International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 4628-4641

https://doi.org/10.70517/ijhsa464306

Topological ordering-guided optimization model and empirical analysis of the interaction between AI music generation and human composition

Qi Liu[1,*]
[1] Department of Art and Technology, School of Music and Dance, Communication University of Zhejiang, Hangzhou, Zhejiang, 310018, China
Corresponding authors: (e-mail: 20140048@cuz.edu.cn).

Abstract The rapid development of digital music industry promotes the in-depth application of artificial intelligence (AI) in the field of music creation. Aiming at the problems of insufficient emotion expression and limited human-computer interaction in AI music generation, this paper constructs a topological sort-guided optimization model for AI music generation and human-composition interaction. Methodologically, a topological network structure characterization is used to establish a guitar chord generation mechanism, the quality of music generation is optimized by Deep Convolutional Generative Adversarial Network (DCGAN) combined with unilateral label smoothing and feature matching, emotion-driven music creation is realized based on the emotion-guided diffusion model, and a hierarchical attention mechanism is designed to enhance the rhyme and emotional expression of the lyrics. The experimental results show that the model achieves an excellent performance of 4.5698, 0.2485, 0.0455, 0.0198 on seven objective evaluation indexes such as PR, PE, PH, SC, etc., and the total subjective evaluation score is 4.3485, with a mean value of 7.7419 and 8.3089 on the Lakh MIDI and MUT MIDI datasets, respectively. The study verifies that the effectiveness of the topological ordering guidance mechanism in improving the quality of AI music generation and human-computer interaction experience, which provides a new technical path for intelligent music creation and promotes the development and application of AI music generation technology.

Index Terms Topological sorting, AI music generation, deep convolutional generative adversarial network, emotion guidance, diffusion model, hierarchical attention mechanism

I. Introduction

In today's era of rapid technological development, Artificial Intelligence (AI) has permeated all areas of our lives, and music composition is no exception. In the past, music creation often relied on the inspiration, talent and years of professional training of human musicians [1]. Behind every classic piece of music, the creator's countless days and nights of effort and emotion are condensed. However, the birth of AI music generation technology breaks the limitations of the traditional creation mode, which makes music creation no longer only the patent of a few professionals, and also provides a platform for those who love music but lack professional skills to show themselves [2]-[5]. However, AI music generation has also caused some controversy and thinking, which weakens the creativity and uniqueness of human musicians, resulting in the music works becoming the same, and the generated music works must also undergo post-reassembly if they want to be loved by the audience, because the music generated by AI has no emotion, and there is a lack of an effective interaction framework between artificial creation and AI generation, which requires the optimization of the interaction between the two creative methods [6]-[9].

Topological sort, as an algorithm for linearly ordering vertices of directed acyclic graphs, has a wide range of applications in scenarios such as project management, task scheduling, and course scheduling, and is especially suitable for planning task sequences with dependencies. Therefore, it can realize the optimization of AI-generated music and manual creation interaction, and provide mutual coordination between AI in music creation and manual creation, so as to create better music works [10]-[13].

In this study, by introducing the topological ordering theory to construct an ordered relational mapping between music elements, we designed a music generation framework based on deep convolutional generative adversarial networks, and integrated the emotion guidance mechanism and diffusion modeling techniques. Specifically, the topological network structural feature analysis method is used to establish a mathematical model for guitar chord generation, the spatial feature extraction capability of deep convolutional generative adversarial network is utilized to improve the quality of music generation, the emotion-guided diffusion model is used to achieve controllable and emotional music creation, and the layered attention mechanism is designed to enhance the rhythmic and emotional expression effect in lyrics generation. The whole technology route focuses on multimodal fusion and human-computer collaboration, and strives to improve the intelligence level and user experience of music creation while ensuring the generation quality.

II. AI music generation based on topological ordering guidance

II. A. Guitar Chord Generation

II. A. 1) Topological network structure characteristics

The degree of a node is a simple but important concept in complex networks. The degree K_i of a node i is defined as the number of other nodes connected to that node i. Its mathematical significance is expressed in terms of adjacency matrix as shown in equation (1):

$$K_i = \sum_{j}^{N} w_{ij} \tag{1}$$

For directed networks, the degree of a node is categorized into the in-degree $K_i^{in} = \sum_{j}^{N} w_{ij}$ and the out-degree $K_i^{out} = \sum_{j}^{N} w_{ij}$. The out-degree refers to the number of edges pointing away from the current node i, and the in-degree refers to the number of edges pointing to that node i. The average degree is the average of the degrees K_i of all nodes i in the network, denoted as $< k >$. Intuitively, a larger degree of a node indicates that the node is more important to some degree.

II. A. 2) Average shortest path and network diameter

The number of connected edges between node i and node j is defined as the distance between the two nodes and the minimum value in the distance is defined as the shortest path d from node i to node j. The maximum value of all the shortest paths in the network is defined as the diameter D of the network. The mathematical formulation of the network diameter is shown in equation (2):

$$D = \max_{i,j} d_{ij} \tag{2}$$

The average of the shortest distances between any two nodes in the network is defined as the average shortest path L, viz:

$$L = \frac{1}{\frac{1}{2}N(N+1)} \sum_{i \geq j} d_{ij} \tag{3}$$

where N represents the number of nodes in the network, and for ease of handling, equation (3) includes the distance from the node itself to itself (the distance is zero). In recent years it has been found that in many networks of considerable size, the average shortest path of the network is indeed surprisingly small.

II. A. 3) The rich man's club and the coefficient of congruence

There are a small number of nodes in the network that have a large degree, these nodes are called “rich nodes”. These rich nodes prefer to connect with each other, a phenomenon known as “rich club”. The “rich club” phenomenon can be characterized by the rich club connectivity $\phi(r)$, which represents the ratio of the actual number of edges L between the top r nodes with the largest degree in the network to the total number of possible edges $r(r-1)$ between these r nodes, i.e.,:

$$\phi(r) = \frac{L}{r(r-1)} \tag{4}$$

The degree correlation described above can also be described by the covariance coefficient. The definition of the covariance coefficient is shown in the following equation:

$$r=\frac{M^{-1}\sum_{i} j_i k_i-\left[M^{-1}\sum_{i}\frac{1}{2}(j_i+k_i)\right]^2}{M^{-1}\sum_{i}\frac{1}{2}(j_i^2+k_i^2)-\left[M^{-1}\sum_{i}\frac{1}{2}(j_i+k_i)\right]^2} \tag{5}$$

where j_i and k_i are the degrees of the nodes at the two endpoints of the i th edge, respectively, and M is the number of edges in the network. If the coefficient of congruence $r>0$, then the network is congruent and nodes with larger degrees in the network are interconnected. If $r<0$, the network is heterocompatible, points with larger degrees in the network are not very connected, and nodes with larger degrees have lower average degrees of their neighboring nodes.

