International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-20

https://doi.org/10.70917/ijcisim-2026-0043
Article

A Melody Generation Algorithm for Popular Music Based on Multidimensional Spectral Analysis and Its Innovation on Compositional Patterns

Hongxu Kang *

School of Music and Dance, Fuyang Normal University, Fuyang, Anhui, 236041, China; k13965563289@163.com

Abstract: In this paper, by performing Fourier transform on the original signal of the musical melody, obtaining its audio signal, and arranging the result to form a speech spectrogram, and utilizing multidimensional spectral analysis method for this to extract the musical melody. On this basis, a convolutional neural network model is added to extract the musical melody by means of the temporal harmonic map convolutional network model. After the pre-training of BERT model, the new musical melody is generated by Transformer, which defines the audio symbols and assists in the musical arrangement. The total number of segments above 1 point in the music clips generated using the model of this paper are 1349, 1047, and 687, respectively. More than 90% of the passages were audibly acceptable, indicating that the structure of the melodic flow generated in this paper has a high degree of similarity to the musical chords, and is subjectively recognized by the listener. The violin audio and the flute audio reached their maximum values in the first and second frames respectively, with the highest frequencies of 0.465 and 0.0198, respectively, and the violin timbre was richer.

Keywords: fourier transform; multidimensional spectral analysis; convolutional neural network modeling; musical melody; musical arrangement

1. Introduction

Arranging and orchestrating is an essential part of pop music production, and traditional accompaniment arranging requires mastering the principles of using various instruments and spending a long time [1-2]. In order to further save labor and time costs, and reduce the threshold of music creation, the introduction of intelligent algorithms to generate melodies and accompaniment to assist musicians in creating pop music has become a mainstream in the field of music creation [3-5]. Music AI combined with intelligent algorithms can automatically generate music melodies, harmonies, arrangements, etc., thus improving the efficiency and quality of music creation [6].

From the perspective of music creation, intelligent algorithms to create music breaks the traditional music creation mode, jumps out of the usual music creation thinking, brings more creative inspiration to music creators, and becomes an auxiliary tool for music creation and analysis [7-10]. On the other hand, the traditional way of composing music requires more professional music knowledge, computer-generated music reduces the threshold of music creation, and helps non-professional music enthusiasts to complete the music production [11-12]. In terms of music accompaniment generation research, a complete music production no longer requires a professional team to arrange music for different instruments, which to a certain extent reduces the music creation cycle, simplifies the music creation process and saves human resources at the same time [13-14]. Although artificial intelligence algorithms have already caught up with humans in some fields, it is undeniable that the current level of computers cannot surpass human artists in artistic creation [15-16]. Art injects soul to machine and machine brings inspiration to art, so it is only important to explore the generation of music melody

supported by intelligent algorithms and its innovation to composition mode [17-18].

In this paper, music audio signals are analyzed based on time domain analysis and frequency domain analysis methods under multidimensional spectral analysis. The conversion between the two is categorized as a speech spectrum, and the phase spectrum is used to calculate the instantaneous frequency, which reflects the true frequency of a single signal at a given moment. The harmonic information and timing information in the popular music melody are extracted and associated with the convolutional neural network model in the deep learning model to form a graphical convolutional network model based on timing harmonics to extract the popular music melody. The extracted music melody is preprocessed using BERT, and the new popular music melody is generated by the MMGPNet model. Define the basic music theory knowledge, mark the note symbols, key positions and harmonies sequentially, and complete the pop music composition through the sheet music representation and arrangement. Evaluate the results of music melody generation through online + offline performance evaluation mode, and at the same time, analyze the score spectrum of the generated music melody.

2. Popular Music Melody Generation Based on Multidimensional Spectral Analysis

2.1. Music Melody Extraction Based on Multidimensional Spectral Analysis

2.1.1. Speech Spectrum Analysis

Time-domain and frequency-domain analysis are the main tools for speech analysis, as well as the analysis of music audio signals, but both have certain shortcomings, as the time-domain analysis does not allow a direct understanding of the frequency of the sound, and in the frequency-domain, there is a lack of analysis of the change of the speech signal in time [19-20]. Sound is a time-varying signal, and therefore its spectrum also changes over time. In the description of acoustic characteristics, usually in the original signal with a fixed window length of the sampling window and the window shift for the Fourier transform, so as to focus only on each of the "frame" as the time unit of the signal, the results of the Fourier transform of each frame is arranged to form the speech spectrum, which through the two-dimensional time and frequency axes represent the three-dimensional information, which represents the change in the spectral state of speech over time, and the frequency of the speech spectrum. It represents the spectral state of speech over time, and the intensity of the frequency component is represented by the brightness of the corresponding position.

2.1.2. Phase Spectrum

A spectrum of speech can be obtained by converting between the time and frequency domains, but the spectrum obtained in this way is not a complete representation of the overall information, mainly because the spectrum only represents the amplitude of the corresponding sinusoidal wave, and does not contain phase information. In a sinusoidal waveform $A\sin(wt+\theta)$, amplitude, frequency and phase information are indispensable. The value obtained from the Fourier transform is the complex number $a+i\omega$, and the variation of the phase φ of each component in the spectrogram with the angular frequency is called the phase spectrum of the signal. The phase spectrum is calculated as shown in equation (1):

$$\varphi = \arg\tan\left(\frac{\alpha}{\omega}\right) \tag{1}$$

In audio analysis, the phase spectrum is often used to calculate the instantaneous frequency (IF). Mathematically, the instantaneous frequency is defined as the rate of change of the phase angle with respect to time, and VanderPol defines the instantaneous frequency as shown in equation (2):

$$w(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt} \tag{2}$$

The instantaneous frequency truly reflects the true frequency of a single signal at a given moment in time, so calculating the instantaneous frequency also plays a large role in the correct pitch of a melody frame at a given moment in time.

2.2. Deep Learning Overview

2.2.1. Convolutional Neural Networks

(1) Convolutional layer

The convolutional layer realizes feature extraction and generates feature maps through convolutional operations, which is the most crucial part of the convolutional neural network. The convolution kernel can be regarded as a two-dimensional array, in the convolution operation, first cover the convolution kernel on a position of the input data, and multiply the value in the convolution kernel with the corresponding value in the input data, and then add up the obtained values to get the feature value of the position, and finally obtain the feature value of each position to generate the feature map.

(2) Pooling layer

In convolutional neural networks, a pooling layer is usually added between adjacent convolutional layers to filter the features of the output of the convolutional layer, thus effectively reducing the number of parameters in the network, allowing the model to converge faster and prevent overfitting problems. Commonly used pooling models include maximum pooling, average pooling, random pooling, spatial pyramid pooling, and so on.

(3) Fully connected layer

The fully connected layer is usually located at the end of the convolutional neural network, and its function is similar to a classifier. Generally, before the fully connected layer, the feature matrix will be expanded into the form of a vector, and each neuron in the fully connected layer will be fully connected to the vector, so that it can better capture the feature information left after convolution and pooling, and get the classification results [21].

2.2.2. Long and Short-Term Memory Networks

In the structure picture, at any moment t, x_i is the input of the previous kind of transcendental neuron, and h_{i-1} denotes the received input of the pre - neuron. y_i is the output of the current neuron state, and h_i is the output passed to the next neuron. It can be seen that at each moment t the total input is the output of the moment $t-1$ plus the input of the moment t. h_i is used to map y_i to the linear layer by dimensionality, and then the classification is performed using the softmax function to obtain the desired data. Where h_i and y_i are represented as shown in equation (3), (4):

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1}) \tag{3}$$

$$y_t = g(W_{ky}h_t) \tag{4}$$

where W_{hy}, W_{xh}, W_{hh} are the weight matrices, W_{hy} is related to the hidden state h_t, W_{zh} is related to input x_i, W_{kh} is related to h_{t-1} at the moment of $t-1$, $f(u)$ and $g(v)$ They are the hidden layer and the output layer activation function.

However, the traditional RNN cannot fully utilize the historical information, so in order to better utilize the temporal information in the spectrogram, a long-short-term memory unit (LSTM) can be used for memory. Compared with RNN, LSTM adds three gates: forgetting gate, input gate, and output gate. The forgetting gate selectively removes some information from the cell, the input gate determines how much new information has been added to the cell, and the output gate outputs an output value based on the state of the cell. The inputs to the gates are controlled by weighting them to determine how much the input of $0 \sim t-1$ before t near and far from this point in time has an effect on the input of t. Each gate has a sigmoid nonlinear gating unit capable of compressing the inputs between 0 and 1.

