Multimedia Systems (2024) 30:4
https://doi.org/10.1007/s00530-023-01245-0

REGULAR PAPER

An automatic music generation method based on RSCLN_Transformer network

Yumei Zhang[1,2,3] · Xiaojiao Lv[1,2] · Qi Li[1,2] · Xiaojun Wu[1,2,3] · Yuping Su[1,2] · Honghong Yang[2,3]

Received: 8 November 2022 / Accepted: 9 December 2023 / Published online: 12 January 2024

Abstract

With the development of artificial intelligence and deep learning, a large number of music generation methods have been proposed. Recently, Transformer has been widely used in music generation. However, the structural complexity of music puts forward higher requirements for music generation. In this paper, we propose a new automatic music generation network which consists of a Recursive Skip Connection with Layer Normalization (RSCLN) model, a Transformer-XL model and a multi-head attention mechanism. Our method not only alleviates the gradient vanishing problem in the model training, but also increases the ability of the model to capture the correlation of music information before and after, so as to generate music works closer to the original music style. Effectiveness of the RSCLN_Transformer-XL music automatic generation method is verified through music similarity evaluation experiments using music structure similarity and listening test. The experimental results show that the RSCLN_Transformer-XL music automatic generation model can generate better music than the Transformer-XL model.

Keywords Music generation · Deep learning · Transformer · Attention mechanism

1 Introduction

In recent years, artificial intelligence technology has achieved significant breakthroughs in both academic and industrial fields. However, it mainly focuses on face recognition technology, driverless technology, natural language processing, and so on [1]. In the field of music generation, various methods and theories have been proposed. Leonard Isaaeson and Lejaren Hiller [2] proposed the use of the Makarov process to create music, it is pioneer in intelligent music creation. Music is not only the art of sound, but also the art of time. It is difficult and impossible to present it truthfully in an intuitive and concrete form, which makes music an abstract art with a relatively complex structure. Therefore, it is a challenging task to transform the existing algorithms of computer vision into automatic music generation.

The mainstream of music generation methods can be classified into two classes, one is the traditional music generation algorithm, such as rule-based knowledge base system method, Markov chain, and genetic algorithm. The rule-based knowledge base system method creates different rules according to different types of music, and the time cost is relatively high [3, 4]. Markov chain method can only

Communicated by J. Gao.

✉ Honghong Yang
yanghonghong0615@163.com

Yumei Zhang
zym0910@snnu.edu.cn

Xiaojiao Lv
1759147201@qq.com

Qi Li
943405861@qq.com

Xiaojun Wu
xjwu@snnu.edu.cn

Yuping Su
ypsu@snnu.edu.cn

[1] School of Computer Science, Shaanxi Normal University, Xi'an, China

[2] Key Laboratory of Intelligent Computing and Service Technology for Folk Song, Ministry of Culture and Tourism, Xi'an, China

[3] Key Laboratory of Modern Teaching Technology, Ministry of Education, Shaanxi Normal University, Xi'an, China

produce subsequences that exist in the original data, and the generated music lacks coherence [5]. Genetic algorithm composition has strong subjectivity in defining fitness function, which is easily affected by the bias of designers [6]. On the other hand, the deep learning-based method has been proved to be more creative in music generation.

The music generation methods mentioned above have some limitations. For instance, the representation of music fails to capture higher-level music information, unable to capture long-term dependencies in music sequences. Moreover, the generated music exhibits lower similarity to the original style, and there is an issue of gradient vanishing during model training. Therefore, in order to overcome the above shortcomings, this paper proposes the RSCLN_Transformer-XL model for music generation. We represent Musical Instrument Digital Interface (MIDI) music using REvamped MIDI-derived events (REMI), introducing more music information and structured representation to enhance the effectiveness of music generation. Additionally, the proposed model not only obtains richer music information through multi-head attention mechanism, but also incorporates the RSCLN module to address the issue of gradient vanishing during model training and capture the dependencies of music information over long sequences.

The main contributions of this paper are as follows:

(1) To overcome the limitation of MIDI event representation method in constructing music rhythm structure, we use REMI to convert music into event sequence for processing.
(2) The proposed model uses the multi-head attention mechanism to enhance the attention to the pitch, rhythm, and structure of music and improve the effect of music generation.
(3) We design RSCLN module, which improves the optimization process of the network by skip connection and alleviates the phenomenon of gradient decline. Add layer standardization repeatedly, so that more music input information can be modeled.

2 Related work

A large number of deep neural network models have been proposed lately for music generation, which learn the relevant characteristics of music and the relationship between notes through training neural network, and then generate music according to the trained neural network. Recurrent Neural Network (RNN), Long Short-Term Memory Network (LSTM), Generative Adversarial Nets (GAN), and other sequence models can be used to simulate music information and produce new music works [7–12]. Self-attention-based architectures are becoming more and more popular in generating music because they can model correlations on multiple time scales of long sequences. Attention mechanism has become an integral part of forced sequence modeling and transduction models in various tasks, allowing the modeling of dependencies without considering their distance in the input or output sequence. The Transformer network model proposed by Google in 2017 completely relies on the attention mechanism to draw the global dependency between input and output [13], which shows great potential in temporal data modeling. Transformer network model was initially applied to natural language processing. Deng et al. [14] applied it to create music with long-term structure for the first time, which shows its potential in music generation.

Huang et al. [15] combined the relative attention mechanism with Transformer, and the proposed Music Transformer can generate music with long-term structure. The Pop Music Transformer constructed by Huang et al. [16] can generate expressive popular piano performances with a coherent structure of up to one minute. Choi et al. [17] used Transformer encoder and decoder to coordinate or generate the accompaniment of a given melody. Wu et al. [18] used Jazz Transformer to generate Jazz style music. Donahue et al. [19] proposed the LakhNes model use Transformer to generate multi-track music.

3 Data pre-processing

3.1 Data representation

In music generation, music is usually represented as Note-On, Note-Off, Time-Shift, and Velocity events [16] in a MIDI-like manner, which converts continuous music sequences into event sequences represented by discrete variables. Although the MIDI-like representation is effective in extracting pitch values of notes, it has certain limitations in constructing the rhythmic structure of music. In the process of music composition, composers often use bars, beats, and sub-beats to define the rhythmic structure and organize repetitive patterns with periodic intervals. However, in the event representation of MIDI-like, this structure is implicit, leading to inefficiency in training models for generation tasks.

Specifically, MIDI-like representation lacks high-level musical information such as tempo and chords. In contrast, REMI adopts a more detailed representation method. It uses durations to replace the note relationships in MIDI encoding and employs the combination of Bar and Position to replace Time-Shift events. Here, Bar represents the start

and end of measures in the music, while Position indicates the location of notes within the measures [20]. The "Note-Duration" events in REMI replace the "Note-Off" events to simulate the rhythm of notes. To mimic the expressive rhythm found in music, REMI also introduces Tempo events, with a Position event preceding each Tempo event.

