Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5845689, 7 pages
https://doi.org/10.1155/2022/5845689

Research Article

Wireless Communications and Mobile Computing Multitrack Music Generation Network Based on Music Rules

Yun Tie,[1,2] Tao Wang,[1] Cong Jin,[3] Xiaobing Li,[2] and Ping Yang[4]

[1]*School of Information and Engineering, Zhengzhou University, 450001, China*
[2]*Department of Music Artificial Intelligence, Central Conservatory of Music, 100031, China*
[3]*School of Information and Communication Engineering, Communication University of China, 100024, China*
[4]*Beijing Polytechnic, 100029, China*

Correspondence should be addressed to Xiaobing Li; lxiaobing@ccom.edu.cn

Received 26 September 2021; Accepted 19 July 2022; Published 29 December 2022

Academic Editor: Peng Li

Multitrack music generation technology is becoming more and more mature, but the existing generation technology cannot reach the desired effect in terms of harmony and matching degree, and most of the generated music does not conform to the music theory knowledge. In order to solve these problems, we propose a multitrack music generation network based on transformer to produce music with high musicality under the guidance of music theory rules. This paper uses an improved version of transformer to learn the information inside a single-track sequence and between different tracks. Then, a combination of music theory rules and crossentropy loss is used to guide the training of the generated network, and the well-designed loss objective function is optimized while the discrimination network is trained. Compared with other multitrack music generation models, the validity of our model is proved.

1. Introduction

As the human demand for music increased, intelligent composition technology emerged. Generally, musical instruments can be divided into single-track and multitrack types, while music generation models can be divided into symbolic [1, 2] and audio techniques [3, 4]. However, piano, guitar, and bass serve as the primary instruments in contemporary music. As such, this study investigated the generation of multitrack symbolic music.

End-to-end sequence models, such as recurrent neural networks (RNNs), long short-term memory (LSTM), and hierarchical RNNs, are a common technique used for intelligent music composition in previous studies. The music models proposed for multitrack music generation include HRNN [5], MiDiNet [6], and MuseGAN [1]. A large number of experiments have found that these models only allow the network to learn the relationship between note features from the real music data, but not the harmony and rules that the composer needs to follow from the whole music. As a result, the resulting music seems to lack harmony and to be incongruous with human hearing habits.

Therefore, in order to solve all the problems encountered above, we propose a novel network, which is improved on the basis of transformer [7] to get a crosstrack transformer network which can learn the information between different tracks well and combined with discrimination network to produce multitrack music in line with the public's musical literacy under the guidance of music rules. Finally, a set of music evaluation indexes is proposed. Through the evaluation, it is found that the model proposed by us is closer to the real music works than the benchmark and other multitrack music generation models.

Our main contributions are as follows: (1) based on transformer network, a generative network based on music theory knowledge is proposed to guide the generation of confrontation network in line with human music literacy. (2) In view of the importance of the internal information of single-track sequence and the information between different track sequences in the generation of multitrack music,

transformer is improved to produce works satisfying the correlation of internal information of single-track and the harmony between different tracks. (3) In view of the importance of music theory rules in music, a new discriminant method combining music rule mathematical model and discrimination network is put forward.

2. Related Work

Many methods of music generation have been proposed by researchers. For example, in 2016, Mogren proposed a continuous recurrent neural network (C-RNN-GAN) model with confrontation training based on an RNN, to generate melody [8]. In 2018, Roberts et al. established a hierarchical RNN to generate 16-bar musical notes [9]. However, common RNN and LSTM networks cannot solve the problem of long-term dependence between contexts. As such, Huang et al. modified the relative attention mechanism in a transformer sequence model used for text translation [10] or text continuation [11] and generated a musical clip with the same pitch, length, and interval structure [12].