The nodes of the LSTM can be computed by the mathematical recursion of (5)~(10), where $t = 1, 2, ..., T$ denotes the time:

$$i_t = \sigma(W^{(x)}x_t + W^{(hi)}h_{t-1} + W^{(ci)}c_{t-1} + b^{(i)}) \tag{5}$$

$$f_t = \sigma(W^{(xf)}x_t + W^{(hf)}h_{t-1} + W^{(cf)}c_{t-1} + b^{(f)}) \tag{6}$$

$$g_t = \tanh(W^x x_t + W^{(hc)} h_{t-1} + b^{(c)}) \tag{7}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{8}$$

$$o_t = \sigma(W^{(xo)} x_t + W^{(ko)} h_{t-1} + W^{(co)} c_t + b^{(o)}) \tag{9}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{10}$$

where i_t denotes the input gate at moment t, f_t, g_t denote the forgetting gates, c_t, h_t denote the long and short memories o_t denote the output gates, $\sigma(\cdot)$ denotes the stgmotd function, w is the weight matrix of each gate, and b is the bias vector.All the parameters of LSTM and RNN can be updated by back propagation algorithm (BPTT).

2.2.3. Attention Mechanisms

Attention modeling (AM) was first used in the field of machine translation, and in deep learning research, attention mechanisms have increasingly become a key aspect that helps to further improve the performance of neural networks. Figure 1 shows the graph encoder-decoder architecture.

The structural composition of the encoder-decoder model is shown in Fig. (a). Both the encoder and decoder are an RNN, the encoder reads the input sequence $(x_1, x_2, ..., x_t)$ and encodes it into a fixed length vector $(h_1, h_2, \ldots, h_t)$. The decoder takes a fixed-length vector h, as input, and decodes it into an output sequence $(y_1, y_2, \ldots, y_t)$. The h_t and s_t denote the hidden states of the encoder and decoder at moment t, respectively. However, such a structure may lead to information loss when the encoder compresses complex input information into a fixed-length vector h_t, and the decoder is unable to selectively attend to relevant input objects when generating the output sequence.

Therefore, the key of the attention mechanism is to introduce the weight of attention into the input sequence by prioritizing a set of positions with associated information. Figure (b) shows the structure of the encoder-decoder network incorporating the attention mechanism. The attention module in this structure mainly assigns different weights α_{ij} to the neurons on the vectors, which automatically obtains associations between h_i and s_j. The vector C is then output by multiplying the generated weight vector with the original vector, which is in turn used as an input to the decoder. At each decoding position C, C_j is the sum of all the hidden states of the encoder and their respective weights, which is computed as shown in equation (11):

$$C_j = \sum_{i=1}^{r} \alpha_{ij} h_t \tag{11}$$

Currently, the commonly used encoder-decoder architecture mainly uses CNNs as encoders and RNNs and LSTMs as decoders, and this architecture is especially practical in multimodal work. In this paper, we also introduce the attention mechanism into the task of melody extraction based on such an encoder-decoder framework. Its main purpose is to weight and merge the outputs of the encoder and input them into the existing decoder, so that more contextual information can be obtained from the original data and the calibration of the outputs and inputs can be guaranteed.

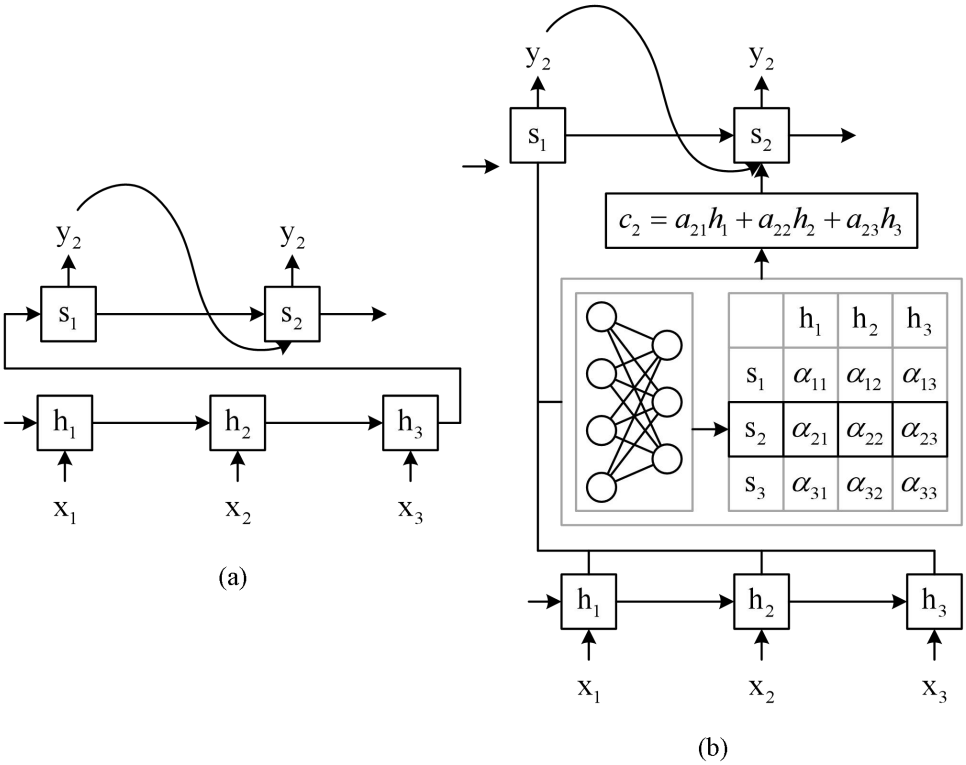


Figure 1. Encoder-Decoder Architecture (a): Traditional model (b): Attention Model.

2.2.4. Activation Functions

In deep neural networks, the nonlinear functional relationship between the nodes of the upper and lower layers of the network for mapping has is called the activation function. In recent years Relu-like functions (Leaky-ReLU, P-ReLU, R-ReLU, etc.) have been widely used in the design of neural networks because of their fast training speed and much faster convergence than sigmoid and tanh.

The ReLU function expression is shown in equation (12):

$$L = \frac{1}{N}\sum_{i} L_i = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic}\log(p_{ic}) f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \tag{12}$$

The Leaky-ReLU function (also denoted as LReLU function) is a variant of the ReLU function with the formula shown in equation (13). Compared to the ReLU function, LReLU makes the partial value of the input less than 0 negative with a very small non-zero gradient, and the value of the non-zero gradient is usually taken as 0.01:

$$f(x) = \begin{cases} 0.01x & x < 0 \\ x & x \geq 0 \end{cases} \tag{13}$$

The neural network constructed in this paper for the melody extraction task is essentially a multi-categorization of pitches the melody extraction work is actually a multi-categorization of pitches the output layer of the network can be used softmax activation function maps the domain of real numbers into the range of 0 to 1, and the probability of each category sums up to 1. The formula for the softmax function is shown in equation (14):

$$softmax(z_j) = \frac{e^{z_i}}{\sum_{K} e^{z_i}} \tag{14}$$

2.3. Music Melody Extraction Based on Temporal Harmonic Map Convolutional Network

2.3.1. Harmonic Information Extraction

The harmonic information extraction part of this algorithm uses an undirected graph to model potential connectivity relationships between different frequencies originating from the same pitch. The nodes of this undirected graph are the frequency points of the spectrum, and the edges of the graph represent the connectivity relationships between the frequency points. The undirected graph can be defined as $\mathrm{G}=(\mathrm{V},\mathrm{E})$. where V represents the set of frequency points in the spectrum of the music signal, and E represents the potential connectivity between the fundamental frequency points and their respective harmonic frequency points. X_t is the CQT spectrum at moment t, then the output of the harmonic information extraction part is [22]:

$$G_i = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X_i\theta \tag{15}$$

$$\tilde{A} = A + I \tag{16}$$

where $G_t \in \mathbb{R}^d$ is the output probability matrix, A is the adjacency matrix of the graph G, and I is the unitary matrix isotypical to A.

θ is a parameter that can be trained to update, and represents the complex mapping relationship between the input x, and the output probability G,.

