Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on AI music generation and melody optimization based on topological sorting algorithm

Qi Liu[1,] ✉

[1] *Department of Art and Technology, School of Music and Dance, Communication University of Zhejiang, Hangzhou, Zhejiang, 310018, China*

ABSTRACT

In this paper, DAG is utilized to represent the dependencies between musical features, and a topological sorting algorithm based on layer order relationships is used as the sampling algorithm for AI music generation models. The feature de-entanglement mechanism of VAE is utilized to learn multiple feature representations, and Transformer-XL is used as the encoder and decoder of the model to design the Control-VAE model that manipulates the latent variable representations to change the music structure. Statistical autocorrelation coefficients, spectral analysis, and diversity auto-assessment metrics data were used to evaluate the model performance in terms of three dimensions: melody, timbre, and diversity. The feasibility of Control-VAE model AI music generation and melody optimization is examined through the evaluation of practical application effects. The results show that the autocorrelation coefficients and frequency amplitudes of the music generated by Control-VAE model are basically consistent with the original music, and reach human-like PPL values, seIf-BLEU values and Zipf coefficients near p=0.95.The music pieces generated by Control-VAE model have a certain degree of musicality, and the melody-optimized music is clear, accurate and novel and interesting.

Keywords: topological sorting algorithm, VAE, Transformer-XL, AI music generation

1. Introduction

With the continuous development of society and the improvement of people's living standards, music as an art is also evolving and growing. The historical evolution of music can be traced back to the Gregorian chant in the Middle Ages, through Baroque music with its mathematical nature, metamorphosed into classical music, Romanticism, and then Impressionism and modern music, and so on for many periods [1]. From the initial monotonous music to the introduction of dominant and polyphonic forms of music, from church music gradually to the masses, from stone knapping to bowed

✉ Corresponding author.
E-mail address: 20140048@cuz.edu.cn (Q. Liu).
Received 25 February 2024; Revised 05 April 2024; Accepted 15 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-388

string instruments to electronic instruments, music has continued to diversify and evolve [2-4]. The production of music requires a wealth of specialized knowledge and relevant musical skills. The process of creating a song includes many aspects such as composing, lyrics writing, arranging, and post-mixing [5]. And outstanding musical works require even more creative inspiration from musicians, which shows that music creation is an extremely time-consuming and laborious process. Professional knowledge of music theory provides the basis for music production, while the skillful use of musical skills determines the quality of the final work [6]. Among them, melody is the soul of music, which is an organic combination of notes in time within a certain pitch range, presenting a musical form rich in beauty and artistry [7]. Melody makes people empathize and emotionally resonate with music, and is often an important part of creation [8].

The process of music creation is both a technical work and requires the creator's deep understanding of music and sensitive grasp of emotions. Therefore, the development of music is not only the progress of technology, but also the result of the creator's unremitting efforts and inspiration, through the computer technology to assist in the creation or realization of the automatic generation of music can greatly reduce the workload of music creators [9]. And as deep learning has achieved good results in the fields of natural language processing, computer vision, etc., it has also been gradually applied to music generation, and a new cross-discipline-Artificial Intelligence (AI) music has been gradually formed [10]. The application of AI music has had a far-reaching impact on music creation, industrial development, and so on. Different generative models can be used as auxiliary tools for creators, providing them with creative inspiration, generating scores, innovating styles and melodies, customizing the generation of music, and exploring new musical forms [11-14]. Through intelligent algorithms, AI systems are able to learn and imitate different styles of music, helping creators to improve efficiency and expand their ideas in creation, and injecting new forces into the music production industry [15]. It can be seen that both from the perspective of theoretical research and practical application, music generation is of great significance.

The process of generating music is usually divided into three stages, creating a score, performing the score, and generating audio. The technical generation of music, on the other hand, usually involves learning existing data, training and then extracting features such as notes and melodies to further generate the target music. For example, literature [16] takes music therapy as a starting point and creates statistical machine learning models for generating new compositions that are similar to a given repertoire structure, style, melody, and chords. Literature [17] created a language model-guided diffusion model MeLoDy in order to achieve high-speed sampling and infinite continuation, and at the same time, the model generates works with better musicality and quality compared to the current state-of-the-art music language models.

In the field of music generation, deep learning is widely used. Literature [18] uses Recurrent Neural Networks (RNN) to learn and train existing melodies or songs to generate new music, but due to the gradient vanishing problem of RNN, it is defective in the long-term consistency of the generated music structure. Whereas Long Short-Term Memory Neural Networks (LSTMs) have advantages in modeling long sequences, literature [19] uses LSTMs to learn input music fragments that have been converted to different tunes, which can be learned to obtain arbitrary notes, which are picked by LSTMs to synthesize satisfactory music. Literature [20] utilizes a convolutional neural network to recognize and classify human facial emotions, followed by a two-layer LSTM to build a music generation model, which together accomplish emotion-focused music generation. However, the defect of prediction accuracy of LSTM in music generation is obvious. Literature [21] utilizes Adaptive Crocodile Optimization Algorithm (ACOA) to optimize LSTM and constructs a new ACOA-LSTM automated music generation model, which not only improves the music prediction accuracy, but also is friendly to the quality and speed of the generated music. In addition, Variational Auto-Encoder (VAE), as a generative model in deep learning, can also decode long sequences, specifically recognizing and learning to pass the probability distribution of a given track in order to generate a new track. Low-fidelity music generation

studied in the literature [22] is the use of the encoder of the VAE model to learn the latent vectors of the music structure, which is computed by the decoder to generate new music samples.

The Transformer model can also be used for generation with two main stages, encoder and decoder, where the encoder inputs the music sequence and converts it into a series of vector representations, and outputs these vectors through a decoder. Literature [23] constructed a multi-track music generation model using a bidirectional encoder representation from a Transformer network, which is the result of optimization on a deep learning model and has significant advantages in pitch prediction.The Transformer model is quite successful in background music generation, literature [24] innovated a controllable music transducer for the tempo of the rhythm between the video and its background music relationship between the video and its background music and music type and user-selected instruments are controlled locally and globally, respectively, as a way to generate background music that conforms to the rhythmic melody of the video. And literature [25] articulates different background music segments in the video through the Transformer model, so that the background music of the video will not appear abrupt and breakpoints. In the form of decoder input, the Transformer model inputs in the form of traditional music sequences, while the Generative Adversarial Network inputs in the form of images. The generative adversarial network transforms music data into images and then performs feature extraction, alternates training with the generator and discriminator, and repeats to obtain the best training results, which does have better results in music generation. For example, literature [26] gives a learning automata-self-attention generative adversarial network model for generating music with real-time emotions, which utilizes the self-attention mechanism to generate music-like structures while eliminating redundant structural fragments with a learning automata iterative algorithm. Literature [27] describes a method for automatic music generation given the music duration, i.e., an adversarial network generated conditionally by inception modeling, which is done jointly with a convolutional neural network considering the music duration distribution and an inception model considering the music quality. Literature [28] introduces deep neural networks, machine learning algorithms, VAE, long and short term memory networks and Transformer model to construct a hybrid framework for music generation which processes in terms of tempo, pitch, and spectrum and subsequently generates high quality music.