By introducing more music information and structured representation, REMI generally offers greater expressive capabilities compared to MIDI-like. It can accurately capture the rhythm, emotion, and expressiveness of music, resulting in more realistic and expressive music generation outcomes. Table 1 demonstrates the representation methods of MIDI-like and REMI events.

In this paper, miditoolkit tool is used to extract the note information in the MIDI type music file in a sequence manner, and the extracted information is converted into a one-dimensional vector represented by REMI events. The specific process is shown in Fig. 1.

According to the music representation method above, music information in each piece of MIDI music is represented by the music events shown in Fig. 1. Convert the music event sequence into the digital sequence of the corresponding index of the music event. The length of the sequence depends on the duration of the MIDI music file itself.

Table 1 Representation of MIDI-like and REMI events

Meaning of the event	MIDI-like	REMI
Start position of note	Note-On	Note-On
End position of note	Note-Off	Note-Duration
Time grid	Time-Shift	Bar&Postion
Sound intensity	Velocity	Velocity
Rhythm change	×	Tempo

3.2 Data augmentation

Data augmentation (DA) is combined with automatic music generation model to enhance the input data. Through random and small adjustments to the data of some training sets, diversity of training datasets is dynamically enhanced, and then the effect of training model got improved. The data augmentation used in this paper includes the following four methods: changing the pitch of notes, extending, shortening or moving notes in time, adding or deleting notes, and splitting or merging notes. We adopt the method of probabilistic MIDI music data enhancement, which selects one-third of the dataset for data enhancement by random probability. Using one-third of the data allows for effectively balancing the ratio between the augmented dataset and the original dataset. This approach increases the diversity of the dataset while avoiding the issues of data distortion or overfitting that can occur with excessive augmentation. The choice of data enhancement method is also to randomly select one of the above four data enhancement methods by random probability.

Take fine-tuning pitch as an example, as shown in Fig. 2, the left picture shows the original MIDI music clip, and the right picture shows the MIDI music clip after adjusting the pitch category of a certain note.

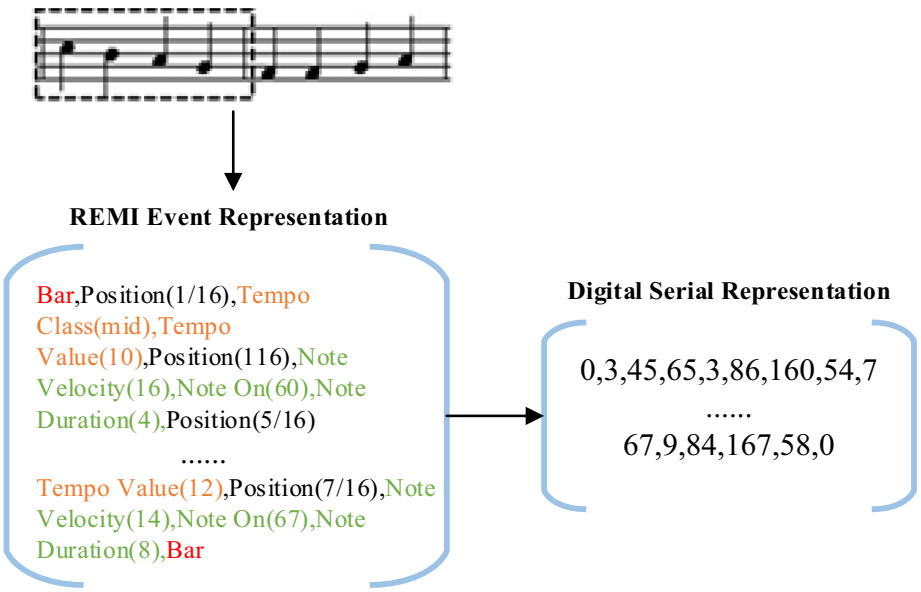

Fig. 1 Musical representation

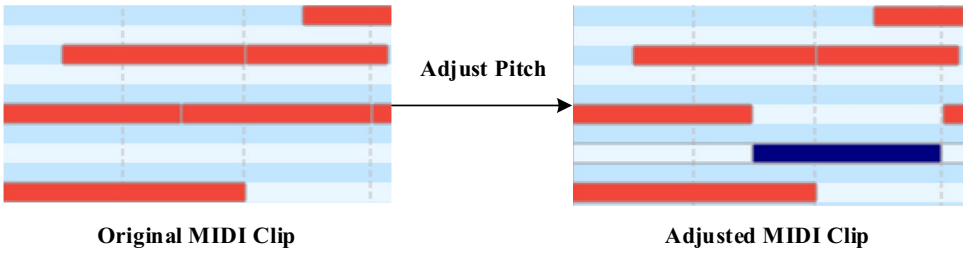

Fig. 2 MIDI file segment after fine-tuning pitch

4 RSCLN_Transformer-XL music generation network model

4.1 Network structure

4.1.1 RSCLN_Transformer-XL network structure

To enhance the quality of music generation, this paper adopts the Transformer-XL network model [21] as the baseline architecture and builds the RSCLN_Transformer-XL model for automatic music generation. The model encodes the input music event digitized vectors and maps them to an embedding layer through an embedding matrix. In order to capture the temporal dependencies in the music sequences, positional encoding is also added to the embedded sequences. The RSCLN_Transformer-XL model primarily consists of multi-head attention modules, RSCLN modules, and feed-forward network layers, as shown in Fig. 3.

Music is an art with complex structure, and the correlation between the sequences is very strong. However, the traditional Transformer model has a limitation when processing long sequence data: the input sequence length cannot exceed the fixed length of the model. This means that when computing hidden states, the model can only rely on limited contextual information from positions before each location.

