International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 627-638

https://doi.org/10.70517/ijhsa46346

Research on Music Style Migration and Melody Generation Techniques in the Framework of Composite Computational Methods

Liangzhu Shao[1,*]
[1] Music and Dance College, Xinyang Normal University, Xinyang, Henan, 464000, China
Corresponding authors: (e-mail: shaoliangzhu821011@163.com).

Abstract This paper constructs an efficient computational framework for style migration and melody generation. A decoding architecture from MIDI audio to CQT spectrogram is proposed based on the diffusion model, and a one-dimensional U-Net structure is introduced to optimize the noise prediction process and improve the inference efficiency of the traditional diffusion model. Utilize VAE to map high-dimensional audio data to low-dimensional space to save computational cost. Design the conditional diffusion model based on cross-attention mechanism to realize high-fidelity migration of music styles. Propose a melody generation method based on LabVIEW random numbers to balance creative inspiration and melodic structural integrity. The model of this paper is applied to the practice of music style migration and melody generation to verify the practical value of the model. The results show that the quality of CQT spectrograms generated by the model in this paper is more than 80%, and the style migration rate is more than 90%. Using the piano roll window to visualize the music melody generated after style migration can enhance the melody intuition and the flexibility of segmentation adjustment. In the subjective evaluation, the generated melodies of this paper's model get the best results in two dimensions: coherence and emotional expression, which can effectively realize the music style migration and high-quality melody generation.

Index Terms diffusion model, CQT spectrogram, one-dimensional U-Net structure, cross-attention mechanism, LabVIEW random numbers

I. Introduction

Music style migration, as an important branch in the field of music generation, has attracted increasing attention in the past decade. It is often difficult for traditional methods to capture the multilevel and multimodal characteristics of musical style, because music is an art form that conveys emotions and thoughts through sound waves, which is fundamentally different from visual arts such as images [1]-[3]. Meanwhile, the definition of music style is a vague concept, which covers various aspects such as music genres, arranging styles, performance techniques and harmonic characteristics [4]. Usually, music style is considered to be music genres, such as classical music, pop music, jazz, etc. Sometimes music style is also considered to be the difference of arrangement and harmony, and of course, music style can also be classified according to the timbre of the instruments played [5]-[7]. Since the meaning of musical styles may vary in different contexts, this makes it difficult to accurately model and transform with traditional music style migration methods [8], [9].

Thanks to the rise of deep learning, traditional music style migration faces new opportunities [10]. Modern neural networks are better able to capture complex features in music, and the training of deep learning models gives them the ability to model musical styles and melodies more carefully and accurately [11]-[13]. The powerful processing and learning capabilities of neural networks make music style migration and melody generation more feasible, providing music creators with a wider creative space [14]-[16]. Through deep learning, the temporal relationship, harmonic structure, and timbre changes of music can be better captured, thus realizing more natural and expressive music style migration and melody generation [17], [18]. In addition, neural networks perform well in processing large-scale music data, laying the foundation for more comprehensive music style research [19].

A large number of scholars have carried out research work on applying deep learning models to music style migration. Cífka, O. et al. used an end-to-end supervised learning model for music style migration to propose a one-time style transformation method for notated music, and examined the efficacy of its application to accompaniment style migration in pop and jazz music [20]. Wu, S. L. and Yang, Y. H. combine the transformer model and the variational autoencoder (VAE) to construct a music generation framework suitable for the symbolic domain, which can migrate specific musical attributes to different musical roles and show high effectiveness [21]. Yin, Z. et al.

applied a deep learning approach to the task of automatic music generation by symbolically labeling target styles for music style migration and melody generation, and evaluated the performance differences between different algorithms [22]. Li, S. and Sung, Y. examined an automatic music generation method based on the pre-trained model MRBERT, which overcomes the widespread problem of long range dependency in traditional music generation methods, while enabling diverse representations of musical melodies and rhythms [23]. Yu, W. designed a multi-style music generation model, through the introduction of the style migration mechanism to adjust the weights of different music styles, to realize the seamless transition of chord styles, can be flexible and innovative to generate a variety of styles of music melody [24]. Wang, W. et al. investigated a controlled music generation model based on a deep learning algorithm that generates complete musical compositions from scratch according to the target musical style, while introducing an adaptive strategy to improve the quality of music melody generation [25]. Chen, J. et al. designed audio generation-oriented style migration model based on the image neural style migration algorithm, by converting audio images into speech spectrograms and performing image style migration in order to generate audio melodies with another style [26]. Li, S. et al. showed that the text-guided image style migration method is controllable and of high quality, and its application to a musical style migration task allows for the creation of musical melodies with specific styles by providing accurate descriptions of musical styles [27]. Overall, the music style migration methods studied above exist both in terms of conversion between symbolic scores and changes between audio signals, however, there are some limitations in the results of the application of these methods. In order to further increase the efficiency, quality, and diversity of music generation for music style migration modeling, the processing methods in the framework of composite computational methods are explored.

The complexity of music style migration and melody generation puts forward higher requirements for related computational techniques. In this paper, we construct a technical model related to music style migration and melody generation through multimodal feature fusion and generation model optimization. Combined with the diffusion model to realize the conversion of MIDI audio to CQT spectrogram. The one-dimensional U-Net structure is used to optimize the inference efficiency of the diffusion model and improve the quality of CQT spectrogram generation. Reduce the computational cost of style migration by a VAE encoder with perceptual compression. In the migration model part, the semantic information of the CQT spectrogram of the target instrument is encoded to guide the gradual denoising of the latent space noise to realize the complete migration of the music style. The melody generation method based on LabVIEW is proposed to constrain the random number generation process through the mapping relationship between the gray level histogram of the CQT spectrogram and the scale array to grasp the randomness and controllability of melody generation. The technical model of this paper is practically analyzed to judge the effectiveness of the model.

II. Technical support for musical style migration and melody generation

This chapter systematically constructs a technical framework for music style migration and melody generation from three dimensions: feature extraction, migration modeling, and melody generation based on LabVIEW random numbers.