Melody is the focus of a musical composition, and there are few studies that consider melody in music generation. Literature [29] constructed a generative pre-trained-2 model for music text using generative pre-trained-2 text generation model and transfer learning, which can generate music melody that possesses a variety of information such as soprano, rhythm, and pause. Literature [30] proposed a statistically supported music generation method, which showed significant advantages in musical rhythm and melodic consistency. Literature [31] proposes an Expectation-Recurrent Neural Network framework for interactive music generation, which can specify a certain meta-constraint and has advantages in realizing individuality requirements such as melody for music composition. Some of the above algorithms try to minimize the search space through simple music representations that are prone to loss of diversity and quality. Since the audio can be converted into a spectral image for input, the music generation model can be realized by learning the generation of spectrograms. And the basic idea of topological ordering algorithm is that by continuously selecting the vertex with incidence 0 and removing that vertex from the graph, the final sequence of vertices obtained is the topological ordering of the graph [32]. This algorithm belongs to the graph algorithm, through the two methods of depth and breadth-first search, it can find out the order that satisfies all the dependencies, and carry out the target ordering of notes, melodies, etc., which is no longer a simple representation of music, and satisfies the diversified generation of music and quality assurance.

In this paper, we describe and sort out the common topological sorting algorithm process, and propose the definition of LB tree based on the layer order relationship. The iterative transformation of LB tree is utilized to complete the output of topological order, and the topological sorting based on layer order relationship is realized. The topological ordering algorithm based on hierarchical

relationship is used as a sampling algorithm to isolate potential variable representations about rhythmic and tonal features relying on the feature de-entanglement mechanism combining the independent encoder and generating adversarial loss function constraints. With the help of Transformer-XL's segment-level looping and relative position encoding mechanism, the effective latent variable information is utilized to generate the final music sequence and realize the design of Control-VAE model. The Vanilla Control-VAE model using the traditional Top-k sampling algorithm is selected as the baseline model, and the melodic and timbral effects of the music are generated using the autocorrelation coefficient and spectral analysis considerations. PPL, self-BLEU and Zipf coefficient metrics are introduced to examine the performance level of the proposed model in terms of music generation diversity. The Control-VAE model is used to generate music and optimize melody respectively to explore its application effect.

2. AI music generation model based on topological sorting algorithm

For a directed acyclic graph (DAG), the goal of topological sorting is to output a sequence of all nodes in V that is guaranteed to satisfy u before v for every edge (u,v) belonging to E, denoted $u<v$. The topological sorting problem is a fundamental problem in graph algorithms, and this algorithm is widely used in scheduling problems. In AI music generation models, the sampling algorithm determines how to generate the actual music content from the model. In this paper, DAG is used to represent the dependency relationship between music features, and the topological ordering algorithm based on layer ordering relationship is used as the sampling algorithm of AI music generation model, which transforms the potential space generated by the model into concrete music elements.

2.1. Topological sorting algorithm

2.1.1. DFS-based topology sorting. Graph theory is the science of graphs, which are composed of a number of vertices and edges connecting these vertices. Traversal of a graph means accessing all the vertices in the graph, and it requires accessing them sequentially along the directed edges connecting the vertices. Currently, the commonly used algorithms to implement the traversal operation of a graph are depth-first search algorithm (DFS) as well as breadth-first search algorithm (BFS).

DFS means starting from a vertex, traversing all the vertices along a path, then backtracking to the previous level and continuing to traverse all the vertices on the next path. This is repeated until the initial vertex is reached. BFS, on the other hand, starts from a vertex and traverses all the neighboring vertices of the initial vertex, then goes to the next level and traverses the neighboring vertices of the neighboring vertices. This is repeated until all vertices are visited. Examples of the basis of the two traversal methods are shown in Figure 1, with DFS on the left and BFS on the right, and the numbers in the figure indicate the order in which the vertices are visited.

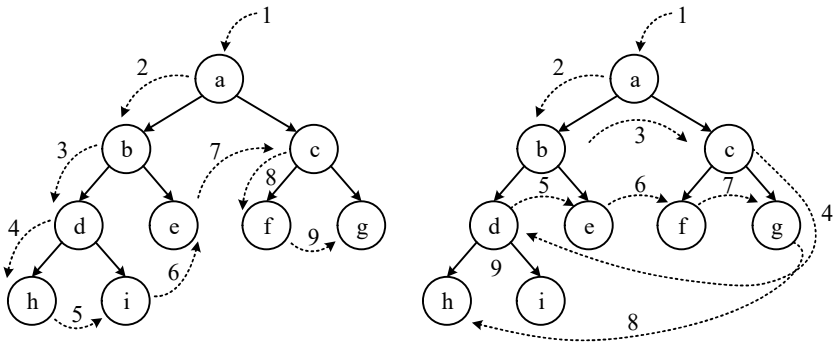


Fig. 1. Shows an example of two different traversal methods

A graph exists with loops if the above two methods traverse to nodes that have already been visited, when the loops correspond to loop-forming flows in percolation fields. And there is a conclusion in graph theory that if a directed graph has loops, its topological ordering does not exist, and the opposite holds.

Based on the above knowledge, for a directed acyclic graph G consisting of n vertex and m directed edges has vertex set V:

$$V=\{v_1,v_2,\cdots,v_{n-1},v_n\},n>1 \tag{1}$$

The bounded set E:

$$E=\{e_1,e_2,\cdots e_{m-1},e_{m=1}\},m\geq 1 \tag{2}$$

A topological ordering is to give a sequence of vertices about Figure G:

$$S_{Topo}=\left(v_{\sigma_1},v_{\sigma_2}\cdots,v_{\sigma_{n-1}},v_{\sigma_n}\right) \tag{3}$$

And for any directed edge in the set of edges E, it is necessary to ensure that the arc head of that directed edge always precedes the arc tail in the sequence S_{Topo}.

One way to obtain the topological ordering is to traverse the graph based on DFS. Each time a path is searched to the last vertex of the path, the vertex is then stored in sequence σ' until all vertices have been traversed and sequence σ' will contain all vertices. At this point a kind of topological sequence σ of the original graph is the inverse of sequence σ'.

2.1.2. Topological ordering based on layer order relationships. The partial order relationship of each node on the DAG is called layer order. Based on the layer order, this paper proposes the definition of LB tree.