2.3.2. Timing Information Extraction

The algorithm in this chapter uses GRU to extract temporal information. The most common network model for processing temporal signals is the recurrent neural network RNN, which may have the problem of gradient disappearance or gradient explosion when processing temporal information.LSTM and GRU and some other models for avoiding the above mentioned problems have come into being.The basic principles of LSTM and GRU algorithms are more or less the same, which are applying the gating mechanism to memorize the longest possible sequential information, however LSTM has a relatively complex structure, more parameters, and takes longer to train. GRU, on the other hand, has a relatively simple structure, a small number of parameters, short training time and other advantages. So the algorithm in this chapter uses GRU to extract the timing information.

2.3.3. Time-Series Harmonic Map Convolutional Networks

In order to simultaneously obtain the temporal and harmonic information in the music signal spectrum, the algorithm in this chapter combines the harmonic graph convolution network (HGCN) with the gated recurrent unit (GRU) to form a new network model temporal harmonic graph convolution network (THGCN). Its overall flow is shown in Fig. 2. The computational kernel is shown in Fig. 3.

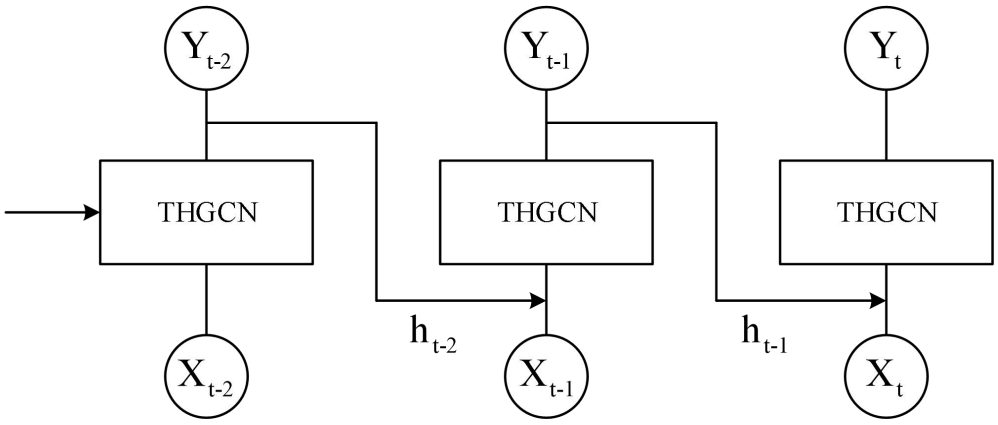


Figure 2. Temporal harmonic graph The overall process of the convolutional network.

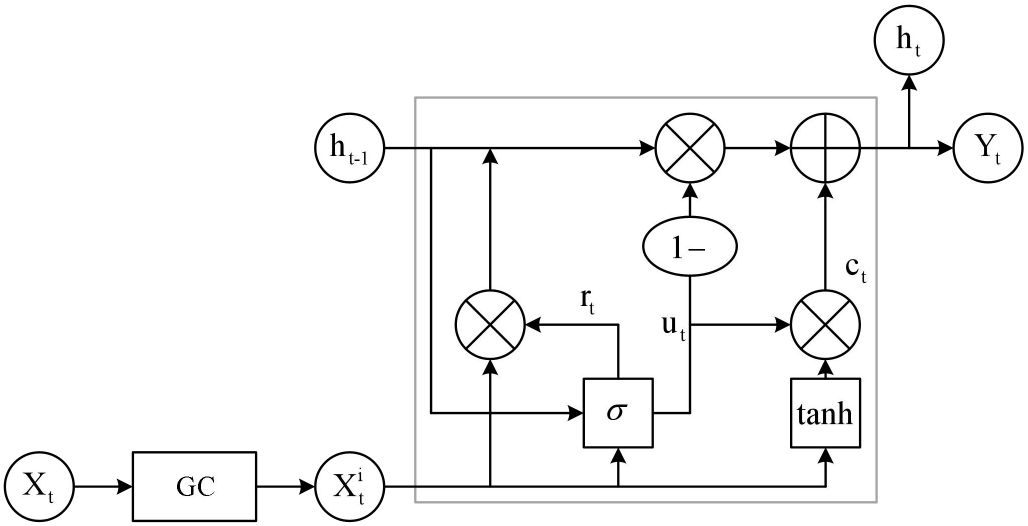


Figure 3. Specific Structure of the THGCN unit.

The specific calculation procedure for the network model is as follows:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot c_t \tag{17}$$

$$u_t = \sigma(V_{gz} G_t + U_{hz} h_{t-1} + b_u) \tag{18}$$

$$r_t = \sigma(V_{gr} G_t + U_{hr} h_{t-1} + b_r) \tag{19}$$

$$c_t = \varphi(V_{gc} G_t + U_{hc}(r_t \odot h_{t-1}) + b_c) \tag{20}$$

where V_g, V_{gr}, V_{gc} are the weight parameters of the update gate, reset gate and the next GRU that are connected to the HGCN, U_{kz}, U_{hr}, U_{hc} are the weight parameters of the current time t and the previous time $t-1$, b_z, b_t, b_c are the update gates , reset the bias parameters for the gate and connect to the next layer of GRU, $\sigma(\cdot)$ is the Sigmond activation function, and $\varphi(\cdot)$ is the Tanh activation function.

2.4. Popular Music Melody Generation Based on MMGPNet Modeling

2.4.1. BERT-Based Pre-Training Model for Music

(1) Zero-filling

Since the trained data needs to be kept of fixed length, for the single track orbital sequence of the score, some empty positions of insufficient length are filled by the algorithm using the fill symbol PAD. For the composite symbol representation, each symbol has 6 attributes, so the positions with insufficient length need to be filled for each of the 6 attributes. In order to differentiate with the value range of each of the 6 attributes, set bool (PAD) = 3, p (PAD) = 128, v (PAD) = 121, d (PAD) = 81, b (PAD) = 101, o (PAD) = 65, and PAD means zero filling.

(2) Embedded modules

The single-track sequence of the composite representation is first received as input. For each symbol of the input sequence multiple attributes are obtained as embeddings separately through their respective linear embedding layers, and then these multiple embeddings are merged Concat, and finally the symbol embedding x with all the attribute information is obtained through a linear layer Linear.

The positional embedding layer Emb_{pos} algorithm uses the relative positional embeddings for encoding to obtain the next embedding x_{emb} with positional information as shown in Equation (21):

$$x_{emb} = Emb_{pos}(x) = x + pos(x) \tag{21}$$

where: pos is the relative position embedding encoding and x is the output of the symbol embedding

layer.

(3) BERT music pre-training module

The music pre-training module uses a slightly modified BERT model as the model backbone, which is a classical multilayer bi-directional Transformer encoder with 6 layers of multi-head self-attention, 16 heads in each layer, and the dimension of the hidden space of the self-attention layer is 1024.The position-encoded embeddings are fed into the multi-head attention layer, and the hidden vector hid is obtained at the output as shown in Eq. (22):

$$hid = \max(x_{emb}) \tag{22}$$

where mal is the multiple attention layer.

The obtained hidden vectors are fed to the dense layer Prob to predict the masked attribute symbols, and the predicted probability distribution prob is obtained as shown in equation (23):

$$prob = [prob^{bool}, prob^{p}, prob^{v}, prob^{d}, prob^{b}, prob^{o}] = \Pr ob(hid) \tag{23}$$

For the music pre-training module of the model, the pre-training task is needed to set the objective function for learning. Therefore, a reconstruction task is constructed to train the music pre-training module. Given a composite symbol $y_j^i = [y_j^{bod}, y_j^p, y_j^v, y_j^d, y_j^b, y_j^o]^i$ ($i = 1, 2, \cdots, n_t$ and $j = 1, 2, \cdots, len_i$) the symbol is from the j th position of the i th track sequence, the algorithm chooses to randomly mask 15% of the attributes (type, pitch, volume, duration, measure, and intra-measure offsets) before reconstructing the attributes of the masked symbol. For each different instrument, an instrument-specific music pretraining model is trained. As for the masked symbols, 80% of the masked symbols are used MASK masked symbols, 10% are used randomly selected symbols, and the remaining symbols are left unchanged.