In the generation of multitrack music, researchers have started to use neural network with VAE, GAN, and transformer for multitrack music generation [8, 13, 14]. In 2016, Chu et al. proposed an RNN-based hierarchical model (HRNN), in which the lower structure generated melody while the high-level structures generated chords and percussion for accompaniment; compared with the traditional music generation method, the rhythm has been greatly improved [5]. In 2019, Zhang proposed a technique to generate multitrack music using a decoding structure with a transformer serving as the generator and an encoding structure functioning as the discriminator [15]. However, these models have no regular limits in terms of melody and rhythm, and the resulting samples are not ideal. Therefore, in 2020, Jin et al. proposed the MTMG model method that could well learn the relationship between different sound tracks [16]. However, the existing models of music generation are deficient in melody, rhythm, overall harmony, and matching degree, and most of the generated music does not conform to the basic knowledge of music theory.

Therefore, based on the existing achievements, this paper proposes a multitrack music generation model guided by music rules, combining transformer model and discrimination network according to the process of human music creation, and proves the effectiveness of the model through experiments.

3. Proposed Method

3.1. Data Representation. The inputs and outputs used by the model are MIDI files. In order to adapt the MIDI file to the generation task of this model, it is necessary to extract eight features of MIDI file and encode them into event sequences according to the features (see Figure 1), where each event is represented as a tuple, namely, bar, position, chord, tempo value, tempo class, note on, note velocity, and note duration [17]. Bar is the number of bars, position represents the position of each event type, chord represents the set chord

```
Event(name=Bar, time=None, value=None, text=1)
Event(name=Position, time=0, value=1/16, text=0)
Event(name=Chord, time=0, value=N:N, text=N:N)
Event(name=Position, time=0, value=1/16, text=0)
Event(name=Tempo Class, time=0, value=mid, text=None)
Event(name=Tempo Value, time=0, value=30, text=None)
Event(name=Position, time=480, value=5/16, text=480)
Event(name=Tempo Class, time=480, value=slow, text=None)
Event(name=Tempo Value, time=480, value=0, text=None)
Event(name=Position, time=960, value=9/16, text=960)
Event(name=Chord, time=960, value=G:maj, text=G:maj)
```

Figure 1: An example of a section of a MIDI file being converted into a sequence of events.

progression, tempo class represents tempo type (fast, moderate, and slow), tempo value represents the value that quantifies the rhythm type, note in indicates the start time of the pitch (pitch quantization range "0-127"), note velocity indicates the perceived loudness of notes (quantization range "0-127") and note duration indicates the length of each note.

3.2. Overall Framework. Based on transformer, this paper is oriented by music theory knowledge rules and combined with discrimination network to generate multitrack music (see Figure 2). Firstly, three tracks are encoded into time sequence, respectively, and the internal information of single-track sequence is learned through three generators, and the state y_t of the next moment is generated. Secondly, six CT-transformer modules are used to learn the sound track sequence in pairs, and the piano sequence after learning the guitar track sequence and the bass track sequence is pieced together to obtain the piano sequence containing the information of the other two tracks. The learning of the guitar track sequence and the bass track sequence is the same as that of the piano track sequence. Finally, the real sample sequence and the generated sample sequence were discriminated by discriminator D_φ, and the generation was guided by music theory rules.

3.3. Generation Network. In the generation stage, the model needs to learn two parts: first, single-track sequence information learning and generation (G_p, G_g, and G_b). Second is information learning and generation between multitrack sequences (CT-transformer).

In the single-track sequence information learning section, only the decoding portion of the transformer [10] was used in the single-track generation network. The input feature sequence was mapped to embedding through using a learning embedding matrix, used to group information after the kth step through Ng self-attention blocks (Ng is equal to 5). This masking mechanism ensures that characters refer only to information prior to time k. The output of the last self-attention block is then mapped to a vocabulary space and activated by a softmax layer to produce the output feature distribution (see Figure 3). In the pretraining stage, the single-track generator is trained to minimize the cross-entropy loss between the predicted character and the input characters. In the generation phase, characters are produced individually in an autoregressive manner.