Definition 1 (LB Tree) For DAG graph $G=(V,E)$, tree T is an LB tree and satisfies the following conditions:

1) T is a spanning tree of G.

2) The layer order of node v in tree T is defined as $L(v,T)$. And satisfies, if v is the root of T, then $L(v,T)=1$; otherwise, $L(v,T)=L(w,T)+1$, where w is the father node of node v in T.

Let $T=(V,E_T)$ be an LB-tree of DAG graph $G=(V,E)$. Edges in E_T are called tree edges, while edges in $E\setminus E_T$ are called non-tree edges. It is clear that for any tree edge (u,v), $L(u,T)<L(v,T)$ in T. If $L(u,T)<L(v,T)$, then the edge (u,v) is simply said to satisfy the hierarchical ordering relation. In this paper, non-tree edges are categorized into three kinds of edges, namely, upward edges, downward edges and forward edges. Each kind of edge is specifically defined as follows.

An upward edge is a non-tree edge $(u,v)\in E\setminus E_T$ and satisfies $L(u,T)\geq L(v,T)$. A downward edge is a non-tree edge $(u,v)\in E\setminus E_T$ and satisfies $L(u,T)<L(v,T)$. A forward edge is a non-tree edge $(u,v)\in E\setminus E_T$ and satisfies in T that u is an ancestor of v and $L(u,T)<L(v,T)$.

The structure of an LB tree is shown in Fig. 2, which illustrates an LB tree $T=(V,E_T)$ with input DAG graphs $G=(V,E)$ as well as G. where tree edges in E_T are represented by solid lines and non-tree edges are represented by dashed lines. Each type of non-tree edge has been labeled in Fig. 2.

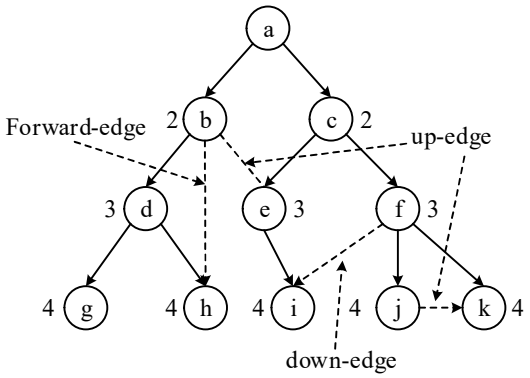


Fig. 2. A LB tree with different kinds of non-tree-edges

Based on maintaining an in-memory structure LB-tree, after iteratively transforming upward edges in the LB-tree into tree edges, the output layer order is a feasible topological order.

2.2. Modeling

2.2.1. Controlled generation of VAE de-entanglement. Variational Auto-Encoder (VAE) is able to learn multiple feature representations by introducing a mechanism for de-entanglement representation and is an effective model for realizing AI music generation due to its latent space manipulability. In this paper, we propose a music generation model called Control-VAE that manipulates latent variable representations to change the structure of music, and the model structure is shown in Fig. 3, which consists of an encoder, a latent space and a decoder. When given a music input sequence $x=\left[Bar_1, Bar_2, \ldots, Bar_T\right]$ containing T consecutive bars, the latent variable representation of each music bar is learned by the encoder $q_\phi(z\,|\,x)$ after a posteriori inference z, and then the input data is reconstructed using the decoder $p_\theta(x\,|\,z)$, which outputs new samples $\tilde{x}$. The model can be optimized by maximizing the Evidence Lower Bound on the Log Likelihood Function $p(x)$ (ELBO) as shown in Eq. (4):

$$\log p(x) \geq \mathrm{E}_{q_\phi(z|x)}\left[\log p_\theta(x\,|\,z)\right] - D_{KL}\left[q_\phi(z\,|\,x)\,\|\,p(z)\right] = \mathrm{L}_{ELBO} \tag{4}$$

where $\mathrm{E}[\cdot]$ is the mathematical expectation, the item indicates the need to maximize the probability value of generating real data, which can be achieved by minimizing the reconstruction loss between input x and output $\tilde{x}$. $D_{KL}[\cdot\|\cdot]$, on the other hand, indicates the need to minimize the KL scatter between the latent distribution $q_\phi(z\,|\,x)$ and the a priori Gaussian distribution $p(z)$.

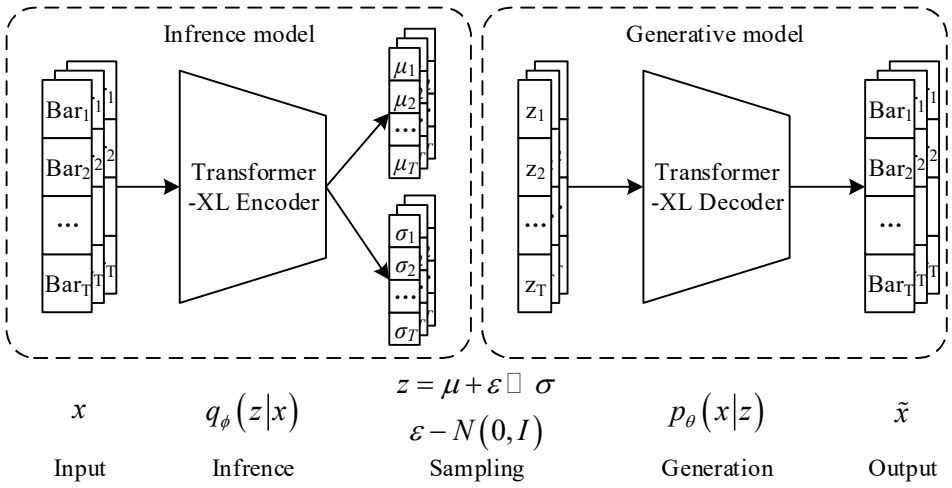


Fig. 3. Macro-structure of Control-VAE model

In this paper, a feature de-entanglement mechanism with independent encoder constraints and generative adversarial loss function constraints is proposed, and the model structure is shown in Fig. 4. Considering the rhythmic feature f_t and tonal feature f_k of the music, two encoder networks E_t and E_k with the same structure are trained to learn the latent variable z_r of rhythmic feature f_r and latent variable z_k of tonal feature f_k from the original sequence of the music, respectively, and to predict the rhythmic feature f_t and the tonal feature f_k using two separate local decoders D_r and D_k. Adversarial training was added to the model based on the idea of GAN. Taking z_r and z_k as inputs to D_k and D_r, respectively, allows D_k and D_r to counteract each other in the training process and make correct feature predictions only for the corresponding latent variable inputs. Thus, the training goal of the process is then to maximize the error between the true and predicted features. Finally, the two latent variables are combined (i.e., $z = Concat[z_t, z_k]$) into the global decoder D_{gbdal} to generate the complete music sequence, which is optimized by maximizing the ELBO.The objective loss function of the Control-VAE model can be defined as:

$$\mathrm{L}_{ELBO} = E_{q_\phi(x_\varphi|x)}\left[\log p_\theta\left(x|z_r, z_k\right)\right] - \mathrm{L}_{KL}^{i} + \mathrm{L}_{Din} \tag{5}$$

where $\mathrm{L}'_{K1} = D_{KL}\left[q_{\phi_i}\left(z_i \mid x\right) \| \, p\left(z_i\right)\right]$, i represent rhythmic features or tonal features. L_{Din} represents the unentanglement loss as shown in equation (6):

$$\begin{aligned} L_{D\mathrm{in}} = {} & E_{q_{\phi_1}(x_i|x)q_{\phi_2}(x_i|x)}\left[\log p_{\varphi_i}\left(f_r|z_r\right)p_{\varphi_1}\left(f_k|z_k\right)\right] \\ & + E_{q_{\phi_1}(x_i|x)q_{\phi_2}(x_i|x)}\left[\log\left(1 - p_{\varphi_i}\left(f_r|z_k\right)\right)\left(1 - p_{\varphi_1}\left(f_k|z_r\right)\right)\right] \end{aligned} \tag{6}$$

The first of these represents the loss when correctly predicting features, and the second represents the adversarial loss, both of which, along with the reconstruction loss in Eq. (5), are implemented using the cross-loss function.

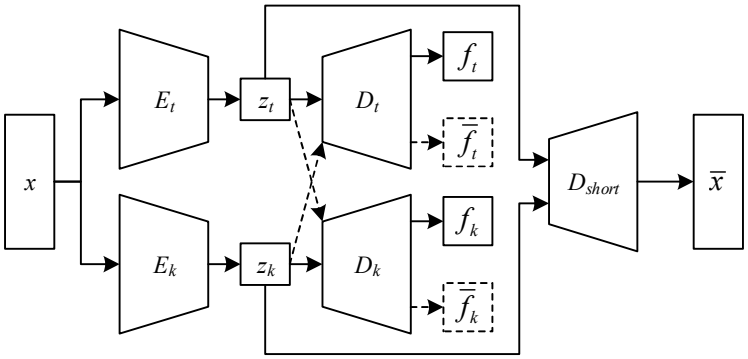


Fig. 4. Musical feature solution structure

Based on the above de-entanglement learning method, the Control-VAE model can effectively isolate latent variable representations about rhythmic features and tonal features from the original music sequence, each of which is independent of each other and does not affect each other, and either latent variable can be manipulated to change the output music.

2.2.2. Encoders and Decoders. To address the problem of limited modeling capability on long sequence data, Control-VAE uses a network structure based on Transformer-XL, which can learn more contextual structures of music bars using its segment-level loop mechanism and relative position encoding.

When Transformer-XL is used as the encoder and decoder of Control-VAE, the music input sequence is divided into multiple segments according to bars, i.e., $x=\left[Bar_1, Bar_2, \ldots, Bar_T\right]$, and each bar contains an equal number of Token. Let $h_{r-1}^{n-1} \in \square^{Lxd}$ be the hidden state of the $\tau-1$ rd bar at layer $n-1$ of self-attention, where L is the sequence length of each bar, d is the dimension of the hidden state, and the hidden state of the τ th bar at layer n can be calculated as in Eq. (7):

$$\begin{aligned} \tilde{h}_{\tau}^{n-1} &= Concat\left[SG\left(h_{r-1}^{n-1}\right)+h_{\tau}^{n-1}\right] \\ Q_r^n, K_r^n, V_r^n &= h_r^{n-1}W_Q^T, \tilde{h}_r^{n-1}W_K^T, \tilde{h}_r^{n-1}W_V^T \\ h_{\tau}^n &= Transformer-Layer\left(Q_r^n, K_r^n, V_r^n\right) \end{aligned} \tag{7}$$

where $SG(\cdot)$ means that the gradient of h_{r-1}^{n-1} is not updated when the next vignette is computed, $Concat(\cdot)$ means that the two hidden state matrices are spliced according to the length of the vignette, and $W_{Q,K,V}$ is a learnable model parameter. Whenever the hidden state h_r^n of the current vignette in layer n is computed, the key-value matrices K_r^n and V_r^n depend on the hidden state h_{r-1}^{n-1} of the previous vignette in layer $n-1$ versus the hidden state of the current vignette in that layer h_r^{n-1}. With this looping mechanism, the contextual information can be extended to all the vignettes and only the hidden state of the previous vignette needs to be cached in memory.

Transformer-XL uses a relative position encoding mechanism to encode the relative position into the attention score when computing the hidden state. Let the query vector of the i st Token in a certain vignette be $q_i=E_{x_i}+P_{x_i}$ and the key vector of the j rd Token be $k_j=E_{x_i}+P_{x_i}$, where E and P are the embedding information and the fully position-encoded information, respectively, then the attention score can be computed according to Equation (8) as:

$$Att_{ij}^{tas}=q_ikj=\left(E_{x_j}^T+P_i^T\right)W_Q^TW_K\left(E_{x_j}+P_j\right) \tag{8}$$

A parameter-free learned sinusoidal encoder matrix $R \in \square^{L \times d}$ is introduced, where each row represents the relative distance encoded between two positions. The relative distances are dynamically injected into the attention scores via Eq. (9):

$$Att_{i,j}^{rel} = \left(E_x^T W_Q^T + u : v\right)\left(W_{K,I} E_{x_i} + W_{k,R} R_{i-j}\right) \tag{9}$$

where $u, v \in \square^{d}$, the trainable parameter vector that replaces $P_i^T W_Q^T$, is multiplied with $W_{K,E}$ and $W_{K,R}$, respectively. $W_{K,E}$ and $W_{K,R}$ are two weight matrices separated from W_K, which are used to map the content information and location information of the key vector, respectively. R_{i-j} is used to describe the relative distance between q_i and k_j. According to the segment-level loop mechanism and relative position encoding, the hidden state of the τ th vignette in layer n can be calculated by Eq. (10):

$$\begin{aligned} Att_{r,j,j}^{n} &= \left(q_{r,i}^{n\,T} + u : v\right)\left(k_{r,j}^{n} + W_{\kappa,R}^{n} R_{i-j}\right) \\ a_r^n &= Soft\max\left(Attt_r^n\right) V_r^n \\ o_r^n &= LayerNorm\left(Linear\left(a_r^n\right) + h_r^{n-1}\right) \\ b_r^n &= FeedForward\left(o_r^n\right) \\ h_r^n &= LayerNorm\left(Linear\left(b_r^n\right) + o_r^n\right) \end{aligned} \tag{10}$$