2.4.2. Transformer-Based Generation Module

(1) Embedding module

Chord embedding transforms a composite chord symbol into a chord symbol embedding, for which multiple attributes of the chord symbols of the input sequence are obtained separately through their respective linear embedding layers, and then these multiple embeddings are merged Concat, and finally obtained through a linear layer Linear operation.

Under the condition of using the music pre-training module, the track embedding module of the generative model is initialized using the music pre-trained weights.

(2) Coding and decoding layers

A generator contains an encoder and a decoder that are responsible for generating the symbols of the tracks, the encoding layer and the decoding layer form the generator module at the same time. For a multi-track music with n_t tracks, the encoding layer has n_i encoders $E^1, E^2, \cdots, E^{n_t}$ and one E^c , and the decoding layer has decoders $D^1, D^2, \cdots, D^n$. In this module, in order to capture the symbolic dependencies between different tracks, the outputs of all the encoders are fused and converted into the inputs of the decoders by means of a joint decoding technique.

(3) Output Layer Module

In the model output module, the outputs of each decoder corresponding to each track are fed into the feed-forward neural network (FNN) and processed by a normalized exponential function, Softmax, to obtain a probability distribution of the current possible symbol contents to be generated. The model will decide whether the newly generated symbol type is a note symbol or a key position symbol based on the first probability value of the probability distribution, and then decide the values of the other attributes within the symbol by probability based on the subsequent probability values of the probability distribution for the weight distribution. This results in a multi-track output $y_{t_t}^i (i = 1, 2, \cdots, n_t)$, where t_i denotes the sampling time corresponding to track i .

(4) Training of generative models

MMGPNet is trained using a TEACHER-FORCING strategy for MMGPNet. In each training step, a multi-track symbol sequence of real dataset data is used as input to each track embedder separately, and the model predicts the next symbol content of the sequence and computes the loss. In this way, the model learns the dependency of the currently generated symbols on the previous content from any track in the previous step or the current step. The training is done using a cross-sense entropy loss function, and the

loss value is the sum of the loss values for each attribute of the symbol.

(5) Generation Algorithm

As the model generates a symbol for only one of the multiple tracks per prediction. During the generation process, in order to be able to rely on the content already generated by other tracks and its own track, at the beginning of each generation step, the algorithm selects the track that has the least playing time for the currently generated content to be generated. This also allows for a more balanced playing time across tracks.

2.5. Evaluation of Music Melody Generation Results

2.5.1. On-Line Evaluation

The online audition effect scoring platform adopts a front-end and back-end separation development method, with the front-end developed in React and the back-end in Java.After the platform development is completed, 10 test piano songs are placed on the platform, 2 of which are from the Demonstration Audio Collection, with the composer being SM, and 3 of which are from the midishow website, which is a professional music sharing and communication platform, where users can upload their own created music and 5 were generated by an automated compositional neural network model. The duration of each piece of music was intercepted for 30 seconds to avoid auditory fatigue of the testers. Testers are invited to listen to the 10 piano pieces online and score them according to their subjective listening experience. The testers invited for online audition evaluation are music lovers, and the scoring method adopts a five-level scoring system.

In the “Composer” column, MS indicates that the piano piece was composed by a musician on the midishow website, and GRU indicates that the piano piece was composed by the automatic composition neural network model in this paper. The testers could only see the test music name of each piano piece, and no other information could be seen.

A total of 30 music lovers were invited to participate in the online audition evaluation, and Table 1 shows the results of the test score ranking. They needed to audition each piece of music, score it according to their subjective listening sensation, and then fill in their own invitation code and click on the submit button to complete this evaluation. The scores for each piano piece are tallied and the average score is calculated.

In the above final evaluation score results, the automatically composed and generated piece Demo_05 entered the top three in the ranking, with a score of 3.974. The scores of Demo_02 and Demo_06 were also ranked in front of the two human compositions, with scores of 3.808 and 3.801, respectively. it shows that the testers could not completely differentiate the human compositions from the automatically generated pieces in this paper, and at the same time it also shows that It also indicates that the automatically generated music in this paper meets the appreciation requirements of the testers. In addition, the scores of the lower ranked ones are also the auto-generated music of this paper, which indicates that there are differences in the quality of auto-generated music, and there is room for further optimization of the network model of auto-composition. In order to get a more professional evaluation, professionals are invited to specify multiple evaluation indexes to comprehensively evaluate each piano piece.

Table 1. Test scores.

Ranking	Test music name	Creative mode	Composer	Score
1	Demo_10	Artificial composition	SM	4.215
2	Demo_08	Artificial composition	MS	4.036
3	Demo_05	Automatic composition	GRU	**3.974**
4	Demo_01	Artificial composition	MS	3.894
5	Demo_02	Automatic composition	GRU	**3.808**
6	Demo_06	Automatic composition	GRU	**3.801**
7	Demo_04	Artificial composition	MS	3.748
8	Demo_09	Automatic composition	GRU	3.348
9	Demo_03	Artificial composition	SM	3.248
10	Demo_07	Automatic composition	GRU	2.436

2.5.2. Evaluation of Offline Performance

The offline performance evaluation invited professionals who have rich experience in piano performance. The professionals designated five indicators for this evaluation, namely "melody, texture, harmony, tension, and aesthetics", and each indicator was worth 100 points, and then 10 pieces of music were performed live, and the scores of the five indicators were recorded for each piece of music.

For the weight size of each indicator, this paper uses the entropy weight method to calculate the specific weight. The entropy method uses information entropy to determine the role size of the indicators in the comprehensive evaluation according to the information provided by each indicator, and then get the specific weight letter.

The assignment step of the entropy weight method is divided into three steps, combined with the evaluation index of this piano song, each step is briefly explained:

The first step is to normalize the data for each indicator. Define 5 indicators as X_1, X_2, X_3, X_4, X_5, where $X_t = \{x_1, x_2, x_3, x_4, \ldots, x_{10}\}$. The values after normalizing the data for each indicator are Y_1, Y_2, Y_3, Y_4, Y_5. where Y_{tj} denotes the value of the i th metric normalized to the j th piano song, calculated as follows:

$$Y_{ij} = \frac{x_{ij} - \min(X_t)}{\max(X_t) - \min(X_t)}, \tag{24}$$

In the second step, it is necessary to calculate the information incitement of each indicator with the following formula:

$$p_{ij} = Y_{ij} / \sum_{i=1}^{n} Y_{ij} \tag{25}$$

$$H_i = -\left(\sum_{i=1}^{n} p_{ij} \ln p_{ij} \right) / \ln(n) \tag{26}$$

In the above formula (26), H_i represents the entropy value of the i th index.

In the third step, according to Eq. (26), the information entropy of the five indicators of "melody, texture, harmony, tension and aesthetics" can be calculated as H_1, H_2, H_3, H_4, H_5. The weight W_i of the i th indicator is calculated as:

$$W_i = \frac{1 - H_i}{5 - \sum H_i} \tag{27}$$

After calculating the weights of the indicators, the formula for the final evaluation score of each song is as follows:

$$S_j = \sum_{i=1}^{5} X_{ij} W_i \tag{28}$$

Table 2 shows the final score ranking of the pieces, SM works ranked first and second, three of the automatically generated piano pieces in this paper ranked in front of the manually composed pieces, and the score gap with the other two pieces composed by MS is small, with scores of 81.536, 81.048, and 80.499, respectively.It shows that the automatically generated pieces in this paper can reach the level of the general manually composed ones, but the score gap is larger compared with that of the famous composers SM's works, the score gap is large. The two scores at the back of the list are also automatically generated by this paper, which indicates that there are differences in the quality of automatically generated music, and the network model of automatic composition needs to be further optimized. Professionals also commented on the monotonous style of the automatically composed music in this paper, with no complex variations and lack of "rhythm", which will be taken into account in the next study to improve the quality of automatically composed music.

Table 2. The final score of the music.

