International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 532-547

https://doi.org/10.70517/ijhsa46338

A deep neural network-based approach to style transformation and emotion encoding in AI music composition in a big data analytics environment

Lianghua Li[1,*]
[1] The College of Music and Dance, Changsha, Hunan Frist Normal University, Hunan, 410205, China
Corresponding authors: (e-mail: lilianghua84816@163.com).

Abstract With the continuous development of artificial intelligence technology, deep neural networks show great potential in the field of intelligent music creation. In this paper, we first extract the CQT and Meier spectral features of music, deform and fill the biphasic information by WaveNet decoder, and realize the overall style migration of emotional music. Then, we design a music emotion representation model that integrates the Plutchik and Thayer bi-emotion models and devise a fusion method for the bimodal emotion results, based on which rhythmic control and tonal conditions are introduced to generate music that contains multiple emotions. The model in this paper can effectively merge audio tracks, and the average style transformation intensity of music of the same style reaches 0.80 and above, and can accurately express negative and positive emotions and transform them into emotional music representations, obtaining a music quality score of 4.1. It adds a scientific supporting theory to the field of intelligent composition research.

Index Terms Music style migration, Meier spectral features, WaveNet decoder, Music emotion representation models

I. Introduction

Edgard Varèse (1883-1965), a 20th century avant-garde composer and the “father of electronic music”, once said that the future of music creation needs to rely on the cooperation between composers and scientists [1]. In recent years, the information technology represented by Artificial Intelligence (AI) has been deeply developed and widely applied, and the combination of science and technology and art has become closer and closer [2]-[4]. It was pointed out at the annual meeting of the China Development High-Level Forum that AI will become the standard of the fourth industrial revolution and an important engine for the development of new quality productivity [5]. The use of AI technology has liberated human mental labor to a certain extent, and has shown great potential in the field of music arranging and creation [6].

The application of computer technology and AI in the field of music arranging began as early as the 1950s [7]. In 1957, Lejaren Hiller (1924-1994) composed the Iliac Suite, which used the probabilistic prediction principle of Markov Chain to predict the pitch of notes and filtered them through the harmonic and polyphonic rules notes, and finally modifying and combining the note material that conforms to the rules to form a string quartet in the traditional style [8]. The piece is also regarded as the world's first computer-generated musical work.In 1974, the International Conference on Computer Music was held at Michigan State University in the U.S.A. The conference explored how computers and music could be further combined, and the conference has become an important platform in the field of computer music [9], [10].

Due to a variety of factors, the 1980s was the “winter” of AI technology development [11]. Until around 2012, with the emergence and rapid development of neural networks, AI ushered in a period of vigorous development [12]. Neural network-enabled deep learning and big data technology has made significant progress in the fields of image, audio, text, etc., making the technology gradually applied to music composition [13]-[15]. To this day, the underlying logic of the vast majority of AI is based on the neural network approach [16], and most of the development of AI music composition is currently based on this. Generally speaking, lyrics, composition, and arrangement are the first and core stage of music creation [17], [18]. AI's extreme ability to generate text, audio and other data provides sufficient conditions for its application in this stage [19]. The programmed and mechanized steps in music creation have been simplified and the difficulty of creation has been reduced due to the support of AI, and at the same time, with the popularization of AI technology, AI music creation has also begun to be popularized and popularized [20]. Music by the support of technology, is showing a new development trend. Through AI technology can make music

regain vitality, realize the combination of spirit and material in music creation, and solve the many dilemmas in the development of current music [21].

In this paper, a deep neural network-based music style conversion and emotion encoding model is designed. Firstly, the CQT features and Meier spectrum of the audio are extracted, and the generative adversarial network is used to transform the two features into styles, and then the phase information of the two features is locked by the WaveNet decoder, which is converted into different styles of music. Based on the Plutchik three-dimensional emotion and Thayer two-dimensional emotion representation, the traditional emotion representation model is improved, and the concepts of music emotion determining coordinates and music emotion offset coordinates are introduced, and the positional difference of the two coordinates is measured using Euclidean distance to measure the reasonableness of the music emotion transmission. Finally, rhythm and modulation were integrated into the traditional music generation model respectively to realize the generation of multi-emotional music. The validity of this paper's model is verified in a music style dataset constructed based on MIDI, and objective and subjective evaluation methods are used to verify the effectiveness of music generation.

II. Knowledge base

II. A. CycleGAN

Generative Adversarial Networks [22] is a deep learning model that is a class of implicitly generative models. The model produces high-quality outputs by learning from the mutual game of two modules in the framework (generative model and discriminative model). The generative model tries to generate fake samples to fool the discriminative model. The discriminative model, on the other hand, tries to distinguish between real data and fake samples. Assume that G is the generator, D is the discriminator, $P_{data}(x)$ is the distribution of real samples and x is sampled from that distribution, and $P_z(z)$ is the distribution of potential codes z for x. The target equation is then:

$$\begin{aligned} G, D = \min_G \max_D \mathrm{E}_{x \sim P_{data}(x)} \left[\log D(x) \right] \\ + \mathrm{E}_{z \sim P_z(z)} \left[\log \left(1 - D\left(G\left(z \right) \right) \right) \right] \end{aligned} \tag{1}$$

CycleGAN is an unsupervised generative adversarial network that learns the mapping between two domains without any pairwise data.CycleGAN contains two generators and two discriminators, each of which needs to learn the mapping from the domain to the corresponding domain. The two discriminators, on the other hand, need to determine whether the data generated by the corresponding domain generator is the data of this domain by learning the real data of their respective domains.The loss function of CycleGAN contains two adversarial losses in addition to a cyclic consistency loss, which is used to preserve the structure of its inputs as shown in equation (2):

$$\begin{aligned} \mathrm{L}_{CYC}\left(G, F \right) = \mathrm{E}_{x \sim P_{real}(x)} \left[\left\| F(G(x)) - x \right\|_1 \right] \\ + \mathrm{E}_{y \sim P_{real}(y)} \left[\left\| F(G(y)) - y \right\|_1 \right] \end{aligned} \tag{2}$$

where G denotes the forward transformation of CycleGAN and F denotes the backward transformation of CycleGAN.

II. B. Time-frequency analysis

II. B. 1) Constant Q-Transform (CQT)

The constant Q transform (CQT) is a time-frequency analysis technique widely used in music signal processing and acoustic research. It decomposes the frequency of a signal by a set of filters, which are characterized by an exponentially regular distribution of center frequencies and different filter bandwidths, but the ratio of the center frequency to the bandwidth is a constant Q. This means that in CQT, the ratio of the center frequency of each filter to its bandwidth is fixed, so that the bandwidth of the filters increases as the center frequency increases in different frequency ranges to keep the value of Q constant.

The CQT's spectral cross-axis frequency uses a logarithmic scale based on a base of 2, rather than a linear scale, which matches the distribution of musical scales and allows the CQT to better capture subtle frequency variations in the audio signal. Since the frequency distribution of music is usually nonlinear, CQT has significant advantages in music signal processing.