II. A.Feature extraction

II. A. 1) Diffusion models

The mainstream generative models include GAN, VAE, Flow-based model, and Diffusion Model.Fig.1 shows the flow of the four generative models.The disadvantages of GAN are mainly reflected in the instability of the training process and the problem of pattern collapse, while the Diffusion Model is stable and diversified but has low inference efficiency.VAE compresses the image information but reduces the dimensionality, while the Diffusion Model destroys the data but maintains the original dimensionality. Flow-based model and diffusion model both map data to Gaussian noise, but Flow-based model requires invertible and differentiable functions and is constrained by grid structure. The diffusion model is more scalable because it can learn forward and backward processes.

The diffusion model is a deep generative model whose mechanism of operation covers a forward diffusion phase and a backward diffusion phase. In the forward diffusion stage, the original input data is gradually perturbed by adding Gaussian noise in multiple steps. In the reverse diffusion stage, the core task of the model is to gradually reverse the diffusion process through learning, so as to gradually return to the original input data. Diffusion models have received much attention due to their excellent performance in terms of quality and diversity of generated samples. Diffusion models, as a class of probabilistic generative models, are centered on learning how to reverse a process that gradually reduces the structure of the training data. Therefore, its training process is naturally divided into two stages: the forward diffusion process and the backward denoising process. Together, these two phases constitute the workflow of the diffusion model.

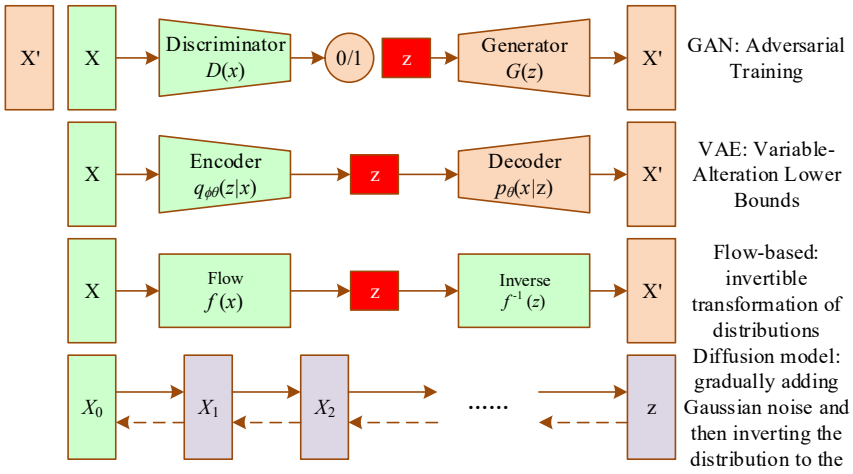

Figure 1: Four types of model generation processes

Forward diffusion consists of multiple steps in which a low level of noise is added to each input image, where the scale of the noise varies at each step. The training data is gradually corrupted until pure Gaussian noise is obtained. Thus each of its moments is only related to the previous one, and this stage can be referred to the Markov chain as shown in equation (1):

$$q\left(x_t \mid x_{t-1}\right)=\mathrm{N}\left(x_t;\sqrt{1-\beta_t}\cdot x_{t-1},\beta_t I\right) \tag{1}$$

where $x_0 \sim p_{data}(x)$ is the training data point, $x_1 \sim x_T$ is the data after adding noise step by step, and $\beta_t \in (0,1)$ is the variance hyper-parameter of Gaussian distribution. Using the reparameterization trick, $x_t=\sqrt{1-\beta_t}x_{t-1}+\sqrt{\beta_t}z_t$ from the first equation, and combining independent random variables $z_1,z_2,\ldots,z_t$ that obey the standard normal distribution into a random variable $\grave{o}$ that obeys the standard normal distribution yields $x_t=\sqrt{\tilde{\alpha}_t}x_0+\sqrt{1-\tilde{\alpha}_t}\grave{o}$. From this, the probability distribution of x_t can be calculated from x_0. As shown in equation (2):

$$q\left(x_t \mid x_0\right)=\mathrm{N}\left(x_t;\sqrt{\tilde{\alpha}_t}\cdot x_0,\left(1-\tilde{\alpha}_t\right)\cdot I\right) \tag{2}$$

where $\alpha_t=1-\beta_t$ and $\tilde{\alpha}_t=\prod_{i=1}^{t}\alpha_i$. When T is large enough, $\tilde{\alpha}_t$ tends to 0 and the distribution of x_T approximates the standard normal distribution $\pi(x_T)\sim \mathrm{N}(0,1)$.

The forward stage is the process of adding noise, while the inverse stage removes the noise. If the distribution $q(x_{t-1}\mid x_t)$ of the inverse process can be obtained, then by inputting a Gaussian noise $x_T \sim \mathrm{N}(0,1)$, a true sample will be generated. Since it is not possible to directly infer $q(x_{t-1}\mid x_t)$, the deep learning model p_θ is used to fit the distribution $q(x_{t-1}\mid x_t)$ as shown in Eqs. (3) and (4):

$$p_\theta\left(x_{0:T}\right)=p\left(x_T\right)\prod_{t=1}^{T}p_\theta\left(x_{t-1}\mid x_t\right) \tag{3}$$

$$p_\theta\left(x_{t-1}\mid x_t\right)=\mathrm{N}\left(x_{t-1};\mu_\theta\left(x_t,t\right),\Sigma_\theta\left(x_t,t\right)\right) \tag{4}$$

With x_0 known, $q(x_{t-1}\mid x_t)$ can be obtained from Bayes' formula as:

$$q\left(x_{t-1}\mid x_t,x_0\right)=\mathrm{N}\left(x_{t-1};\tilde{\mu}_t\left(x_t,x_0\right),\tilde{\beta}_t I\right) \tag{5}$$

where the variance $\tilde{\beta}_t$ and the mean $\tilde{\mu}_t$:

$$\begin{aligned}\tilde{\mu}_t(x_t,x_0)&=\frac{\sqrt{\tilde{\alpha}_{t-1}}\beta_t}{1-\tilde{\alpha}_t}x_0+\frac{\sqrt{\tilde{\alpha}_t}(1-\tilde{\alpha}_{t-1})}{1-\tilde{\alpha}_t}x_t\\&=\frac{1}{\sqrt{\alpha_t}}\left(x_t-\frac{\beta_t}{\sqrt{1-\tilde{\alpha}_t}}\grave{o}\right),\grave{o}\sim\mathrm{N}\left(0,I\right)\end{aligned} \tag{6}$$

$$\tilde{\beta}_t=\frac{1-\tilde{\alpha}_{t-1}}{1-\tilde{\alpha}_t}\beta_t \tag{7}$$

Since β_t is predefined, it is only necessary to estimate $\grave{o}$ using the denoising network $\grave{o}_\theta(x_t,t)$ to obtain the mean value $\mu_\theta(x_t,t)=\tilde{\mu}_t(x_t,x_0)$.

