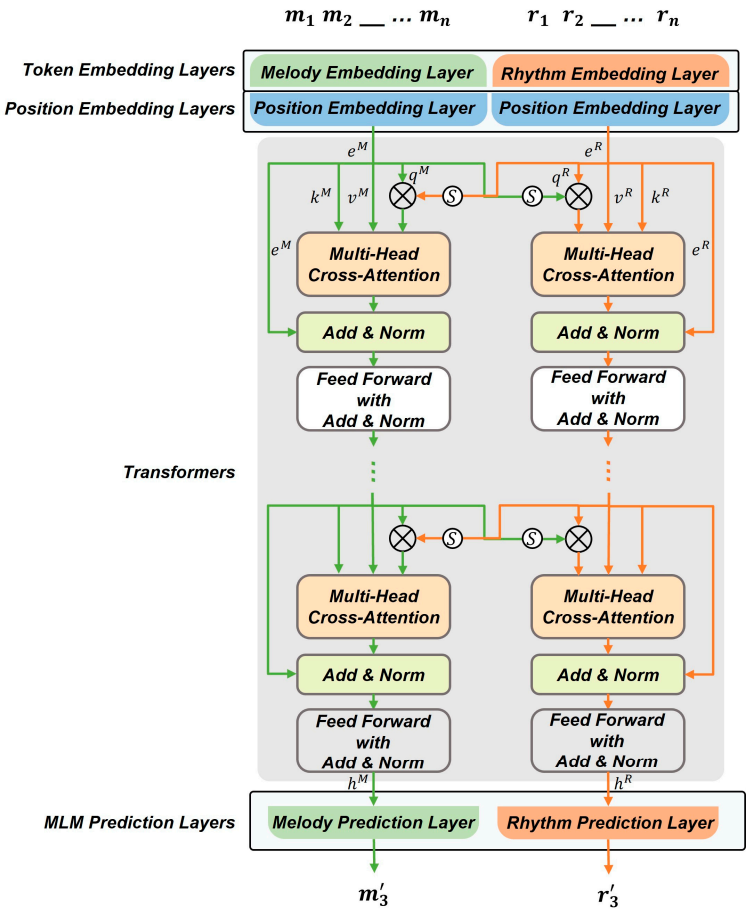


**Figure 2.** Pipeline of pre-training of MRBERT.

The pre-training strategy of this paper refers to the MLM proposed by BERT, which follows that 15% of the tokens in the sequence are randomly masked: (1) 80% of the selected tokens are replaced by MASK; (2) 10% are replaced by randomly selected tokens; (3) the remaining 10% remain unchanged. Furthermore, to enhance the performance of the pre-training, this paper refers to BERT-like models and other related studies, drops the next sentence prediction pre-training task, and uses dynamic masking [27].

### *3.3. Fine-Tuning of Generation Tasks*

To address the diverse generation tasks, the MRBERT is fine-tuned with three downstream tasks, namely autoregressive, conditional, and Seq2Seq generation. Furthermore, after fine-tuning for each task, joint generation can be achieved by executing the three generation methods simultaneously.

#### 3.3.1. Autoregressive Generation Task

To accomplish the autoregressive generation task, its generation pattern should be known, which can be summarized as a unidirectional generation similar to a Markov chain [28] $P(t_i|t_1, t_2, t_3, \ldots, t_{i-1})$, where the probability of the token $t_i$ depends on $t_1$ to $t_{i-1}$. Autoregressive generation reveals that the tokens are predicted in order from left to right, and the current token is predicted based on the previous tokens. First, <BOS> (the beginning of the sequence, which is a special token in vocabulary) is passed into the MRBERT. Next, the output layers, which are a pair of fully connected layers, predict the melody and rhythm based on the hidden state from the MRBERT. Finally, the predicted melody and rhythm are used to calculate the cross-entropy loss for backpropagation. When backpropagation ends, the input token sequences are incremented by one time step, and the model predicts the melody and rhythm of the next time step until <EOS> (the end of the sequence, a special token corresponding to <BOS>) is generated. The ground truth label data are easily obtained by shifting the input sequences to the right by one time step. The pre-trained model and output layer continuously shorten the gap between the prediction and the label data through fine-tuning. After fine-tuning, whenever the melody and rhythm are generated, generations are added to the end of the sequence to form a new input, as displayed in Figure 3.

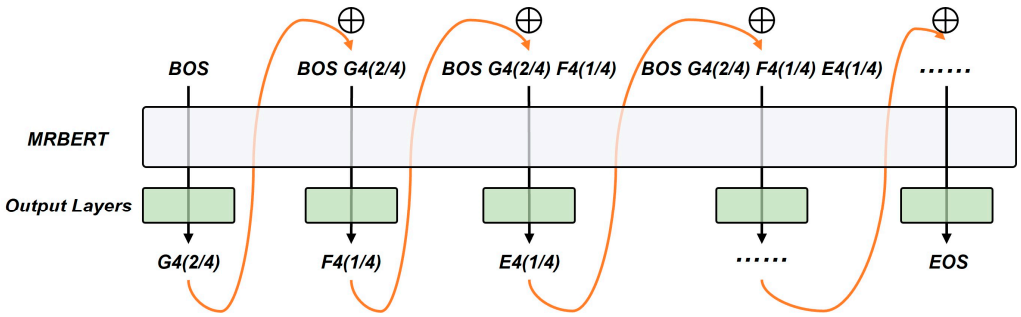


**Figure 3.** Pipeline of autoregressive generation. The orange arrows represent the predicted melody and rhythm should be continuously added to the end of the input.