II. B. Music generation based on deep convolutional generative adversarial networks

II. B. 1) DCGAN model based solution idea

Deep Convolutional Adversarial Network (DCGAN) is a derivative model that combines convolutional neural network and generative adversarial network, which introduces the idea of convolutional operation into the generative model for unsupervised training, uses convolutional neural network and transposed convolution as the network layer structure of generator and discriminator, and improves the training effect of generative adversarial network and the quality of generated image results with the help of the spatial feature extraction ability of convolutional operation. The structure is shown in Figure 1 [14].

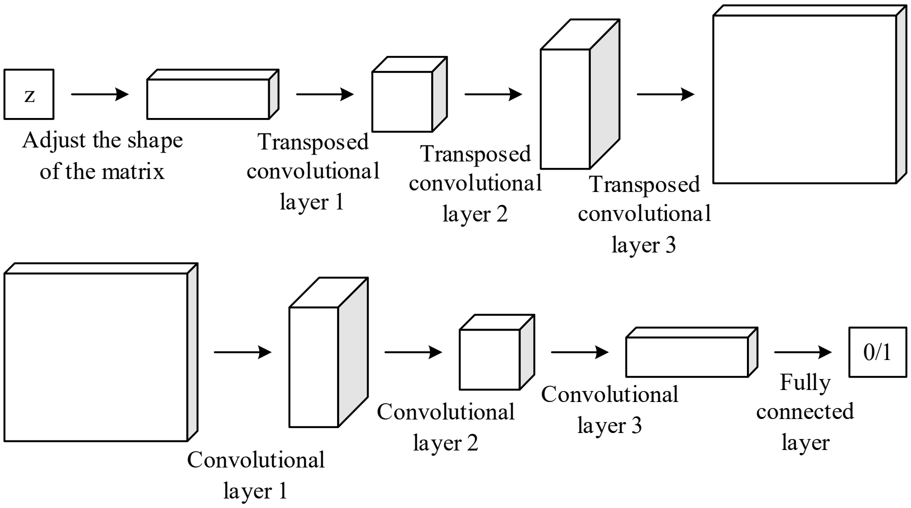
z
Adjust the shape of the matrix
Transposed convolutional layer 1
Transposed convolutional layer 2
Transposed convolutional layer 3
Convolutional layer 1
Convolutional layer 2
Convolutional layer 3
Fully connected layer
0/1

Figure 1: Structure of the generator and discriminator network of DCGAN

Compared to the original generative adversarial model, DCGAN has the following features:

(1) The discriminator uses convolutional steps instead of spatial pooling, and transposed convolutional operations are used in the generator to expand the data dimensions, both of which discard the pooling layer of CNN.

(2) The generator and discriminator use a (BN) layer after each convolutional layer, i.e., a batch normalization layer, which is performed independently on each dimension of a batch of data. The batch normalization primitive arithmetic formula is as follows:

$$y^{(k)}=\gamma^{(k)}\frac{x^{(k)}-\mu^{(k)}}{\sqrt{(\sigma^{(k)})^2+\varepsilon}}+\beta^{(k)} \tag{6}$$

Here, k is the dimension of the data, $x^{(k)}$ is the data before the batch normalization operation, $y^{(k)}$ is the data after the batch normalization operation, $\mu^{(k)}$ is the mean of the input data batch, $\sigma^{(k)}$ is the standard deviation of the input data batch, $\beta^{(k)}$ is a learnable translation parameter, $\gamma^{(k)}$ is a learnable scaling parameter, and ε is a very small value that prevents the denominator from being zero.

The advantage of the batch normalization operation is that it is applicable to a variety of deep neural network structures, can improve the accuracy and generalization ability of the model, effectively avoid model overfitting, and the effect is more obvious when there are few training samples or uneven distribution.

II. B. 2) Structure of the DCGAN-based music generation model

The architecture of the generator consists of a fully connected layer, a transposed convolutional layer and a convolutional layer. An input batch is 72, so the input random vector z is a Gaussian noise vector of size 72 × 100. z goes through the fully connected layer to change its shape and then enters the transposed convolutional layer. In the first three transposed convolutional layer operations, the convolution kernel size is (2, 1) and the step size is set to (2, 2). In the fourth transposed convolutional layer, the convolutional kernel size is (1, 128) and the step size is set to (1, 2). Since the structure of DCGAN is then used, each transposed convolution is processed using a batch normalization operation, and the activation function for each layer is a ReLU function. In contrast, the startup music bars are input vectors of size 128×16 that enter the convolutional layers, in the first convolutional layer the convolutional kernel size is (1, 128) with a step size of (1, 2), and in the second to fourth convolutional layers the convolutional kernel size is (2, 1) with the step size set to (2, 2). It can be seen that the process of starting the convolution operation of the music bars in the generator can be seen as the inverse operation of the generator transposed convolution, and also with reference to the structure of DCGAN, the batch normalization operation is carried out for each convolution, while the activation function of each layer is used is the Leaky-ReLU, with the formula:

$$\mathrm{LReLU}(x_i)=\begin{cases}x_i, & x_i>0\\ \alpha_i\cdot x_i, & x_i\le 0\end{cases} \tag{7}$$

Here, α is the coefficient required when the input x is negative and i is the number of convolution channels.

When greater than 0, Leaky-ReLU is a linear function, and when less than 0 it will control the data through the set parameters, introducing a nonlinear transformation while mitigating the gradient vanishing problem. In order to maintain the sensitivity of the training process model and help capture more complex data patterns, the α parameter is set to 0.2 in this model.

II. B. 3) Optimization and adaptation of the loss function and model structure

In order to solve the problems of training instability and mode crash that are prone to generative adversarial networks, this chapter further improves the architecture and loss function of the DCGAN model. For the loss function, the method of one-sided label smoothing and feature matching is used, and for the model structure, the small batch discriminator is used to replace the original discriminator structure.