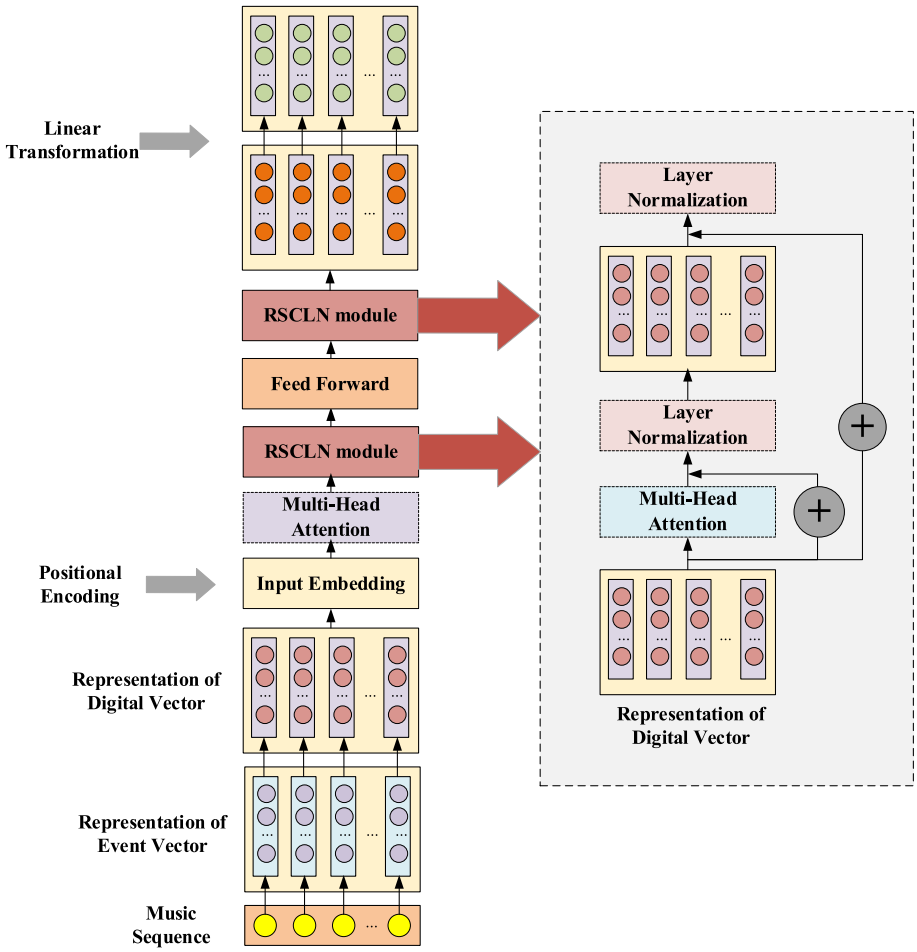

Fig. 3 The RSCLN_Transformer-XL music automatic generation network model structure

While the problem of input sequence length being smaller than the fixed length can be solved by padding, when the sequence length is larger than the fixed length, the sequence is divided into multiple segments without considering the natural boundaries of sentences. Instead, division is based on the fixed length. During the training process, each segment is trained individually without considering the contextual information between adjacent segments. As a result, the semantic coherence between segments is incomplete. This fragmentation of context limits the model's ability to model long-term dependencies effectively. To address this issue, the Transformer-XL model introduces a recursive mechanism to handle extremely long music sequence data. As shown in Fig. 4, the Transformer-XL model introduces the concept of recursion into the deep self-attention mechanism network. Instead of calculating the hidden state of each segment from scratch, it reuses the hidden state obtained in the previous segment. The hidden state obtained in the previous segment will be fixed and saved in the memory form of the current segment, and a circular connection will be built between segments. This method makes the learning dependency exceed the fixed length without destroying the temporal coherence. Therefore, Transformer-XL is capable of handling extremely long sequences and has an advantage in modeling long-term dependencies. This enables it to better capture the long-term structure and dependencies in music sequences, thereby improving the effectiveness of music generation. Compared to the traditional Transformer model, Transformer-XL exhibits superior performance in this regard.

During the training process of neural network models for music generation, the issue of gradient vanishing often arises. In order to avoid the gradient vanishing in the process of training the network, the method of RSCLN is introduced into this module, which uses the advantages of Skip Connection and Layer Normalization [22, 23] recursively to solve the gradient vanishing problem in the process of network optimization. Input the output results of the RSCLN module into the feed-forward network layer, and then use the RSCLN method again for Skip Connections and Layer Normalization. After performing multiple encodings, the output is linear regression. Finally, softmax layer is used as the output layer, and the probability distribution of notes is output to obtain the output sequence and convert it into music sequence.

4.1.2 Multi-head attention mechanism

The multi-head attention mechanism is introduced to record the importance of different information in music generation [24]. For different styles of music, the proportion of each music feature classification is different. Through the multi-head attention mechanism to improve the ability of centralized processing of important data, we can optimize the effect of music generated by the model.

In order to process the input sequence $X = [x_1, x_2, \cdots, x_N]$ of music events with variable length, the feature-based self-attention mechanism is utilized to dynamically generate weights for different connections, which are used to compute attention weights. These weights represent the degree of association or importance between different positions. They are then used to compute a weighted sum of the input vectors, generating the final output representation. Calculation of the query vector Q, the key vector K and the value vector V in the self-attention mechanism is as shown in formulas 1–3.

$$Q = W_q X \tag{1}$$

$$K = W_k X \tag{2}$$

$$V = W_v X \tag{3}$$

where W_q, W_k and W_v represent the weight matrices corresponding to the music. Formula (4) is the calculation formula of the self-attention mechanism.

$$SelfAttention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

where d_k represents number of dimensions of the key vector. Firstly, output of the self-attention mechanism obtains the

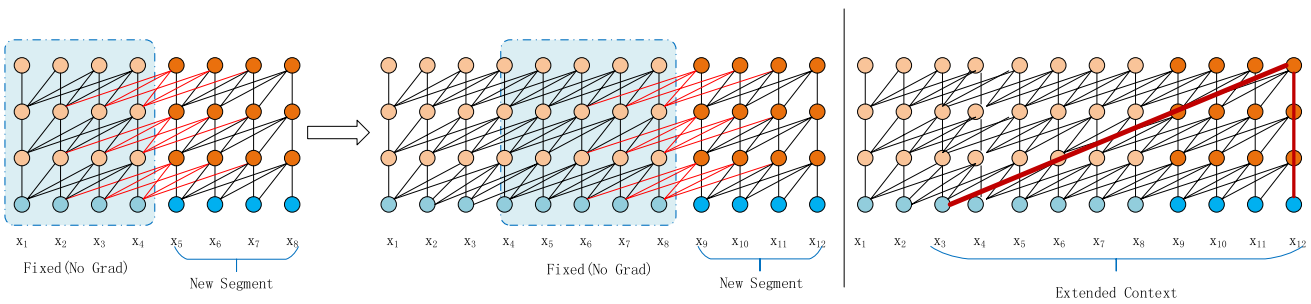

Fig. 4 Recursive mechanism in Transformer-XL

query vector Q, the key vector K and the value vector V by multiplying the input sequence X of the music with the three weight matrices W_q, W_k and W_v respectively. The scaled dot product is used as the attention scoring function to calculate the scoring value, the *soft* max function is used for normalization, and finally the output of the attention mechanism is calculated by combining the value vector V. While in the multi-head self-attention mechanism, the input X is input into N "heads" in parallel, and each "head" is calculated by a separate self-attention mechanism. Output of each "heads" are merged together, and the final output of the multi-head attention mechanism is obtained through the final addition weight matrix W_o. Calculation process of the multi-head attention mechanism is shown in formulas 5–7, and the steps to obtain the multi-head attention are shown in Fig. 5.

$$MultiHead(X) = \left[head_1, head_2, \ldots, head_N\right] W_o \tag{5}$$

$$head_n = SelfAttention(Q_n, K_n, V_n) \tag{6}$$

$$\forall n \in \{1, \cdots, N\}, Q_n = W_q^n, K_n = W_k^n, V_n = W_v^n \tag{7}$$

where $MultiHead(X)$ is the final output of the multi-head self-attention mechanism, and $head_n$ is each "head" output calculated by the self-attention mechanism.