#### 3.3.2. Conditional Generation Task

Unlike in autoregressive generation, in conditional generation, not only previous tokens but also future tokens are considered when predicting unknown tokens. The model should consider the bidirectional contextual information of the unknown tokens. To realize this task, a generation pattern such as a denoising autoencoder [29] is used, $P(t_j|t_1, t_2, \ldots, t_{j-1}, t_{j+1}, \ldots, t_i)$, where the unknown token $t_j$ should be predicted based on the known tokens. Fine-tuning for conditional generation is highly similar to pre-training. However, since multiple tokens are masked, when predicting one of the tokens, it is assumed to be independent of the other masked tokens. To address this problem, shorter sequences are used and only a pair of melody and rhythm tokens is masked in fine-tuning. The cross-entropy loss is calculated by the predictions (melody or rhythm) and ground truth labels, which are then used for fine-tuning. After fine-tuning, the MRBERT and the output layer of the conditional generation fill in the missing parts according to the contextual information obtained from the given melody and rhythm as displayed in Figure 4.

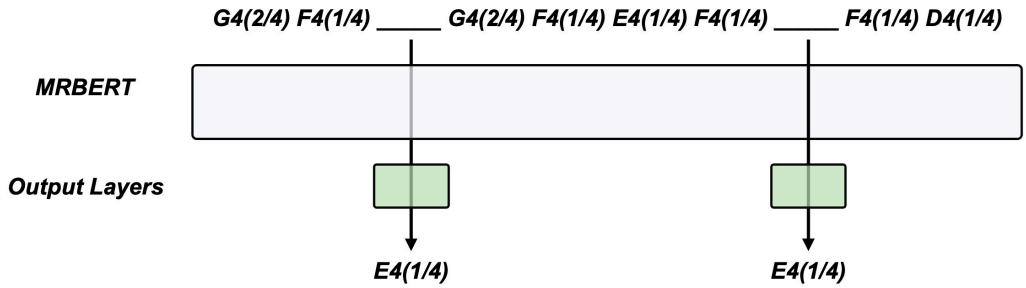


**Figure 4.** Pipeline of conditional generation. The underline represents the missing part of the music.

#### 3.3.3. Seq2Seq Generation Task

When the melody and rhythm are created, chords should be added to make it sound less monotonous. This generation pattern can be summarized as $\mathrm{P}(t_1, t_2, \ldots, t_i \mid t'_1, t'_2, \ldots, t'_i)$, where $t'$ represents the given tokens, and $t$ represents the tokens that should be predicted. The probability of $t$ for the position 1 to $i$ is based on the given $t'$ of 1 to $i$. In fine-tuning, the melody and rhythm sequences are input into the MRBERT, and the chords of the corresponding position are predicted by the output layer of the Seq2Seq generation. The cross-entropy loss calculated from the predicted chords and ground truth label data is used for fine-tuning. After fine-tuning, the MRBERT can accept the melody and rhythm, and subsequently generate chords through the output layer of the Seq2Seq generation, as displayed in Figure 5. The continuous output of the same chord symbol indicates that the same chord is continuing until a different symbol appears.

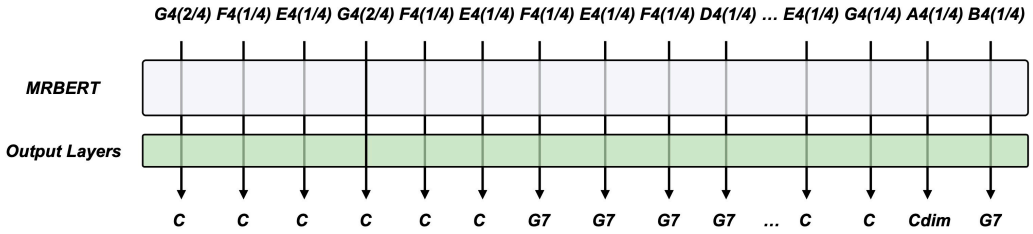


**Figure 5.** Pipeline of Seq2Seq generation. Melody and rhythm can be of any length, and the length of the generated chords vary accordingly.

#### 3.3.4. Joint Generation

Users can use the MRBERT with three generation tasks interactively, as displayed in Figure 6. A simulated use case reveals how the three generation approaches operate simultaneously. First, the melody and rhythm can be generated under the autoregressive generation task. Next, the user can adjust the tokens in the generated melody and rhythm through conditional generation. Finally, the chords are matched to the generated melody and rhythm through the Seq2Seq generation task.