When the discriminator thinks that the generated sample is too poor, or too different from the real sample, the error sample from the model cannot be stimulated to get close to the real data, therefore, the positive label needs to be smoothed from 1 to α , and the sublabel is set to 0 where α only needs to be smoothed to a number slightly less than 1, and 0.9 is taken here to replace the original discriminator:

$$D^*(x)=\frac{\alpha p_{data}(x)+\beta p_g(x)}{p_{data}(x)+p_g(x)} \tag{8}$$

The original generative adversarial network objective function needs to maximize the output of the discriminator, and feature matching is achieved by specifying a new objective that requires the generator to produce results that match the real samples as closely as possible, while the discriminator is used to specify only the data that is worth matching as a way of avoiding overfitting of the generator in response to the requirements of the discriminator. The specific implementation method is to let the generated sample through the discriminator intermediate layer when the features and the features of the real sample as much as possible the same, in the objective function to introduce a penalty term. Under this premise, the objective function of the generator at this point in training is:

$$\begin{aligned} loss &= loss_G + loss_{FM} \\ &= loss_G + \| E_{x\sim p_{data}(x)} f_D(x) - E_{z\sim p_z(z)} f_D[G(z)] \|_2^2 \end{aligned} \tag{9}$$

where $loss_G$ is the generator loss function and $loss_{FM}$ is the feature matching penalty term.

f_D is the output of the middle layer of the discriminator. In this model, two L2 regularization terms are actually introduced as penalty terms to force the generated pianoroll matrix to be close to the real music bar matrix.

$$\begin{aligned} loss &= loss_G + loss_{FM} \\ &= loss_G + \lambda_1 \| EX - E[G(z)] \|_2^2 + \lambda_2 \| E_{x \sim p_{data}(x)} f_D(x) - E_{z \sim p_1(z)} f_D[G(z)] \|_2^2 \end{aligned} \tag{10}$$

The f_D function in the regularization term takes the first convolutional layer in the discriminator, i.e., it takes the result after the first convolution as the feature of the generated result and the real data, and the λ_1 and λ_2 empirical parameters are taken as 0.1 and 0.01 respectively.

The discriminator can only process one sample independently at a time, and there is no coordination between the gradients. In the event of pattern collapse phenomenon, the samples are under similar or the same pattern, resulting in the gradient update information fed back from the discriminator to the generator also pointing to the same direction. The idea of improving the small batch discriminator is to let the discriminator no longer consider a sample independently, but consider a batch of samples at the same time.

Assume that the feature vector of a sample x_i at a certain layer of the discriminator network is $f(x_i)$, multiply $f(x_i)$ by a tensor parameter T^{A*B*C} to get a tensor M_i^{B*C}, and then compute the L1 paradigm number for each M_i, i.e. $c_b(x_i, x_j)$, and then sum all $c_b(x_i, x_j)$ to get $o(x_i)_b$, and $o(x_i)_b$ denotes the difference between the sample x_i and that of the sum of the differences of the b th feature of the other samples in the batch, and then all $o(x_i)_b$ are combined to obtain a vector $o(x_i)$ of size B:

$$o(x_i)_b = \sum_{j=1}^{n} c_b(x_i, x_j) \in \mathbb{R} \tag{11}$$

$$o(x_i) = [o(x_i)_1, o(x_i)_2, \cdots, o(x_i)_B] \in \mathbb{R}^B \tag{12}$$

$$o(X) \in \mathbb{R}^{n \times B} \tag{13}$$

Among them:

$$c_b(x_i, x_j) = \exp(-\| M_{i,b} - M_{j,b} \|_{L.l}) \in \mathbb{R} \tag{14}$$

The vectors $o(x_i)$ and $f(x_i)$ are combined as inputs to the next layer of the discriminator network. Compared to the original discriminator structure, here there is an additional mini-batch layer whose input is $f(x_i)$ and whose output is $o(x_i)$, with a learnable tensor parameter T in between. The mechanism of the original discriminator is to determine the probability that a sample originates from the training dataset. The discriminator that introduces a small batch of discriminative layers has no change in the training principle, but its output no longer depends on a single sample only, but sends information about the differences between samples in a small batch to the next layer along with this batch of samples, which can effectively avoid the situation where the generator is trapped in a single pattern sample.

When the generator needs to be updated for a mode collapse condition, the generator is first generated into a batch of samples $\{G(z)_1, G(z)_2, \ldots G(z)_m\}$, and the mini-batch layer results are computed by the small-batch discriminator $\{o(G(z)_1), o(G(z)_2), \ldots o(G(z)_m)\}$. In the case of a mode collapse, this batch of samples is in the same mode, and the computed results of the mini-batch layer are bound to be very different from those of the real dataset used to train the discriminator. The minibatch layer result will be very different from that of the real data set used to train the discriminator, and the captured difference information will keep the minibatch discriminator $D(G(z_i))$ value from being too low, so the minibatch discriminator will not simply give the same gradient direction to all the samples, which ultimately serves to stabilize the training and attenuate the phenomenon of mode collapse.

In this model, similar to the feature matching method mentioned above, here $f(x_i)$ is still taken from the resultant features of the first convolutional layer of the discriminator as the input of the mini-batch layer, and at the same time, in order to simplify the model, the original method of the mini-batch discriminator is not used, and another version of the mini-batch discriminator is used, which is similar to the above mentioned method in idea, except that it simplifies the computation and does not introduce new tensor parameters that require additional learning. For a batch of input samples $\{G(z)_1, G(z)_2, \ldots. G(z)_m\}$, the first convolutional layer of the discriminator is used as the features of the samples, and the standard deviation of each dimension is computed and the mean is used as the output of the mini-batch layer:

$$\text{o} = \frac{1}{n}\sum_{i=1}^{n}(\sigma_i) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\frac{1}{m-1}\sum_{j=1}^{m}(f(x_j)_i - \hat{f}_i)2} \tag{15}$$