By using different parameters N times to do a linear transformation on the query vector Q, key vector K and value vector V, we can learn the relevant information of music features from different dimensions and representation subspaces. Input the results into the scaled dot product attention to obtain different attention outputs, and connect them according to formulas (5–7) to obtain the final output.

4.1.3 Recursive skip connection with layer normalization

Skip connections propagate linear components in the neural network layer to alleviate the optimization difficulties caused by non-linearity, and add identity mapping from neural network input to output to realize information transmission and integration. In the RSCLN_Transformer-XL

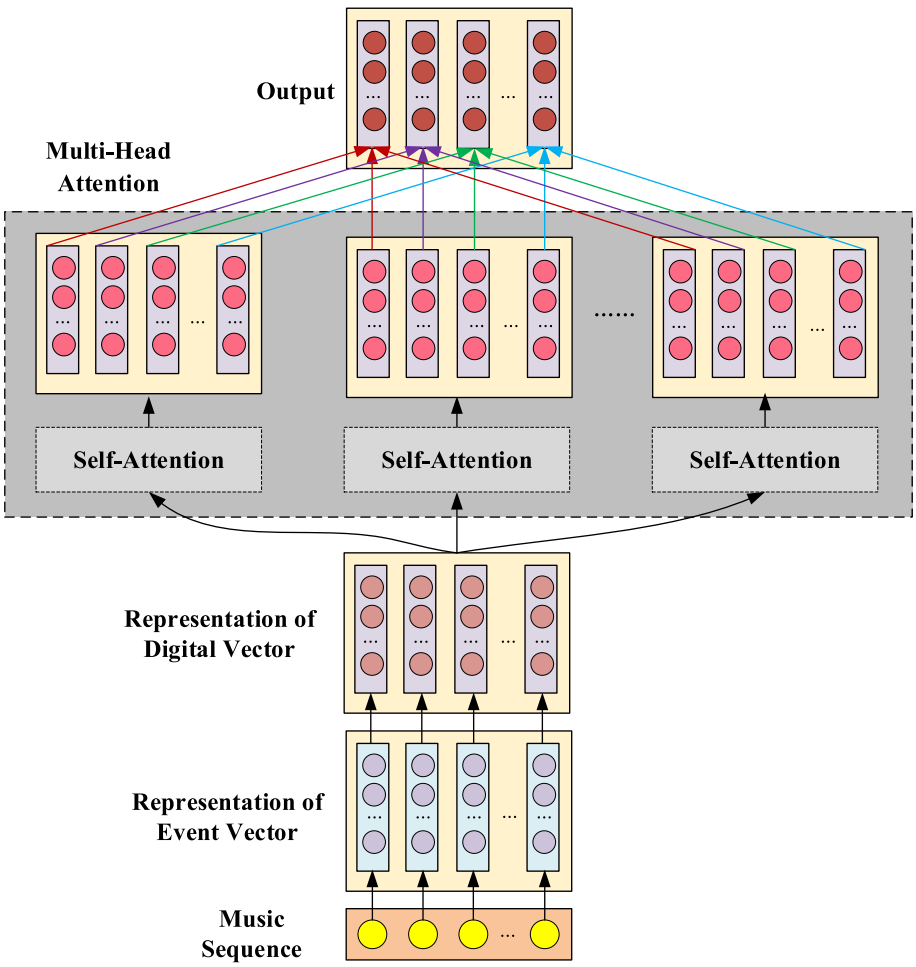

Fig. 5 Calculation process of the multi-head attention mechanism

music generation network model, skip connections can be expressed as formula (8):

$$y = x + Attention(Q, K, V) \tag{8}$$

where x represents the input of the music sequence after the position embedding operation, $Attention(Q, K, V)$ represents calculation function of multi-head attention, and y represents the output after skip connection.

In the process of skip connection, the skip connection and normalization method are combined, which can be expressed as formula (9):

$$y = G(\lambda x + Attention(Q, K, V)) \tag{9}$$

where x represents the input of the sequence of musical events, λ represents the relatively important modulation factor controlling the skip connection or shortcut, $Attention(Q, K, V)$ represents the operation process of the multi-head attention mechanism, G represents the normalized function, and y is the output after normalized function calculation. However, the modulation factor λ will not always be 1, especially when the $Attention(Q, K, V)$ is not well trained. RSCLN combines the advantages of skip connections and layer normalization in a recursive way. It adds music input information and standardizes it in a recursive way, so that more music input information can be modeled. Recursion is defined as formula (10):

$$y_\lambda = LN(x + \lambda_{y-1}) = LN(x + LN(x + Attention(Q, K, V))) \tag{10}$$

where λ represents an integer not less than 1, for example, when $\lambda = 1$, it regresses to the residual block [25] used in Transformer and conforms to the situation that no scaling adjustment is required after skip connection. Figure 6 shows the RSCLN of $\lambda = 2$ when $y_\lambda = LN(x + \lambda_{y-1}) = LN(x + LN(x + Attention(Q, K, V)))$. RSCLN module makes RSCLN_Transformer-XL music generation network model uses layer normalization many times to improve the optimization process and can incorporate more information from the music input sequence x through skip connections. Furthermore, this method has better performance than simply adjusting the proportion of input information in skip connections, because each recursive step can essentially build a different characteristics distribution, and the recursive structure can learn the adaptive ratio of the music input sequence x and the multi-head attention mechanism $Attention(Q, K, V)$. The ratio of the input x and $Attention(Q, K, V)$ is $x/Attention(Q, K, V) = \sigma_1/\omega_1 + 1$, where σ_1 is the standard deviation of $Attention(Q, K, V)$ and ω_1 is the inner normalization gain parameter.