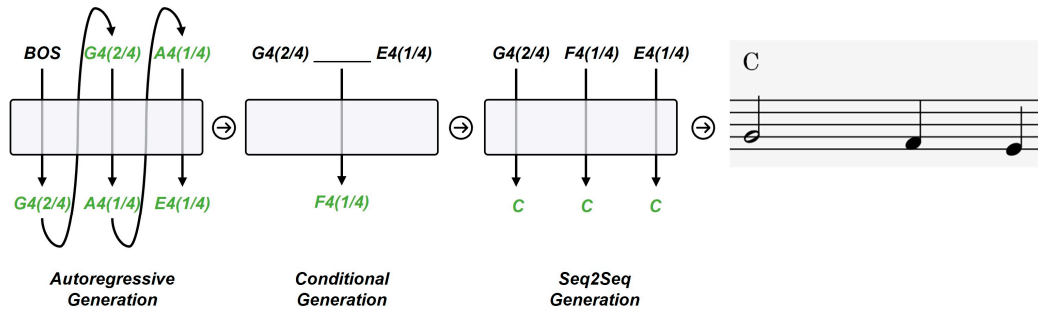


**Figure 6.** Human–interactive use case of automatic music generation.

Among the predictions provided under the aforementioned three tasks, in addition to the prediction with the highest probability, other candidates and their corresponding

probabilities are also given because, in music, a fixed answer rarely exists. Although the high-probability prediction is the most reasonable for analyzing after the model has learned the music corpus, it may not be the most appropriate. Users can choose the candidate they think is the most suitable.

## 4. Experiments

The MRBERT was first trained to convergence through the pre-training task MLM. Next, ablation experiments were conducted on three generation tasks based on the pre-trained MRBERT. BERT, which is a traditional language pre-trained model, was used as the baseline for the ablation experiments.

### *4.1. Dataset*

The EWLD (Enhanced Wikifonia Leadsheet Dataset) is a dataset of music leadsheets containing various metadata about composers, works, lyrics, and features. It is designed specifically for musicological and research purposes. OpenEWLD [26] is a dataset extracted from EWLD, containing only public domain leadsheets, which is used as the dataset for training in this paper. As shown in Figure 1, each leadsheet contains the melody, rhythm, and chords required for training. A total of 502 leadsheets from different composers are included in OpenEWLD, and 90% of these were selected for training, with the remaining 10% used for evaluation.

### *4.2. Experimental Environment*

The ablation experiment includes *w/o cross-attn.* (BERT + separate embedding), which used separate embedding and original self-attention instead of semi-cross attention; *w/o separate embed.* (BERT), that is, the melody and rhythm shared a common embedding layer and only used self-attention (*w/o* means "without"). Furthermore, experimental results on RNNs (and BiRNNs) without any pre-training techniques were also listed to detail the effect of pre-training. HITS@k [21] (k = 1, 3, 5, and 10), which can calculate the proportion of the correct answer included in the k candidates, was used as the evaluation metrics. HITS@k was calculated as shown in formula (2), where n represents the number of samples; $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the rank of the correct answer is less than k, and 0 otherwise.

$$HITS@k = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(rank_i \leq k) \tag{2}$$

Table 1 presents the hyperparameters of the MRBERT (with ablation model) in pre-training and fine-tuning. During pre-training, most of the hyperparameters were set to the same values as those in RoBERTa-base [27], with slight differences in the *Number of Layers*, *Learning Rate Decay*, *Batch Size*, *Max Steps,* and *Warmup Steps*. The *Number of Layers* in the MRBERT was set to $6 \times 2$ because it has two sets of transformer blocks corresponding to the melody and rhythm separately, while ensuring that the number of parameters in the model is on the same level as in the ablation experiments. In terms of the *Learning Rate Decay*, *power* was used rather than linear, that is to make the change in the learning rate smoother and more conducive to convergence. While the settings of the *Batch Size*, *Max Steps,* and *Warmup Steps* were adjusted according to the music corpus used.

**Table 1.** Hyperparameters for pre-training and fine-tuning of MRBERT (with ablation model).

| Parameters | MRBERT | w/o Cross-Attn. | w/o Separate Embed. |
|---|---|---|---|
| [1] Number of Layers | $6 \times 2$ [3] | 12 | 12 |
| Hidden size | 768 | 768 | 768 |
| FFN inner hidden size | 3072 | 3072 | 3072 |
| Attention heads | 12 | 12 | 12 |
| Attention head size | 64 | 64 | 64 |
| Dropout | 0.1 | 0.1 | 0.1 |
| Batch Size | 32 | 32 | 32 |
| Weight Decay | 0.01 | 0.01 | 0.01 |
| Max Steps | 10 k | 10 k | 10 k |
| Warmup Steps | 1 k | 1 k | 1 k |
| Learning Rate Decay | power | power | power |
| Adam $\epsilon$ | $1 \times 10^{-6}$ | $1 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Adam $\beta_1$ | 0.9 | 0.9 | 0.9 |
| Adam $\beta_2$ | 0.98 | 0.98 | 0.98 |
| [2] Melody Vocab Size | 68 + 4 = 72 [4] | 68 + 4 = 72 | - |
| Rhythm Vocab Size | 17 + 4 = 21 | 17 + 4 = 21 | - |
| Melody + Rhythm Vocab Size | - | - | 68 + 17 + 4 = 89 |
| Chord Vocab Size | 795 + 4 = 799 | 795 + 4 = 799 | 795 + 4 = 799 |