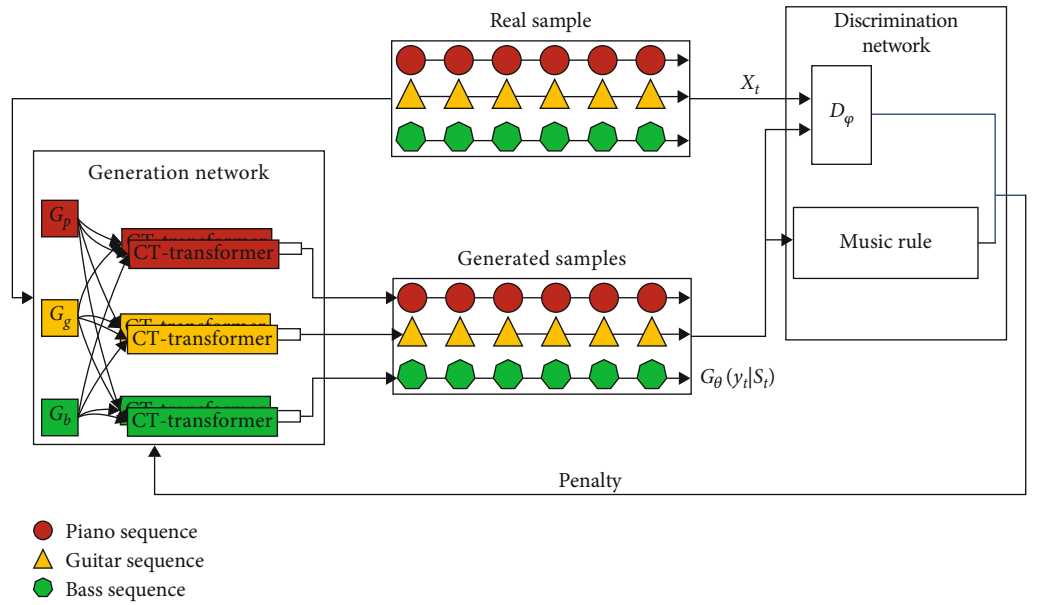

Figure 2: Multitrack music generation network framework.

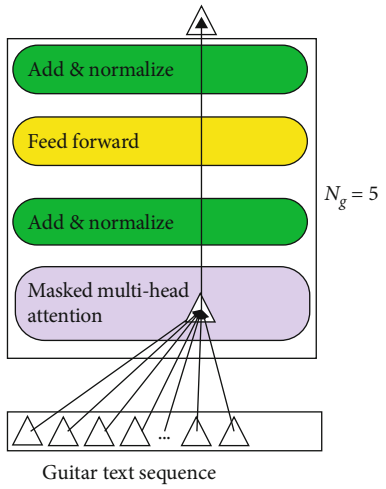

Figure 3: Single-track music generation structure.

In the part of information learning and generation between multitrack sequences, this module improves the self-attention mechanism based on transformer, namely, CT-transformer, and the core part is the crosstrack attention mechanism. The piano sequence is taken as the learning object, and the guitar sequence is taken as the learning object, respectively, represented as $X_p \in R^{T_p \times d_p}$ and $X_g \in R^{T_g \times d_g}$. $T_{(.)}$ and $d_{(.)}$ represent sequence length and feature dimension, respectively. Let us define the query as $Q_{(p)} = X_p W_{Q_p}$ and the key-value pairs as $K_{(g)} = X_g W_{K_g}$ and $V_{(g)} = X_g W_{V_g}$, where $W_{Q_p} \in R^{d_p \times d_k}$, $W_{K_g} \in R^{d_g \times d_k}$, and $W_{V_g} \in R^{d_g \times d_v}$ are weights (see Figure 4). $Z_{p \longrightarrow g}^{[i]}$ is the state obtained at the moment i, $i = [1, 2, \cdots, 6]$. At time i, the character gets $\widehat{Z}_{p \longrightarrow g}^{[i]}$ after passing through the crosstrack learning module CT-transformer. After layer normalization, it gets $f_{\theta_{p \longrightarrow g}^{[i]}}(\mathrm{LN}(\widehat{Z}_{p \longrightarrow g}^{[i]}))$ through the feedforward layer and sums with the sequence normalized by the layer to get the state $Z_{p \longrightarrow g}^{[i]}$ at the next time, as shown in Equation (1), where $Z_{p \longrightarrow g}^{[i]}$ is the piano sequence learning the guitar sequence, and the multihead crosstrack attention output sequence through layer i is shown in Equation (2); $\widehat{Z}_{p \longrightarrow g}^{[i]}$ is the sum of the previous time state after the previous time state is normalized with the layer through crosstrack learning.