Based on the definition of the VAE latent distribution $q(z \mid x)$, the mean and variance of representation $q(z \mid x)$ are first learned, and then the latent variables are sampled from this distribution in a process that can be expressed as in Equation (11):

$$\begin{aligned} h_r^{pool} &= Avgpool\left(\left[h_{r,1}^N, h_{r,2}^N, \ldots, h_{r,L}^N\right]\right) \\ \mu &= h^{pool} W_\mu \log\sigma^2 = h^{pool} W_\sigma \\ z &= \mu + \varepsilon \square\ \sigma\varepsilon \sim \mathrm{N}\ (0, I) \end{aligned} \tag{11}$$

where $h_\tau^{pool} \in \square^{d}$, represents the overall hidden state of the τ nd vignette, which is obtained by average pooling of the hidden states at all locations. $\mu, \sigma \in \square^{T \times f}$, is the mean and standard deviation vectors of the latent distributions about all vignettes learned from the hidden states, l is the set latent variable dimension, W_μ, $W_\sigma \in \square^{d \times l}$ are the parameter matrices. $z \in \square^{T \times l}$ is the latent variable about all vignettes sampled according to the reparameterization technique. For decoding generation, latent variable z is transformed to model uniform dimension by linear layer and input to the subsequent self-attention layer by summing with the hidden state output from the previous layer, so as to generate the final music sequence by utilizing the effective latent variable information.

3. Study of the effects of modeled music generation

3.1. Performance evaluation

This section examines the effectiveness of the Control-VAE model based on the topological sorting algorithm for generating AI music by designing controlled experiments. In this paper, the Vanilla Control-VAE model is chosen as the baseline model for the experiments, and the model references the design in Section 2.2, but uses traditional Top-k sampling as the sampling algorithm.

The MAESTRO dataset is introduced to train the model, and a piece of music is transformed to get a digital sequence, in which all the information of the piece of music is contained. In this paper, this digital sequence is called the music signal time sequence, and a music sequence in the MAESTRO

dataset is analyzed and simulated as the research object of this study, and this music sequence is called the original music sequence. For such a music sequence, this paper introduces three statistical features, namely autocorrelation coefficient, spectral analysis, and diversity auto-assessment metrics, in order to express various aspects of music information.

3.1.1. Comparison of autocorrelation coefficients. Firstly, the autocorrelation coefficients are extracted for the music sequences simulated using the two models respectively and compared with the autocorrelation coefficients of the original music, and the autocorrelation coefficients of the whole sequences are visualized as shown in Fig. 5. The horizontal coordinate is the bit difference τ taking values from 0 to 500, and the vertical coordinate is the autocorrelation coefficient. To simplify the description, the simulated music 1 described below refers to the music generated based on the Vanilla Control-VAE model simulation, and the simulated music 2 refers to the music simulated based on the Control-VAE model. By looking at Fig. 5, we can see that the fundamental period of the music corresponding to the three figures is essentially the same, all around 110. The simulation periods are also all approximated to be around 50, and the fluctuation frequencies of the autocorrelation coefficient curves are also basically the same. However, it can also be seen that the fundamental period of the simulated music 1 is not as obvious as that of the other two, which indicates that the melody and rhythm of the simulated music 1 and the other two are different.

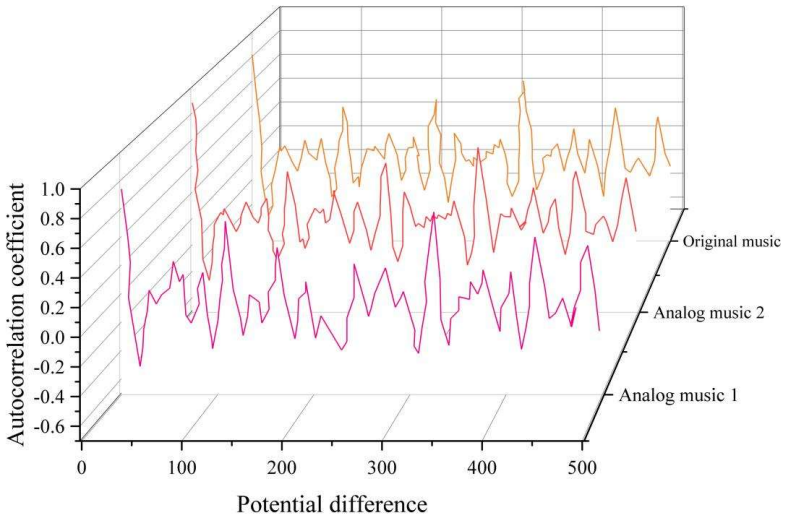
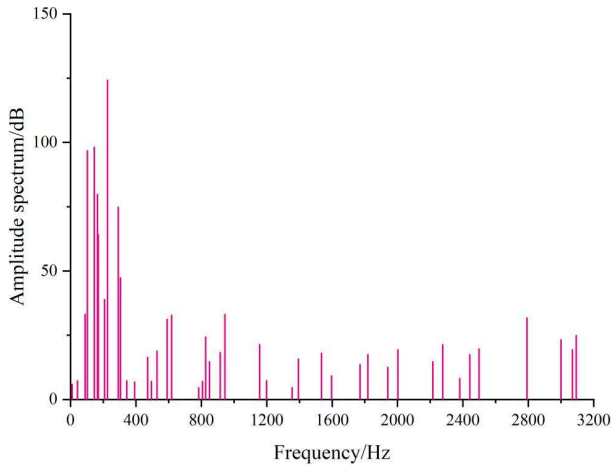
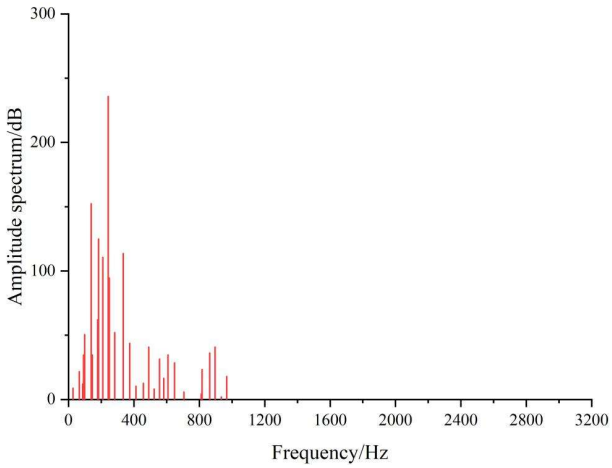