For a constant Q filter, the ratio of the center frequency to the bandwidth is a fixed value and can be expressed by the following equation:

$$Q = \frac{f_c}{\Delta f} \tag{3}$$

where Q is the value of the constant Q, f_c is the center frequency of the filter, and Δf is the bandwidth of the filter.

The bandwidth of the filter and the spacing between neighboring frequencies are adjusted to ensure that the frequency resolution over different frequency ranges is adaptable to changes in signal characteristics. For low-frequency waveforms, CQT will use a narrower filter bandwidth to enhance the resolution of notes with small frequency intervals. For high frequency waveforms, on the other hand, the CQT will use a wider filter bandwidth to enhance the temporal resolution for rapidly changing overtones.

By definition, the frequency bandwidth δ_f at frequency f, also known as frequency resolution, indicates the filter bandwidth at that frequency. In CQT, the bandwidth of the filter varies with frequency to ensure that the frequency resolution adapts to changes in signal characteristics over different frequency ranges.

Assuming that the lowest tone to be processed is $f_{\min}$, the frequency f_k of the k th frequency component can be expressed by the following equation:

$$f_k = f_{\min} \cdot 2^{k/b} \tag{4}$$

where b denotes the number of spectral lines contained within each octave, e.g., $b = 36$ means that there are 36 spectral lines within each octave and three frequency components per semitone.

In CQT, the frequency resolution δ_f can be expressed in terms of the bandwidth of the filter. For a frequency bandwidth that is at frequency f_k, it is usually defined as:

$$\delta f_k = Q \cdot f_k \tag{5}$$

where Q is the constant Q value that represents the ratio of the center frequency f_k of the filter to the bandwidth δf_k. Then it is known from the above equation:

$$Q = \frac{f}{\delta f} = \frac{1}{2^{1/b} - 1} \tag{6}$$

Therefore, the value of Q is related to b.

According to the given conditions, the window length N_k with frequency can be calculated as follows:

$$N_k = \left\lceil Q \frac{f_s}{f_k} \right\rceil, k = 0,1,\cdots,K-1 \tag{7}$$

where $\lceil x \rceil$ denotes the upward rounding function, f_k is the frequency of the k th semitone, f_s is the sampling frequency, and K is the total number of semitones.

To summarize, so in CQT, the k th semitone frequency component of the n th frame can be expressed as:

$$X^{CQT}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} x(n) w_{N_k}(n) e^{-j\frac{2\pi Q}{N_k} n} \tag{8}$$

where $X^{CQT}(k)$ denotes the frequency component of the k th semitone. $x(n)$ is the sampled value of the input signal in the time domain. $w_{N_k}(n)$ is a window function of length N_k.

II. B. 2) Short-Time Fourier Transform (STFT)

Fourier Transform [23] is an important tool for signal processing which is mainly used to convert signals from time domain to frequency domain, but it cannot provide local characterization of signals in time domain. To solve this problem, STFT divides the signal into multiple time segments and uses a window function to weight the signal in each time segment and then performs a Fourier transform to locally analyze the signal in the time-frequency domain.

The process of STFT is to multiply the signal by a time-limited window function $h(t)$ before the signal is Fourier transformed. This window function serves to limit the time horizon of the signal in the time domain and assumes

that the signal is smooth within the analysis window. The signal is then analyzed segment by segment by shifting the position of the window function $h(t)$ on the time axis to obtain a set of localized spectral information.

The mathematical expression for the STFT is:

$$X(t,\omega)=\int_{-\infty}^{\infty}x(\tau)\cdot h(\tau-t)\cdot e^{-j\omega\tau}d\tau \tag{9}$$

where $X(t,\omega)$ denotes the spectral component at time t and frequency ω. $x(\tau)$ denotes the original signal. $h(\tau-t)$ is a window function, usually 1 at the moment t and decaying to zero at other moments, used to limit the range of the signal in time. This step helps to minimize spectral leakage and ensures a smooth transition of the signal to zero at the window boundary. The window function is generally chosen as a Hanning window, a Hamming window, etc. In particular, when the window function is taken as $h(t)\equiv 1$, the STFT is equivalent to a conventional Fourier transform.

To obtain the best localization performance, the width of the window should be adjusted according to the characteristics of the signal. For example, for sinusoidal-type signals, it is appropriate to choose a larger window width, while for impulse-type signals, it is appropriate to choose a smaller window width. This ensures a better resolution in the time-frequency domain.

The advantage of the STFT is that its basic algorithm is the Fourier transform, which is easy to interpret in its physical meaning. However, its disadvantage is that the window width is fixed and cannot be adjusted adaptively. This means that the size of the window needs to be determined beforehand when processing the signal, otherwise it may lead to loss of information or insufficient resolution.

II. C.WaveNet

The WaveNet network [24] models conditional probabilities using CC (causal convolution) and DCC (dilated causal convolution). In this case, causal convolution means that the state at the current moment is determined only by the historical state before the current moment, i.e., the convolution is computed using only the data from the past timesteps and not the data from the future timesteps. The basic principle is that the time-series data is composed of one sample point, and each sample point is subject to the constraints of historical sample points. The joint probability density $p(x)$ of the entire time series data can be expressed as:

$$p(x)=\prod_{t=1}^{N}p\left(x_{t+1}|\ x_1,x_2,\cdots,x_t\right) \tag{10}$$

In order to obtain longer history information, it is necessary to further increase the sensory field of CCN, so it is necessary to increase the number of network layers or increase the size of the convolution kernel, which will lead to too many CCN training parameters and the model itself becomes extremely complex, which leads to the inefficiency of the training, and therefore the introduction of the DCC. In DCC, keep the size of the convolution kernel of the causal convolution unchanged, and increase the sensory field by changing the diffusion rate. The DCC computational formula $F(t)$ at t moment can be expressed as:

$$F(t)=\sum_{i=0}^{k-1}f(i)x_{t-d\cdot i} \tag{11}$$

Where, f - one-dimensional convolution kernel for causal convolution, k - size of one-dimensional convolution kernel for causal convolution, $t-d\cdot i$ - the tensors involved in the inflationary causal convolution, d - the expansion coefficient.

Stacking the DCCs can form a WaveNet network.The WaveNet network structure generally increases the diffusion rate by an exponential power of 2 from the lower to the higher layers, using lower convolutional layers for short-term data learning and higher convolutional layers for long-term data learning. Compared to general CCNs, WaveNet networks can process long sequence data more efficiently.

III. Deep neural-based music style transformation and emotion encoding

III. A. Music Style Migration Model Architecture

This section describes the basic architecture of the model and the processing flow of the model. The model is based on CycleGAN and WaveNet decoder. The model processing flow is as follows:

1) Extract the Mel spectral features and CQT features of the audio.

2) Combine the extracted Mel spectral features and CQT features into two layers and input them into the CycleGAN model, and then CycleGAN generates the Mel spectral features and CQT features after style migration.

3) The two layers of features are input into the pre-trained WaveNet decoder to produce audio.