The above model inference process has no input signals, so the generated data is unconstrained and the user has no control over the generated results. Introducing conditions can bias the generated data towards the user's desired outcome. There are many ways to introduce conditions. For example, for an image generation task, a conditional branching-guided diffusion model can be introduced that uses its gradient to guide the image generation to favor a particular semantics, so that the model can generate the appropriate image given the label. Input images or text can also be used to guide image generation.

The optimization objective of the conditional diffusion model is:

$$\min_{\theta} E_{t,x_0,c,\grave{o}}\left[\left\|\grave{o}-\grave{o}_{\theta}(x_t,t,c)\right\|_2^2\right] \tag{8}$$

where the condition c is either discrete finite (e.g., finite class condition) or continuous embedding (e.g., text condition).

The biggest problem with diffusion models is that they are extremely “expensive” in terms of both time and economic costs, and Stable Diffusion, a variant of diffusion models, was developed to solve these problems. Figure 2 shows the flow of the Stable Diffusion model, which consists of three parts: a variational auto-encoder VAE, a UNet, and a text encoder. Unlike learning denoised image data directly in pixel space, instead of going directly to learn and remove noise at the pixel level of the image, StableDiffusion first transforms the image into a low-dimensional latent space representation via the VAE. Then, the process of adding and removing Gaussian noise is performed on this low-dimensional potential space, and finally the processed potential representation is decoded back into the pixel space. In the forward diffusion process, Gaussian noise is iteratively applied to the compressed potential representation. Each denoising step is accomplished by a ResNet-containing UNet architecture, where the latent representation is obtained by denoising from forward diffusion in the opposite direction. Finally, the VAE decoder generates the output image by transforming the representations back into pixel space.

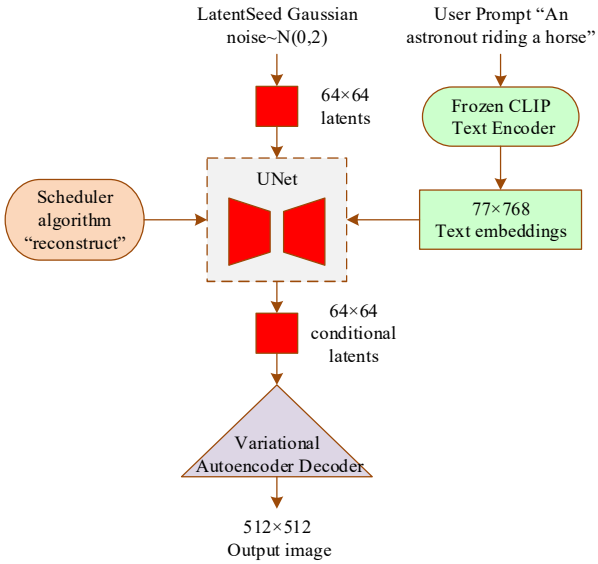

Figure 2: Stable Diffusion model flow

II. A. 2) MIDI to CQT spectra

For the input of symbolic music in MIDI format, at this stage of the paper an encoder-decoder architecture (Midiff) is designed. The first encoder consists of a self-Attend layer and an MLP layer, which is responsible for receiving a series of symbolic note events, which can be note events containing any number of instruments. The second encoder, on the other hand, also consists of a self-Attend layer and an MLP layer, which can optionally use the early part of the Meier spectrogram as contextual information. This information is passed to the decoder, whose task is responsible for generating a CQT spectrogram corresponding to the input note sequence. Inspired by recent successes in the field of imaging such as STABLE diffusion and DALL-E2, here in this paper we investigate the training of diffusion models as decoders.

Diffusion model is a probabilistic generative model that iteratively generates data from noise by reversing the Gaussian diffusion process. The model consists of two main parts, the forward noise addition process and the reverse denoising process. In the forward process, the input signal x is converted to noise $\varepsilon \sim N(0,I)$, which occurs at diffusion time steps $t \in [0,1]$. In this way, the resulting noise figure X_t is given by the following equation:

$$X_t = \alpha_t X + \sigma_t \varepsilon \tag{9}$$

where $\alpha_t \in [0,1]$ and $\alpha_t \in [0,1]$ are parameters in the noise table that are used to mix the original signal and the noise in diffusion time. In this work, the decoder $\grave{o}_\theta$ is trained to predict the additive noise given noisy data. To achieve this, this paper minimizes the objective loss function by minimizing a loss function of the form

$$L_{Midiff} = \underset{X,c,\grave{o},t}{E} w_t \left\| \grave{o}_\theta \left(X_t, c, t \right) - \grave{o} \right\|_1^1 \tag{10}$$

where w_t is a set of loss weights to weight the losses for different diffusion time steps. c is additional condition information for the decoder. These weights w_t, the parameter α_t and the time step t are hyperparameters used to selectively emphasize specific steps in the backward diffusion process.

During the sampling process, the paper follows the reverse diffusion process. Starting with independent Gaussian noise for each frame and frequency bin, noise estimation is used iteratively to gradually reduce the noise content and generate a new CQT spectrogram. Since the process of training using raw audio data and Mel spectrograms in audio context is slow and requires a huge amount of computational memory. Therefore, in order to speed up the diffusion process, the training target and the Mel spectrogram of the audio context need to be scaled before use. In this work, this paper compresses the input in the range $[-1,1]$. During the inference process, this paper scales the model output to the expected CQT spectrogram range.