Ranking	Test music name	Creative mode	Composer	Score
1	Demo_10	Artificial composition	SM	93.485
2	Demo_03	Artificial composition	SM	89.796
3	Demo_08	Artificial composition	MS	83.615
4	Demo_01	Artificial composition	MS	82.466
5	Demo_02	Automatic composition	GRU	81.536
6	Demo_05	Automatic composition	GRU	81.048
7	Demo_06	Automatic composition	GRU	80.499
8	Demo_04	Artificial composition	MS	78.636
9	Demo_09	Automatic composition	GRU	74.539
10	Demo_07	Automatic composition	GRU	69.348

2.6. Auditory Analysis of Melody Generation in Popular Music

2.6.1. Melody Generation Results

Since the results in this paper are based on the expression form of melodic flow whose structure is different from the traditional melodic detection results, and for many reasons such as the lack of standard evaluation platforms in this field, the results are evaluated here utilizing both objective and subjective routes. First of all, we will look at the objective accuracy evaluation, taking 30ms frame length as the basic processing unit. For the result evaluation of recognizing the traditional melody line, it is defined that if the frequency of a component in the melody stream obtained at a certain moment is consistent with the base frequency of the standard melody line at this moment, it is regarded as correct, as shown in Table 3.

Since the instantaneous structure of the resultant melodic flow at each moment is a chord-like structure (superposition of chromatic components), the melodic flow is considered to have "captured" the traditional standard if, for any moment, there exists a certain chromatic component in the chord-like structure of the moment that is chromatically identical to the chromatic component of the standard melodic line. The melodic flow is considered to have "captured" the traditional standard melodic line, i.e., the result is recognized as correct with respect to the method of measuring results with melodic lines. This means that the results of the musical melodic flows generated in this paper "cover" more than 76.023% of the traditional melodic lines. Without considering the multi-candidate problem caused by the composite structure of the melodic flow, the method in this paper can recognize and generate traditional melodic lines very well.

Table 3. Discovery results of melody lines in the database.

Music type	Total frame	Correct frame	Coverage ratio
Classical	57900	47400	81.865%
Light music	57500	48050	83.565%
Rock and roll	32500	21495	66.138%
Jazz	32400	23498	72.525%
Population	180300	140443	76.023%

Table 4 shows the results of chromatic recognition. The model in this paper is dragged down by rock and jazz tunes, and the overall precision of chromatic recognition is at 59.67%, while the recall is only 65.891%. There are more misrecognitions for these two types of music, and the problem is most obvious for rock music, which has complex orchestrations and intense rhythms. Jazz, on the other hand, is affected by the greater looseness of the singer's interpretation (transposition, more off-key). Light music performed better in terms of chromatic recognition, which is directly related to its stable orchestration, soothing rhythm, and melodic prominence.

Table 4. Semi-sound recognition.

/	Classical	Light music	Rock and roll	Jazz	Population
Frame number	57900	57500	32500	32400	180300
Half tone number	160160	123450	89310	60230	433150

Correct number	112485	92780	43650	41795	290710
Error number	62348	41530	53048	27498	184424
Rejection number	48039	30640	56189	18385	153253
Recall rate	70.045%	75.185%	48.849%	69.485%	65.891%
Accuracy	64.249%	69.048%	45.198%	60.185%	59.67%

Table 5 shows the recognition of semitones without temporal harmonic extraction. Comparing the above two experiments, it can be seen that the number of misrecognized semitones increases dramatically in all types of music without post-processing of the results of temporal harmonic recognition, which directly leads to the overall accuracy dropping to 50.224%, and the rock music is even lower than the low point of 36.148%, and the number of correctly recognized semitones is also declined to different degrees, thus reflecting the most significant effect of the post-processing mechanism is to control the number of misrecognition well and to reduce the cases of refusal to recognize. The number of correctly recognized semitones also decreased to different degrees, which shows that the post-processing mechanism is most effective in controlling the number of misrecognitions and reducing the number of rejections.

Table 5. The semi-sound recognition of unorthodox sequence harmonic extraction.

/	Classical	Light music	Rock and roll	Jazz	Population
Frame number	57900	57500	32500	32400	180300
Half tone number	160160	123450	89310	60230	433150
Correct number	104798	88165	40198	39481	272642
Error number	103584	62100	70958	32105	268747
Rejection number	55349	35289	49385	20318	160341
Recall rate	65.485%	71.485%	44.948%	66.269%	62.047%
Accuracy	50.851%	58.549%	36.148%	55.348%	50.224%

2.6.2. Subjective Hearing

Obviously, it is unscientific to evaluate the strengths and weaknesses of the melody generation technology in this paper solely from the statistical point of view of the objective experimental results, because music as an art form has no meaning when it is separated from the subjective perception of human beings. Therefore, this paper adopts a subjective evaluation mechanism: firstly, the melody flow generation result is synthesized into a virtual electroacoustic piano playing section by using the scoring software Overture, and the corresponding test music is segmented according to the three time levels of 3 seconds, 4 seconds and 6 seconds, and the listener (who does not have any professional music experience) is asked to listen to and score the result of the segmented section and the original music section one by one in each time scale. Scoring. The scoring method used here is different from the traditional scoring system of three grades: 0 (not similar), 1 (acceptable), and 2 (very similar). This scoring method helps the listener to distinguish the degree of similarity in the sense of hearing, and will not confuse the sense of hearing because of too many scoring grades, thus making the scoring results more objective. Figure 4 shows the subjective listening scores for different segment lengths.

From the above table, it can be seen that in the case of the highest score of only 2 points, the scores of all types of music are above 1 (acceptable) on average, and the total number of segments with more than 1 point in the 3s, 4s, and 6s music clips are 1349, 1047, and 687, respectively, and the number of segments that are acceptable in the listening sense is more than 90% of the total, which indicates that the structure of the melodic flow generated in this paper has a high degree of similarity to the musical chords and is subjectively recognized by the listener. This indicates that the melodic flow structure generated in this paper is similar to the musical chords and is subjectively recognized by the listeners.

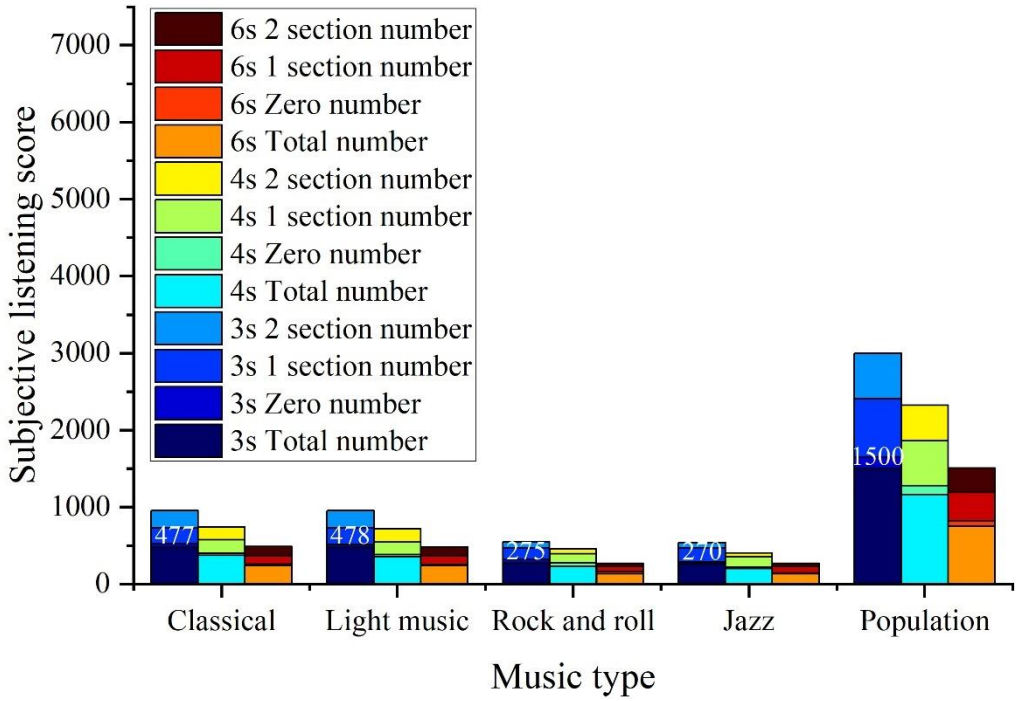


Figure 4. Subjective hearing score of different sections.