III. A. 1) Problem definition and data preprocessing

Given a music dataset dataset, a single music sample is sampled from dataset. The music sample is represented as a two-dimensional matrix M_i, with matrix M_i of size $m \times n$, where m denotes the length of the music segment over the time series, and n denotes the number of channels of the music. For a particular piece of music M_i in the training set, where $i \in \{1,2,3,\cdots,k\}$, k denotes the number of music in the training set. For a single music sequence M_i, the music sequence is transformed at the same window size and step size to obtain the spectrogram Q and the Mel spectrum T, assuming that the size of the spectrogram Q is $i \times j$, and that the size of the spectrogram T is $i \times s$, the matrices T and matrices Q can be expressed as:

$$T = \begin{bmatrix} t_1^1 & t_1^2 & \cdots & t_1^s \\ t_2^1 & t_2^2 & \cdots & t_2^s \\ \vdots & \vdots & \ddots & \vdots \\ t_i^1 & t_i^2 & \cdots & t_i^s \end{bmatrix} \tag{12}$$

$$Q = \begin{bmatrix} q_1^1 & q_1^2 & \cdots & q_1^j \\ q_2^1 & q_2^2 & \cdots & q_2^j \\ \vdots & \vdots & \ddots & \vdots \\ q_i^1 & q_i^2 & \cdots & q_i^j \end{bmatrix} \tag{13}$$

Since the window size and step size used in both transformations are the same, $j > s$, the matrix T is added with $j - s$ column vectors, viz:

$$Tp = \begin{bmatrix} t_1^1 & t_1^2 & \cdots & t_1^s & 0_1^{s+1} & \cdots & 0_1^j \\ t_2^1 & t_2^2 & \cdots & t_2^s & 0_2^{s+1} & \cdots & 0_2^j \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_i^1 & t_i^2 & \cdots & t_i^s & 0_i^{s+1} & \cdots & 0_i^j \end{bmatrix} \tag{14}$$

The filled matrices Tp and Q are merged into two layers to form a 3D vector X with vector size $i \times j \times 2$. Since CycleGAN is an unsupervised algorithm, i.e., there is no corresponding labeling, the vectors X of the two style domains are directly used as the inputs and outputs of CycleGAN for training in the experiments.

III. A. 2) Design of CycleGAN

In this paper, the model processes translations between two domains at a time, so they are referred to as $domain_x$ and $domain_y$, which correspond to music from two different genres. Since the transmission is supposed to be symmetric, the samples are transmitted from $domain_x$ to $domain_y$ and from $domain_y$ to $domain_x$ at the same time.The basic loss function of CycleGAN's $X \to Y$ is shown in the following equation:

$$\begin{aligned} \mathrm{L}_{GAN}(G, D, X, Y) = & \mathrm{E}_{x \sim p_{data}(x)}\left[\log D(x)\right] \\ & + \mathrm{E}_{y \sim p_{data}(y)}\left[\log\left(1 - D\left(G(y)\right)\right)\right] \end{aligned} \tag{15}$$

In addition to this, CycleGAN also adds identityloss. Experiments have shown that when this loss is not added, the generated spectrogram loses its color component, which is manifested as a large noise in the final generated audio, so adding this loss facilitates the model to generate higher quality spectrograms.

$$\begin{aligned} \mathrm{L}_{identity}(G, F) = & \mathrm{E}_{y \sim P_{data}(y)}\left[\left\|G(y) - y\right\|_1\right] \\ & + \mathrm{E}_{x \sim P_{data}(x)}\left[\left\|F(x) - x\right\|_1\right] \end{aligned} \tag{16}$$

Add cycleconsistency loss and identity loss, where λ_1 and λ_2 denote the weights occupied by cycleconsistency loss and identity loss, respectively. Then the total loss function is:

$$\begin{aligned} \mathrm{L}\left(G,F,D_X,D_Y\right)=\mathrm{L}_{GAN}\left(G,D_Y,X,Y\right)+\mathrm{L}_{GAN}\left(F,D_X,Y,X\right) \\ +\lambda_1\mathrm{L}_{cyc}\left(G,F\right)+\lambda_2\mathrm{L}_{identity}\left(G,F\right) \end{aligned} \tag{17}$$

The weights λ_1 and λ_2 due to cycleconsistency loss and identity loss can significantly affect the generated audio. In the case of λ_1, when λ_1 is too large, Cycle GAN chooses simple, low-latitude transformations. When λ_1 is too small, Cycle GAN will seek transformations with higher complexity, and the resulting transformations are varied but not easy to control. In this paper, λ_1 is set as a fixed value in the experiment. As for λ_2, it is shown that the value of λ_2 should decrease as the number of iterations of the algorithm increases. Usually the decay of λ_2 is linear, but in this paper, it is found through experiments that the curvilinear decay of λ_2 is more in line with the style transformation algorithm of music. Therefore, this paper tries to propose a nonlinear decay function for λ_2. Compared with the linear decay function, the nonlinear decay function proposed in this paper makes the model show better robustness. Assuming that the algorithm iterates a total of t steps, then at the n th step, λ_2 is:

$$\lambda_2=\frac{\ln t+\sqrt{t}}{n+\sqrt{t}} \tag{18}$$

Since CycleGAN employs the operation of back-convolution, this can lead to severe checkerboard artifacts in the generated spectrograms, which manifests itself as severe indirect noise on the audio.

For this reason this study uses nearest neighbor interpolation and regular convolution instead of anti-convolution. The first approach ensures that the kernel size used is divided by the step size to avoid overlap problems. But despite the recent success of this technique in image super-resolution, the inverse convolution is still prone to results with checkerboard artifacts. Another approach is to isolate higher resolution upsampling from the convolution to the computed features. The input is first resized by nearest-neighbor interpolation and then entered into the convolutional layer. Compared to the first method, this method not only works well in image super-resolution tasks, but also makes the resulting spectrogram less prone to results with checkerboard artifacts.

III. A. 3) WaveNet Decoder

Since the WaveNet loss function is a tanh activation function, and the waveform represented by the tanh function is between $[-1,1]$, and the values of the CQT spectrograms and Meier spectra generated by audio taking the natural logarithm conform to the normal distribution between $[-6,2]$. Therefore, it is necessary to globally normalize the input data before training so that the distribution of the input data conforms to the $N(0,1)$ distribution.

It is difficult to predict the phase directly from the spectrogram in the time-frequency representation, so it is necessary to discard the phase information of the Meier spectrum and the CQT spectrogram, and then deform and complementary zero-filling operations are performed on the phase information of the two layers in order to merge the two layers.

The network structure of the WaveNet decoder in this paper maintains the same structure as the original WaveNet state, but changes its inputs to spectrograms with a network expansion rate of 2^k (k denotes the layer the network is in). For all bandhole convolution and causal convolution layers, a convolution kernel of size 3 was used. For all residual blocks, the length of the far-hop connections and residual connections is 256. In addition to this, each residual layer contains a ReLu nonlinear function.