Fig. 5. Comparison of autocorrelation coefficients of the whole sequence

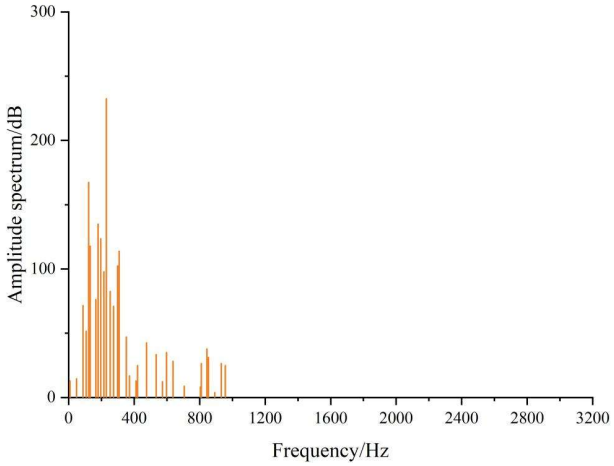
3.1.2. Spectrum analysis. In order to compare the difference between the timbre frequencies of the two simulated music, this paper carries out the Fourier transform and makes the spectrum diagram, and the results of the spectrum analysis are shown in Fig. 6. From the figure, it can be seen that the frequencies of the three pieces of music are mainly concentrated in 0~800, while the spectral distribution of the simulated music 2 and the original music is more similar, and the amplitude value of the frequency is basically zero outside of the 0~1200m, while the amplitude value of the frequency of the simulated music 1 after the 1200Hz always shows a periodic increase, which indicates that the difference between the timbre of the simulated music 1 and the original music is larger than that of the simulated music 2 and the original music. This indicates that the difference between the timbre of analog music 1 and the original music is larger than that between analog music 2 and the original music.



(a) Analog music 1



(b) Analog music 2



(c) Original musie

Fig. 6. Fourier transform comparison of whole sequence

3.1.3. Diversity of music generation. In order to validate the model's enhancement of generating task diversity, the following automatic evaluation metrics are introduced for consideration.

First, quality metrics are examined. Quality and fluency are evaluated by calculating the confusion degree (PPL) of the generated samples. The lower the PPL score, the more fluent it is.

Second, diversity indicators were examined. Diversity was evaluated by calculating the self-BLEU (4, 5) value between generated samples.The lower the Self-BLEU value, the greater the difference between generated samples, the higher the diversity.

Third, the characterization index is examined. The characteristics of the generated samples are evaluated by calculating the Zipf coefficient. The closer the metrics are to human metrics, the better they are, and the closer they are to the behavior of human samples.

The results of the PPL indicators of the generated music are shown in Figure 7. It can be seen that when the p-value of the diversity parameter is small, the PPL of the generated music is much lower than the human level, showing obvious degradation phenomenon, and the generated music will show a lot of repetitive behaviors, with insufficiently novel content and poor diversity. And when the p-value increases, the Vanilla Control-VAE model is unable to reach that level although it can gradually approach the human PPL, which indicates that, for music generation, the upper limit of diversity of the Vanilla Control-VAE model is low, and there is a certain gap from the human level. As a comparison,

the Control-VAE model can not only reach a PPL value similar to that of humans near 0.95, but also closer to the human value than the Vanilla Control-VAE model. This indicates that the proposed algorithm can effectively combat the degradation problem in music generation and achieve PPL metrics similar to humans. This increase in diversity is not achievable by the Vanilla Control-VAE model.

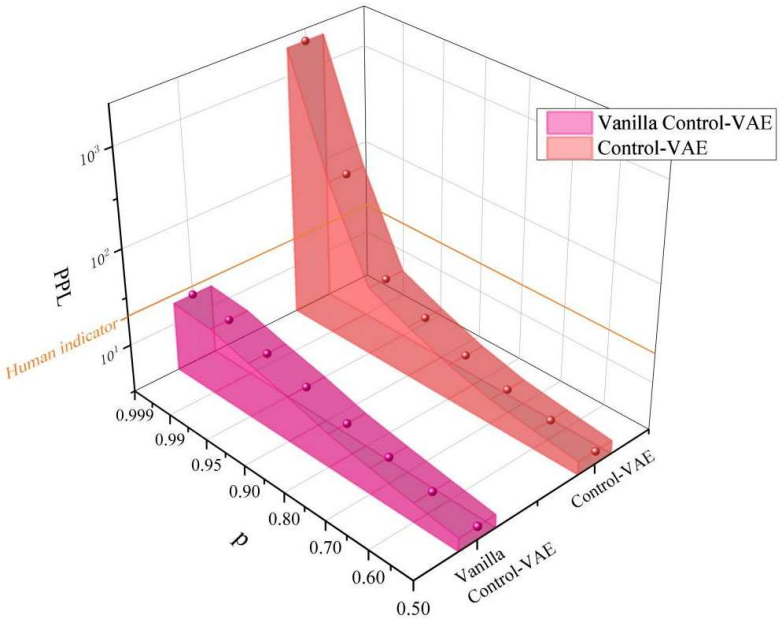
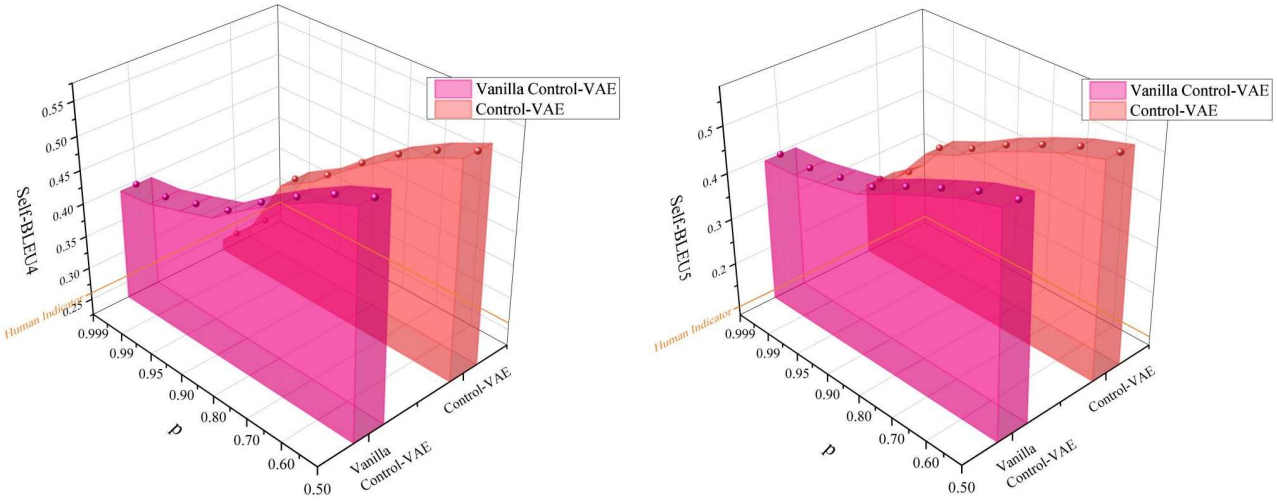


Fig. 7. Experimental results of generating musical PPL indicators

The self-BLEU metrics for the generated music are shown in Figure 8. In this case, the human metrics are the self-BLEU values calculated for all 1735 pieces of music on the MAESTRO dataset (taking the first 1024 tokens for each piece of music), with a self-BLEU4 metric of 0.26 and a seIf-BLEU5 metric of 0.10, and the Vanilla Control-VAE model achieves a seIf- BLEU values are significantly higher and diversity is worse. Note that the Vanilla Control-VAE model has much higher selfBLEU values than human levels. This is highly consistent with the PPL metrics, suggesting that musical diversity is more demanding and difficult to achieve using the Vanilla Control-VAE model. As a comparison, the Control-VAE model can achieve close to human levels on the seIf-BLEU4 metric and outperforms the Vanilla Control-VAE model. For the se1f-BLEU5 metric, the Control-VAE model can still achieve lower scores than the Vanilla Control-VAE model, obtaining higher diversity and thus having an advantage.