II. C.Music generation network based on emotion guidance-diffusion modeling

II. C. 1) Diffusion models

Diffusion models are a class of probabilistic generative models that learn the mapping of noise to data by iteratively refining the noise samples. Denoising Diffusion Probabilistic Model (DDPM) is a typical diffusion approach [15]. DDPM defines a forward process that gradually transforms the input into Gaussian noise, while learning an inverse process for recovering the input. Specifically, in the forward diffusion process, the original sample x_0 is gradually added with noise after T steps to generate a series of noisy samples $x_1, x_2, \ldots, x_T$. At each time step t, the conditional probability distribution of the sample x_t is determined by the sample x_{t-1} at the previous moment, which has the mathematical form:

$$q(x_t \mid x_{t-1}) = N(x_t \mid \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{16}$$

where $\beta_1, \ldots, \beta_1, \ldots, \beta_T$ is a predefined noise scheduling parameter. Based on the nature of Gaussian distribution, it can be derived:

$$q(x_t \mid x_0) = N(x_t \mid \sqrt{\bar{\alpha}_t x_0}, (1-\bar{\alpha}_t)I) \tag{17}$$

where $\bar{\alpha}_t = \Pi_{s=1}^{t}\alpha_s, \alpha_t = 1-\beta_t$. By sampling $\varepsilon \sim N(0, I)$ and using the reparameterization trick, one can obtain the sample $x_t = \sqrt{\bar{\alpha}_t x_0} + \sqrt{1-\bar{\alpha}_t}\varepsilon$. Under certain conditions, the distribution $q(x_T)$ of the final step is approximated as a standard Gaussian distribution.

The inverse generation process starts with pure noise samples x_τ and progressively denoises and reconstructs $x_{T-1}, x_{T-2}, \ldots, x_0$, and finally obtain a realistic sample. The inverse process is defined as a conditional probability distribution $p_\theta(x_{t-1} \mid x_t)$, which is learned by a neural network for approximating $q(x_{t-1} \mid x_t, x_0)$. To learn $p_\theta(x_{t-1} \mid x_t)$, simply train the model output $\varepsilon_\theta(x_t, t)$ to recover the noise ε that was added in generating x_t. The loss function for training the diffusion model is $E_{t,x_0,ò}[\| \varepsilon - \varepsilon_\theta(x_t, t) \|^2]$. In inference, given x_t and the predicted noise, it can be sampled from $p_\theta(x_{t-1} \mid x_t)$ by the following equation:

$$x_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)) + \beta_t z \tag{18}$$

Of these, $z \sim N(0, I)$.

II. C. 2) Human emotion-guided music generation

The overall architecture consists of two components, the VAE module and the emotionally guided diffusion-based bootstrap module.The VAE module is used to compress the representation $S \in R^N$ of a musical source containing N sample waveform domains into a compact and contiguous latent space while ensuring that the reconstruction result is perceptually indistinguishable from the original source. Given an input signal S, the encoder maps it to a posterior distribution:

$$\psi_{enc}(S) = N(\cdot \mid \mu_z(S), \sum_z(S)) \tag{19}$$

where $\mu_z(S) \in R^{C \times N/D}$ is the potential a posteriori mean, $\sum_z(S)$ is the a posteriori covariance matrix, D=320 is the time-domain downsampling factor, and C=80 is the potential spatial dimension.

After encoding, the signal S is reconstructed by sampling $z \sim N(\cdot \mid \mu_z(S), \sum_z(S))$ and feeding it to the decoder. To further simplify the extraction of potential features, this paper directly uses the posterior mean $z_s = \mu_z(S)$ as the potential representation.

Modeling the process of music generation in the latent space, this paper further introduces music emotion information to guide the generation of latent variables based on the latent diffusion model. Musical affective potency is an important feature of musical expression [16]. Russell's two-dimensional model of affect uses the dimensions of pleasantness and arousal as the broadest indicators, whose continuity and and tunability are well suited to capture such dynamic changes in musical affect. By directly modulating the values of pleasantness and arousal, variables in the latent space can be guided to generate musical content with specific emotional characteristics.

Let the potential representation of multiple input music sound sources be $Z = (z_1, z_2, ..., z_K) \in R^{K \times C \times N/D}$, where z_i is the potential representation of the i th music segment and K is the number of segments. Music emotion information is generated by an emotion encoder that maps the emotion description to a feature matrix $E \in R^{M \times C}$ in the potential space, where M is the number of emotion features. During the generation process, a cross-attention mechanism is used to combine the sentiment information with the latent representation:

$$A = \text{softmaxg}(\frac{ZE^T}{\sqrt{c}} g), \tilde{Z} = Z + AE \tag{20}$$

where $A \in R^{K \times M}$ is the attentional weight and $\tilde{Z}$ is the latent representation after incorporating emotional information. In the diffusion process, according to the time scheduling $\sigma(t) = t$, the forward diffusion process is defined as:

$$d\tilde{Z}(t) = -\sigma(t)\nabla_{\tilde{Z}(t)} \log pl(\tilde{Z}(t)r)dt \tag{21}$$

where $\tilde{Z}(t) = N(\tilde{Z}(0), \sigma^2(t)I), \tilde{Z}(0) = \tilde{Z}$. And then sample $\tilde{Z}$ by solving the ODE inverse process:

$$d\tilde{Z}(t) = \sigma(t)\nabla_{\tilde{Z}(t)} \log pl(\tilde{Z}(t)r)dt \tag{22}$$

where the score term $\nabla_{\tilde{Z}(t)} \log p(\tilde{Z}(t))$ is approximated by the neural network $S^{\theta}(\tilde{Z}(t), \sigma(t))$, and trained by the score matching loss.

III. Emotionally and rhythmically enhanced lyric generation

III. A. Lyrics and Rhymes

Lyrics, as an important part of a song, serve to clarify the main idea of the song. Lyrics are derived from poems, and the creation of lyrics needs to follow rules similar to those of poems, such as formatting (word pattern) and rhyming. Lyrics generation is a sub-task of text generation, and unlike open domain text generation, lyrics generation requires effective control of word pattern, rhyme and emotional expression, so that the generated lyrics can complement the music. Grammar is the formatting rule that lyrics need to follow strictly to ensure that they can be smoothly integrated into the rhythm of the music and are easy to sing. Rhyme is also a key consideration in the process of generating lyrics. Maintaining appropriate rhymes within a certain range can make the lyrics catchy and enhance the sense of rhythm and fluency. In terms of emotional expression, as the lyrics usually most directly reflect the emotion of the entire song, it is important to ensure that the generated lyrics can match the emotional tone of the music and maintain emotional consistency throughout the lyrics.