III. B. Music Emotion Representation Model Based on Chunking Matrix

III. B. 1) Chunked representation of emotions

In this section, a music emotion representation model based on the fusion of the Plutchik emotion model and the Thayer emotion model is designed. The fusion of 25 discrete emotion characteristics is grouped into one category for every 5 groups of emotions, and the degree of motivation of each category of emotions increases step by step, and the degree of positivity of each group of emotions increases step by step. Therefore, the 25 discrete emotions are mapped into vectors, with the horizontal coordinates of the vectors indicating the degree of motivation of each group of emotions and the vertical coordinates of the vectors indicating the intensity of each group of emotions.

A matrix of 5×5 is used in this project to represent the corresponding emotions. Each emotion is represented by two coordinates, which take values from 0 to 4, one value for each interval of 1 unit, and the significance of the value indicates the degree of positivity of the emotion, where 0 is the most negative emotion, and 4 means the most positive. Emotion group 1 to emotion group 5 is from positive emotion to negative emotion, where the most positive emotion is group 1, so the emotion of group 1 is expressed as:

$$emotion_{group1} = \{(4, y_i)\} \tag{19}$$

where y_i denotes the degree of sentiment within the group, taking values from 0 to 4, with values taken at 1 unit intervals, and the degree ranging from weak to strong. The most negative emotion is group 5, which is represented using 0, so the emotion for group 5 is represented as:

$$emotion_{group5} = \{(0, y_i)\} \tag{20}$$

where y_i takes the same value as in group 1.

III. B. 2) Sentiment representation model based on chunking matrices

Music emotion extraction task is a classification task, in this topic, the music emotion is categorized into 25 categories, so it is a multi-categorization task. In this project, the classification task is performed for 25 types of emotions, but the output of the classification model is a tensor of $5*5$, which has three advantages, the first one is that it can reduce the output layers of the neural network, if the output is a tensor of $1*n$ according to the conventional method, then 25 output layers are needed. If the output is a $5*5$ tensor, only 5 output layers are needed. The second advantage is that it is convenient to apply the two-dimensional sentiment model, there is no need to perform a complex matrix decomposition of the matrix, but only need to call the sentiment determination algorithm to get the type of sentiment. The third advantage is the convenience of bimodal emotion fusion. In the musical expression emotion extraction task, it is necessary to fuse the emotion extracted from the audio with the extraction of the lyrics.

III. B. 3) Sentiment Normalization Algorithm

The process of music emotion normalization is to fuse the probability distributions of positive and strong music emotions respectively to get the music emotion chunking matrix, and then the chunking matrix leads to a definite coordinate representation of music emotion. Music emotion can be represented as a chunking matrix. The final output of the deep learning based multi-classification model is the probability distribution over each classification, and normalization is required in order to map the probability distribution of the emotion categories to the emotion coordinates.

$$emotion_{music} = \begin{bmatrix} (x_1, y_{11}) & \cdots & (x_1, y_{15}) \\ \vdots & \ddots & \vdots \\ (x_5, y_{51}) & \cdots & (x_5, y_{55}) \end{bmatrix}_{5\times5} \tag{21}$$

$$emotion_{active} = [x_i], \text{Which } 0 \le i \le 5, \sum_{i=0}^{5} x_i = 1 \tag{22}$$

$$emotion_{strong} = [y_i], \text{Which } 0 \le i \le 5, \sum_{i=0}^{5} y_i = 1 \tag{23}$$

For emotion representation, Eqs. (21) and (22) need to be fused into Eq. (23).

The horizontal coordinate of each element in the tensor represents the intensity of the baseline emotion (the probability distribution over the emotion groups, where $\sum_{i=1}^{5} x_1 = 1$), and the vertical coordinate represents the positivity of the baseline emotion (the probability distribution over the emotions in each emotion group, where $\sum_{j=1}^{5} y_{ij} = 1$, where i takes values from 1 to 5), which is processed by the two-dimensional matrix-based representation model to take out the maximum of $\max(x_i)$, and then the maximum of $\max(y_{ij})$ in the i th row.

$$p(music_{emotion}) = (\max(x_i), \max(y_{ij})) \tag{24}$$

The coordinates for determining musical emotion are expressed as equation (25):

$$music_{emotion} = (i-1, j-1) \tag{25}$$

The music emotion determination coordinates are $(i-1, j-1)$, and mapping the music emotion determination coordinates into the music emotion model is the final baseline music emotion representation. The above is the process of music emotion normalization.

III. B. 4) Algorithm for emotional offset coordinates

The probability distribution of positive degree and the probability distribution of intensity of music emotion can be obtained by emotion normalization to determine the coordinates of music emotion. Since this project studies the transmissibility of music emotion, so combined with the probability distribution of music emotion in positive degree and intensity, this project designs an emotion offset coordinate algorithm to measure the relationship of emotion transmission through the emotion offset coordinate, and uses the Euclidean distance to verify the correctness of the music emotion offset coordinates.

(m_{offset}, n_{offset}) is the emotion offset coordinate, and then use the Euclidean distance to verify the correctness of the emotion offset coordinate. It is necessary to calculate the distances of the positive and negative degrees of the emotion from the baseline emotion coordinates. In the k neighborhood (KNN) algorithm, Euclidean distance and Manhattan are commonly used to calculate the distance between nodes. The Euclidean distance is defined in the following equation:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \cdots + (y_{n1} - y_{n2})^2} \tag{26}$$

Manhattan distance [25] definition:

$$d = |x_1 - x_2| + |y_1 - y_2| \tag{27}$$

In this topic, Euclidean distance is used to calculate the distance from coordinates (x_i, y_j) to 25 sets of base points, and the base point with the shortest distance is the sentiment feature of coordinates (x_i, y_j). Of course, this algorithm can be further optimized, since the 25 sets of base points have been given, just find the four base points with the closest distance from (x_i, y_j), and the base point with the shortest distance is the emotional representation of point (x_i, y_j). If (x_i, y_j) is the same as the emotion determination coordinates, then the music emotion offset coordinates are also reasonable.

III. B. 5) Bimodal sentiment outcome fusion methods

Bimodal subtask fusion requires multiple considerations and needs to balance multiple objectives to prevent loss of accuracy in subtask results. There are five objectives for balancing bimodal subtasks; the loss magnitude of different tasks needs to be similar, the similar rate learning of different tasks needs to be similar, the data magnitude of different tasks is close, the proximity assessment of the importance level of different tasks needs to be similar, and the estimation of uncertainty of different tasks needs to be similar. Most previous multi-task learning methods focus on only the following two criteria:

(1) Network architecture, focusing on how to share data across multiple self-tasks. Multi-task learning architecture should express both task-sharing properties and task-specific properties.

(2) Loss function, which focuses on how to balance losses across multiple tasks. The multitask loss function weights the relative contribution of each task and should be able to learn all tasks with equal importance without allowing learning to be dominated by simpler tasks.

The goal of the bimodal sentiment fusion algorithm is to fuse two outcomes with the goal of scaling up or down the outcome data.