$$Z_{p \longrightarrow g}^{[i]} = f_{\theta_{p \longrightarrow g}^{[i]}}\left(\mathrm{LN}\left(\widehat{Z}_{p \longrightarrow g}^{[i]}\right)\right) + \mathrm{LN}\left(\widehat{Z}_{p \longrightarrow g}^{[i]}\right), \quad (1)$$

$$\widehat{Z}_{p \longrightarrow g}^{[i]} = \mathrm{CT}\left(\mathrm{LN}\left(Z_{p \longrightarrow g}^{[i-1]}\right), Z_{p \longrightarrow g}^{[0]}\right) + \mathrm{LN}\left(Z_{p \longrightarrow g}^{[i-1]}\right), \quad (2)$$

$$\mathrm{CT}\left(X_p, X_g\right) = W[\mathrm{head}_1, , \mathrm{head}_h], \quad (3)$$

$$\mathrm{Head}_h = \mathrm{softmax}\left(\frac{Q_p K_g^T}{\sqrt{d_k}}\right) V_g. \quad (4)$$

After getting multihead crosstrack attention, in order to make the output sequence and the input sequence have the same dimension, the output sequence is normalized by layer, and then, input the feedforward sublayer to make residual connection with the normalized output sequence, so as to get the output sequence $Z_{p \longrightarrow g}^{[i]}$ after the learning module of layer i. Similarly, take the piano track and learn the bass track information $Z_{p \longrightarrow b}^{[i]}$.

Finally, $Z_{p \longrightarrow g}^{[i]}$ and $Z_{p \longrightarrow b}^{[i]}$ 144 are spliced to obtain the piano sequence Z_p containing the information 145 of guitar sequence and bass sequence, as shown in Equation (5).

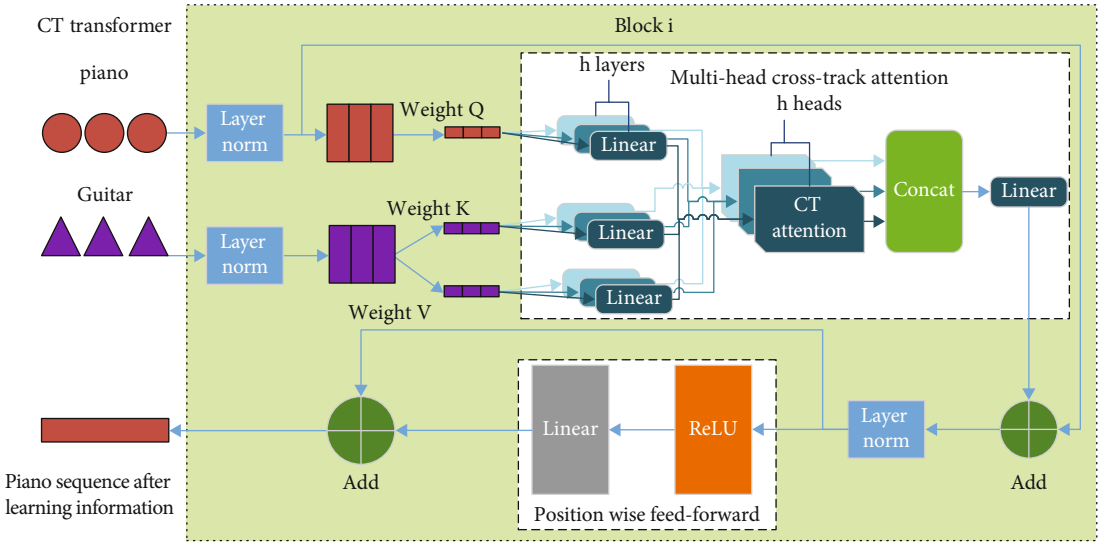

Figure 4: Piano sequence learning guitar sequence module.