(a) Self-BLEU4 (b) Self-BLEU5

Fig. 8. Experimental results of generating music self-BLEU indicators

The results of the Zipf coefficient metrics for the generated music are shown in Figure 9. Where the human Zipf coefficient is 0.95. It is clear that the Vanilla Control-VAE model requires a p relaxation close to 1.0 to reach the human Zipf coefficient, while the Control-VAE model reaches the human Zipf coefficient near 0.95. This result also reveals a higher requirement for diversity in music, with a human Zipf coefficient of 0.95 indicating a flatter distribution of human music and therefore a higher requirement for diversity. And with Control-VAE modeling, the diversity can be significantly improved, and thus a Zipf coefficient closer to the human level can be achieved.

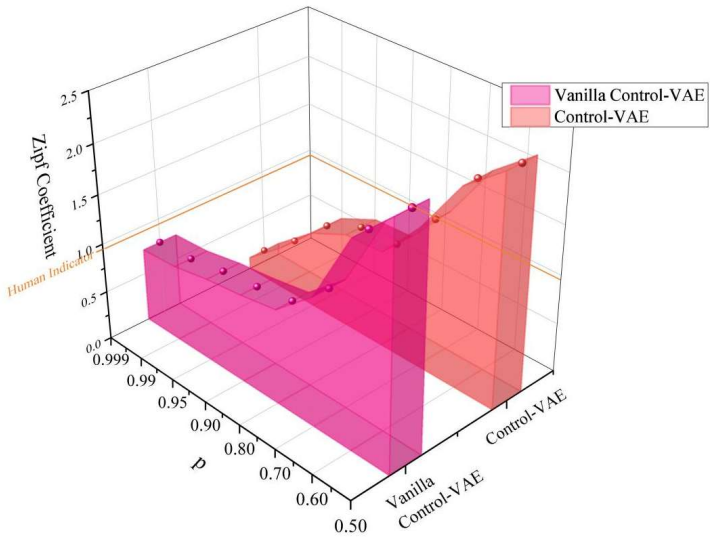
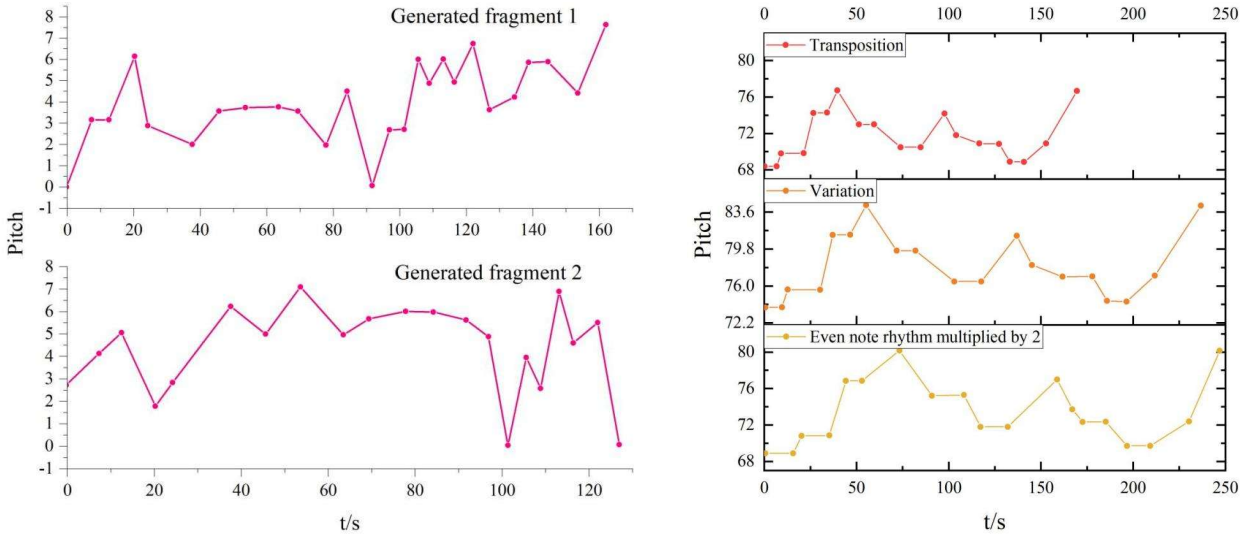


Fig. 9. Experimental results of generating Zipf coefficient index for music

3.2. Analysis of application effects

In order to investigate the practical application of the Control-VAE model, two musical compositions were generated from the drawn melodic directions, and three musical compositions were generated by extracting an existing melody and adjusting the melodic directions. The three pieces are a detuned version of an existing melody, a slower version, and a version of an even-numbered-note tempo multiplied by two, respectively. In this section, these five music clips are selected for the analysis of the results, and these music pieces are presented in a figurative manner. The melodic direction diagrams and corresponding musical results, and the melodic direction diagram modifications and corresponding musical results are shown in Fig. 10(a)(b), respectively.



(a) Melody direction generation (b) Melody direction modification

Fig. 10. Music generation results

Listening to the music generated by the model in the direct melodic direction drawing mode, we can feel that the works generated by the Control-VAE model have certain musicality. The generated notes conform to certain tonal and cyclic relationships, and the generated musical works are more musically structured than other music generation models, and are more in line with the human listening experience. By listening to the music generated by the model in the mode of modification of existing melodic fragments, we can feel that the Control-VAE model has been modified in the details of pitch range, note density, pitch mean, rhythmic complexity, etc. The modified music results are clear, accurate, novel and interesting.

4. Conclusion

This paper proposes an AI music generation model based on topological sorting algorithm, and sets up control experiments to explore its performance and application effects.