III. B. Modeling

III. B. 1) Lyrics Text Input Representation

In this chapter, we explicitly model the clauses, internal positions, rhymes and moods of lyrics to form a joint input representation, which enhances the expressive power of such information in the input. The following is an example of the input representation of the lyrics “your tears/softly hurt” (excerpted from Jay Chou's “Chrysanthemum Terrace”, using “/” to indicate the position of the break):

(1) Sentence marking:

$$SEG = \{s_0, s_0, s_0, s_0, \langle /s \rangle, s_1, s_1, s_1, s_1, s_1, \langle /s \rangle, \langle eos \rangle\} \tag{23}$$

where s_i denotes the token of the i th sentence, $\langle /s \rangle$ is the inter-sentence separator, and $\langle eos \rangle$ is the sentence termination marker. By setting the sentence separator token, the model can explicitly learn the association information of different sentences in the lyrics.

(2) Internal position markers:

$$POS = \{p_3, p_2, p_1, p_0, \langle /s \rangle, p_4, p_3, p_2, p_1, p_0, \langle /s \rangle, \langle eos \rangle\} \tag{24}$$

where p_i denotes the penultimate $i+1$ word of the sentence (i.e., i is the number of remaining words of the sentence). In the lyrics generation task, the number of words in each sentence is strictly specified according to the word frame, and the function of setting the internal position markers is to enable the model to learn the information of the number of words in each sentence, so as to avoid incomplete forced truncation in the generation process.

(3) Rhyme markers:

$$RHY = \{c, c, c, r_m, \langle /s \rangle, c, c, c, c, r_n, \langle /s \rangle, \langle eos \rangle\} \tag{25}$$

where c is the general character in the sentence, r_m denotes the rhyme of the rhyming word at the end of the sentence, and $m, n \in \{0, 1, \cdots, 13\}$, denotes one of the 14 rhymes. In particular, if the rhyming pattern of the lyrics is uncertain and no fixed rhyme scheme exists, the end word of the stanza is still denoted by c.

(4) Sentence-level mood markers:

$$EMO = \{e_m, e_m, e_m, e_m, \langle /s \rangle, e_n, e_n, e_n, e_n, \langle /s \rangle, \langle eos \rangle\} \tag{26}$$

where $m, n \in \{0, 1, \cdots, 7\}$, denotes one of the 8 emotion categories.

After constructing the input tokens as above, the tokens for each part are mapped into fixed-length vector representations through the embedding layer, and then the embedding vectors of each type are summed to obtain the initial hidden state representation H^0 at layer 0 as in Equation (27):

$$H_t^0 = E_{w_t} + E_{SEG_t} + E_{POS_t} + E_{RHY_t} + E_{EMO_t} + E_{g_t} \tag{27}$$

where t is the position index, E_* is the embedding vector of input *, w is the original input token, and g is the global position index, as implemented in Transformer.

In order to enable the model to capture the dynamic information of the global sequence for word pattern and rhyme, the global information representation F^0 is introduced as in Equation (28):

$$F_t^0 = E_{SEG_t} + E_{POS_t} + E_{RHY_t} \tag{28}$$

In order to make the sentiment information always noticed during the model generation process, this chapter also introduces the global sentiment representation S^0, as in Equation (29):

$$S_t^0 = E_{EMO_t} + E_{g_t} \tag{29}$$

III. B. 2) Hierarchical attention mechanisms

After the inputs are embedded for representation, in order to make the model learn the rhyme and emotion information in the lyrics more fully, two layers of attention mechanisms are introduced in this chapter, where the first layer is the masked self-attention and emotion attention, and the second layer is the global attention, and the hierarchical attention structure is shown in Fig. 2. The design details of each layer of the attention mechanism are described in detail next.

(1) Masked self-attention

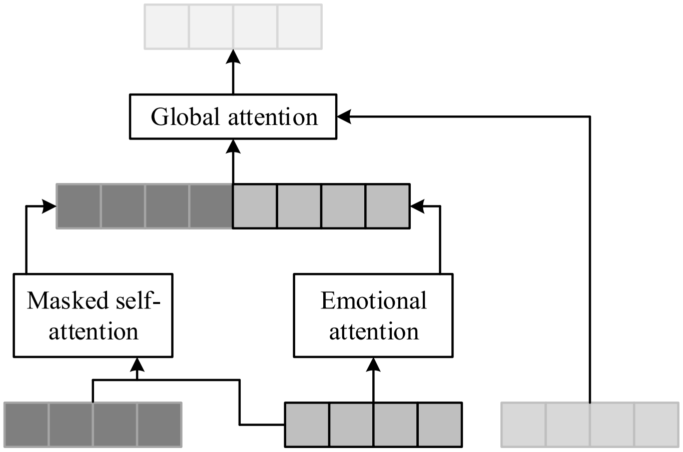

Figure 2: Hierarchical Attention Structure

$$\begin{aligned} &Q^0, K^0, V^0 = H^0 W^Q, H^0 W^K, H^0 W^V \\ &C_t^1 = LayerNorm(Attention(Q_t^0, K_{\le t}^0, V_{\le t}^0) + H_t^0) \\ &C_t^1 = LayerNorm(FFN(C_t^1) + C_t^1) \end{aligned} \tag{30}$$

where W^* is the optimizable parameter matrix, and LayerNorm(·), Attention(·) and FFN(·) are the layer normalization, attention, and feed-forward network, respectively. In order to prevent the information after time step t from being input to the model in advance and causing information leakage, only the content from that time step and before is considered in time step t, which is denoted by subscript $\le t$.

(2) Emotional attention

Emotional attention is calculated in a similar way to the masked self-attention in the same layer, as shown in Equation (31):

$$\begin{aligned} &Q^0, K^0, V^0 = H^0 W^Q, S^0 W^K, S^0 W^V \\ &T_t^1 = LayerNorm(Attention(Q_t^0, K_{\le t}^0, V_{\le t}^0) + H_t^0) \\ &T_t^1 = LayerNorm(FFN(T_t^1) + T_t^1) \end{aligned} \tag{31}$$

After obtaining the first layer representations C_t^1 and T_t^1, they are spliced and passed through the FFN module, which is used as the input information for the global attention as shown in Equation (32):

$$\begin{aligned} &U_t^1 = [C_t^1 : T_t^1] \\ &U_t^1 = LayerNorm(FFN(U_t^1) + H_t^0) \end{aligned} \tag{32}$$

(3) Global attention

In order for the model to capture the global dynamic information from F^0, the global attention is set as shown in Equation (33):

$$\begin{aligned} &Q^1, K^1, V^1 = U^1 W^Q, F^0 W^K, F^0 W^V \\ &H_t^1 = LayerNorm(Attention(Q_t^1, K^1, V^1) + U_t^1) \\ &H_t^1 = LayerNorm(FFN(H_t^1) + H_t^1) \end{aligned} \tag{33}$$

After two layers of attention, the hidden layer representation H_t^1 is obtained. In the specific model training process, the number of attention layers is set to L, then the final hidden layer representation is obtained as H^L.