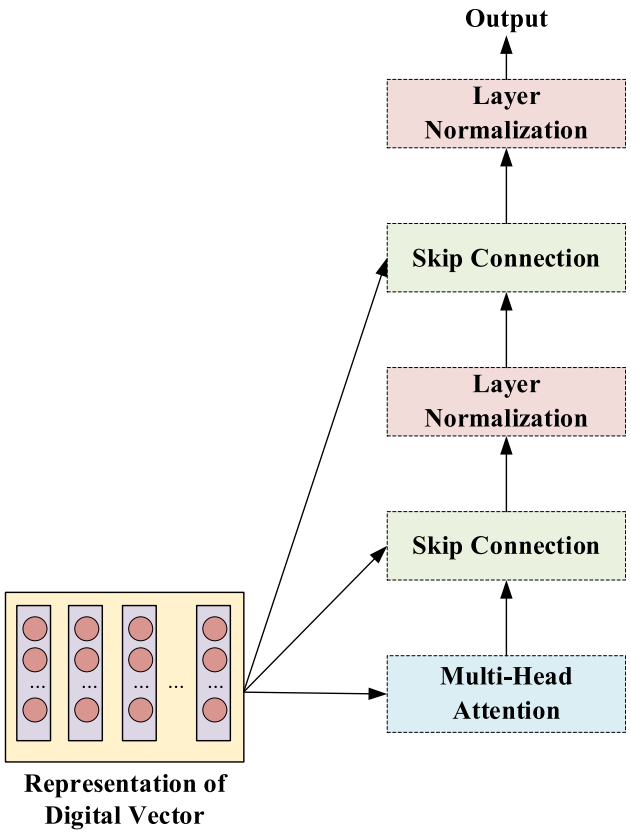

Fig. 6 Recursive skip connections with layer normalization

4.2 RSCLN_Transformer-XL music automatic generation process

The process of the proposed automatic music generation method is shown in Fig. 7: (1) Collecting the MIDI music data training set; (2) Performing online dynamic data enhancement on the MIDI training set, selecting part of the data in the training set for fine-tuning; (3) Parse the input MIDI through miditoolkit to obtain music information; (4) Convert the obtained music information into a REMI event sequence, and convert it by comparing the reference files; (5) Train the RSCLN_Transformer-XL music automatic generation model; (6) Use the trained model to generate music.

5 Experimental setup

The RSCLN_Transformer-XL music automatic generation network model is trained on a training set consisting of 775 MIDI music [26]. The dataset is popular piano music composed of Japanese anime, Korean pop songs and western pop songs. The average length of songs is about 4 min, with a total of about 48 h. All songs are in 4/4 beats (4 beats per bar). We split dataset into train–test–validation sets with a ratio of 8: 1: 1.

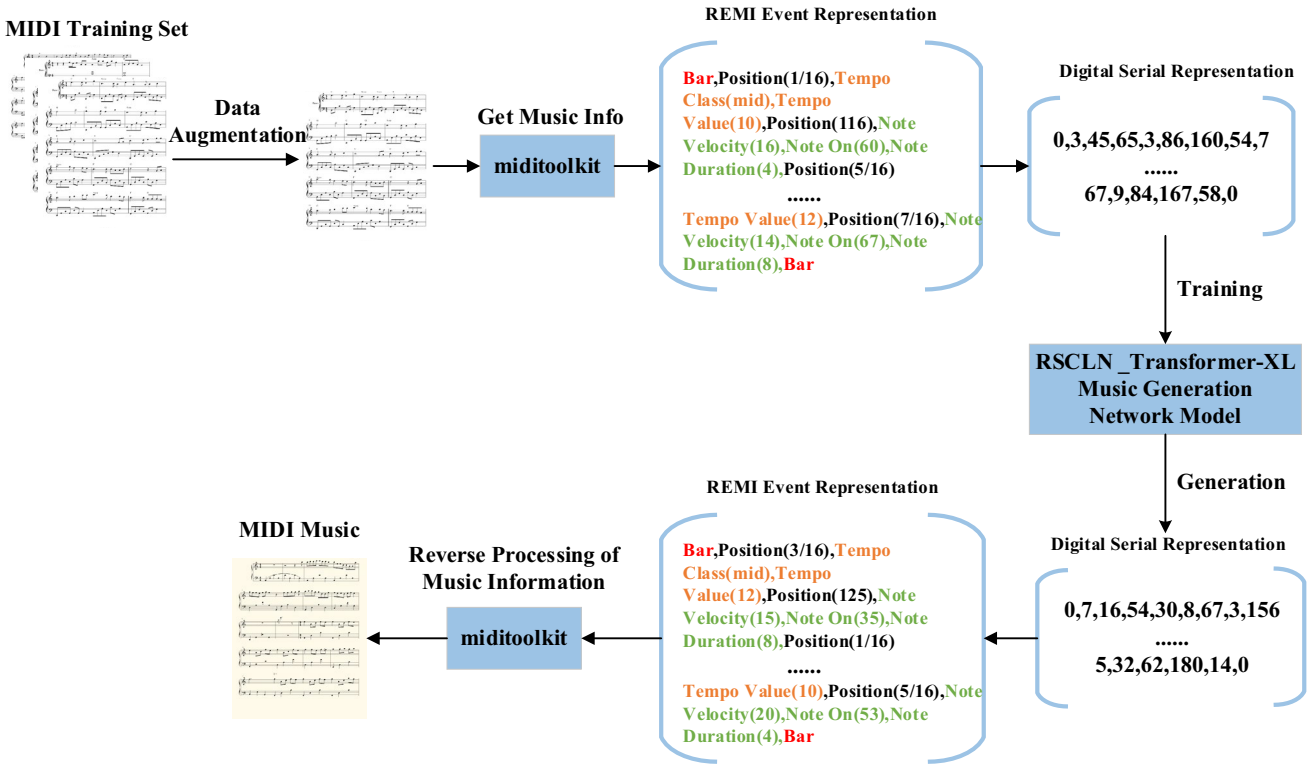

Fig. 7 Automatic music generation process

Our model was implemented using PyTorch 1.7.0. We employed the cross-entropy loss function [27] to measure the difference between two probability distributions, specifically the difference between predicted and true probability distributions. To minimize the loss during training, we utilized the Adam optimizer with a learning rate of 1e-5 and trained the model for 600 epochs. The RSCLN_Transformer-XL model employed cosine annealing learning rate [28] to control the parameter update rate and mitigate training issues caused by excessively large or small learning rates. Our complete model was run on hardware equipped with an NVIDIA GeForce RTX 1080 Ti GPU. Algorithm 5–1 illustrates the pseudo-code during the training of the RSCLN_Transformer-XL model.

Algorithm 5–1 RSCLN_Transformer-XL music generation model

1. Definition: RSCLN_Transformer-XL network model parameter θ_T , MIDI music training set and number of training rounds *epoch*.
2. Data augmentation for MIDI music;
3. Convert MIDI music files to REMI-type music event vectors;
4. Parameter θ_T is initialized randomly
5. $N \leftarrow epoch$
6. for $i = 0 \rightarrow N - 1$ do
7. REMI-type musical event sequences are input to an automatic music generation network model;
8. Calculate the cross-entropy loss value after each forward propagation;
9. Adam optimizer adjusts network model parameter θ_T through loss value;
10. End for
11. Training of automatic music creation model .