In this work, this paper uses a one-dimensional U-Net architecture at the decoder stage. Figure 3 shows the one-dimensional U-Net architecture. Where (R) represents the residual 1D convolutional unit, which is used to learn the features efficiently; (M) represents the modulation unit, which is used to change the channel for a given feature at different diffuse noise levels; (I) represents the injection item, which connects the external channel to the current depth in order to transfer the influence; (A) represents the attention item, which is used to share the contextual information; and (C) represents the cross-following item, which is used to learn the text embedding conditions. Compared with 2DU-Net, 1DU-Net has more efficient computational efficiency because it uses 1D convolutional kernels. Here, in this paper, each frequency is considered as a different channel, which allows U-Net to be successfully applied to CQT spectrograms. In this paper, various repetition terms are used at each resolution of U-Net. Among them, the injection term is only applied at a specific depth of the first stage decoder to influence the latent conditions. The attention and cross-attention terms, on the other hand, are only used in the internal blocks of the second stage U-Net for learning audio data and condition information.

During the forward process of the training phase, this paper uses a uniformly sampled noise time step ranging between $[0,1]$ for each training example to obtain a noise example. This noise example is then used as input and Eq. (10) is used to train the model to predict the components of Gaussian noise. In the inference process, the diffusion process is run in reverse and 1000 linear interval steps in $[0,1]$ are used in this paper.

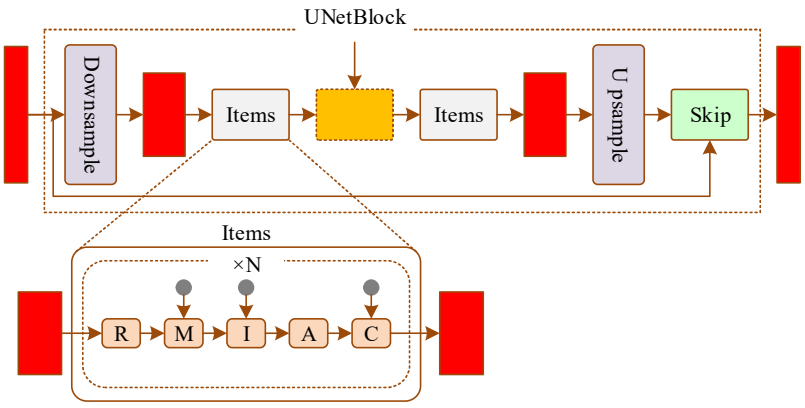

Figure 3: One-dimensional U-Net architecture

II. B.Migration models

In this section, the proposed timbre style migration architecture is described. The architecture takes as input the CQT spectrogram corresponding to the source instrument and generates a target CQT spectrogram corresponding to it. This target CQT spectrogram is obtained by playing the same music using the target instrument. The tone migration model consists of three main components. The first is the variational autoencoder (VAE) model in pixel space, whose task is to sense the information of the input and output CQT spectrograms for efficient compression and reduction. The second is a diffusion model in potential space, whose main function is to realize timbre migration.

The model introduces a cross-attention mechanism that is able to map the information from the conditional mechanism into the denoising UNet to accomplish domain transformation. Finally, the conditional mechanism is used to learn the information from the CQT spectrogram of the target instrument and pass it into the potential space to provide key information for the whole timbre transfer process.

II. B. 1) Perceptual compression

In order to improve the efficiency of training the Diffusion model for generating high quality CQT spectrograms, perceptual compression is introduced in this study. This makes the Diffusion model computationally more efficient since the sampling is done in a low dimensional space.

Formally, in this paper, a convolutional variational autoencoder (VAE) is used to encode the CQT spectrogram $X \in \square^{T \times F}$ into the potential space Z, where $Z \in \square^{C \times \frac{T}{r} \times \frac{F}{r}}$, T and F are the time and frequency dimension sizes, C is the number of channels, and r is the compression level of the potential space, respectively. In order to have high computational efficiency and sample quality, C and r are set to 8 and 4, respectively, in this paper.Both the encoder E and the decoder D consist of stacked convolutional modules, while each block consists of convolutional layers and residual connections. In the generation phase, the decoder is used to reconstruct Z into a CQT spectrogram $\tilde{X} \in \square^{T \times F}$ for a given potential representation.

In this paper, three loss functions are constructed to train the VAE, namely, the CQT spectrogram reconstruction loss L_{rec}, the adversarial loss L_{adv}, and the Gaussian constraint loss KL_{Gau}. The CQT spectrogram reconstruction loss is used to compute the average error between the input sample X and the reconstructed CQT spectrogram. The adversarial loss is used to improve the reconstruction quality through the PatchGAN discriminator. Gaussian constraints, on the other hand, are applied to the latent space of the VAE to encourage the VAE to learn a continuous, structured latent space instead of an unorganized one to better capture the underlying structure of the data. In summary, the overall training goal of the VAE can be described as:

$$\mathrm{L}_{VAE} = \mathrm{L}_{rec}\left(x, D\left(\varepsilon\left(x\right)\right)\right) + \lambda \mathrm{L}_{adv}\left(\psi D\left(\varepsilon\left(x\right)\right)\right) + \gamma KL_{Gau}\left(\mu;\sigma^2\right) \tag{11}$$

where L_{rec} is the reconstruction loss, L_{adv} is the confrontation loss, KL_{Gau} is the Gaussian constraint loss, ψ is the discriminator used in the confrontation, and μ and σ denote the mean and the variance of the potential space of the VAE.

II. B. 2) Conditional mechanisms

The conditional mechanism consists mainly of the encoder τ_θ, which is constructed similarly to the encoder in perceptual compression. For the input CQT spectrogram y of the target instrument, it is encoded by the encoder τ_θ and projected to an intermediate representation $\tau_\theta(y) \in \square^{M \times d_\tau}$. This intermediate representation is then mapped into the intermediate layer of U-Net through the cross-attention layer to enable the generation of a CQT spectrogram containing the timbre of the target instrument according to condition y. The cross-attention mechanism is implemented as Eq:

$$Attention\left(Q,K,V\right) = soft\max\left(\frac{QK^T}{\sqrt{d}}\right)\cdot V \tag{12}$$

where $Q = W_Q^{(i)} \cdot \varphi_i(z_t)$, $K = W_K^{(i)} \cdot \tau_\theta(y)$, and $V = W_V^{(i)} \cdot \tau_\theta(y)$. $\varphi_i(z_t) \in \square^{N \times d_\delta^i}$ denotes the U-Net intermediate representation that realizes ε_θ. $W_V^{(i)} \in \square^{d \times d_\delta^i}$, $W_K^{(i)} \in \square^{d \times d_\tau}$ and $W_Q^{(i)} \in \square^{d \times d_{tau}}$ are projection matrices that map intermediate representations from $\tau_\theta(y)$ into the target domain, allowing for the migration of tones. The objective function can be rewritten as:

$$\mathrm{L}_{LDM}(\theta) = \mathrm{E}_{\varepsilon(x),y,\delta\sim \mathrm{N}(0,1)}\left\|\delta - \delta_\theta\left(z_t,t,\tau_\theta\left(y\right)\right)\right\|_2^2 \tag{13}$$

where τ_θ and δ_θ can be co-optimized by an objective function.