[1] Hyperparameters for pre-training. [2] Hyperparameters for fine-tuning. [3] 6 transformer layers of melody and 6 transformer layers of rhythm. [4] 4 represents the number of special tokens: <BOS>, <EOS>, <UNK>, <PAD>.

In fine-tuning, the *Melody Vocab Size*, *Rhythm Vocab Size*, and *Chord Vocab Size* determine the dimension of the probability distribution given by the output layer. The melody and rhythm have 72 and 21 candidates, respectively, which contain four special tokens (<BOS>, <EOS>, <UNK>, <PAD>). In the ablation experiment of *w/o separate embed.*, since the melody and rhythm share an embedding layer, the number of candidates is 89. Furthermore, the number of chord candidates reached 799.

### *4.3. Results of Autoregressive Generation*

When evaluating autoregressive generation, the pre-trained MRBERT with the output layer of the autoregressive generation task predicts the next melody and rhythm at each time step based on the previous. Figure 7 displays the generated melody and rhythm.

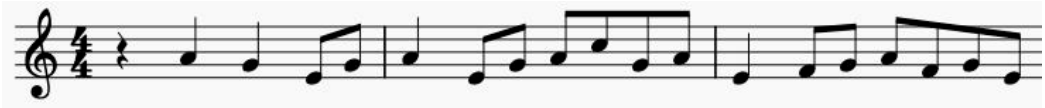

**Figure 7.** Leadsheets of the generated melody sequence.

Table 2 presents the generated melody and rhythm, and the probabilities of the predictions at each time step. The top prediction of the rhythm occupies a higher proportion, whereas the probabilities of all the melody predictions are balanced. The model is more confident in the rhythm prediction. This result is consistent with the analysis results of the music data. Music typically has obvious rhythm patterns, whereas the progression of the melody is complex and changeable.

**Table 2.** Details of autoregressive generation.

| Time Step | Pitch | Probabilities of Melody | | | | Rhythm | Probabilities of Rhythm | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | <BOS> | | | | | <BOS> | | | | |
| 2 | Rest | Rest:0.100 | G4: 0.098 | F4: 0.092 | D4: 0.089 | 1/4 | 1/4: 0.309 | 1/8: 0.263 | 1/2: 0.146 | 1/16: 0.054 |
| 3 | A4 | A4: 0.111 | G4:0.104 | D4: 0.095 | Rest: 0.095 | 1/4 | 1/4: 0.519 | 1/8: 0.205 | 1/2: 0.114 | 3/4: 0.046 |
| 4 | G4 | G4: 0.127 | E4: 0.114 | A4: 0.087 | F4: 0.079 | 1/4 | 1/4: 0.501 | 1/8: 0.202 | 1/4: 0.104 | 3/4: 0.054 |
| 5 | E4 | E4: 0.132 | A4: 0.098 | F4: 0.081 | D4: 0.072 | 1/8 | 1/8: 0.364 | 1/4: 0.364 | 1/2: 0.097 | 3/4: 0.070 |
| 6 | G4 | G4: 0.161 | A4: 0.153 | D4: 0.079 | B4: 0.069 | 1/8 | 1/8: 0.427 | 1/4: 0.356 | 1/2: 0.073 | 3/8: 0.042 |
| 7 | A4 | A4: 0.187 | E4: 0.146 | B4: 0.080 | D4: 0.077 | 1/4 | 1/4: 0.423 | 1/8: 0.398 | 1/2: 0.065 | 3/8: 0.037 |
| 8 | E4 | E4: 0.152 | A4: 0.136 | G4: 0.125 | D4: 0.104 | 1/8 | 1/8: 0.465 | 1/4: 0.308 | 1/2: 0.076 | 3/4: 0.049 |
| 9 | G4 | G4: 0.157 | E4: 0.147 | A4: 0.118 | D4: 0.112 | 1/8 | 1/8: 0.412 | 1/4: 0.313 | 1/2: 0.072 | 3/8: 0.061 |
| 10 | A4 | A4: 0.164 | D4: 0.100 | E4: 0.089 | C5: 0.066 | 1/8 | 1/8: 0.355 | 1/4: 0.344 | 1/2: 0.110 | 3/8: 0.056 |
| 11 | C5 | C5: 0.125 | G4: 0.107 | D4: 0.093 | F4: 0.087 | 1/8 | 1/8: 0.385 | 1/4: 0.370 | 1/2: 0.112 | 3/8: 0.038 |
| 12 | G4 | G4: 0.177 | A4: 0.148 | E4: 0.139 | D4: 0.088 | 1/8 | 1/8: 0.569 | 1/4: 0.267 | 1/2: 0.056 | 3/8: 0.045 |
| 13 | A4 | A4: 0.163 | E4: 0.113 | D4: 0.106 | Rest: 0.086 | 1/8 | 1/8: 0.405 | 1/4: 0.338 | 1/2: 0.071 | 3/8: 0.048 |
| 14 | E4 | E4: 0.131 | A4: 0.108 | F4: 0.085 | D4: 0.074 | 1/4 | 1/4: 0.453 | 1/8: 0.319 | 1/2: 0.082 | 3/8: 0.029 |
| 15 | F4 | F4: 0.148 | A4: 0.102 | G4: 0.090 | C5: 0.086 | 1/8 | 1/8: 0.497 | 1/4: 0.263 | 1/2: 0.075 | 3/4: 0.046 |
| 16 | G4 | G4: 0.212 | A4: 0.142 | E4: 0.116 | D4: 0.088 | 1/8 | 1/8: 0.519 | 1/4: 0.259 | 1/2: 0.082 | 3/8: 0.031 |
| 17 | A4 | A4: 0.156 | E4: 0.116 | D4: 0.088 | F4: 0.076 | 1/8 | 1/8: 0.445 | 1/4: 0.349 | 1/2: 0.056 | 3/8: 0.039 |
| 18 | F4 | F4: 0.144 | E4: 0.104 | G4: 0.087 | C5: 0.079 | 1/8 | 1/8: 0.452 | 1/4: 0.286 | 1/2: 0.093 | 3/8: 0.045 |
| 19 | G4 | G4: 0.148 | A4: 0.134 | E4: 0.103 | D4: 0.099 | 1/8 | 1/8: 0.489 | 1/4: 0.329 | 1/2: 0.065 | 3/8: 0.034 |
| 20 | E4 | E4: 0.139 | A4: 0.120 | C5: 0.093 | F4: 0.077 | 1/8 | 1/8: 0.495 | 1/4: 0.296 | 1/2: 0.082 | 3/8: 0.041 |