6 Music generation similarity evaluation

To verify RSCLN_Transformer-XL music automatic generation network model generates music. This paper will adopt objective and subjective evaluation methods to conduct similarity evaluation experiments on Transformer-XL, Transformer-XL + RSCLN, Transformer-XL + DA, and Transformer-XL + RSCLN + DA.

6.1 Objective evaluation of music generation and results

6.1.1 Objective evaluation index of music

The objective evaluation of music generation is an effective method to evaluate the task of music generation. This paper demonstrates the validity of RSCLN_Transformer-XL model from three aspects: pitch-related metrics, rhythm-related metrics, and music structure indicators.

(1) Pitch-related metrics.

Pitch range (PR).: It is defined as the range of pitch.

Polyphony (PP).: Used to measure the frequency at which at least two tones are played simultaneously.

Polyphony rate (PPR).: It is defined as the playback rate at which two tones are played simultaneously.

Pitch-in-scale rate (PSR).: It is defined as the rate of pitch change in the scale.

Scale consistency (SC).: It is obtained by calculating the pitch proportion of all standard scales and reporting the proportion of the best matching scale.

Pitch-class histogram entropy (PH).: In information theory, entropy is a measure of probability distribution [29], it is used as an index of pitch distribution in music. Collecting notes within a bar, build a 12-dimensional pitch-class histogram $\vec{h}$ according to notes' pitch classes (such as C, C#, …, A#, B). After normalization through the total number of notes in the bar, explore the usage of different pitches. The entropy of the pitch histogram $\vec{h}$ is calculated by formula (11).

$$\mathrm{H}(\vec{h}) = -\sum_{i=0}^{11} h_{i\log_2(h_i)} \tag{11}$$

(2) Rhythm-related metrics.

Empty-beat rate (EBR).: It represents the ratio of empty beats in the rhythm of music.

Grooving pattern similarity.: The rhythm pattern similarity index is used to represent the rhythm in the music. A rhythmic pattern indicates there is at least one note begin within a bar, denoted by $\vec{g}$. Formula (12) defines the similarity between a pair of rhythmic patterns $\vec{g}^a$ and $\vec{g}^b$, Q represents the dimension of $\vec{g}^a$, $\vec{g}^b$, $XOR(.,.)$ represents the exclusive OR operation, and the value of $\mathcal{GS}(.,.)$ is always between 0 and 1.

$$\mathcal{GS}(\vec{g}^a, \vec{g}^b) = 1 - \frac{1}{Q}\sum_{i=0}^{Q-1} XOR(g_i^a, g_i^b) \tag{12}$$

(3) Structure indicators (SI).

Structure of music is caused by the repetitive musical content in a composition, and it can involve multiple granularities, from instant musical ideas to entire sections. From a psychological point of view, the repetitive structures is the nature of catchy and thought-provoking of music [30].

The structural indicators in this paper are based on fitness plot, aiming at capturing the most significant repetition in a specific duration interval, indicating the state of repetitive structure in music. For the sake of concise mathematical expression, we assume that the sampling frame rate of fitness plot matrix S is 1 Hz, and define the structure index as shown in formula (13).

$$SI_l^u(S) = \max_{\substack{l<i<u \\ 1\le j\le N}} S \tag{13}$$

where l, u is the lower and upper limit of a segment duration interval (in seconds), N is the segment duration in seconds, and i and j represent the subscripts of the matrix S. In our experiment, we choose SI_3^8, SI_8^{15} and SI_{15} structural indicators to represent the short-term, medium-term, and long-term structure of music respectively.

Table 2 The mean value of each evaluation dimension index

Model	Pitch-related						Rhythm-related		SI		
	PR	PP	PPR	PSR	SC	PH	EBR	GS	SI_short	SI_mid	SI_long
Baseline	58.33	6.0218	0.9335	0.6288	0.9600	2.8793	0.0834	0.7086	0.3962	0.3971	0.3651
Transformer-XL	33.03	3.2269	0.5498	0.6721	0.9899	2.7984	0.1394	0.8016	0.2879	0.0971	0.0231
Transformer-XL + RSCLN	35.66	3.5972	0.6166	0.6757	0.9899	2.8021	0.1178	0.7941	0.2799	0.0974	0.0251
Transformer-XL + DA	38.42	3.9208	0.6762	0.6751	0.9861	2.8557	0.0991	0.6241	0.2879	0.1022	0.0427
Proposed model	39.96	4.0071	0.6957	0.6813	0.9876	2.8039	0.0839	0.6811	0.2959	0.1036	0.0414

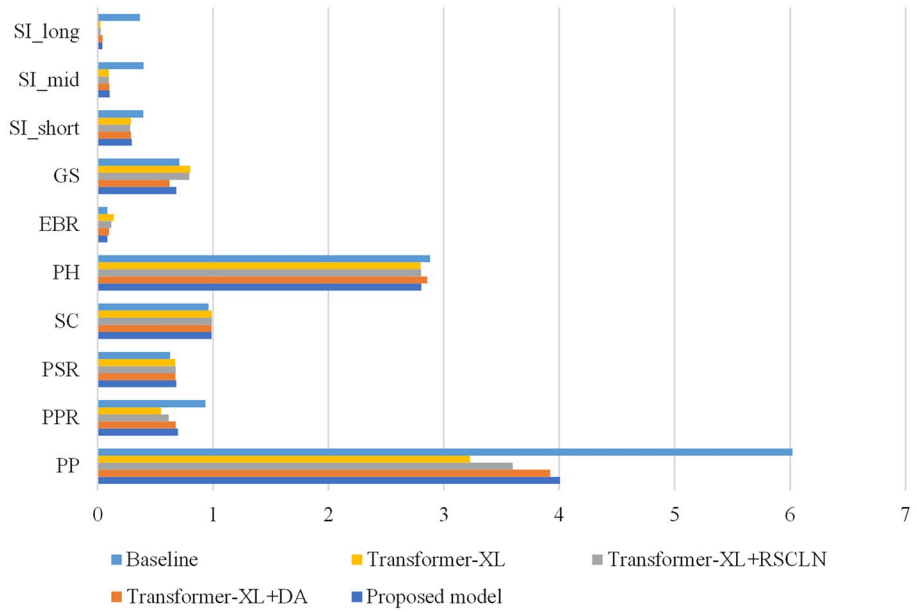

Fig. 8 Bar chart of the mean value of each evaluation dimension index

6.1.2 Analysis of objective evaluation of experiment results

In the experiment, 100 pieces of music are generated by the trained model respectively. The mean values of the music generated by four trained model in each evaluation dimension are shown in Table 2 and Fig. 8.