3. Spectral Composition Based on Multidimensional Spectral Analysis

3.1 Definition of Symbols

3.1.1. Note Symbols

A note symbol is denoted by Note and is written as y^n. A Note contains three attributes as shown in equation (29):

$$y^n = [y^{np}, y^{nv}, y^{nd}] \tag{29}$$

where $y^{np} \in \{0,1,2,\cdots,127\}, y^{nv} \in [0,120]$ and $y^{nd} \in \{0,1,2,\cdots,80\}$ denote the pitch, the volume, and the note duration, respectively. For the encoding of note duration, 1/8 of a quarter note is defined as the smallest basic unit of duration, and note symbols with durations less than this duration value are ignored. For multiple consecutive notes of the same pitch in the original score, a single note symbol with the same duration is used to represent them.MuseSeq does not encode rest notes.

3.1.2. Key Position Marking

A key position marker is denoted as y^k. y^k is a binary group as shown in equation (30):

$$y^k = [p^{bar}, p^{off}] \tag{30}$$

where $p^{bar} \in \{0,1,\cdots,100\}$ is the bar position and $p^{off} \in \{0,1,\cdots,63\}$ is the number of offsets within the bar. Similar to traditional position markers, the KPS is generally located in front of the note symbols in the sequence, and serves to specify the moment of performance of the note symbols in the sequence. Another role of the KPS in the generative model is to align symbols in multiple sequences in a temporal order. Unlike traditional positional markers, KPSs are inserted in front of only a small number of key notes.

3.1.3. Harmony Markers

Harmonic markers are notated as y^c and are used to indicate the type of harmony at a particular position in a sequence. Unlike note notation, the harmonic marker does not use a separate key position symbol to mark the position, but encodes the position information directly into the symbol content. Thus, the structure of the harmonic marker is shown in equation (31):

$$y^c = [p^{bar}, p^{off}, y^{ch}] \tag{31}$$

where $y^{ch} \in \{0,1,2,\cdots,83\}$ is an integer number to indicate the harmonic type. According to music theory, the harmonic type y^d is determined by two factors: the pitch of the root note and the harmonic color.

3.2. Score Representation and Arrangement

$$S = [Y^c, Y_1^t, Y_2^t, \cdots, Y_{n_t}^t] \tag{32}$$

where Y^c is the harmonic marker sequence and $Y_i^t (i = 1,2,\cdots,n_t)$ is the note sequence of the i th track.

The sequence of harmonic markers Y^c consists only of harmonic symbols, as shown in equation (33):

$$Y^c = [y_1^c, y_2^c, y_3^c, \cdots, y_{n_c}^c] \tag{33}$$

The i th note sequence contains two symbol types: the position symbol y^p and the note y^n, as shown in equations (34), (35):

$$Y_i^t = [y_1^t, y_2^t, y_3^t, \cdots, y_{n_{n_1}}^c] \tag{34}$$

$$y_t^t = y_t^p \text{ or } y_t^n \tag{35}$$

Within each track sequence, the note symbols are arranged in ascending order according to the start time of the symbol's performance. In front of each note symbol, a KPS symbol may exist to indicate the position of the note. However, it is not necessary that every note will have a corresponding KPS present in front of it.

3.3. Horizontal Cumulation of the Frequency of Musical Pieces

3.3.1. Horizontal Accumulation of Pitch Shift Frequency

The frequency of pitch transfer is counted in notes and accumulated horizontally. Fig. 5 shows the horizontal accumulation of pitch transfer frequency, and the highest horizontal accumulation frequency is found in pitch theory domains 5 and 6, and the horizontal accumulation frequency of pitch theory domains from 0 to 6 is 4→5→9→15→11→15→6→7, 4→5→9→15→10→13→11→7, respectively.

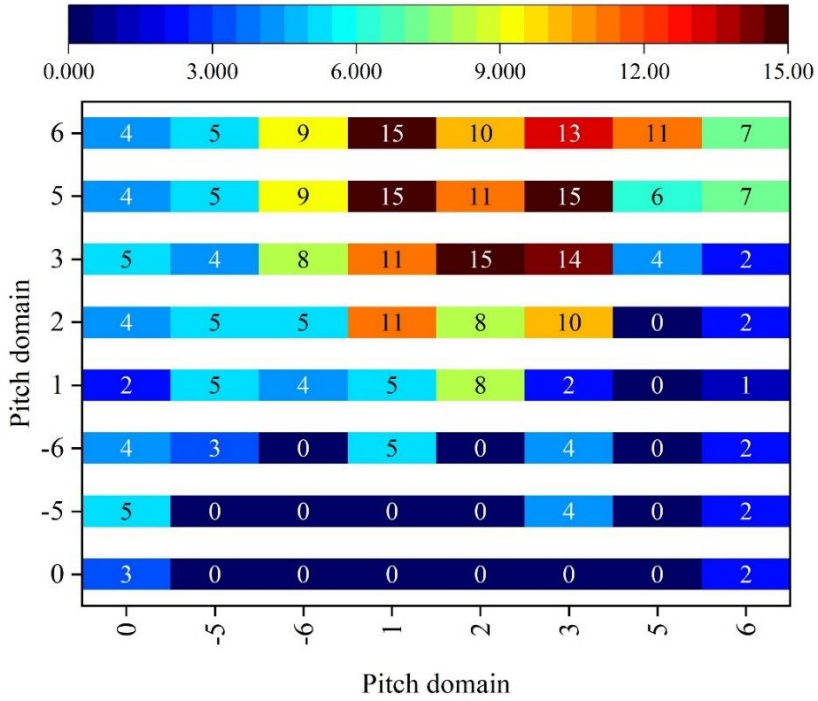


Figure 5. The horizontal accumulation of the frequency of pitch transfer.

3.3.2. Horizontal Accumulation of Shift Frequencies in Time Series

Using the same method to carry out the transverse accumulation of the transfer frequency of the time-value sequence in terms of subsections, Figure 6 shows the transverse accumulation of the transfer frequency of the time-value sequence, with the increase of the time-value sequence coding, the transverse accumulation of the transfer frequency also shows a stepwise growth, when the time-value sequence coding is 14, the transverse accumulation reaches the full coding cumulative, the cumulative value of the value of the value of the value of the sequence is from 1~3 ranges.