Guitar sequence Z_g and bass sequence Z_b are similar to piano sequence Z_p.

$$Z_p = \text{Concat}\left(Z_{p\longrightarrow g}^{[i]}, Z_{p\longrightarrow b}^{[i]}\right). \tag{5}$$

3.4. Discrimination Network. After the corresponding predicted token $\hat{y}_t$ is obtained by generating the network, this paper takes the note vector X_{t+1} input at the next moment as the target value $\hat{y}_{t+1}$ at the current moment; that is, it forms a supervised learning environment and updates the model parameters according to the predicted value and the original sample. In this paper, softmax layer is taken as the output layer, that is, the probability distribution of the output notes, so crossentropy is used to construct the loss function. As shown in Equation (6), when multitrack music is generated, the parameters can be greatly optimized and the quality of music works can be improved through cross-entropy training of the model.

$$D_\varphi = -\frac{1}{I}\sum_{i=1}^{I}[y_t \log \hat{y}_t + (1-y_t)\log(1-\hat{y}_t)]. \tag{6}$$

The emotion represented by each mode in the music is different. When composing music, the composer needs to set the range of notes in a music in advance. Once there are notes higher or lower than the range, the quality and emotion of a music will be greatly reduced. In order to improve the quality of generated music, this paper adds this restriction to the music theory rules. In pop music, the pitches of the piano, guitar, and bass tracks are within the C2-C6, E2-E3, and E2-C4 ranges, respectively. In Equation (7), $y_{\min}$ and $y_{\max}$ are the lowest and highest notes set in advance according to the musical mode, y_t is the pitch at time$t$, and $R^{m_1}(S_{1:t}, y_t)$ represents the reward value of the state at time$t$ (we set different reward values according to the impact on the music itself caused by conforming to this rule and not conforming to this rule. The reward value in subsequent rules is also different. The setting of the value is as follows: first, find the rule that has the greatest impact on the quality of generated music, set its reward value to +1 or -1, and then get its reward value according to the comparison between each rule and this strongest rule).

$$R^{m_1}(S_{1:t}, y_t) = \begin{cases} 0.1, & y_t \in [y_{\min}, y_{\max}], \\ -0.6, & y_t \notin [y_{\min}, y_{\max}]. \end{cases} \tag{7}$$

Furthermore, the number of notes for the piano, guitar, and bass is fewer than 8, 6, and 1, respectively. In as Equation (8), a_t represents the number of notes at time t, n represents the maximum number of notes sounded, and $R^{m_2}(a_t)$ represents the bonus value for the number of notes within the required range.

$$R^{m_2}(a_t) = \begin{cases} 0.2, & a_t \le n, \\ -0.5, & a_t > n. \end{cases} \tag{8}$$

The chords set in this paper are triads, such as F-G-Am-F. For chord notes, in strong music, the strong beat is basically in the odd beat position. Assume that C_1^t, C_2^t, and C_3^t are chord notes at the moment t, and y_t represents the notes selected by the generation network at that moment t, as shown in Equation (9), where $R^c(S_{1:t}, y_t)$ is the reward value provided by this rule (we set the corresponding reward value according to the compliance with this rule, and for subsequent rules, the reward value is determined by the degree of impact on the quality of the generated music).

$$R^C(S_{1:t}, y_t) = \begin{cases} 0.7, & y_t \in \left(C_1^t, C_2^t, C_3^t | t\%2 = 1\right), \\ 1, & y_t \notin \left(C_1^t, C_2^t, C_3^t | t\%2 = 1\right). \end{cases} \tag{9}$$

Different weight ratios are assigned according to the importance of different rules in music, as shown in Equation (10), where R_G represents the reward value for meeting the set rules and α_i represents the weight of rule i, $i = 1, 2, 3$.

$$R_G = \alpha_1 R^{m_1}(S_{1:t}, y_t) + \alpha_2 R^{m_2}(a_t) + \alpha_3 R^C(S_{1:t}, y_t). \quad (10)$$