II. C.Principle study of LabVIEW-based random number generation melody

A melody consists of a series of tones with pitch, duration of continuation, timbre, and volume, arranged in a sequence on a musical score. A randomized melody or piece of music is formed by numbering selected notes or musical segments, selecting numbers based on the number of points that appear on a throw of a die, and throwing the die multiple times to combine the notes or musical segments in the order in which the dice points appear. Programmed in LabVIEW, a random number is used to index an array into which selected pitches have been entered, and the indexed frequency values form a sequence of pitches that can be used for automatic composition. Setting the vocal duration of the piece, the frequencies in the array are selected in LabVIEW by using random numbers, and a melody is obtained by inputting it into Beep.vi.

The study of the relationship between CQT spectrogram and melody reveals that melody is the main component of music, and the melody line describes and analyzes the music melody in the form of a curve, and expresses the feelings and constitutes the artistic image through the movement of the melody line. The creation of melody relies on inspiration, which cannot be obtained by pure reason, so composing requires a certain degree of randomness. However, the algorithm of dice music generating melody can not complete the control of the direction of the generated melody, and can not predict the direction of the generated melody, it is necessary to introduce material to limit the random number, and to limit the development of the pitch change by the change of the height of the line. According to the visual properties of the human eye, the grayscale histogram carries the characteristic of linear variation in two-dimensional coordinates, which can be associated with the melodic line. In pentatonic scores, a series of notes representing a melody connected in sequence will form a melodic line.

The grayscale histogram of a CQT spectrogram is an important parameter reflecting the characteristics of the image. The grayscale histogram consists of a series of data, which also has a tendency to go up and down, and can be transformed into a musical sequence. For different CQT spectrograms, the grayscale histogram presents a randomly varying form, so the grayscale histogram in the CQT spectrogram can be used instead of random numbers to control the generation of the melody.

The gray level histogram of the image constitutes an array denoted as $Array_1$, there are 256 elements in $Array_1$ and the process of playing 256 values is lengthy, so it is necessary to select values for control. A specific gray level is selected to control the number of random numbers with the formula:

$$T = \left[256 / c\right] \tag{14}$$

where: T is the number of taken values; c is the interval of selected data.

The gray scale histogram of CQT spectrogram is obtained by counting the pixels of each gray level, the size of the values of the gray scale histogram of different images varies, and some of the values are much larger than the maximum index of the array, in order to simultaneously make the ups and downs of the size of the random values close to the ups and downs of the trend of the gray scale histogram, and to control the output of a new array of $Array_2$, Eq:

$$A_i = \sum_{n=i\cdot c}^{(i+1)\cdot c-1} L_n / p \tag{15}$$

where: L_n is the number of pixels with an image gray level of n; p is the total number of pixels of the image being processed; A_i is the i th element in the array $Array_2$, $i = 0 \sim T-1$.

In order to reduce the uncertainty of the generated melody, the frequency values of 8 tones are selected to form the array. The selected frequency values of the C major scale are 261.64Hz, 293.67Hz, 329.62Hz, 349.24Hz, 393Hz, 441Hz, 493.89Hz, and 523.26Hz, and the index values are arranged in order from 0 to 7 to create the C major scale array. In order to improve the harmony of the generated melody, an array is created based on the principle of triad composition. The triad composition tone frequency values of 261.64Hz, 329.62Hz, 393Hz, 523.26Hz, 784.88Hz, 1045.1Hz, 1306.6Hz, and 1568.8Hz are selected, and the index values are arranged in order from 0 to 7 to establish the triad array.

A melody can be obtained by indexing and playing the C major scale array or triad array through the elements in the array $Array_2$, and the frequency generated by the index value generates the array $Array_3$. The formula is:

$$B_i = \left(S-1\right) A_i / A_{\max} \tag{16}$$

where: S is the size of the array formed by the notes; $A_{\max}$ is the maximum value in the array $Array_2$; B_i is the i th element in the array $Array_3$.

III. Analysis of musical style migration and melody generation practices

This chapter applies the model constructed in the previous section to actual music style migration and melody generation practices, and analyzes the advantages of its application through experiments.

III. A. Time-frequency analysis of music

III. A. 1) Audio Acquisition

In order to have obvious timbre features, the music was selected as much as possible with a single instrument as the main instrument. Due to too much data in the training set, this paper only uses Beethoven's String Trio No. 1 in E-flat major and Telemann's Flute Fantasia flute version as examples to introduce the whole audio data analysis process.

The core of the study in this section is to convert MIDI audio into CQT spectrograms, so when selecting data for the training set, the audio duration should preferably be of the same size and should not be too long. Since there are too few datasets online that meet the requirements and not enough samples, the first step in this experiment is

to segment the audio. Take Beethoven's String Trio No. 1 in E-flat major as an example, each audio clip is 7 seconds long. After obtaining enough small segments of audio, the time and frequency domains are analyzed. Again, two small clips are selected as examples. Figure 4 is the first 7-second clip based on Beethoven's String Trio No. 1 in E-flat major (hereafter referred to as beethoven.wav). Figure 5 is a time-domain plot of the first 7-second fragment of the Fantasia for Flute (hereafter Telemann). Its horizontal coordinate is time and its vertical coordinate is amplitude. Observing the time-frequency plots in Figs. 4 and 5, it can be found that the time-domain plots of the different music clips have clear amplitudes with 7 seconds as the segmentation duration, which indicates that this duration is more appropriate to choose.