In Table 2 and Fig. 8, baseline represents the average value of training set data in each evaluation dimension. From Table 2 and Fig. 8, it can be seen that the music generated by RSCLN_Transformer-XL model has higher similarity with the training set data.

In order to intuitively obtain the similarity between the generated music and the training set, calculate the difference between the music generated by each model and the data in each dimension of the music in the training set, as shown in Table 3.

Based on Table 3 and Fig. 9, it can be seen that the music generated by Transformer-XL combined with RSCLN or DA is better than the music generated by Transformer-XL network model alone. By comparing the music data generated by four different models with the data of the training set through eleven evaluation metrics (PR, PP, PPR, PSR, SC, PH, EBR, GS, SI_short, SI_mid, SI_long). The results show that the difference between the music obtained by using RSCLN_Transformer-XL model and the training set is smaller than that of the other three models. Therefore, the structure of the music generated by the RSCLN_Transformer-XL model is closer to the structure of the music in the training set.

It can be seen from the objective data that the music generated by the RSCLN_Transformer-XL music automatic generation network model proposed in this paper is more similar to the training set.

6.2 Subjective evaluation of music generation

Listening test is used for the subjective evaluation. During the experiment, after listening to a piece of music, subjects need to judge the music is created by human or generated by a computer, and should give score to the music according to their subjective feelings. A total of 65 people participated in the experiment, including 40 professionals and 25 non-professionals. Among them, the professionals are composed of music teachers and music college students. Evaluation results given by subjects to music pieces are shown in Fig. 10 and Table 4.

Table 3 The absolute value of the difference between the music data generated by each model and the training set data

Model	Pitch-related						Rhythm-related		SI		
	PR	PP	PPR	PSR	SC	PH	EBR	GS	SI_short	SI_mid	SI_long
Transformer-XL	25.3	2.7949	0.3837	0.0433	0.0299	0.0809	0.056	0.093	0.1083	0.3	0.342
Transformer-XL + RSCLN	22.67	2.4246	0.3169	0.0469	0.0299	0.0772	0.0344	0.0855	0.1163	0.2997	0.34
Transformer-XL + DA	19.91	2.101	0.2573	0.0463	0.0261	0.0236	0.0157	0.0845	0.1083	0.2949	0.3224
Proposed model	**18.37**	**2.0147**	**0.2378**	0.0525	0.0276	0.0754	**0.0005**	**0.0275**	**0.1003**	**0.2935**	**0.3237**

Minimum difference results are marked in bold

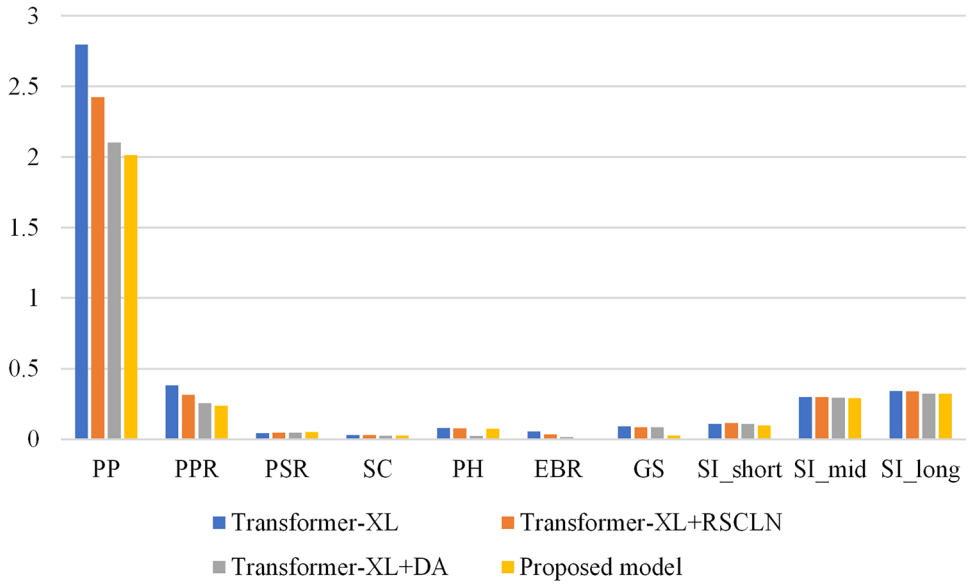

Fig. 9 The histogram of absolute value of the difference between the music data generated by each model and the training set data

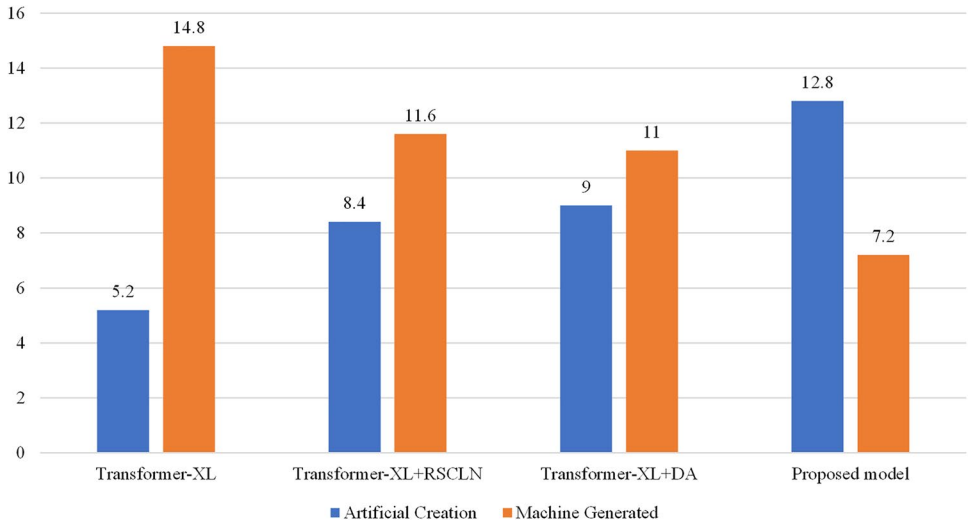

Fig. 10 The subjective evaluation results to each music generation model

Table 4 The average scores for the subjective preference test on a three-point scale from 1 (like the least) to 5 (like the most)

Average scores	Transformer-XL	Transformer-XL + RSCLN	Transformer-XL + DA	Proposed model
Professionals	2.84	3.61	3.88	**4.15**
Non-professionals	3.22	3.72	3.91	**4.09**
All participants	2.99	3.65	3.89	**4.13**

Maximum scores are marked in bold

As Turing test results shown in Fig. 10, an average of 83.38% subjects judged that music generated by the Transformer-XL model was generated by machine, and only 16.62% of subjects believe that it was created by human. While an average of 84.62% subjects judged the music generated by the RSCLN_Transformer-XL model was created by human, only an average of 15.38% believed that it was generated by machine. This indicates that the music generated by the RSCLN_Transformer-XL model is more realistic.