The objective function of the model is obtained by assigning different weights to the reward function and the crossentropy loss function through the discrimination network, as shown in Equation (11), where J_{G_θ} is the objective function value after assigning weight values β_1 and β_2 to the reward value and the loss function value.

$$J_{G_\theta} = \beta_1 R_G + \beta_2 D_\varphi. \quad (11)$$

The Adam optimizer used in the experiment sets the learning rate as 0.0001. The number of iterations set in the training process is 20000. If the number of iterations of training is less than 20000, the value of loss function converges, and the model training is terminated immediately. If the number of iterations of training has reached 20000, but the loss function value has not converged, the model training will end automatically. In the process of model training, when the model obtains the reward value of identifying network feedback at each step, it will automatically update the network parameters, so as to maximize the long-term reward of the objective function. This section uses the objective function constructed by Equation (11) to update the gradient of the generated network parameter θ of the model, as shown in Equation (12). The network parameters θ of the generated network can be optimized according to Equation (13). After the model is trained, the test set of 500 MIDI files is used to test the model. The event of one section at the beginning of each MIDI file is used as the input, and the predicted section event is used as the input of the next round of prediction, and the model is continued in turn.

$$\nabla_\theta J_{G_\theta} = \sum_{t=1}^{T} G_\theta(y_t | S_{1:t}) * R_G(S_{1:t}, y_t), \quad (12)$$

$$\theta \longleftarrow \theta + \nabla_\theta J_{G_\theta}. \quad (13)$$

4. Experiment and Analysis

4.1. Data Set and Implementation Details. All experiments in this paper used the Lakh MIDI data set [18], which included 176,581 different multitrack MIDI files. The music containing piano, guitar, and bass was screened out from the data set, and then, these three tracks were extracted through the pretty MIDI library and combined to obtain 55,213 MIDI files. Finally, the MIDI files of 4/4 beats are selected. At this point, the data set contains only 34,610 MIDI files. We used 24,610 MIDI files as the training set and 10,000 MIDI files as the test set. The generation and discrimination networks in the proposed model were trained using an Adam optimizer and a reward network, to minimize the crossentropy level and optimize the output. The learning rate ε was set to 0.0002, and the number of iterations was 10000. This section tests the MuseGAN model, the MultINN network, and MTMG using our data set and compares the results generated by our proposed network with the same character length generated by all four methods.

Table 1: Human vs. AI evaluation results.

	Average (pro)	Average (all)
Human	8.02	7.93
AI	7.83	8.11

Table 2: Score results of four models.

Indicators	Our	MuseGAN	MTMG	HRNN
Rhythm	8.05	5.79	6.36	5.33
Melody	8.63	7.75	8.21	6.47
Emotion	7.88	8.36	7.29	6.36
Harmony	8.22	5.96	7.48	4.50

Table 3: To quantitatively compare the different modes of multitrack music generation.

Method	Chord matching	Harmony	BLEU
Our	0.681	0.765	0.660
MTMG	0.527	0.731	0.523
MuseGAN	0.579	0.699	0.641
HRNN	0.363	0.525	0.592

4.2. Analysis of Experimental Results

(1) Subjective evaluations: we divided all participants into two groups, professional composers and noncomposers. Participants in the professional groups are those with degrees in music creation or electronic music creation and production education, including the Central Conservatory of Music, Communication University of China, and Zhengzhou University

Human and AI. We prepared a mix of five pieces of music by professional human composers and five pieces created by our model for people to decide whether they were created by humans or by AI [19]. Forty professional composers were asked to rate each piece of music they heard in terms of musical creation theory, while 60 noncomposers were asked to rate their subjective feelings. Each listener will evaluate and score the test samples (points 1 to 10).

Among professional composers, the average score for human music was higher than AI (see Table 1). However, across all participants, our AI music scored higher than real human works (8.11 vs. 7.93), indicating that the quality of our AI music creation was quite close to that of human composers. According to a few single ratings, there are even works that transcend 8 human work. Interestingly, for all

Table 4: Comparison of music theory characteristics.