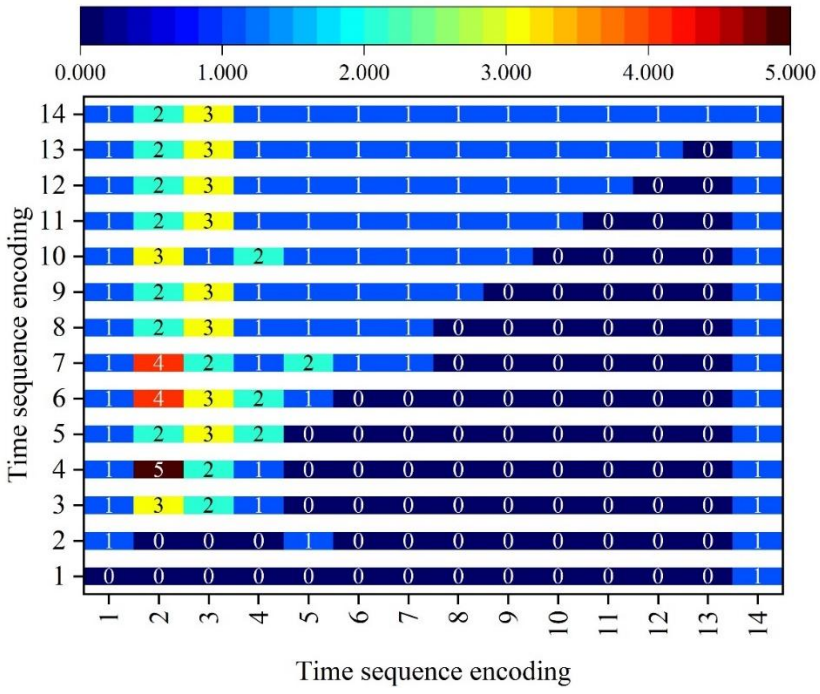


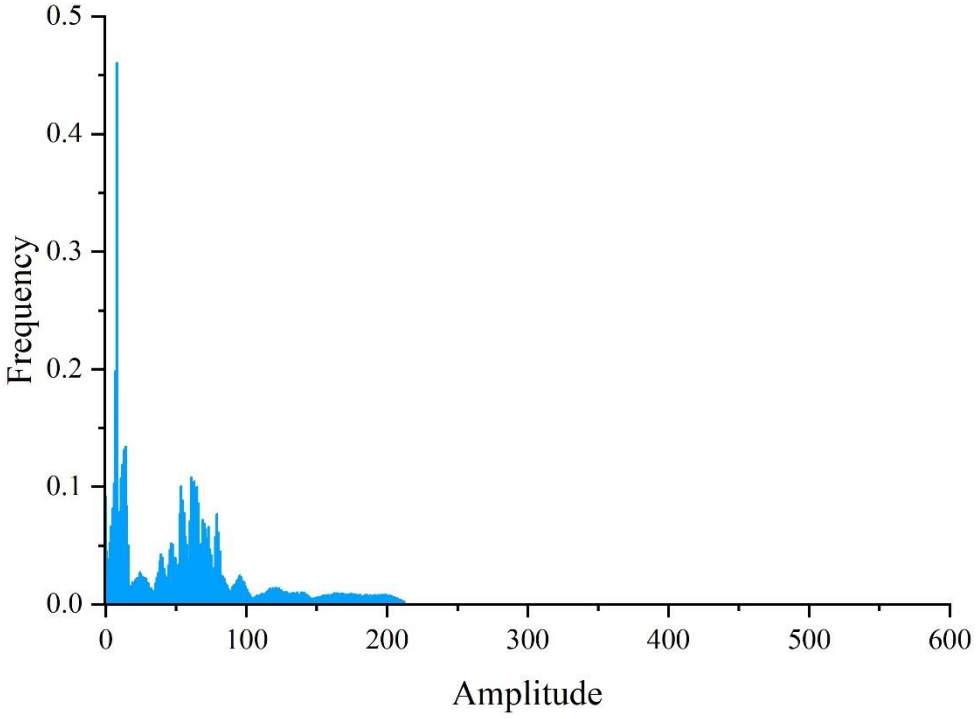
Figure 6. The horizontal accumulation of the time sequence transfer frequency.

3.4. Spectral Analysis of the Score

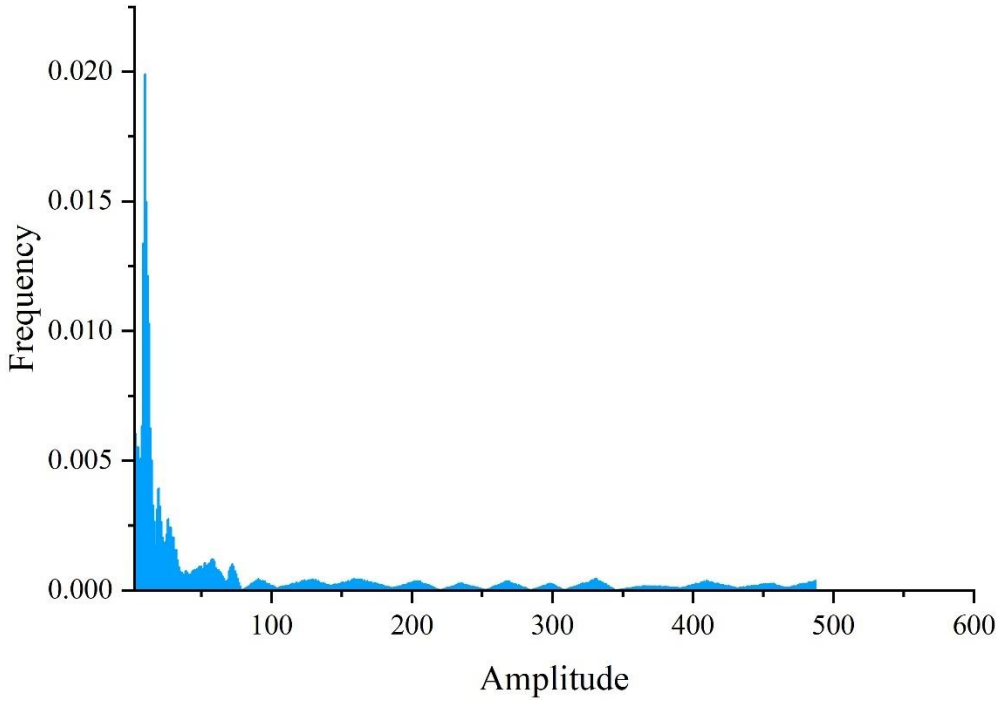
3.4.1. Single-Frame Spectral Analysis

The first frame of the violin audio and the second frame of the flute audio are taken out, and the single-frame spectrograms are plotted respectively. Fig. 7 shows the single-frame spectral analysis, Fig. (a) shows the first frame of the violin audio spectrum, and Fig. (b) shows the second frame of the flute audio spectrum.

According to the audio spectra of violin and flute, it can be clearly seen that the amplitude difference between the two is large, so their timbre and pitch are different, at the same time, the amplitude of these two instruments in this frame may have reached the highest, the highest frequency of 0.465 and 0.0198, respectively. in the spectrum of the violin, when the amplitude value is the largest, the amplitude is still fluctuating and amplitude finally zero, indicating that the signal has stopped finally. In the spectrum of the violin, after the maximum amplitude value, the amplitude still fluctuates greatly and the amplitude value finally reaches zero, indicating that the signal has finally stopped. In the spectrum of flute, when the amplitude value reaches the maximum, the fluctuation of amplitude is almost none, but there is a signal all the time, which indicates that the overtone frequency of violin is higher compared with that of flute, and thus the violin has richer color, so for the music fragment with shorter length, the violin can be chosen as the main melody of the piece of music to be played by the automatic compositions.