(4) Loss function

The lyrics generation task in this chapter is a typical autoregressive generation, which adopts the negative log-likelihood as the loss function, as shown in equation (34) [17]:

$$\mathrm{L}_{NLL} = -\sum_{t=1}^{n} \log P(y_t \mid y_{<t}) \tag{34}$$

where n is the number of samples.

IV. Experimental design

IV. A. Data sets and data preprocessing

The dataset used in this chapter is the open-source Chinese lyrics dataset MusicLyricChatbot, which has been used in previous studies. The raw data of MusicLyricChatbot contains the lyrics of 140,694 songs, separated by lines. The vast majority of the lyrics in this dataset are in Chinese, and a small number of lyrics in other languages such as English, Japanese, and Korean are also included. In order to fit the tasks in this chapter, the dataset is cleaned and preprocessed in the following steps:

(1) Remove non-Chinese lyrics

The lyrics containing non-Chinese characters are matched and removed using regular expressions, and it should be noted that this operation also removes Chinese lyrics containing a small number of other languages. As the research in this chapter involves the processing of rhymes in lyrics, the mixed rhymes of multiple languages have some complexity in the determination, in order to simplify the task, this chapter only retains the lyrics that are entirely in Chinese in the data.

(2) Remove punctuation and other irrelevant characters

The lyrics themselves are written on a line-by-line basis, and the separation between sentences does not need to be recognized by punctuation marks, so this chapter removes all punctuation from the data.

(3) Remove data of too short a length

There are some too short lyrics in the dataset, which are usually invalid data, so this chapter removes the lyrics in the dataset that are too short in length (the number of sentences is less than 4).

(4) Sentence-level rhyming annotation

In order to achieve rhyme-enhanced lyrics generation, this chapter pre-labeled the rhyming feet of each lyrics data automatically, rather than just letting the model implicitly learn the rhyming rules during the training process. Lyrics are a special form of poetry, and the rhyming rules in lyric writing are more similar to those of poetry. In this chapter, the Chinese New Rhyme (fourteen rhymes), which is currently commonly used in the literary world, is chosen as the reference for rhyme labeling. Specifically, for the last word of each line of the lyrics, the open-source pinyin conversion tool pypinyin is utilized to obtain the pinyin rhyme of the word and align it to one of the 14 types of rhymes [18].

IV. B. Performance evaluation

IV. B. 1) Objective assessment

(1) Entropy distribution

Two nonlinear features, their association dimension and Kolmogorov entropy, are extracted from the generated music of four emotions: excitement, calmness, tension and sadness, and Fig. 3 shows the Kolmogorov entropy distribution of the generated music.

The difference of different emotion music in the two nonlinear features of association dimension and Kolmogorov entropy is obvious, especially the sad emotion, in the interval of embedding dimension 2~18, the Kolmogorov entropy distribution is in [1854.0724,2551.8948], which indicates that the music generated by the emotion-guided DCGAN model designed in this paper has a certain effect of emotion expression.

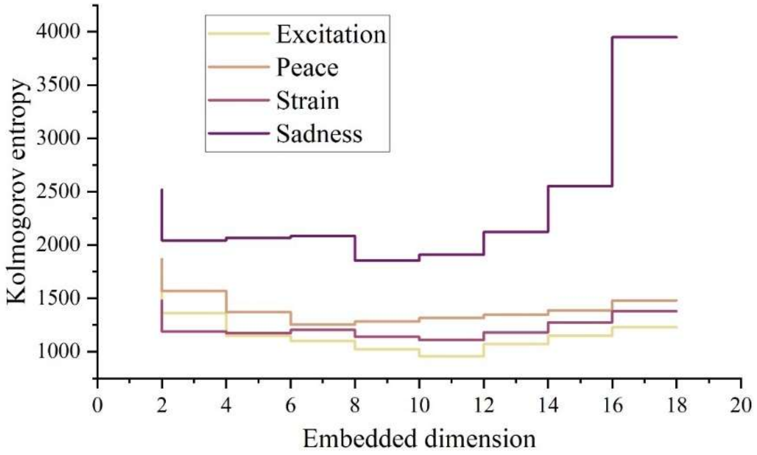

Figure 3: The Kolmogorov entropy distribution of music

(2) Music Quality

In order to verify the validity of the model proposed in this paper, the model proposed in this paper was compared with the Transformer -XL model, the CEG-Transformer model, and the CP Transformer model. Eight pieces of music for each of the four emotions of excitement, calmness, tension and sadness were generated using these four models. The 32 pieces of generated music were evaluated objectively against the music in the original dataset in terms of pitch, tempo, and structure to obtain the mean data of the different models on the 10 evaluation metrics. Finally, the indexes of generated music and the indexes of music in the original dataset are made differences respectively, and the difference results of the evaluation indexes of music generated by different models and real data are obtained as shown in Table 1.

The values of the seven objective evaluation indexes of PR, PE, PH, SC, EBR, SI_mid and SI_long for the music generated by this paper's model are 4.5698, 0.2485, 0.0455, 0.0198, 0.0969, 0.0866, 0.0966, respectively, which are the closest with the music in the original dataset, which indicates that the model proposed in this paper generates better music than the other three models.