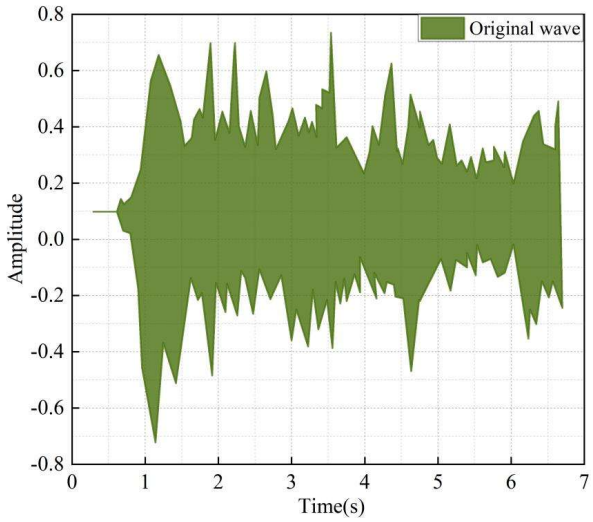

Figure 4: beethoven.wav time domain diagram

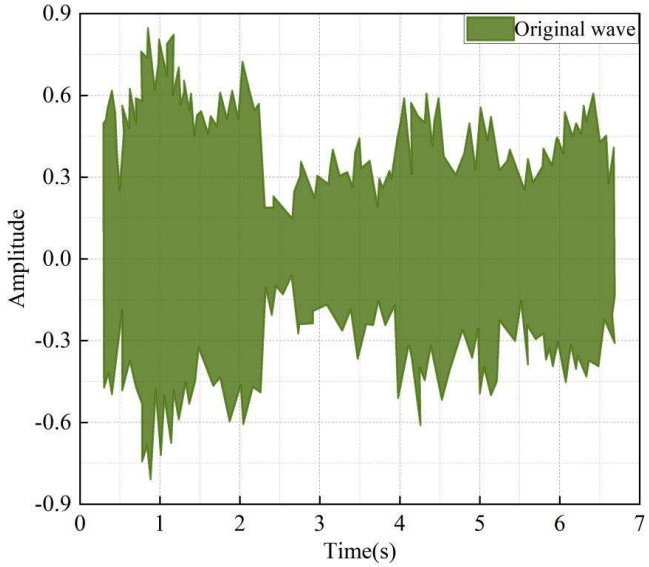

Figure 5: Telemann domain diagram

III. A. 2) Time-frequency processing

The audio in the training set is segmented with a time duration of 7 seconds and the corresponding time-frequency maps of the music segments are obtained. After that, the time-frequency map is processed by adding Kaiser window. Take 8 of the audio samples as an example, set the parameter α=0.6, and show the time-frequency map and frequency domain map obtained after the audio waveforms are processed with Kaiser window. Figure 6 is the time domain of Kaiser window processing. Figure 7 is the frequency domain of Kaiser window processing. From the time-frequency plots in Fig. 6 and Fig. 7, it can be seen that the time and frequency domain ranges of the eight selected audio samples are clearer and clearer after the Kaiser window processing, which lays a good foundation for the subsequent CQT spectrogram conversion, as well as music style migration and melody generation.

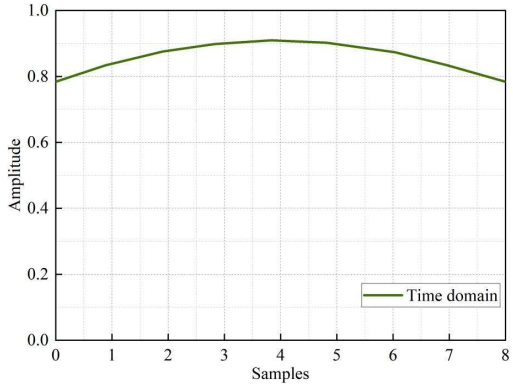

Figure 6: Kaiser window processing in the time domain

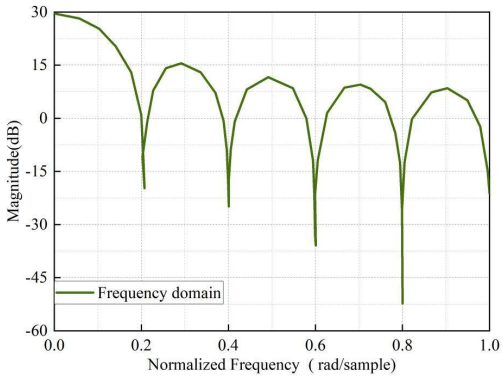

Figure 7: Kaiser window handles the frequency domain

III. B. CQT Spectral Style Migration Quality Analysis

The style migration quality of the target CQT spectrogram can verify the validity of this paper's model, and also determines the final quality of music style migration and melody generation. Therefore, this section analyzes the style migration quality of CQT spectrograms of this paper's model. The CQT spectrograms are generated using this paper's model for the time-frequency diagrams of the processed audio training set, and the audio training set is categorized into six classes according to music styles. The CQT spectrogram generation quality qualification rate (AQR) and style migration rate (TR) of different music genre styles are judged with little difference in audio quality.

Table 1 shows the CQT spectrogram generation quality qualification rate and style migration rate of the model in this paper. Analyzing Table 1, it is found that in the process of mutual migration of styles of six types of music genres, the CQT spectrogram generation quality qualification rate of this paper's model is more than 80% and the style migration rate is more than 90%, both in the forward diffusion stage and the reverse diffusion stage. It shows that the use of the model in this paper for the generation of target CQT spectrograms and the migration of different music styles can obtain better results.The CQT spectrograms can be further applied to the melody generation method based on LabVIEW to obtain high quality music melodies after style migration.

Table 1: Quality pass rate and style mobility of CQT spectrum generation

Origin domain → Object domain	Forward diffusion stage		Reverse diffusion stage	
	AQR (%)	TR (%)	AQR (%)	TR (%)
Pop→Classical	86.25±5.67	93.61	83.16±4.60	92.35
Pop→Blues	87.13±4.85	96.75	86.90±3.31	94.76
Blue→Country	82.36±5.22	93.77	81.44±8.06	91.52
Folk→Jazz	88.68±5.51	92.43	84.09±8.21	90.08
Jazz→Classical	82.77±6.96	91.70	85.48±8.38	91.15
Pop→Folk	84.24±4.67	94.41	81.77±6.13	92.41

III. C. Melodic piano roll window visualization

After combining high-quality CQT spectrograms to realize the style migration of music, LabVIEW random number method is further utilized to generate the corresponding melodies. The generated audio melodies are presented in the form of a visualization of a piano roll window.

