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

Application of Artificial Intelligence-based Music Generation Technology in Popular Music Production

Fan Wang[1,✉]

[1]*Art College, Zhengzhou Shengda University, Zhengzhou, Henan, 451191, China*

ABSTRACT

Artificial Intelligence has been applied in many aspects of life, however, AI algorithms have been less used in the field of music. In this paper, a multi-track based pop music generation model MuseGAN is proposed, due to its poor contextualization and excessive tempo jumps in generating pop music samples. In this paper, a new multi-track pop music generation model-Recurrent Feature Generation Adversarial Network RFGAN is proposed. the model addresses the temporal relevance of the music structure and the repetitive nature of the musical section, and proposes a temporal model that enhances the contextual relevance of the music samples in terms of the time series, and improves the generative model according to this temporal model by converting the unidirectional structure in the original model to a recurrent structure, adding the feature extractor to the previous level of training information, which is combined with arbitrary noise and passed to the next training. An average pooling layer is added at the end of the generative model as a solution to the situation where the model generates too much noise for pop music samples. The improved model is superior to the pre-improvement model in terms of stability, convergence speed, and overfitting in pop music generation. In the audience scoring experiment, 60% of the top 5 pop music scores were generated using the RFGAN model proposed in this paper, indicating that the pop music generated using the RFGAN model has reached a high level comparable to the level of artificial pop music composition.

Keywords: Contextualization; MuseGAN; RFGAN; Generative Adversarial Networks; Multi-track Music Generation

1. Introduction

The music industry, as a practical field of music culture and theory, has always been a sensitive "reactor" of technological innovation and social development, and the dynamics of the industry in the fields of music creation, music production, music dissemination, music performance, and music education profoundly reflect the self-improvement and development of the art of music. From the birth of recording technology

✉ Corresponding author.
E-mail address: 13598000435@163.com (F. Wang).
Received 10 June 2024; Revised 08 September 2024; Accepted 13 October 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-038

to the rise of digital music, every technological innovation has had a profound impact on the structure and operation of the music industry. With the rapid development of the current generative technology language, the deep combination of music and AI makes music technology upgrade and revolutionize itself once again [1-2].

The music production industry as a process link in the music industry, especially obviously reflects the technical characteristics, so the innovation diffusion of AI technology products will change the overall situation of the industry more rapidly. This has led to the "diffusion of innovation effect", resulting in the birth of a variety of software tools [3-4]. These tools not only reduce production costs and make high-quality music production easy to realize, but also provide musicians with efficient means of creation, thus improving the overall quality of musical works [5]. From the melody formed by each note, to the accompaniment woven by each instrument, to post-recording, editing and mixing, AI technology not only provides content products for the music industry, but also promotes the value-added of the industry [6]. It can be said that AI technology has not only changed the technical landscape of music production, but also had a profound impact on the music industry's business model and creative approach.

Lin, T. F. et al. point out that AI is of great importance to the music industry, and that generative AI can help creators move away from traditional structural frameworks for music, allowing creators to explore new possibilities in the creative process [7]. Suh, M. et al. explore the role played by AI in human-AI musical co-collaboration, and experiments have shown that AI can provide creative collaboration with a common ground, act as a psychological safety net for creative activity, provide strength for group progress, reduce interpersonal stagnation and friction, and also change the collaborative and creative roles of users [8]. Bryan-Kinns, N. et al. explored the process of interactive generative AI systems for creative practice in different musical genres, in addition to the AI having the roles of an audience and critics to enable the musicians to work in collaboration with the AI to collaborative creative process to accomplish creative practice [9]. Eric, T. G. et al. analyzed the impact of generative AI music modeling on composers and used an ethnographic approach to explore the practical needs of composers in order to design human-AI collaborative tools that reflect personal ambitions, socio-cultural contexts, and distinctive genre characteristics [10]. Deruty, E. et al. investigated the contemporary production of popular music in the AI music tool design and use, concluded that audio-based AI tools are more relevant to the creative workflow of music creators than piano roll or MIDI-based tools [11].