Table 3 presents the ablation experimental results of HITS@k in the autoregressive generation task. For the melody prediction, in HITS@k (k = 1, 3, 5, and 10), the MRBERT achieved the average of 51.70%, 2.77% higher than *w/o cross-attn.*, and 3.65% higher than *w/o separated embed.*, and 7.94% higher than the RNN. For the rhythm prediction, it achieved the average of 81.79%, 0.37% higher than *w/o cross-attn.*, and 0.78% higher than *w/o separated embed.*, and 2.56% higher than the RNN.

**Table 3.** Ablation experimental results of the autoregressive generation task.

| Model | HITS@1 (%) | | HITS@3 (%) | | HITS@5 (%) | | HITS@10 (%) | |
|---|---|---|---|---|---|---|---|---|
| | Mel. | Rhy. | Mel. | Rhy. | Mel. | Rhy. | Mel. | Rhy. |
| MRBERT | 15.87 | 51.53 | 42.03 | 83.01 | 61.53 | 92.81 | 87.36 | 99.81 |
| w/o cross-attn. | 14.74 | 51.44 | 38.96 | 82.65 | 57.45 | 91.88 | 84.58 | 99.80 |
| w/o separate embed. | 14.27 | 51.16 | 38.14 | 82.17 | 55.90 | 90.91 | 83.88 | 99.79 |
| RNN | 12.51 | 48.24 | 33.60 | 79.28 | 50.28 | 89.67 | 78.63 | 99.72 |

The experimental results revealed that the MRBERT outperformed the models of the ablation experiment in all metrics, especially in the melody prediction. Since *w/o cross-attn.* utilized separate embedding, the performance is slightly higher than that of *w/o separated embed.* Furthermore, pre-training considerably improved the prediction of the melody and rhythm.

### *4.4. Results of Conditional Generation*

In the conditional generation, the melody and rhythm dropped at random positions were used as the evaluation data. The pre-trained MRBERT with the output layers of the conditional generation predicted the missing part of the melody and rhythm based on a given melody and rhythm. Figure 8 displays the predictions of the model and correct answers for the missing parts of the head, middle, and tail of a piece of music. The leadsheet reveals that the missing part in the middle of the bar (or measure) could be easily predicted, but misjudgments occurred at the position at which the bar was switched.