In this subsection, an approach based on correction factor fusion is designed based on weighted fusion, the core idea of which is to utilize the rate of change of LOSS to balance the speed of multi-task learning, and the correction factor weighted fusion formula is as follows:

$$L = \sum_i w_i * L_i \tag{28}$$

The correction coefficient w_i varies with the amount of raw data of the subtask or the model accuracy of the subtask result. When the sub-task A model accuracy result is higher than that of the B task, multiplying the A model result by the correction factor $w_i\,(w_i > 1)$, and adding the result of the self B task afterward is the final result of the sub-task fusion as in Eq:

$$emotion_{song} = \{x_1 \cdots x_5\} w_i + \{x_1 \cdots x_5\} \tag{29}$$

III. C. Multi-emotion music generation model

Music generation models [26] encode melodic sequences and chord sequences separately and feed them into a decoder to obtain new chord sequences. For the task of matching melody with chords, most of the models generate chords according to fixed time intervals, but harmonic rhythms are not only related to the melody that develops according to the fixed time intervals, but also depend on the tempo. Moreover, two adjustable parameters, tempo and modulation, can control the generation of music with different emotions; therefore, in this chapter, tempo and modulation are also integrated into the generative model to realize the generation of multi-emotional music.

In this paper, rhythmic control and tuning conditions are introduced, so a chord rhythm model has to be trained first, on which the chord generation model is trained by controllable rhythm and tuning. Therefore, the network architecture in this chapter consists of two parts, the first part is the chord rhythm generation model, which realizes the chord generation with tuning and rhythm control. The second part is the chord generation model, in which the chord sequences with control conditions from the previous model and the chords from the original music are used to train the new chord generation model, which realizes the chord generation with tuning and tempo control, and then combines with the melodic sequences with tuning control to generate music with different emotions.

First of all, the chord rhythm model is mainly composed of three parts: melodic encoder, metronomic encoder and chord rhythm decoder. The melody encoder and the beat encoder use the encoder part of Transformer, while the chord rhythm encoder corresponds to the decoder part of Transformer.

Given a melodic sequence of length T $m_{1,T} = \{m_1, m_2, ...m_T\}$ and the corresponding beat sequence $b_{1,T} = \{b, b_2, ...b_T\}$, if the length of each event sequence in the melody is T, the pitch histogram vector can be represented as a $T \times 12$ dimensional vector sequence, which is spliced with the melodic sequence before being fed into the melodic encoder to realize the control of the musical key. The output of the melody encoder and the output of the beat encoder are spliced together with the chord rhythm sequence of the previous time step and fed into the chord rhythm decoder to obtain the chord rhythm sequence $r_{1,T} = \{r_1, r_2, ...r_T\}$. When the time step $t \in \{1, 2, ..., T\}$, the model generates the chord rhythmic labeling r_t for the current time step based on the input sequences $m_{1,T}$, $b_{1,T}$, the pitch histogram vector p, and the sequence $r_{1,t-1}$ generated at the previous time step:

$$r_t^n = M_R\left(m_{1,T}, b_{1,T}, p, \theta_R\right) \tag{30}$$

where M_R is the chord rhythm model, i.e., the Transformer model, and θ_R is a parameter in it. Since the chord rhythm model can refer to melodic information $m_{t+1,T}$ and rhythmic information $b_{t+1,T}$ after time step t, the model will make longer-term choices in generating r_t. This approach is also consistent with the compositional methods of most composers, who also often base their current chord arrangements on the melody and beat number that follows.

As with the chord representation of the chord generation model, each chord c_i of the chord generation model in this section is still encoded as a combination of four vectors c_i^{1st}, c_i^{2nd}, c_i^{3rd}, and c_i^{4th}. The structural diagram of this part of the model is the same as the diagram of the chord generation model based on the dual encoding of melody and chord, except that the input melodic sequence becomes composed of the output sequence $r_{1,T}$ of the previous model and the chord output sequence of the previous time step, and the sequence $r_{1,T}$ is passed through the chord encoder and fed to the chord decoder together with $c_{1,t-1}$ to obtain the new chord sequence. The structure of the chord model is shown in Figure 1.

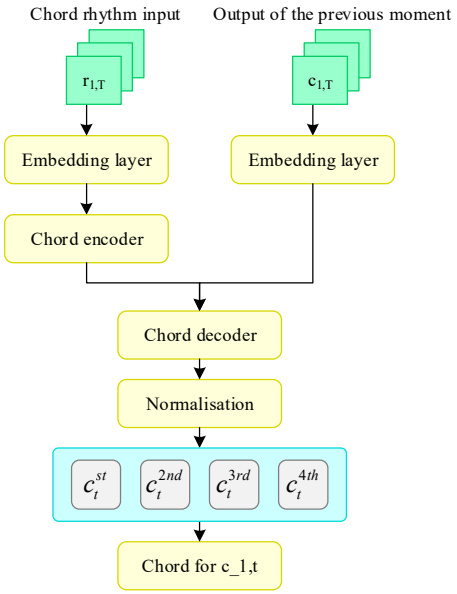


Figure 1: Chord generation model

Although two models are used to realize chord generation, the two models contain both melodic and chord encoding, therefore, the system can also be called a music generation system based on dual encoding of chord and melody. The output vector formula is:

$$c_t^n = M_P\left(r_{1,T}, c_{1,T-1}, \theta_P\right) \tag{31}$$

where M_P denotes the chord model, θ_P is a parameter in it, and $n \in \{1,2,3,4\}$ is the index of the chord vector. The chord sequence c_t at time step t can be combined from the four outputs at time step t.

Similarly, since the output of the chord model is still four, the chord model still corresponds to four loss values in this section. In order not to change the nature of the output chords, we do not set weights for the four loss values in this paper.

In general, modern language models not only require a lot of effort from the researcher to design a generatively controllable structure, but also require a large amount of annotated data for training. However, in some generative tasks, certain properties of the generated sequences are strongly correlated with specific tokens. For example, holding tokens are highly correlated with the rhythmic properties of music. Therefore, controlled generation of note density can be achieved by adjusting the frequency of occurrence of tokens in the sequence. This not only makes it easy to introduce a priori knowledge into the sampling process, but also eliminates the need to redesign the structure or retrain the model with annotated data.

The tempo is related to the note density; the higher the note density, the faster the tempo, and the lower the note density, the slower the tempo. In order to realize a chord generation task with controlled chord tempo, this paper uses a tangent function to modify the logarithmic probability of holding a token according to the parameter density $d \in (0,1)$: the lower the value of d, the fewer chords are generated, and vice versa. The basic idea of the method is to increase or decrease the probability of a given token by a given d, the value of which can be calculated using the following equation.

$$p_h^* = p_h^{\tan\left(\frac{\pi d}{2}\right)} \tag{32}$$

$$p_i^* = \left(p_h - p_h^*\right) \cdot \frac{p_i}{\sum p \setminus h} + p_i \tag{33}$$

where p_h is the original probability of holding the token, p_h^* is the new probability of holding the token, p_i is the original probability of not holding the token, p_i^* is the new probability of not holding the token, and $i \in \setminus h$.