Nicholls, S. et al. investigated the application of AI in the collaborative process of creating and producing a piece of recorded music by building models and data training to synergize the interactive process between songwriters for the long-term integration of technology and music [12]. Louie, R. et al. showed that although interactive music AIs are capable of generating a large amount of content for music creation, their outputs are subject to uncertainty, and so developed an AI oriented tool consisting of speech channels that uses sliders of instances to control the similarity of the generated content to existing examples, and semantic sliders to push music generation in an advanced direction [13]. Kaliakatsos-Papakostas, M. et al. introduced AI methods for deep abstraction and modeling of music, on the basis of which they proposed the fusion of music non-adaptive generation methods, implicit or explicit data learning probabilistic methods, and intuitively interactive evolutionary methods [14]. Cádiz, R. F. et al. proposed two image-based AI models for music generation, the variational self-encoder network TimbreNet capable of generating audio-based musical chords through training, and the generative adversarial network StyleGAN Pianorolls was able to create short music excerpts and was evaluated to be creative for the proposed generative model [15]. Li, F. constructed a set of creative and artistic music generative models using Hidden Markov Models for music chord recognition and proposed algorithms based on Multi-Style chord music generative networks to generate music with stylistic features [16].

This paper introduces a classical multi-track pop music model-MuseGAN, which adopts DCGAN as a framework while incorporating an optimized WGAN-GP model. To address the shortcoming that the music phrases generated by the MuseGAN model are not contextually relevant. The original generator with unidirectional structure is proposed to be converted into a generator with cyclic structure. The

randomness of the generated samples and the innovativeness of the generated popular music are guaranteed. Then a new temporal generation model is proposed, and a feature extractor is added to the front end of the generator, an average pooling layer is added to reduce the jumping amplitude between the values in the regions in the resultant sample matrix, and the generated samples are organized and filtered, so as to achieve the purpose of generating smooth pop music.

2. Artificial Intelligence in Music Generation

Music generation, or "AI composition", is also known as artificially intelligent music creation. With the rise of deep learning techniques, especially the emergence of models such as Recurrent Neural Networks (RNN) and Generative Adversarial Networks (GAN), AI composition has made great progress. These models can learn the features and laws of music from a large amount of music data, and thus generate musical compositions with a high degree of complexity and creativity. Currently, AI composition has covered a wide range of styles and genres of music, including classical, pop, and electronic music. With the development and application of AIGC technology, the degree of automation of music generation systems can be categorized into two types. One is the use of machine learning as a music generation model. In this case, the overall music creation activity relies more on the intervention of the composer.

The other is the use of deep learning techniques as a generative model. In this case, the system is capable of automatically generating complete or more complete music works, including complete generation, partial generation, or human-machine organic interaction. Based on the higher degree of automation of the music generation system, according to the different degrees of human-computer interaction, there are two specific methods of "human-machine collaboration": one is the systematic autonomous generation of the machine, with a lower degree of human-computer interaction. For example, OpenAI MuseNet, a deep learning model developed by OpenAI, aims to generate music through artificial intelligence techniques.

There is also compositional collaboration, in which humans and computers work together to create musical works. In this approach, computers usually provide a variety of tools and resources to help composers with tasks such as music theory analysis, harmonic processing, and melody generation during the composition process. This collaborative model can greatly enhance the creativity and efficiency of the compositional process.

In addition to traditional music generation models and deep learning techniques, several new approaches and applications have emerged that further expand the possibilities and scope of music creation.

One of these new trends is music generation systems that incorporate emotion recognition techniques. These systems are not only capable of generating music, but also of recognizing and expressing emotions, thus making musical works richer and more expressive. For example, by analyzing elements such as rhythms, chords, and pitches in music and combining them with emotion recognition algorithms, the system can generate music works that meet emotional needs based on emotional guidance input from the user. In this way, music creation is no longer limited to a single musical element, but pays more attention to the transmission and expression of emotions, thus enhancing the infectiousness and attractiveness of musical works.

Another new direction is the music generation system based on large-scale data. With the popularization of the Internet and digital technology, the acquisition and storage of music data has become more convenient and abundant. Using these large-scale data, combined with machine learning and deep learning technologies, more complex and intelligent music generation models can be constructed to generate more diverse and personalized music works. In this way, the music generation system can learn and extract the laws from the massive music data, constantly optimize and improve its own generation ability, and provide users with a more personalized and high-quality music experience.