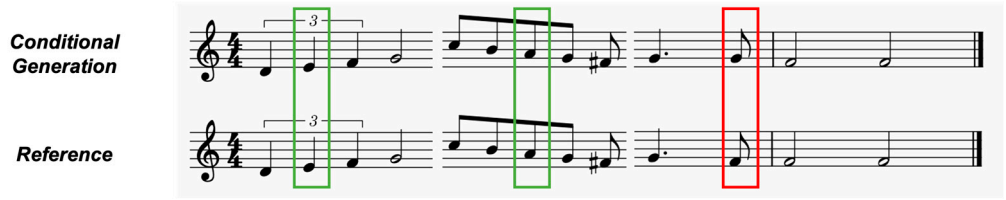


**Figure 8.** Leadsheets of conditional generated results and reference.

Table 4 presents the details of the predictions in Figure 8. The model presents strong confidence in the rhythm prediction with a high accuracy, whereas the probabilities of the melody candidates did not differ considerably. Although the model predicted *F4* as *G4*, *F4* appeared as the second candidate immediately after. Furthermore, the rhythm *1/8* was accurately predicted at this time but the probability of the first candidate did not have an absolute advantage because, during the bar switching stage, the prediction of the rhythm fluctuates, which is a normal phenomenon.

**Table 4.** Details of conditional generation.

| Masked Pitch Sequence | Probabilities of Pitch | Masked Rhythm Sequence | Probabilities of Rhythm |
|---|---|---|---|
| <BOS>, D4, E-4 [1], F4, G4, . . . | E-4: 0.276; G4: 0.130; B-4: 0.118; A-4: 0.114; F4: 0.087; Rest: 0.069 | <BOS>, 1/6, 1/6, 1/6, 1/2, . . . | 1/6: 0.626; 3/16: 0.098; 1/4: 0.094; 1/2: 0.048 |
| . . . , C5, B4, A4, G4, F#4, . . . | A4: 0.229; Rest: 0.164; G4: 0.160; C5: 0.141; B4: 0.096; D5: 0.033 | . . . , 1/8, 1/8, 1/8, 1/8, 1/8, . . . | 1/8: 0.785; 1/4: 0.109; 3/8: 0.040; 1/2: 0.038 |
| . . . , G4, F4, F4, F4, <EOS> | G4: 0.280; F4 [2]: 0.127; A4: 0.116; E4: 0.105; D4: 0.086; F#4: 0.083 | . . . , 3/8, 1/8, 1/2, 1/2, <EOS> | 1/8: 0.280; 1/2: 0.197; 1/4: 0.086; 3/8: 0.080 |

[1] The underline "__" indicates the covered pitch or rhythm. [2] Model predicted G4, but the correct answer is F4.

Table 5 presents the ablation experimental results of HITS@k in the conditional generation task. For the melody prediction, in HITS@k (k = 1, 3, 5, and 10), the MRBERT achieved the average of 54.86%, 1.49% higher than *w/o cross-attn.*, and 5.22% higher than *w/o separated embed.*, and 9.95% higher than the BiRNN. For the rhythm prediction, it achieved the average of 81.85%, 0.55% higher than *w/o cross-attn.*, and 2.09% higher than *w/o separated embed.*, and 3.16% higher than the BiRNN.

**Table 5.** Ablation experimental results of the conditional generation task.

| Model | HITS@1 (%) | | HITS@3 (%) | | HITS@5 (%) | | HITS@10 (%) | |
|---|---|---|---|---|---|---|---|---|
| | Mel. | Rhy. | Mel. | Rhy. | Mel. | Rhy. | Mel. | Rhy. |
| MRBERT | 18.67 | 51.14 | 45.86 | 82.78 | 65.05 | 93.69 | 89.84 | 99.79 |
| w/o cross-attn. | 18.07 | 50.93 | 43.94 | 82.02 | 63.35 | 92.55 | 88.10 | 99.69 |
| w/o separate embed. | 15.69 | 48.61 | 40.27 | 80.11 | 57.68 | 90.73 | 84.91 | 99.57 |
| BiRNN | 13.07 | 48.11 | 34.91 | 78.48 | 51.95 | 89.03 | 79.71 | 99.12 |

The experimental results revealed that the MRBERT outperformed the other ablation models, and the accuracy of the rhythm prediction was higher than that of the other models. Compared to the autoregressive generation, since information from two directions was considered in the conditional generation, the accuracy was slightly higher.

### *4.5. Results of Seq2Seq Generation*

In the Seq2Seq generation, the melody with the chords was used as the evaluation data. Figure 9 shows an example of the real chords and predicted chords based on the pre-trained MRBERT with the output layer of the Seq2Seq generation. The predicted chords contained "F," "BbM," and "C7." They were all included in the real chords.