When $d < 0.5$, the gap between the revised probability and the original probability gradually decreases and the probability of holding the token increases. When $d > 0.5$, the gap between the modified probability and the original probability gradually increases and the probability of holding the token decreases. When $d = 0.5$, the probability of holding the token remains unchanged.

IV. Experiments and analysis of results

IV. A. Introduction to the experimental environment and dataset

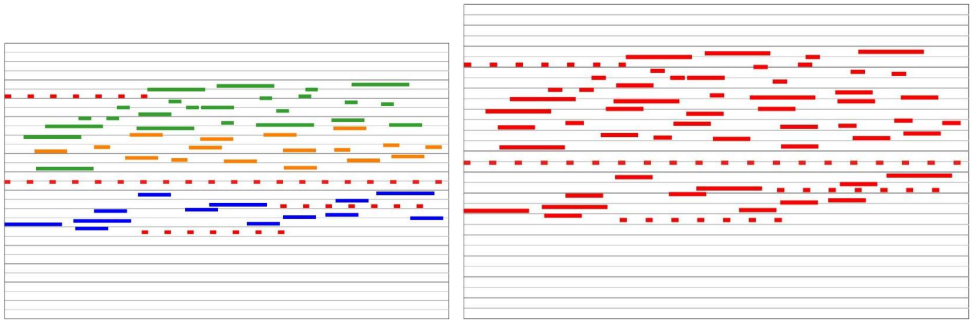
It is generally believed that music of the same genre has the same style, this paper collects and downloads MIDIs labeled with genres from related music platforms on the Internet, and constructs a performance style dataset, which contains four styles, namely classical, country, pop and jazz. All the MIDIs used in this paper are in 4/4 time, and the first beat starts from 0.

The performance style transformation network built in this paper is implemented using the Keras library based on the TensorFlow backend, which is a high-level modular deep learning library that relies on a dedicated backend engine to deal with low-level arithmetic problems, and currently supports the TensorFlow backend, the Theano backend, and the CNTK backend, which can be switched arbitrarily. TensorFlow was developed by Google and is widely used in most deep learning tasks. With TensorFlow, Keras can run seamlessly between CPUs and GPUs.

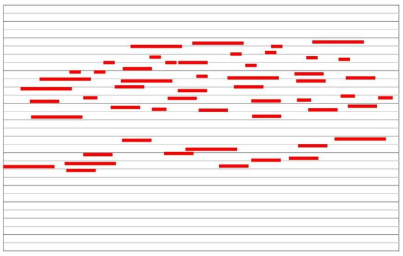
IV. B. Experiments on music emotion representation

Merge multi-track MIDI into mono-track MIDI, and save the merged mono-track MIDI as a new MIDI. the piano rollup before and after merging the tracks is shown in Figure 2, which is the piano rollup of tropic_twilight_lg.mid before and after merging the tracks drawn with MidiEditor, which uses different colors to differentiate between different tracks, and a rectangular bar of different lengths along the horizontal axis time step to indicate the different notes. Along the horizontal axis of the time step using different lengths of rectangular bars to represent the different notes, the length of the rectangular bar represents the note timing, merging tracks only changes the number of tracks in the MIDI file, before and after the merging of the tracks, the pitch of all the notes in the MIDI, the timing, the intensity and its position in the piano rollup should not be changed.

In Fig. 2, Fig. (a) represents the piano rollup before track merging, Fig. (b) represents the piano rollup after merging all tracks, and Fig. (c) represents the piano rollup after removing the percussion track and merging all remaining tracks. Comparing figures (a) and (b), it can be seen that the audio track merging method described in this article effectively merges the audio tracks, and the positions of different notes do not change after merging the audio tracks, and it is found that the percussion audio track in this MIDI is composed of a series of drum beats, usually without a definite pitch, and the piano roll diagram of the percussion track removed when merging the audio tracks is shown in Figure (c), showing that the percussion audio track is effectively removed in this paper.



(a) The piano scroll in front of the sound rail (b) Combine all the piano rolls after the track



(c) Remove the piano scroll of the percussion track

Figure 2: The sound track is combined with the piano roll

IV. C. Music Style Transition

Set the batch size to 64, input the note matrices of different styles in the training set to the playing style conversion network for training, the loss function of each style is adopted as the mean square error, and optimized using the Adam optimizer, with an initial learning rate of 0.001, and train the playing style conversion network by normalizing the true intensity matrices of the different styles as the prediction target of the playing style conversion network for 300 iterations. The loss change curve of the playing style conversion network is shown in Fig. 3, the horizontal axis indicates the iteration rounds Epochs, and the vertical axis indicates the loss Loss, the final loss of the training set is 0.0032, and the loss of the validation set is 0.0056.

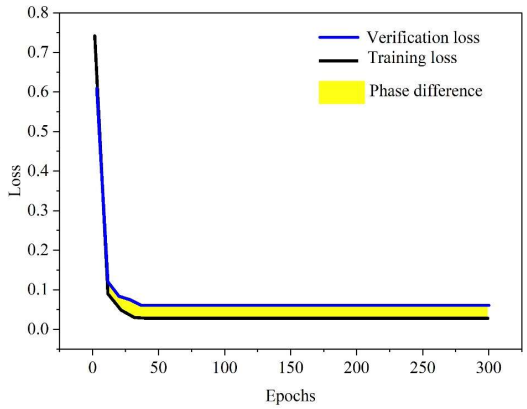


Figure 3: The performance style transformation network loss curve

In order to balance the dataset, the country style with the least number of samples is taken as the benchmark, and 9,900 samples are taken from different styles of MIDI, totaling 39,645 samples to form the strength classification dataset, of which 80% is used as the training set and 20% is used as the test set, and the classification cross entropy is used as the loss function to train the strength classifier. The final accuracy of the strength classifier is 94.25% on the training set and 92.28% on the test set, and the classification accuracy for each style on the test set is shown in Table 1. It can be seen that the trained strength classifier is most ideal for classical styles, while the classification of country and pop is somewhat poor, but basically meets the classification needs of this paper.

Table 1: Accuracy distribution on test set

Style	Classical	Country	In fashion	Sir
Accuracy rate/%	94.55	89.53	91.56	93.46

The average style conversion strength of the test set is shown in Table 2. For the same style, the playing style conversion network can basically keep the original style unchanged after conversion, and the average style conversion strengths are all above 0.80. For different styles, the performance style conversion network is affected by the original style of the song, among which, the conversion intensity between jazz and pop styles is the lowest, which may be due to the high similarity of the intensity distribution of the two styles, because some jazz is easily categorized as pop, and this paper considers the performance styles from the angle of intensity alone, and the performance styles conversion effect is better in general.

Table 2: Test set average style conversion strength

		Predictive style			
		Classical	Country	In fashion	Sir
True style	Classical	0.95	0.53	0.64	0.70
	Country	0.69	0.91	0.67	0.63
	In fashion	0.71	0.56	0.89	0.48
	Sir	0.66	0.58	0.47	0.84

IV. D. Evaluation of music generation results

IV. D. 1) Objective assessment of music quality

Currently researchers have proposed a large number of objective evaluation metrics to judge the quality of generated music, in order to comprehensively evaluate the generated music, this paper objectively evaluates the generated music from the following aspects:

(1) Perplexity (PPL): PPL is a common metric for evaluating the performance of a language model and is calculated as follows:

$$PPL = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(x_i)\right) \tag{34}$$

The lower PPL indicates that the music generated by the model is closer to the real music, and the generation model is more effective.