Method	Ours	MTMG	MuseGAN	HRNN	Ours (without music rules)
Note repetition	19.4%	52.6%	0.660	25.4%	35.1%
The notes are out of tune	5.1%	13.4%	0.523	7.2%	8.7%
Unique maximum note	54.6%	47.6%	0.641	51.7%	52.3%
Unique minimum note	57.1%	47.9%	0.592	48.3%	59.2%

the reviewers, the music of the human composers was considered to have been produced by artificial intelligence.

Contrast experiment. Our second test was to compare the generated samples with the three baseline models. We did a hearing test comparing MTMG, MuseGAN, and HRNN. Participants (15 composers and 30 noncomposers) received 20 pieces of music from four different models, each of which generated five pieces of music, but were given the same starting notes and instrumental timbers. The participants were then asked to rate and rate the music in terms of melody, harmony, rhythm, and emotion. After summarizing the scores, the participants were given another round of assessment, which was repeated three times in turn to summarize the final score.

Compared with the three generation models of MuseGAN, MTMG, and HRNN, the overall quality of our model has been significantly improved (see Table 2). Except for emotion, the scores of other indicators are significantly higher than those of the other three models, indicating that the music we generate is more in line with the requirements and rules of composition. It reflects the need to strengthen research on emotion in future work.

(2) Objective evaluations: in this paper, a test set containing 500 MIDI files is used to analyze and evaluate our model, MuseGAN, MTMG, and HRNN models from three aspects of harmony degree, chord accuracy, and BLEU score (the BLEU score is used to measure the similarity between the test set and the generated sample [20]). The model parameters and training set are the same. In order to test whether the model proposed in this paper improves chord accuracy, chord accuracy is defined to evaluate the accuracy between the chords of the samples generated by the model and the specified chords, as shown in Equation (12), where P is the number of segments, $\tilde{y}_m$ is the m chord detected in the generated melody, y_m is the corresponding m chord in the given chord progression, and $E(y_m, \tilde{y}_m)$ represents the error value between $\tilde{y}_m$ and y_m:

$$\text{Chord matching} = \frac{\sum_{m=1}^{P} E(y_m, \tilde{y}_m)}{P}, \quad E(y_m, \tilde{y}_m) = \begin{cases} 1, & \tilde{y}_m = y_m, \\ 0, & \tilde{y}_m \neq y_m \end{cases} \tag{14}$$

The harmony of music is the basic standard to evaluate the quality of music, so it is meaningful to evaluate the harmony degree. We also analyzed the harmony degree of the samples generated by the four models and defined that the two tracks have similar chord progression; that is, the two tracks are harmonious, as shown in Equation (15), where P and K, respectively, represent the number of segments generated by music and the number of instrument tracks and C_P^K is the chord corresponding to the P section of the K instrument track.

$$\text{Harmony} = \frac{\sum_{p=1}^{P} \delta \cap {}_1^k C_P^K}{P}, \tag{15}$$

$$\delta(a) = \begin{cases} 1, & a \neq \varnothing, \\ 0, & a = \varnothing. \end{cases} \tag{16}$$

The model we proposed is higher than the other three models in terms of chord matching degree, harmony degree, and BLEU, indicating that the introduction of rules into the discrimination network can guide the generation of music to a certain extent, and CT-transformer can also be of great help in learning information between different tracks (see Table 3).

Comparison of music theory characteristics: in order to verify whether the music rules and their rewards and punishments in our model play a guiding role in music generation, this section quantifies their expression forms according to the set music rules and carries out a series of comparative experiments on the music theory rules. This experiment compares 500 music works generated by three models: MTMG, MuseGAN, HRNN, and our model without music theory rules. In order to ensure the fairness of the experiment, the parameters of the three models, the number of bars, and starting notes of the generated music are the same when generating samples. Select a series of effective feature information from the music samples of the above three models, compare the music theory rules, and summarize the specific statistics, as shown in Table 4.

It can be seen from Table 4 that our model effectively reduces the repetition of notes compared with the music works generated by other models. Compared with the model without music theory rules, the complete model has certain advantages in many indicators, which also shows that the music theory rules after mathematical modeling play a certain guiding role in generating music.