Table 4 shows the average score of subjective preference test. It can be seen that both professionals and non-professionals have higher satisfaction with the model proposed in this paper. In the average scores of all participants, the average

score of music generated by Transformer-XL model is only 2.99, while the average score of music generated by RSCLN_Transformer-XL model is 4.13. Experiments show that the music generated by RSCLN_Transformer-XL model is easier to get people's satisfaction.

7 Conclusions

In order to improve quality of music generated by deep learning, an automatic music generation model RSCLN_Transformer-XL based on multi-head attention mechanism is proposed. Taking Transformer-XL model as the basic structure for automatic music generation, the multi-head attention mechanism helps process long and non-fixed musical event sequences. RSCLN is introduced into Transformer-XL model to improve the optimization process of the model and alleviate the gradient vanishing phenomenon in the model. At the same time, it can improve the ability of combining more music information and optimize performance of establishing long-term dependencies in context. Through adding MIDI music data augmentation to the model, to the fixed training set, diversity of the training set can be enhanced and training effect of the music generation model is optimized. The subjective and objective evaluation experiments to the generated music were conducted. The experimental results show that the music generated by the RSCLN_Transformer-XL model is better than the music generated by the Transformer-XL model.

Acknowledgements This work is partially supported by the National Natural Science Foundation of China (No. 62377034, 11872036), the Shaanxi Key Science and Technology Innovation Team Project (No. 2022TD-26), the Fundamental Research Fund for the Central Universities (No. GK202101004, GK202205035), the Science and Technology Plan of Xi'an city (No. 22GXFW0020), Shaanxi Science and Technology Plan Project (No. 2023YBGY158), and the Key Laboratory of the Ministry of Culture and Tourism (No. 2023-02).

Author contributions YZ contributed to conceptualization, resources, validation, supervision, writing—review and editing. XL contributed to methodology, software, visualization, writing—original draft. QL performed methodology and writing—review and editing. XW and HY performed supervision, validation, writing—review and editing. YS was involved in writing—review and editing.

Data Availability The dataset used during the current study can be obtained from reference [26].

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
2. Hemalatha, E.: Artificial music generation using LSTM networks. Int. J. Eng. Adv. Technol. **9**(2), 4315–4319 (2019)
3. Kemal, E.: An expert system for harmonizing chorales in the style of J. S. Bach. J. Logic. Program. **8**(1), 145–185 (1990)
4. Salas, H., Gelbukh, A., Calvo, H.: Automatic music composition with simple probabilistic generative grammars. Polibits. **44**(9), 59–65 (2011)
5. Feng, Y., Zhou, C.L.: Advances in algorithmic composition. J. Software. **10**(2), 209–215 (2006)
6. Cao, X.Z., Zhang, A.L., Xu, J.C.: Intelligent music composition technology research based on genetic algorithm. Comput. Eng. Appl. **44**(32), 206–209 (2008)
7. Todd, P.M.: A connectionist approach to algorithmic composition. Comput. Music. J. **13**(4), 27–43 (1989)
8. Hochreiter, S., Schmidhube, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Eck, D., Schmidhuber, J.: A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale. **103**(4), 48 (2002)
10. Li, S., Sung, Y.: INCO-GAN: variable-length music generation method based on inception model-based conditional GAN. Mathematics. **9**(4), 102–110 (2021)
11. 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 31th Association for the Advance of Artificial Intelligence Conference, pp. 212–225 (2018)
12. Yang, L.C., Chou, S.Y., Yang, Y.H.: MidiNet: a convolutional generative adversarial network for symbolic-domain music generation. In: Proceedings of the 18th International Society for Music Information Retrieval Conference, pp. 324–331 (2017)
13. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Adv. Neural. Inf. Process. Syst. **30**(13), 123–130 (2017)
14. Deng, X., Chen, S.J., Chen, Y.F., Xu, J.: Multi-level convolutional transformer with adaptive ranking for semi-supervised crowd counting. In: Proceedings of the 4th International Conference on Algorithms, Computing and Artificial Intelligence, pp. 28–34 (2021)
15. Huang, C., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A., Hoffman, M., Dinculescu, M., Eck, D.: Music transformer: generating music with long-term structure. In: International Conference on Learning Representations, pp. 364–375 (2019)
16. Huang, Y.S., Yang, Y.H.: Pop music transformer: beat-based modeling and generation of expressive pop piano compositions. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1180–1188 (2020)
17. Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., Engel, J.: Encoding musical style with transformer autoencoders. In: International Conference on Machine Learning, pp. 254–267 (2020)
18. Wu, S.L., Yang, Y.H.: The Jazz Transformer on the front line: exploring the shortcomings of AI-composed music through quantitative measures. In: Proceedings of the 21th International Society for Music Information Retrieval Conference, pp. 451–463 (2020)
19. Donahue, C., Mao, H.H., Li ,Y.E., Cottrell, G. W., Mcauley, J.: LakhNES: improving multi-instrumental music generation with cross-domain pre-training. In: Music Information Retrieval Conference, pp. 685–692 (2019)

20. Oore, S., Simon, I., Dieleman, S., et al.: This time with feeling: learning expressive musical performance. Neural Comput. Appl. **32**(4), 955–967 (2020)
21. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.: Transformer-XL: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2978–2988 (2019)
22. Liu, F., Ren, X., Zhang, Z., Sun, X., Zou, Y.: Rethinking Skip Connection with Layer Normalization in Transformers and ResNets. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1324–1332 (2020)
23. Zhang, B., Sennrich, R.: Root mean square layer normalization. NeurIPS. **13**(27), 12360–12371 (2019)
24. Xiong, R., Yang, Y., He, D., Zheng, K., Liu, T.Y.: On layer normalization in the transformer architecture. In: International Conference on Machine Learning, pp. 10524–10533 (2020)
25. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 464–468 (2018)
26. Huang, Y.S., Yang, Y.H.: Pop music Transformer: Beat-based modeling and generation of expressive pop piano compositions. In: ACM Multimedia, pp. 1180–1188 (2020)
27. Ma, N., Zhang, X., Liu, M., et al.: Activate or not: learning customized activation. Comput. Vision Pattern Recogn. **21**(5), 145–157 (2020)
28. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. Comput. Sci. **46**(7), 122–127 (2014)
29. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(4), 623–656 (1948)
30. Levitin, D.J.: This is your brain on music: the science of a human obsession. Plume/Penguin, New York (2006)