(2) Pitch category (UPC): UPC indicates the number of different pitches used in a measure, which can reflect the pitch diversity of the generated samples.

(3) Rhythmic Consistency (GC): GC refers to the degree of rhythmic similarity between neighboring sections, the higher the GC value, the more smooth and powerful the music sounds, calculated as follows:

$$GC = 1 - \frac{1}{T-1}\sum_{i=1}^{T-1} d\left(G_i + G_{i+1}\right) \tag{35}$$

(4) Empty Beat Rate (EBR): EBR indicates the proportion of empty beats in a piece of music to the total number of beats. The higher the EBR, the looser the rhythmic feel of the music, and vice versa, the stronger the rhythmic feel of the music, and the formula is as follows:

$$EBR = \frac{empty\_beats}{beats} \tag{36}$$

When assessing the quality of music, in order to avoid the interference of different emotions on the music assessment, 25 pieces of music are generated for each type of emotion in the 4Q classification, i.e., each model generates a total of 100 kinds of music with different emotions, and this paper uses the method provided by the MusPy package to compute the final results to take the arithmetic mean, in order to calculate the scores of the relevant indexes in the real dataset, the same in the validation set is randomly 100 samples with different emotions were selected, and the results of comparing the music generated by different models with the database data in terms of objective indicators are shown in Table 3.

Comparing the results in the table, it can be seen that in the index PPL, the score of this paper's model is much higher than that of m LSTM, which indicates that this paper's music generation model is able to better memorize

the dependency relationship between long sequences when dealing with long music sequences, and the score of this paper's model is the lowest, which is only 1.68, which indicates that the music generated by the model is the closest to the distribution of the music in the real dataset, and the degree of realism is higher.

In addition, in the UPC, GC and EBR indexes, the scores of this paper's model are closer to those of Pop Music Transformer and Compound Word Transformer, but better than the m LSTM model, which indicates that the model architecture of this paper is reasonable, and the scores of this paper's model in the UPC and EBR indexes are the closest to those of the dataset, and have a better score than other models. The scores of this model in UPC and EBR metrics are the closest to the dataset, and there is a large gap with other models, which indicates that the model generates music with better pitch diversity, rhythmic consistency, and structural stability, which further proves that the music generated by this model is closer to human creation.

Table 3: Music quality assessment results

Model	PPL	UPC	GC	EBR
mLSTM	2.18	9.39	0.63	0.19
Pop Music Transformer	1.83	9.26	0.84	0.15
Compound Word Transformer	1.78	9.34	0.86	0.16
Dataset	1.74	8.17	0.79	0.16
This model	1.68	8.15	0.89	0.11

IV. D. 2) Objective assessment of emotional accuracy

In order to facilitate the calculation of the accuracy of the model-generated emotional music, each model generates 100 pieces of music for different emotional categories, and calculates the proportion of the correctly predicted music samples to the total samples, so as to obtain the emotional accuracy of the model, and the results of the emotional accuracy of different models are shown in Table 4.

The music generated by the models in this paper are all higher than the other three models by more than 10 percentage points in terms of emotional accuracy, and the accuracy of emotional expression has been substantially improved, which indicates that the emotional music generated by the models in this paper is more capable of reflecting the target emotion compared to the baseline model, and has a stronger ability to learn more about the emotional conditions of the emotion and more about the potential characteristics of the music emotion.

Table 4: Evaluation of emotional accuracy of music

Model	Accuracy/%
mLSTM	26
Pop Music Transformer	59
Compound Word Transformer	68
This model	85

Figure 4 illustrates the piano roll diagrams of the music generated by the model in this paper under different emotional conditions. With time steps, the model is able to generate repeated music segments (as shown in the green round box in the figure), which indicates that our model is able to learn the repetitive structure of the music and express the emotion of the music by repeatedly emphasizing specific segments of the music, and that the generated music is structurally stable, which is similar to the real music composition, where the emotion of the music has to be lyrical by repeatedly repeating the musical segments of the music to express the creator's emotion, and the experimental The results further indicate the authenticity of the generated music.

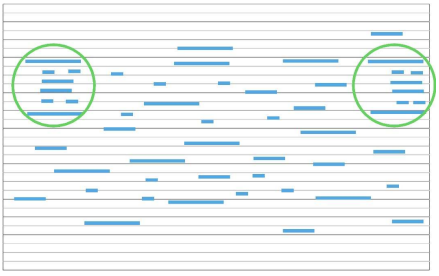


Figure 4: The emotional condition is the piano screen drawing of music

IV. D. 3) Subjective evaluation

Many objective evaluation indexes have been put forward, but music as a product of artistic creation, its evaluation still needs human participation, because it is impossible to use quantitative hard indicators to judge a work of art, only the subjective evaluation of human beings is the most persuasive, for this reason, this paper has designed the relevant subjective evaluation indexes to more comprehensively evaluate the generated music, including Valence: whether the emotion is positive or negative, (Arousal): whether the emotion is calm or excited, (Truth): the degree of similarity with human creation, (Harmony): the degree of melodic smoothness and harmony, (Overall quality): the overall quality level of music. Therefore, this paper designs relevant subjective evaluation indexes to evaluate the generated music more comprehensively, including Valence: whether the emotion is positive or negative, Arousal: whether the emotion is calm or excited, Truth: the degree of similarity with human creation, Harmony: the degree of melodic smoothness and harmony, and Overall quality: the level of the overall quality of the music.

Before the start of the experiment, it is necessary to select the appropriate experimenter, and for all participants in the experiment will need to provide their basic information, including name, age, gender and music experience. The music experience is divided into 5 levels, 1-5 are "I have not studied any music theory or practice", "I have studied music theory or practice in two years", "I have studied music theory or practice for two to five years", "I have studied music theory or practice for more than five years" and "I have a music degree". Forty participants were carefully selected for subjective evaluation of the generated music clips. Participants rate musical compositions on the 5 indicators presented, with the ratings increasing from 1 to 5. Table 5 shows the scoring results.

The model in this paper is better than other models in subjective listening experiments, generating music with more authenticity and harmony, and better overall quality, and in emotional expression, when the provided emotion is positive or negative, and the emotion is excited or calm, the model is able to generate specific emotional music according to the provided emotional conditions, indicating that the model can keenly perceive the change of emotional conditions, and fully learn the emotional characteristics of the music, and the model is able to generate music according to the provided emotional conditions. The overall quality score of the music generated by the model in this paper is 4.1, which is higher than that of Compound Word Transformer, which has a better performance, by 0.3 points. Therefore, the model proposed in this paper performs better in the task of emotional music generation.