Table 1: Different models generate the differences between music and real data

Model	The indicators of pitch			
	PR	PSR	PE	PH
CP Transformer	5.3545	0.1056	0.3422	0.2969
Transformer-XL	10.8948	0.0645	1.0988	0.7485
CEG-Transformer	9.3645	0.0785	1.0636	0.7796
This model	4.5698	0.1348	0.2485	0.0455
Model	Rhythmic indicators			/
	SC	EBR	GS	
CP Transformer	0.0869	0.0966	0.2249	
Transformer-XL	0.2458	0.1048	0.2136	
CEG-Transformer	0.2348	0.1038	0.2148	
This model	0.0198	0.0969	0.2496	
Model	Music structure index			/
	SI_short	SI_mid	SI_lomg	
CP Transformer	0.0698	0.0966	0.1069	
Transformer-XL	0.0769	0.1248	0.1345	
CEG-Transformer	0.0848	0.1349	0.1386	
This model	0.0636	0.0866	0.0966	

(3) Pitch and note density

Pitch is an important perspective to reflect the information of a piece of music, by considering the proportion of the pitch of each note of the sequel track generated by the model, the similarity between the generated music and the original sample in the information dimension of pitch can be considered. 50 sample music sequences were used as input to the model, and three word embedding methods were used to compose the same samples in the input, and after that, a statistical average was made for all the used song categories are statistically averaged. Figure 4 shows the pitch frequency distribution, the higher pitch ratio of the original sample sequences used is maintained at 70 to 80, and the pitch ratios of the three modes of compositions also show this trend, which indicates that overall the three encoding modes are able to restore the original sample repertoire at this level of pitch better. However, considering from the perspective of standard deviation, the model in this paper is closer to the original samples in almost all the pitch values compared to CP Transformer and Transformer-XL, and the difference between the two pitch values between 70 and 80 is 0.01146 and 0.00693, which indicates that the model in this paper is better for modeling the integration of dataset sequences and better facilitation of extracting pitch information from the input samples.

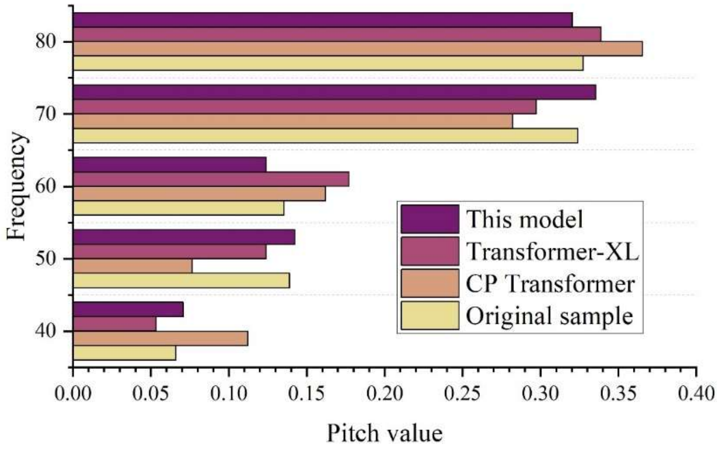

Figure 4: Pitch frequency distribution

Popular music tends to have a relatively stable rhythmic and harmonic melodic repetitive structure both before and after this track, and the overall note density trend over time is relatively stable. It is therefore of great interest to consider the stability of the note density of model compositions over time. In this paper, 50 samples are used to compute note density, defined as the number of notes per window size of 5 seconds in the temporal sense, given a series of identical original samples, composed with models trained on three different word embeddings and their statistical averages calculated. Figure 5 shows the change in note density, the pre-composition period of the model in this paper contains a stable note density, which is stable around 10.5 until 15 time windows. However, as time passes, the overall note density slips over time, which indicates that the melodic stability of the model composition is difficult to be ensured over a longer period of time when the original input samples are used to generate music using the model of this paper. The model compositions using CP Transformer and Transformer-XL have stable note densities on the overall timeline given by the samples, but the CP model has a relatively large change in note densities for each short-term time interval, so in comparison, this proposed paper has the advantage of maintaining stable note densities over a long period of time while having a more small standard deviation, which also reflects the objective level that the model trained in this paper composes more stable rhythms and more melodic layering, which is more in line with the characteristics of a piece of music.

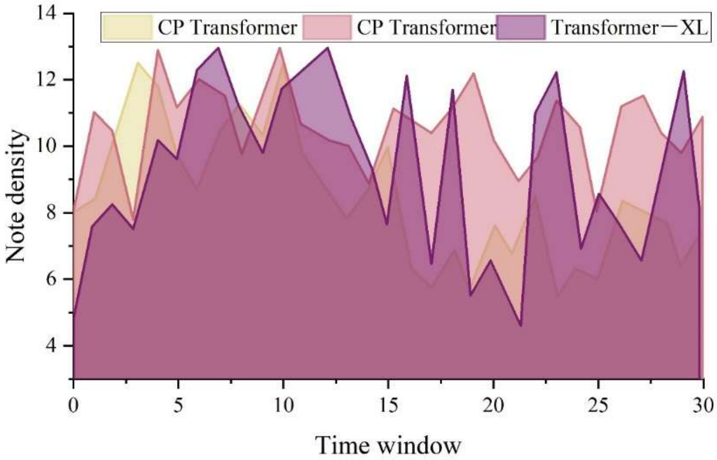

Figure 5: Change in note density

IV. B. 2) Subjective assessment

From the music generated by the four models, namely the model in this paper, the Transformer-XL model, the CEG-Transformer model, and the CP Transformer model, as well as from the original dataset, five pieces of music each, for a total of 20 pieces of music, were selected as evaluation data. To enhance the professionalism of the evaluation experiment results, 50 non-music professionals and 30 music professionals were invited to score and evaluate the music. To analyze the results more intuitively, the scores were weighted. In order to highlight the importance of emotion in the scores, weights were assigned to the five evaluation indexes of harmony, rhythm, pleasantness, fluency and musical structure according to the weight of 2:2:3:2:1, and the weighted scores of each model were calculated. At the same time, the ratings of professionals and non-professionals were assigned in the ratio of 6:4, and the final total score of each model was calculated as shown in Table 2.

According to the weighting calculation method described above, the music generated by the model in this paper has the highest score and is closest to the music in the dataset, with a total score of 4.3485.

Table 2: Subjective music effect evaluation weighted rating results

Model	Professionals	Nonprofessional	General evaluation
Real data	4.5458	4.7489	4.5846
CP Transformer	3.7885	3.9485	3.8588
Transformer-XL	2.6998	2.9485	2.7496
CEG-Transformer	3.5459	3.7496	3.5966
This model	4.2998	4.4488	4.3485

In order to verify the superiority of the models in this paper, the MUT MIDI dataset and the Lakh MIDI dataset are used to train CP Transformer, Transformer-XL, CEG-Transformer, and the models in this paper, respectively, in this section. Finally, given the trained models with the same start notes, music generation is performed, and five musical compositions generated by each of the four models under the two datasets are used as test samples. And the experts were invited to score, and the specific scoring results are shown in Table 3.