The fundamental period of the simulated music 1, simulated music 2 and the original music is basically about 110, and the proposed period is all approximated to 50, and the fluctuation frequency of the autocorrelation coefficient curve is also basically the same, but the fundamental period of the simulated music 2 is not as obvious as that of the other music. The amplitude values of the music generated by the proposed model and the original music are basically zero at frequencies other than 0~1200 m, while the amplitude values of the simulated music 1 are always periodically increased at frequencies after 1200 Hz. The Control-VAE model achieves PPL values and Zipf coefficients similar to those of human beings around p=0.95, while the Vanilla Control-VAE model has a low PPL values, high seIf-BLEU values, poor diversity, and the need to relax p to near 1.0 to reach human Zipf coefficients.

Using the Control-VAE model to generate music, and forensically listening to the musical works generated by the model in the mode of directly drawing the melodic direction map, it can be felt that the works generated by the Control-VAE model have a certain degree of musicality. Listening to the music generated by the model in the mode of modifying the existing melodic fragments, we can feel that the results of the Control-VAE model are clear, accurate, novel and interesting.

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] Nikolsky, A., & Perlovsky, L. (2020). The evolution of music. Frontiers in Psychology, 11, 595517.

[2] Parncutt, R. (2019). Pitch-class prevalence in plainchant, scale-degree consonance, and the origin of the rising leading tone. Journal of New Music Research, 48(5), 434-448.

[3] Nadirova, M., & Aliyeva, H. (2024). EVOLUTION OF MUSICAL INSTRUMENTS: FROM ANCIENT TIMES TO MODERN INNOVATIONS. AGRICULTURAL SCIENCES, 135, 12.

[4] Kamalnathan, S., Mishra, Y., Kumawat, V., & Bangwal, V. (2019). Evolution of different music genres. International Journal of Engineering and Advanced Technology, 9(1), 5138-5143.

[5] Saragih, H. (2019). Co-creation experiences in the music business: a systematic literature review. Journal of Management Development, 38(6), 464-483.

[6] Bates, E., & Bennett, S. (2018). The production of music and sound: A multidisciplinary critique. Critical approaches to the production of music and sound, 1-21.

[7] Scharinger, M. (2022). Melody in speech and music. How language speaks to music: Prosody from a cross-domain perspective, Berlin: De Gruyter.

[8] Zhang, L., Xie, S., Li, Y., Shu, H., & Zhang, Y. (2020). Perception of musical melody and rhythm as influenced by native language experience. The Journal of the Acoustical Society of America, 147(5), EL385-EL390.

[9] Liu, C., Wei, L., & Chen, L. (2021, April). Research on the application of computer technology in music creation. In Journal of Physics: Conference Series (Vol. 1883, No. 1, p. 012031). IOP Publishing.

[10] Briot, J. P., & Pachet, F. (2020). Deep learning for music generation: challenges and directions. Neural Computing and Applications, 32(4), 981-993.

[11] Carnovalini, F., & Rodà, A. (2020). Computational creativity and music generation systems: An introduction to the state of the art. Frontiers in Artificial Intelligence, 3, 14.

[12] 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.

[13] Zhu, H., Liu, Q., Yuan, N. J., Zhang, K., Zhou, G., & Chen, E. (2020). Pop music generation: From melody to multi-style arrangement. ACM Transactions on Knowledge Discovery from Data (TKDD), 14(5), 1-31.

[14] Kang, J., Poria, S., & Herremans, D. (2024). Video2music: Suitable music generation from videos using an affective multimodal transformer model. Expert Systems with Applications, 249, 123640.

[15] Li, S. (2025). The impact of AI-driven music production software on the economics of the music industry. Information Development, 02666669241312170.

[16] Dai, S., Ma, X., Wang, Y., & Dannenberg, R. B. (2022). Personalised popular music generation using imitation and structure. Journal of New Music Research, 51(1), 69-85.

[17] Lam, M. W., Tian, Q., Li, T., Yin, Z., Feng, S., Tu, M., ... & Wang, Y. (2024). Efficient neural music generation. Advances in Neural Information Processing Systems, 36.

[18] Sajad, S., Dharshika, S., & Meleet, M. (2021, September). Music generation for novices using Recurrent Neural Network (RNN). In 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-6). IEEE.

[19] Hewahi, N., AlSaigal, S., & AlJanahi, S. (2019). Generation of music pieces using machine learning: long short-term memory neural networks approach. Arab Journal of Basic and Applied Sciences, 26(1), 397-413.

[20] Madhok, R., Goel, S., & Garg, S. (2018, January). SentiMozart: Music Generation based on Emotions. In ICAART (2) (pp. 501-506).

[21] Kumar, V. B., & Kathiravan, M. (2024). Automatic music generation using a bio-inspired algorithm-based deep learning model. International Journal of System of Systems Engineering, 14(5), 480-503.

[22] Bairwa, A. K., Bhat, S., Sawant, T., & Manoj, R. (2024). MGU-V: A Deep Learning Approach for Lo-Fi Music Generation Using Variational Autoencoders with State-of-the-Art Performance on Combined MIDI Datasets. IEEE Access.

[23] Jiang, R., & Mou, X. (2024). The Analysis of Multi-track Music Generation with Deep Learning Models in Music Production Process. IEEE Access.

[24] Di, S., Jiang, Z., Liu, S., Wang, Z., Zhu, L., He, Z., ... & Yan, S. (2021, October). Video background music generation with controllable music transformer. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 2037-2045).

[25] Hsu, J. L., & Chang, S. J. (2021). Generating music transition by using a transformer-based model. Electronics, 10(18), 2276.

[26] Zheng, L., & Li, C. (2024). Real-Time Emotion-Based Piano Music Generation using Generative Adversarial Network (GAN). IEEE Access.

[27] Li, S., & Sung, Y. (2021). INCO-GAN: variable-length music generation method based on inception model-based conditional GAN. Mathematics, 9(4), 387.

[28] Pricop, T. C., & Iftene, A. (2024). Music Generation with Machine Learning and Deep Neural Networks. Procedia Computer Science, 246, 1855-1864.

[29] Guo, Y., Liu, Y., Zhou, T., Xu, L., & Zhang, Q. (2023). An automatic music generation and evaluation method based on transfer learning. Plos one, 18(5), e0283103.

[30] Goienetxea, I., Mendialdua, I., Rodríguez, I., & Sierra, B. (2019). Statistics-based music generation approach considering both rhythm and melody coherence. IEEE Access, 7, 183365-183382.

[31] Hadjeres, G., & Nielsen, F. (2020). Anticipation-RNN: Enforcing unary constraints in sequence generation, with application to interactive music generation. Neural Computing and Applications, 32(4), 995-1005.

[32] Li, Z., Zhao, H., Deng, W., Li, J., Li, B., & Tian, H. (2017). Study on a Multi-Hierarchy Topological Sort Algorithm for Automatic Calculation. International Journal of Multimedia and Ubiquitous Engineering, 12(1), 191-202.