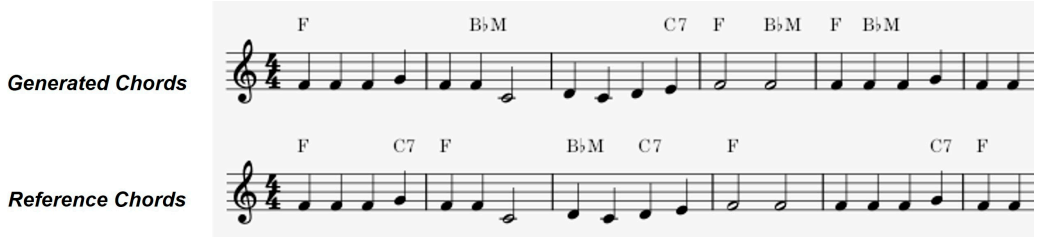


**Figure 9.** Leadsheets of given melody sequence with generated chords and reference chords.

Table 6 presents the ablation experimental results of HITS@k in the Seq2Seq generation task. The MRBERT achieved the average of 49.56%, 0.61% higher than *w/o cross-attn.*, and 1.83% higher than *w/o separated embed.*, and 5.14% higher than the BiRNN.

**Table 6.** Ablation experimental results of Seq2Seq generation task.

| Model | HITS@1 (%) | HITS@3 (%) | HITS@5 (%) | HITS@10 (%) |
|---|---|---|---|---|
| MRBERT | 22.94 | 45.90 | 57.42 | 71.97 |
| w/o cross-attn. | 22.61 | 45.24 | 56.75 | 71.18 |
| w/o separate embed. | 22.15 | 43.46 | 55.12 | 70.17 |
| BiRNN | 19.70 | 39.96 | 51.50 | 66.51 |

The experimental results revealed that the MRBERT outperformed the other ablation models in the Seq2Seq generation task. Separate embedding also improved the performance even when predicting the chords rather than the melody and rhythm.

## 5. Discussion

This paper has conducted ablation experiments for three kinds of tasks, autoregressive generation, conditional generation, and Seq2Seq generation, and has evaluated them at multiple levels by setting different k in HIST@k. The following has been demonstrated by the experimental results: First, pre-trained representation learning can improve the performance of the three kinds of tasks. This is evident in the fact that the performance of the RNN and BiRNN is significantly lower than that of the models using pre-training techniques in all tasks. Second, it is effective to consider the melody and rhythm separately in representation learning. From the ablation results, it can be seen that the model using separate embedding performs better in HITS@k in each task than that not using separate embedding. Third, the assumption that there are weak dependencies between the melody and rhythm is reasonable. The performance of the MRBERT using both separate embedding and semi-cross attention together is slightly higher than that using only separate embedding.

This paper and other music representation learning studies are inspired by language modeling in natural language processing, so this method can only be applied to symbolic format music data. In fact, a large amount of music exists in audio format, such as mp3, wav, etc. This requires the model to be able to handle continuous spectrograms rather than discrete sequences. There have been some studies in computer vision that explore the application of representation learning in image processing [30–32], which is very enlightening for future work.

## 6. Conclusions

This paper proposed MRBERT, a pre-trained model for multitask music generation. During pre-training, the MRBERT learned representations of the melody and rhythm by dividing the embedding layers and transformer blocks into two groups and implementing information exchanging through semi-cross attention. Compared to the original BERT, the MRBERT simultaneously considered the strong dependencies of the melodies and rhythms on themselves and the weak dependencies between them, which allows it to

learn better representations than the original BERT. In the subsequent fine-tuning, the corresponding content was generated according to the tasks. Three music generation tasks, namely autoregressive, conditional, and Seq2Seq generation, were designed to help users compose music, making the composition more convenient. Unlike traditional music generation approaches designed for a single task, these three tasks included multiple functions of melody and rhythm generation, modification, and completion, as well as chord generation. To verify the performance of the MRBERT, ablation experiments were conducted on each generation task. The experimental results revealed that pre-training improves the task performance, and the MRBERT, using separate embedding and semi-cross attention, outperformed the traditional language pre-trained model BERT in the metric of HITS@k.

The proposed method can be utilized in practical music generation applications, including melody and rhythm generation, modification, completion, and chord matching, such as web-based music composers. However, to generate high-quality music, a music corpus composed of leadsheets is used as the training data. These leadsheets must clearly label the melodies, rhythms, and corresponding chords. The problem is that it is difficult to collect this type of data, which limits the expansion of the data volume. In the future, although the application of pre-training techniques in music will continue to be explored, it is equally important to extend the generation tasks to unlabeled music symbolic data and audio data.