In the models proposed in this paper, whether it is the LakhMIDI dataset or the MUT MIDI dataset, the results generated by the models in this paper are higher than the index scores of the CP Transformer and Transformer-XL models on the whole, which indicates the effectiveness of the models in this paper. The mean scores of the proposed model in the two databases are 7.7419 and 8.3089, respectively, and the highest scores in the four indexes of completeness, music form, audibility and smoothness, which indicate that the quality of the improved model has been qualitatively improved. It also visualizes that the MUT MIDI dataset proposed in this paper has certain advantages over other datasets when performing music-related tasks. From the subjective evaluation, the model proposed in this paper does have a great improvement in the overall performance, but from the local performance, the rhythmic and melodic aspects still need to be further improved.

Table 3: Quiz score

Database	Index	CP Transformer	Transformer-XL	CEG-Transformer	This model
Lakh MIDI	Rhythm	5.5478	6.2489	5.2699	7.1985
	Melody	5.0485	5.9486	7.0499	6.2452
	Integrity	6.5458	7.1698	5.6498	8.1685
	Music Form	7.0486	7.4985	8.0496	8.1969
	Audibility	7.0044	6.1248	6.2985	8.4958
	Fluency	7.5966	7.3458	5.7599	8.1465
MUT MIDI	Rhythm	6.2456	7.0499	5.7498	7.3485
	Melody	8.7665	8.7498	7.9485	8.9663
	Integrity	6.8485	7.3969	5.6936	8.0448
	Music Form	7.3485	7.8496	8.2488	8.3485
	Audibility	7.0458	6.6469	6.7485	8.5969
	Fluency	7.6486	7.9798	5.5695	8.5485

V. Conclusion

The AI music generation and manual creation interaction optimization model guided by topological sorting in this study achieves significant performance improvement in multiple dimensions. The objective evaluation results show that the proposed model achieves 0.0969, 0.0866, and 0.0966 in the three key indexes of EBR, SI_mid, and SI_long, respectively, which are all better than the comparison methods, proving the superior performance of the model in terms of the structural integrity of music and rhythmic stability. Through the training and validation of 140,694 lyrics data, the model successfully realizes rhyme-enhanced lyrics generation, which effectively solves the deficiencies of traditional methods in emotional expression and metrical control.

The experiments are tested with 50 sample music sequences, and the results show that the topological network structural characterization method can accurately capture the correlation relationship between musical elements, providing a reliable theoretical basis for chord generation. The application of deep convolutional generative adversarial network combined with unilateral label smoothing and feature matching techniques significantly improves the stability and diversity of music generation. The emotion-guided diffusion model achieves precise emotion control through the cross-attention mechanism, which makes the generated music more infectious in expressing specific emotions. The introduction of the layered attention mechanism further enhances the rhythmic and emotional consistency of lyrics generation, providing a more complete solution for AI music composition. This research lays an important foundation for the development of intelligent music composition technology, and is of great significance in promoting the application of artificial intelligence in the field of artistic creation.

Funding

This research was supported by the Second Batch of 14th Five-Year Plan Undergraduate Teaching Reform Projects in Zhejiang Province in 2024: Exploration and Practice of Digital Teaching System for Music Design and Production Major Empowered by Artificial Intelligence.

References

[1] Civit, M., Civit-Masot, J., Cuadrado, F., & Escalona, M. J. (2022). A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. Expert Systems with Applications, 209, 118190.

[2] Wang, L., Zhao, Z., Liu, H., Pang, J., Qin, Y., & Wu, Q. (2024). A review of intelligent music generation systems. Neural Computing and Applications, 36(12), 6381-6401.

[3] Giuliani, L., De Filippo, A., & Borghesi, A. (2023). Towards Intelligent Music Production: A Sample-based Approach. In CEUR WORKSHOP PROCEEDINGS (Vol. 3519, pp. 50-59). CEUR-WS.

[4] Dash, A., & Agres, K. (2024). Ai-based affective music generation systems: A review of methods and challenges. ACM Computing Surveys, 56(11), 1-34.

[5] 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.

[6] Gupta, S., Marwah, S., & Briskilal, J. (2022). AI Music Generator. Journal of Pharmaceutical Negative Results, 13.

[7] Casini, L., Marfia, G., & Roccetti, M. (2018, September). Some reflections on the potential and limitations of deep learning for automated music generation. In 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC) (pp. 27-31). IEEE.

[8] Atanacković, D. (2024). Artificial Intelligence: Duality in Applications of Generative AI and Assistive AI in Music. INSAM Journal of Contemporary Music, Art and Technology, (12), 12-31.

[9] Mitra, R., & Zualkernan, I. (2025). Music Generation Using Deep Learning and Generative AI: A Systematic Review. IEEE Access.

[10] Ajwani, D., Cosgaya-Lozano, A., & Zeh, N. (2012). A topological sorting algorithm for large graphs. Journal of Experimental Algorithmics (JEA), 17, 3-1.

[11] Zhu, S., Lu, J., & Ho, D. W. (2019). Finite-time stability of probabilistic logical networks: A topological sorting approach. IEEE Transactions on Circuits and Systems II: Express Briefs, 67(4), 695-699.

[12] Gao, T., Li, J., & Ma, H. (2023). A new approach for semi-external topological sorting on big graphs. IEEE Transactions on Knowledge and Data Engineering, 35(12), 12430-12443.

[13] Tran, M. L., Lee, D., & Jung, J. H. (2024). Machine composition of Korean music via topological data analysis and artificial neural network. Journal of Mathematics and Music, 18(1), 20-41.

[14] Zhaopin Su,Guofu Zhang,Zhiyuan Shi,Donghui Hu & Weiming Zhang. (2024). Message-Driven Generative Music Steganography Using MIDI-GAN. IEEE Transactions on Dependable and Secure Computing,21(6),5196-5207.

[15] Hong Huang,Junfeng Man,Luyao Li & Rongke Zeng. (2024). Musical timbre style transfer with diffusion model..PeerJ. Computer science, 10,e2194.

[16] Yu Gao,Shu Ping Wan & Jiu Ying Dong. (2025). A novel similarity-based taste features-extracted emotions-aware music recommendation algorithm. Information Sciences,708,122001-122001.

[17] Tanmoy Debnath & Suvvari Sai Dileep. (2023). An Automatic Lyrics Generation Method Based on Deep Learning. Journal of Research in Science and Engineering,5(8),

[18] Duan Wei,Yu Yi & Oyama Keizo. (2023). Semantic dependency network for lyrics generation from melody. Neural Computing and Applications, 36(8),4059-4069.