Table 5: Subjective evaluation score

	Valence-High	Valence-Low	Arousal-High	Arousal-Low	Truth	Harmony	Overall quality
mLSTM	3.0	3.2	3.6	3.6	3.1	2.5	3.4
Pop Music Transformer	3.4	2.9	4.1	2.7	3.2	4.0	3.7
Compound Word Transformer	3.8	2.9	3.9	2.6	4.1	3.4	3.8
This model	3.8	2.7	4.3	2.3	4.1	4.2	4.1

V. Conclusion

In this paper, we realized intelligent music style transformation and emotion encoding based on deep neural networks, and used artificial intelligence methods to complete automatic music creation.

In the visualization experiment of merging multiple tracks into a single track, the model in this paper effectively merges the tracks and removes the classical track of percussion. The average style conversion strength of the same style of music is above 0.80, which confirms the effectiveness of this paper's method in the music style conversion task.

Objective evaluation experiments show that the multi-emotional music generated by this paper's multi-emotional music generation model outperforms comparison models such as mLSTM in terms of music quality and emotional accuracy. The accuracy of the generated music is tens of percentage points higher than the comparison model. In addition, the emotional music generated by the music generation model in this paper is more realistic and harmonious, and can accurately convey negative or positive emotions. The overall quality score of the generated music reaches 4.1 points, which is 0.3 points better than the better model.

The research in this paper has achieved better results, but there are still many areas of work that have not been addressed, and in future research, it is possible to consider how to further improve the quality of the music after the style conversion and to realize the interaction between the user and the music generation system.

References

[1] Authier, R. (2015). Ecuatorial d'Edgard Varèse: une étude des relations entre le texte et l'organisation musicale. Musurgia, (3), 77-96.

[2] Shen, Y., & Yu, F. (2021). The influence of artificial intelligence on art design in the digital age. Scientific programming, 2021(1), 4838957.

[3] Tao, F., Zou, X., & Ren, D. (2018, October). The art of human intelligence and the technology of artificial intelligence: artificial intelligence visual art research. In International Conference on Intelligence Science (pp. 146-155). Cham: Springer International Publishing.
[4] Zheng, X., Bassir, D., Yang, Y., & Zhou, Z. (2022). Intelligent art: The fusion growth of artificial intelligence in art and design. International Journal for Simulation and Multidisciplinary Design Optimization, 13, 24.
[5] Zhao, J., Guo, L., & Li, Y. (2022). Application of digital twin combined with artificial intelligence and 5G technology in the art design of digital museums. Wireless Communications and Mobile Computing, 2022(1), 8214514.
[6] Pereverzeva, M. V. (2021). The prospects of applying artificial intelligence in musical composition. Russian Musicology, (1), 8-16.
[7] Siphocly, N. N. J., El-Horbaty, E. S. M., & Salem, A. B. M. (2021). Top 10 artificial intelligence algorithms in computer music composition. International Journal of Computing and Digital Systems, 10(01), 373-394.
[8] Posthoff, C. (2024). Artificial Intelligence in the Arts. In Artificial Intelligence for Everyone (pp. 183-186). Cham: Springer Nature Switzerland.
[9] Cheng, L., Leung, C. H., & Pang, W. Y. J. (2022). The International Conference on Music Education Technology 2023: A report. Journal of Music, Technology & Education, 15(2-3), 223-231.
[10] Holbrook, U. A., & Rudi, J. (2022, July). Computer music and post-acousmatic practices. In International Computer Music Conference 2022: Standing Wave (pp. 140-144). International Computer Music Association.
[11] Burlakov, V. V., Dzyurdzya, O. A., Fedotova, G. V., Alieva, A. H., & Kravchenko, E. N. (2020). The modern trends of development of AI technologies. Artificial Intelligence: Anthropogenic Nature vs. Social Origin, 374-383.
[12] Mannuru, N. R., Shahriar, S., Teel, Z. A., Wang, T., Lund, B. D., Tijani, S., ... & Vaidya, P. (2023). Artificial intelligence in developing countries: The impact of generative artificial intelligence (AI) technologies for development. Information Development, 02666669231200628.
[13] Kumar, K., & Thakur, G. S. M. (2012). Advanced applications of neural networks and artificial intelligence: A review. International journal of information technology and computer science, 4(6), 57-68.
[14] Sharifani, K., & Amini, M. (2023). Machine learning and deep learning: A review of methods and applications. World Information Technology and Engineering Journal, 10(07), 3897-3904.
[15] Burlakov, V. V., Dzyurdzya, O. A., Fedotova, G. V., Alieva, A. H., & Kravchenko, E. N. (2020). The modern trends of development of AI technologies. Artificial Intelligence: Anthropogenic Nature vs. Social Origin, 374-383.
[16] Li, K., Lin, Y., Lin, M., & Cheng, X. (2024, December). Discussion on the Foundation of Logic in Artificial Intelligence. In 2024 4th International Symposium on Artificial Intelligence and Intelligent Manufacturing (AIIM) (pp. 474-477). IEEE.
[17] Huang, C. Z. A., Koops, H. V., Newton-Rex, E., Dinculescu, M., & Cai, C. J. (2020). AI song contest: Human-AI co-creation in songwriting. arXiv preprint arXiv:2010.05388.
[18] Corbelli, A. (2024). Beyond the Algorithm. Ethical and aesthetic challenges of AI in music. Itinera, (28).
[19] Mycka, J., & Mańdziuk, J. (2024). Artificial intelligence in music: recent trends and challenges. Neural Computing and Applications, 1-39.
[20] 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.
[21] Sturm, B. L., Ben-Tal, O., Monaghan, Ú., Collins, N., Herremans, D., Chew, E., ... & Pachet, F. (2019). Machine learning research that matters for music creation: A case study. Journal of New Music Research, 48(1), 36-55.
[22] Xinxin Huang,Jialin Liu,Feng Yang,Xu Qiao,Liang Gao,Tingyang Fu & Jianshe Zhao. (2025). Study on GPR Image Restoration for Urban Complex Road Surfaces Using an Improved CycleGAN. Remote Sensing,17(5),823-823.
[23] Wei Hou & Tim Colonius. (2025). Fast and Robust Method for Screened Poisson Lattice Green's Function Using Asymptotic Expansion and Fast Fourier Transform. SIAM Journal on Scientific Computing,47(2),A1198-A1224.
[24] Jin Hui Cao,Chi Xie,Yang Zhou,Gang Jin Wang & You Zhu. (2025). Forecasting carbon price: A novel multi-factor spatial-temporal GNN framework integrating Graph WaveNet and self-attention mechanism. Energy Economics,144,108318-108318.
[25] Xiaowei Wang,Yang Yue,Fan Zhang,Yizhao Wang & Zhihua Zhang. (2025). Active Detection of Interphase Faults in Distribution Networks Based on Energy Relative Entropy and Manhattan Distance. Electric Power Systems Research,241,111397-111397.
[26] Zihao Ning,Xiao Han & Jie Pan. (2024). Semi-supervised emotion-driven music generation model based on category-dispersed Gaussian Mixture Variational Autoencoders.. PloS one,19(12),e0311541.