(a) The first frame spectrum of violin audio

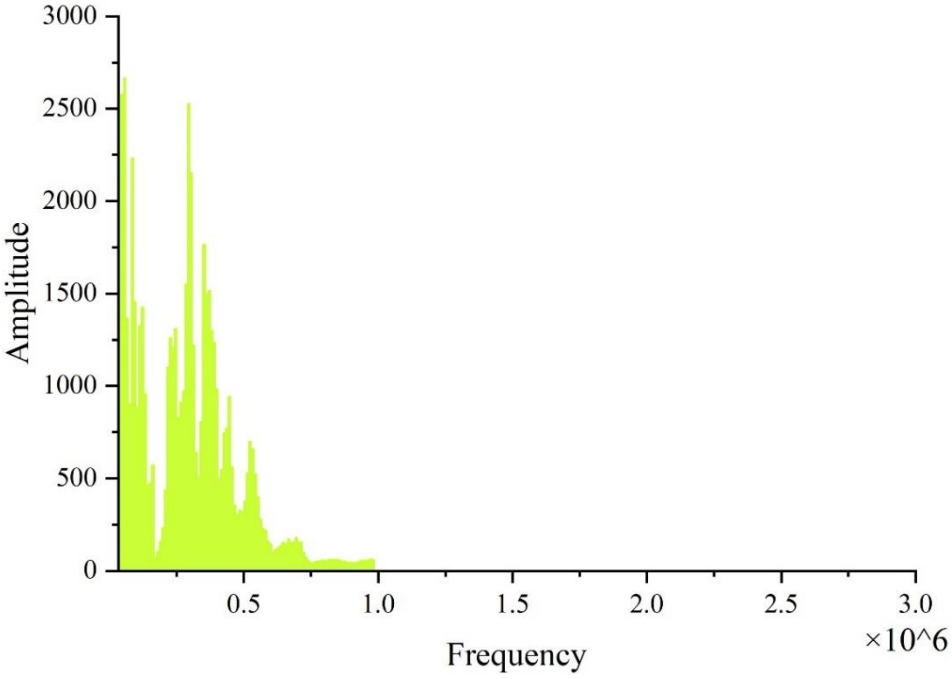


(b) The second frame spectrum of flute audio

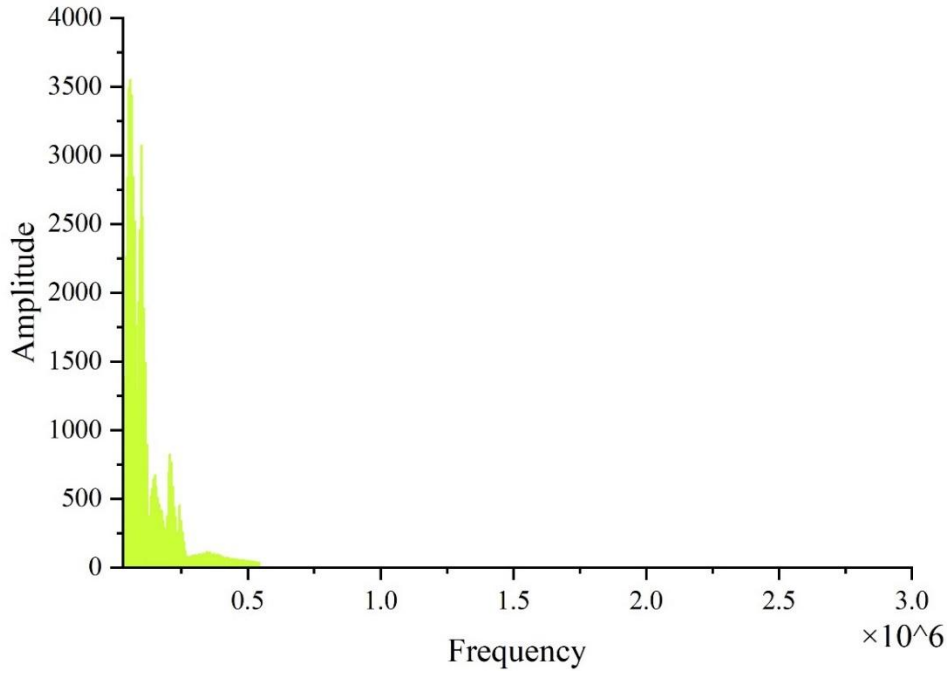
Figure 7. Single frame spectrum analysis.

3.4.2. Spectral Analysis of All Frames

Figure 8 shows the spectral analysis of all the frames of the violin and flute audio, Figure (a) is for violin and Figure (b) is for flute. According to the violin and flute two spectrograms to find out the location of the maximum amplitude point, the sample points are the 58355th and 90485th, respectively, you can find out the frequency of a certain sample point n, in the reading of the audio file can be derived from both the sampling frequency of 44154Hz. first of all, the total number of sample points of the violin is found to be 5012485, and its maximum amplitude of the frequency of the sample point is obtained by using the equation is 563Hz. Then the total number of sample points of the flute is 5064856, and the same formula is used to find out the frequency of the sample point with the largest amplitude is 783 Hz. For the violin and the flute, when their amplitude reaches the highest, the frequency of the flute is obviously higher than the frequency of the violin, and at this time the pitch of the flute is also higher than that of the violin, so the flute is the most important instrument in the automatic composition of the composition of the pitch part of the composition. In automatic composition work, the flute can be used for the creation of the pitch part, which increases the richness of the music.



(a) Violin



(b) Flute

Figure 8. The spectrum analysis of all frames of the violin and flute audio.

4. Conclusion

In this paper, multidimensional spectral analysis methods such as speech spectrum and phase spectrum are used in conjunction with deep learning models to jointly extract a pop music melody. After pre-training the extracted melody, a new pop music melody is generated. The multi-dimensional spectral analysis method is utilized to mark the key positions of note symbols and harmonies to assist in the composition of pop music.

The results of the generated music melody are evaluated through a combination of online and offline

methods, and the evaluation results tell us that the highest rating of the music melody generated by automatic composition using this paper's model is 3.974, which indicates that the automatically generated music in this paper meets the appreciation requirements of the testers. Compared with the human compositions, the score gap of this paper's model is small, the top three automatic composition melody scores are 81.536, 81.048, 80.499, respectively, and the automatically generated music in this paper can reach the level of general artificial composition. Spectral analysis of the score was performed to extract the single-frame and full-frame spectra of the violin and flute. The amplitudes of these two instruments reach the highest in the first and second frames, with the highest frequencies of 0.465 and 0.0198, respectively.The overtone frequency of the violin is higher compared to that of the flute, but it is better in terms of richer timbre.

Funding

This research was supported by the: Anhui Province 2022 Provincial Quality Engineering Continuing Education Teaching Reform Project: Construction and Practice of the Curriculum System of Higher Education in Anhui North under the Background of Rural Revitalization – Taking the Musicology major of Fuyang Normal University as an Example (2022jxjy044); Horizontal Research Project of Fuyang Normal University: Midshore Art innovation technology transformation and service contract (HX2021048); Research topic of social science Innovation and development in Anhui Province in 2024: Research on the logical way of spreading revolutionary songs in Anhui (1921-1949) (2024CX148).

References

1. Knust, M. (2025). The Creation of Music: A Historical Overview of the Composition Process. In Pop Music Made in Småland: Music Production and Entrepreneurship in Sweden (pp. 13-28). Cham: Springer Nature Switzerland.
2. Ng, H. H. (2020). Towards a synthesis of formal, non-formal and informal pedagogies in popular music learning. Research Studies in Music Education, 42(1), 56-76.
3. Hutchings, P. E., & McCormack, J. (2019). Adaptive music composition for games. IEEE Transactions on Games, 12(3), 270-280.
4. Marrington, M. (2017). Composing with the digital audio workstation. The singer-songwriter handbook, 77-89.
5. Liu, C. H., & Ting, C. K. (2016). Computational intelligence in music composition: A survey. IEEE Transactions on Emerging Topics in Computational Intelligence, 1(1), 2-15.
6. Zhu, H., Liu, Q., Yuan, N. J., Qin, C., Li, J., Zhang, K., ... & Chen, E. (2018, July). Xiaoice band: A melody and arrangement generation framework for pop music. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2837-2846).
7. Huang, Y. S., & Yang, Y. H. (2020, October). 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).
8. Liu, W. (2023). Literature survey of multi-track music generation model based on generative confrontation network in intelligent composition. The Journal of Supercomputing, 79(6), 6560-6582.
9. Hernandez-Olivan, C., & Beltran, J. R. (2022). Music composition with deep learning: A review. Advances in speech and music technology: computational aspects and applications, 25-50.
10. Lin, T. F., & Chen, L. B. (2024). Harmony and algorithm: Exploring the advancements and impacts of AI-generated music. IEEE potentials.
11. Deruty, E., Grachten, M., Lattner, S., Nistal, J., & Aouameur, C. (2022). On the development and practice of ai technology for contemporary popular music production. Transactions of the International Society for Music Information Retrieval, 5(1).
12. Wu, J., Liu, X., Hu, X., & Zhu, J. (2020). PopMNet: Generating structured pop music melodies using neural networks. Artificial Intelligence, 286, 103303.
13. Jeong, J., Kim, Y., & Ahn, C. W. (2017). A multi-objective evolutionary approach to automatic melody generation. Expert Systems with Applications, 90, 50-61.
14. Ji, S., Yang, X., & Luo, J. (2023). A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. ACM Computing Surveys, 56(1), 1-39.
15. Goienetxea, I., Mendialdua, I., Rodriguez, I., & Sierra, B. (2019). Statistics-based music generation approach considering both rhythm and melody coherence. IEEE Access, 7, 183365-183382.
16. Ponce de León, P. J., Iñesta, J. M., Calvo-Zaragoza, J., & Rizo, D. (2016). Data-based melody generation through multi-objective evolutionary computation. Journal of Mathematics and Music, 10(2), 173-192.

17. Fukumoto, M., & Hatanaka, T. (2017). A proposal for distributed interactive genetic algorithm for composition of musical melody. Information Engineering Express, 3(2), 59-68.
18. Lam, M. W., Tian, Q., Li, T., Yin, Z., Feng, S., Tu, M., ... & Wang, Y. (2023). Efficient neural music generation. Advances in Neural Information Processing Systems, 36, 17450-17463.
19. Yafeng Zhou,Fan Chen & Fangfang Li. (2025). Personalized sleep aid audio strategy model based on self-attention and time-domain analysis. Journal of Computational Methods in Sciences and Engineering,25(1),312-323.
20. Xiaoying Mao,Ye Tian,Tairan Jin & Bo Di. (2025). Enhancing music audio signal recognition through CNN-BiLSTM fusion with De-noising autoencoder for improved performance. Neurocomputing,625,129607-129607.
21. Xiyuan Gao & Ruohan Gao. (2025). Music signal recognition aids based on convolutional neural networks in music education. Systems and Soft Computing,7,200219-200219.
22. Jichen Yang & Rohan Kumar Das. (2020). Improving anti-spoofing with octave spectrum and short-term spectral statistics information. Applied Acoustics,157,107017-107017.