**Author Contributions:** Conceptualization, S.L., and Y.S.; methodology, S.L. and Y.S.; software, S.L., and Y.S.; validation, S.L. and Y.S.; writing—original draft preparation, S.L.; writing—review and editing, Y.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2021R1F1A1063466).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Restrictions apply to the availability of these data. Data were obtained from https://github.com/00sapo/OpenEWLD and are available accessed on 1 October 2022.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, W.; Wu, Q.J.; Yang, Y.; Akilan, T. Multimodel Feature Reinforcement Framework Using Moore–Penrose Inverse for Big Data Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 5008–5021. [CrossRef] [PubMed]
2. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. In Proceedings of the 34th Advances in Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020; pp. 1877–1901.
3. Dong, H.W.; Hsiao, W.Y.; Yang, L.C.; Yang, Y.H. MuseGan: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 34–41.
4. Li, S.; Jang, S.; Sung, Y. Automatic Melody Composition Using Enhanced GAN. *Mathematics* **2019**, *7*, 883. [CrossRef]
5. Choi, K.; Fazekas, G.; Sandler, M.; Cho, K. Convolutional Recurrent Neural Networks for Music Classification. In Proceedings of the 2017 IEEE 42nd International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2392–2396.
6. Qiu, L.; Li, S.; Sung, Y. DBTMPE: Deep Bidirectional Transformers-Based Masked Predictive Encoder Approach for Music Genre Classification. *Mathematics* **2021**, *9*, 530. [CrossRef]
7. Park, H.; Yoo, C.D. Melody Extraction and Detection through LSTM-RNN with Harmonic Sum Loss. In Proceedings of the 2017 IEEE 42nd International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2766–2770.
8. Li, S.; Jang, S.; Sung, Y. Melody Extraction and Encoding Method for Generating Healthcare Music Automatically. *Electronics* **2019**, *8*, 1250. [CrossRef]
9. McLeod, A.; Steedman, M. Evaluating Automatic Polyphonic Music Transcription. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 23–27 September 2018; pp. 42–49.

10. Jiang, Z.; Li, S.; Sung, Y. Enhanced Evaluation Method of Musical Instrument Digital Interface Data based on Random Masking and Seq2Seq Model. *Mathematics* **2022**, *10*, 2747. [CrossRef]
11. Wu, J.; Hu, C.; Wang, Y.; Hu, X.; Zhu, J. A Hierarchical Recurrent Neural Network for Symbolic Melody Generation. *IEEE Trans. Cybern.* **2019**, *50*, 2749–2757. [CrossRef] [PubMed]
12. Li, S.; Jang, S.; Sung, Y. INCO-GAN: Variable-Length Music Generation Method Based on Inception Model-Based Conditional GAN. *Mathematics* **2021**, *9*, 387. [CrossRef]
13. Makris, D.; Agres, K.R.; Herremans, D. Generating Lead Sheets with Affect: A Novel Conditional Seq2Seq Framework. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
14. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
15. Walder, C. Modelling Symbolic Music: Beyond the Piano Roll. In Proceedings of the 8th Asian Conference on Machine Learning (ACML), Hamilton, New Zealand, 16–18 November 2016; pp. 174–189.
16. Hadjeres, G.; Pachet, F.; Nielsen, F. DeepBach: A Steerable Model for Bach Chorales Generation. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1362–1371.
17. Chu, H.; Urtasun, R.; Fidler, S. Song From PI: A Musically Plausible Network for Pop Music Generation. *arXiv* **2016**, arXiv:1611.03477.
18. Mogren, O. C-RNN-GAN: Continuous Recurrent Neural Networks with Adversarial Training. *arXiv* **2016**, arXiv:1611.09904.
19. Noh, S.H. Analysis of Gradient Vanishing of RNNs and Performance Comparison. *Information* **2021**, *12*, 442. [CrossRef]
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
21. Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; Liu, T.Y. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. In Proceedings of the Findings of the Associations for Computational Linguistics: ACL-IJCNLP, Online, 1–6 August 2021; pp. 791–800.
22. Chou, Y.H.; Chen, I.; Chang, C.J.; Ching, J.; Yang, Y.H. MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding. *arXiv* **2021**, arXiv:2107.05223.
23. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
24. Huang, Y.S.; Yang, Y.H. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1180–1188.
25. Hsiao, W.Y.; Liu, J.Y.; Yeh, Y.C.; Yang, Y.H. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 178–186.
26. Simonetta, F.; Carnovalini, F.; Orio, N.; Rodà, A. Symbolic Music Similarity through a Graph-Based Representation. In Proceedings of the Audio Mostly on Sound in Immersion and Emotion, North Wales, UK, 12–14 September 2018; pp. 1–7.
27. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
28. Shapiro, I.; Huber, M. Markov Chains for Computer Music Generation. *J. Humanist. Math.* **2021**, *11*, 167–195. [CrossRef]
29. Mittal, G.; Engel, J.; Hawthorne, C.; Simon, I. Symbolic Music Generation with Diffusion Models. *arXiv* **2021**, arXiv:2103.16091.
30. Zhang, W.; Wu, Q.J.; Zhao, W.W.; Deng, H.; Yang, Y. Hierarchical One-Class Model with Subnetwork for Representation Learning and Outlier Detection. *IEEE Trans. Cybern.* **2022**, 1–14. [CrossRef] [PubMed]
31. Zhang, W.; Yang, Y.; Wu, Q.J.; Wang, T.; Zhang, H. Multimodal Moore–Penrose Inverse-Based Recomputation Framework for Big Data Analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–13. [CrossRef] [PubMed]
32. Zhang, W.; Wu, Q.J.; Yang, Y. Semisupervised Manifold Regularization via a Subnetwork-Based Representation Learning Model. *IEEE Trans. Cybern.* **2022**, 1–14. [CrossRef] [PubMed]