Piano roll-up window is a method of mapping a sequence of notes to a two-dimensional spatial music representation, where rows represent time, columns represent pitch, and each element indicates whether the pitch is triggered at that moment, and such a representation intuitively reflects the time and pitch dimensions of the musical information in the melody. There are 2 reasons for using the piano roll window: the segmentation is regarded as a 2D image, which allows the model to extract and learn the spatio-temporal characteristics of the melodic information; it is convenient to carry out data enhancement operations such as flipping, random cropping, etc., to optimize the stochastic and structural logic of generating the melody. In the piano roll window segmentation, in view of the deficiencies in the existing segmentation methods, this paper uses a combination of fixed and variable length segmentation methods. Specifically, the piano roll-up window is segmented into large segments according to the time length of the CQT spectrogram, and then the large segments are segmented into fixed-length segments using the fixed-length segmentation method. Using this segmentation method, avoiding the short CQT spectrogram duration in the fixed-length segmentation affects the integrity of the melodic structure, and at the same time, the melody is segmented into fixed-length notes that are easy to train, which is convenient for judging the effect of melody generation.

Fig. 8 shows the piano roll-up window of the generated melody after the Pop→Classical audio in the training set realizes the music style migration. In Fig. 8, it can be clearly seen that the melodic pitches of the generated melodies after style migration are between 65-80 in the time period of 0-250s. Through the piano roll-up window, the researcher can intuitively and quickly understand the situation of the generated melody and make adjustments accordingly so that the final music melody is more in line with the migrated music style.

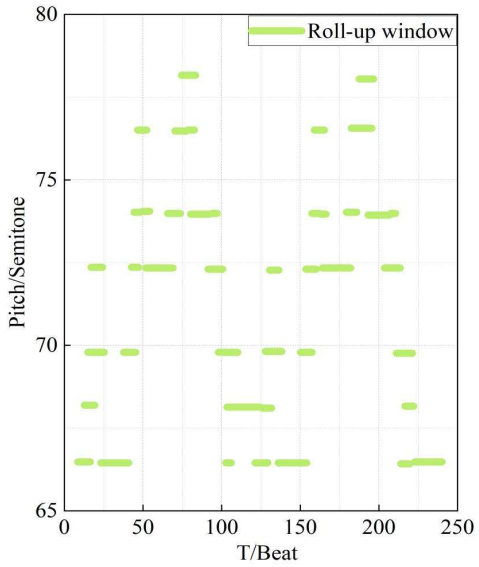

Figure 8: A piano curtain that generates melody

III. D. Subjective evaluation and analysis of the model

In the field of computerized music style migration and melody generation, objective metrics can only reflect the similarity of style migration between the generated music and the target music, but cannot adequately evaluate the melodic beauty of the generated music, so it is necessary to use human auditory evaluation to subjectively evaluate the model.

This paper adopts the user survey, which is mainly used in music research, as the subjective evaluation method of the model. In order to make the evaluation of the experimental results more fair, therefore, diversified groups of evaluators, clear and uniform evaluation indexes, and the combination of objective indexes are used as several means to improve the fairness of the evaluation results. Specifically, 25 people selected from people with different experiences were used as evaluators of musical melodies. The evaluation indexes of musical melody were divided into 2 aspects: emotional expression and coherence. Each aspect was categorized as excellent, moderate, and poor.

First, 16 music pieces (4 pieces per model) were generated using 4 different algorithms: the model of this paper, Midi Nte model, LSTM-RNN model and CNN-GAN model. These musical compositions were randomly numbered to form 16 serial numbers for testing and evaluation. Second, for each model-generated piece, 25 evaluators randomly selected one piece of music for evaluation and marked it on an evaluation form that included the song number, evaluator number, emotional expression of the evaluation, and coherence.

Figure 9 shows the subjective evaluation of the generated music in terms of coherence and emotional expression. Analyzing the subjective evaluations of the evaluators in Fig. 9, the music melodies generated by this paper's model received 15 excellent evaluations, 8 moderate evaluations, and only 2 poor evaluations in terms of coherence. Compared with the comparison model, the musical melodies generated by this paper's model are more natural and smooth between notes, and sound close to a complete piece. In terms of emotional expression, the music melody generated by this paper's model receives 14 excellent ratings, 9 medium ratings, and 2 poor ratings. Compared with the comparison model, this paper's model can more accurately express the emotion, atmosphere and mood contained in the music. From the above analysis, it can be seen that the model of this paper has obvious advantages over the other three models in terms of coherence and emotional expression, and it can generate music melodies that are closer to the style of the original song, and the quality of the music melodies is higher.

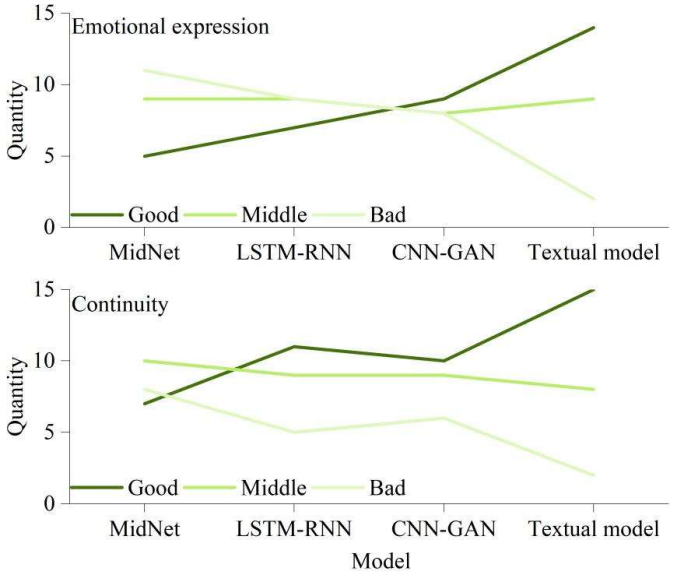

Figure 9: Subjective evaluation of coherence and expression of emotion

IV. Conclusion

This paper constructs a music style migration and melody generation model based on the composite computing method to realize high-quality style migration and melody generation. Using the model in this paper to analyze the audio, a clear time-frequency diagram can be obtained. Further realizing the music style migration, it is found that among the 6 types of music style migration, the CQT spectrogram generation quality qualification rate of this paper's model is greater than 80%, and the style migration rate is greater than 90%, which has an excellent style migration effect. The piano roll window visualization of the generated music melodies ensures the melody generation effect and operation convenience. In 2 aspects of continuity and emotional expression, 25 evaluators gave high evaluation to the generated melodies of this paper's model, indicating that the generated melodies of this paper's model have melodic beauty and meet the aesthetic level of the public. In the future, we can continue to optimize the music style migration computing technology to improve the migration efficiency and quality, and generate more excellent melodies to promote the intelligent development of music generation.