5. Conclusions

In this paper, we propose a novelty model for multitrack music generation. It combines sequence-to-sequence

generation and multitrack learning techniques in a unified framework to achieve optimal convergence of multitrack learning and codecs. In this model, we combine with the discrimination network on the basis of transformer to produce multitrack music in line with the music literacy of the public under the guidance of music rules. The experimental results show that this model has significant advantages over some existing techniques in terms of rhythm, audibility, fluency, and compliance with music rules. In the future, we will strengthen the research on emotion and more than three tracks.

Data Availability

The music data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

There is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by Key Projects of the National Key Research and Development Program of China (2018YFB14039002), National Natural Science Foundation of China (61631016), Beijing Outstanding Young Scientist Program (BJJWZYJH01201910048035), and Fundamental Research Funds for the Central Universities (CUC210B011).

References

[1] H. W. Dong, W. Y. Hsiao, L. C. Yang, and Y. H. Yang, "MuseGAN: multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," *Thirty-Second AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[2] C. Jin, Y. Tie, Y. Bai, X. Lv, and S. Liu, "A style-specific music composition neural network," *Neural Processing Letters*, vol. 52, no. 3, pp. 1893–1912, 2020.

[3] S. Mehri, K. Kumar, I. Gulrajani et al., "SampleRNN: an unconditional end-to-end neural audio 291 generation model," 2016, http://arXiv.1612.07837.

[4] T. Wang, J. Liu, C. Jin, J. Li, and S. Ma, "An intelligent music generation based on variational autoencoder," in *2020 International Conference on Culture-oriented Science & Technology (ICCST)*, pp. 394–398, Beijing, China, 2020.

[5] H. Chu, R. Urtasun, and S. Fidler, "Song from PI: a musically plausible network for pop music generation," 2016, http://arXiv.1611.03477.

[6] L. C. Yang, S. Y. Chou, and Y. H. Yang, "MiDiNet: a convolutional generative adversarial network for symbolic-domain music generation," 2017, http://arXiv.1703.10847.

[7] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, pp. 5998–6008, 2017.

[8] O. Mogren, "C-RNN-GAN: continuous recurrent neural networks with adversarial training," 2016, http://arXiv.1611.09904.

[9] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," *International Conference on Machine Learning*, vol. 80, pp. 4364–4373, 2018.

[10] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," 2018, http://arXiv.1807.03819.

[11] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: attentive language models beyond a fixed-length context," 2019, http://arXiv.1901.02860.

[12] C. Z. A. Huang, A. Vaswani, J. Uszkoreit et al., "Music transformer: generating music with long-term structure," 2018, http://arXiv.1809.04281.

[13] M. Akbari and J. Liang, "Semi-recurrent CNN-based VAE-GAN for sequential data generation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2321–2325, Calgary, AB, Canada, 2018.

[14] C. Donahue, H. H. Mao, Y. E. Li, G. W. Cottrell, and J. McAuley, "LakhNES: improving multi-instrumental music generation with crossdomain pre-training," 2019, http://arXiv.1907.04868.

[15] N. Zhang, *Learning adversarial transformer for symbolic music generation*, IEEE Transactions on Neural Networks and Learning Systems, Piscataway, NJ, USA, 2020.

[16] C. Jin, T. Wang, S. Liu et al., "A transformer-based model for multi-track music generation," *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 11, no. 3, pp. 36–54, 2020.

[17] S. Ji, J. Luo, and X. Yang, "A comprehensive survey on deep music generation: multi-level representations, algorithms, evaluations, and future directions," 2020, http://arXiv.2011.06801.

[18] C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching [Ph.D. thesis]*, Columbia University, Ann Arbor, MI, USA, 2016.

[19] X. Wu, C. Wang, and Q. Lei, "Transformer-XL based music generation with multiple sequences of7 time-valued notes," 2020, http://arXiv.2007.07244.

[20] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 3060–3070, Long Beach, California, USA, 2019.