References

[1] Apolo, M. J., & Mendoza, M. (2023, July). Bimodal Style Transference from Musical Composition to Image Using Deep Generative Models. In International Conference on Human-Computer Interaction (pp. 229-240). Cham: Springer Nature Switzerland.

[2] Luo, M., Zhang, Y., Xu, P., Wang, T., Bo, Y., Jin, X., & Dong, W. (2024). Dance Montage through Style Transfer and Music Generation. In SIGGRAPH Asia 2024 Art Papers (pp. 1-5).

[3] Xing, B., Dou, J., Huang, Q., & Si, H. (2021). Stylized image generation based on music-image synesthesia emotional style transfer using CNN network. KSII Transactions on Internet and Information Systems (TIIS), 15(4), 1464-1485.

[4] Neuman, Y., Perlovsky, L., Cohen, Y., & Livshits, D. (2016). The personality of music genres. Psychology of Music, 44(5), 1044-1057.

[5] Cífka, O., Ozerov, A., Şimşekli, U., & Richard, G. (2021, June). Self-supervised vq-vae for one-shot music style transfer. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 96-100). IEEE.
[6] Huang, R., Ren, Y., Liu, J., Cui, C., & Zhao, Z. (2022). Generspeech: Towards style transfer for generalizable out-of-domain text-to-speech. Advances in Neural Information Processing Systems, 35, 10970-10983.
[7] Kwon, H., Lee, K., Ryu, J., & Lee, J. (2023). Audio adversarial example detection using the audio style transfer learning method. IEEE Access.
[8] Aggarwal, S., Uttam, S., Garg, S., Garg, S., Jain, K., & Aggarwal, S. (2025). Advancements in End-to-End Audio Style Transformation: A Differentiable Approach for Voice Conversion and Musical Style Transfer. AI, 6(1), 16.
[9] Grinstein, E., Duong, N. Q., Ozerov, A., & Pérez, P. (2018, April). Audio style transfer. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 586-590). IEEE.
[10] Mitra, R., & Zualkernan, I. (2025). Music Generation Using Deep Learning and Generative AI: A Systematic Review. IEEE Access.
[11] Wang, Y., Stanton, D., Zhang, Y., Ryan, R. S., Battenberg, E., Shor, J., ... & Saurous, R. A. (2018, July). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In International conference on machine learning (pp. 5180-5189). PMLR.
[12] Sturm, B. L., & Ben-Tal, O. (2017). Taking the models back to music practice: Evaluating generative transcription models built using deep learning. Journal of Creative Music Systems, 2, 32-60.
[13] Chen, K., Zhang, W., Dubnov, S., Xia, G., & Li, W. (2019, January). The effect of explicit structure encoding of deep neural networks for symbolic music generation. In 2019 International workshop on multilayer music representation and processing (MMRP) (pp. 77-84). IEEE.
[14] Zhang, Y. (2025). An IoT-enhanced automatic music composition system integrating audio-visual learning with transformer and SketchVAE. Alexandria Engineering Journal, 113, 378-390.
[15] Zhang, K. (2021). Music style classification algorithm based on music feature extraction and deep neural network. Wireless Communications and Mobile Computing, 2021(1), 9298654.
[16] Hu, Z., Liu, Y., Chen, G., Ma, X., Zhong, S., & Luo, Q. (2024, March). Responding to the Call: Exploring Automatic Music Composition Using a Knowledge-Enhanced Model. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 1, pp. 521-529).
[17] Ning, Q., & Shi, J. (2022). Artificial neural network for folk music style classification. Mobile Information Systems, 2022(1), 9203420.
[18] Lu, C. Y., Xue, M. X., Chang, C. C., Lee, C. R., & Su, L. (2019, July). Play as you like: Timbre-enhanced multi-modal music style transfer. In Proceedings of the aaai conference on artificial intelligence (Vol. 33, No. 01, pp. 1061-1068).
[19] Nowak, K., & Zieliński, T. (2025). Graph Neural Network for Music Style Classification. Frontiers in Artificial Intelligence Research, 2(1), 26-34.
[20] Cífka, O., Şimşekli, U., & Richard, G. (2020). Groove2groove: One-shot music style transfer with supervision from synthetic data. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 2638-2650.
[21] Wu, S. L., & Yang, Y. H. (2023). MuseMorphose: Full-song and fine-grained piano music style transfer with one transformer VAE. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 1953-1967.
[22] Yin, Z., Reuben, F., Stepney, S., & Collins, T. (2023). Deep learning's shallow gains: a comparative evaluation of algorithms for automatic music generation. Machine Learning, 112(5), 1785-1822.
[23] Li, S., & Sung, Y. (2023). MRBERT: pre-training of melody and rhythm for automatic music generation. Mathematics, 11(4), 798.
[24] Yu, W. (2025). The construction of improved GCA multi-style music generation model for music intelligent teaching classroom. Systems and Soft Computing, 200221.
[25] Wang, W., Li, X., Jin, C., Lu, D., Zhou, Q., & Tie, Y. (2022, July). CPS: full-song and style-conditioned music generation with linear transformer. In 2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW) (pp. 1-6). IEEE.
[26] Chen, J., Yang, G., Zhao, H., & Ramasamy, M. (2020). Audio style transfer using shallow convolutional networks and random filters. Multimedia Tools and Applications, 79(21), 15043-15057.
[27] Li, S., Zhang, Y., Tang, F., Ma, C., Dong, W., & Xu, C. (2024, March). Music style transfer with time-varying inversion of diffusion models. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 38, No. 1, pp. 547-555).