In summary, with the continuous development and application of artificial intelligence technology, the degree of automation of the music generation system is constantly improving, and at the same time, it also presents more and more diverse and innovative characteristics. In the future, more novel music generation technologies and applications will bring more possibilities and surprises for music creation and expression.

People are exploring more and more methods on music generation. Generative Adversarial Network, as a very promising generative model at this stage, has attracted the favor of many researchers, and its application to popular music generation will enhance the use of professional music theory knowledge to better improve the generative effect between music tracks. Transforming professional music theory knowledge into data information to guide the model to generate music, this will be the future direction of music generation development.

3. Generative Adversarial Networks and their Derivative Models

3.1. Generating Adversarial Networks

Generative Adversarial Network (GAN) is a common generative model [17]. Generative Adversarial Network is composed of two main parts: generator and discriminator. Among them, the generator's task is to transform random noise into data samples that are closer to the real data distribution as much as possible. The discriminator's task is to correctly discriminate, as much as possible, whether the input data samples are real data or data generated by the generator.

In generative adversarial networks, the generator and the discriminator form a dynamic game process, where the generator and the discriminator are alternately trained and optimized using separate optimizers until a Nash equilibrium is reached, i.e., the discriminator is unable to accurately discriminate the outcome with an accuracy close to 50%. During training, first, a set of potential vectors z is randomly generated, which usually obey some prior distribution. Then a generator is used to convert the potential vectors z into generated samples $G(z)$, after which the discriminator and generator are trained in turn. When training the discriminator, the fake samples generated by the generator are mixed with real samples, allowing the discriminator to distinguish whether these samples are real or generated. When training the generator, let the generator generate results as close as possible to the real samples. The objective function of the generative adversarial network is as follows:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right] \tag{1}$$

where x denotes the real data sample, $p_{data}(x)$ denotes the real data distribution, z denotes random noise, $p_z(z)$ denotes the data distribution in the latent vector, D denotes the discriminator, and G denotes the generator.

Regarding the objective function of the generative adversarial network, it can be understood in two parts. The first part $E_{x\sim P_{data}(x)}\left[\log D(x)\right]$ represents that during the training process of the discriminator, the model will try its best to make the discriminator be able to recognize the real data correctly as much as possible, and when $D(x)$ approaches 0, it indicates that the discriminator fails to recognize the real data correctly. The second part $E_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right]$ represents that during the training process of the generator, the model will try as much as possible to let the samples generated by the generator be determined as real samples by the discriminator, and when $D\left(G(z)\right)$ is close to 0, it indicates that the generator fails to produce enough real fake data. In the training process of the discriminator, the model will try to let the samples generated by the generator be judged as fake samples by the discriminator, i.e., let $D\left(G(z)\right)$ be as close as possible to 0. When $D\left(G(z)\right)$ is close to 0, it indicates that the discriminator is able to correctly discriminate the fake samples; when $D\left(G(z)\right)$ is close to 1, it indicates that the discriminator is not able to correctly discriminate the fake samples, and the discriminating ability is poor.

In addition, the minimization task of generative adversarial networks can also be equated to minimizing the following loss function:

$$Loss = KL\left(P_g \| P_{data}\right) - 2JS\left(P_{data} \| P_g\right) \tag{2}$$

Where, P_g represents the distribution of samples generated by the generator, P_{data} represents the distribution of real samples, KL denotes the KL scatter, and JS denotes the JS scatter. The specific formula for the KL scatter is as follows:

$$KL\left(P_1 \| P_2\right) = E_{X \sim P_1} \log \frac{P_1}{P_2} \tag{3}$$

The specific formula for JS scatter is as follows:

$$JS\left(P_1 \| P_2\right) = \frac{1}{2} KL\left(P_1 \left\| \frac{P_1 + P_2}{2}\right.\right) + \frac{1}{2} KL\left(P_2 \left\| \frac{P_1 + P_2}{2}\right.\right) \tag{4}$$

In the formulas of KL scatter and JS scatter, P_1 and P_2 denote two different distributions, respectively. Both KL scatter and JS scatter can be used to measure the distance between two distributions, and the smaller it is, the closer the two distributions are.

Although generative adversarial networks have achieved a series of results in the field of generation, they also have some limitations in training and generating results. First, the training process of generative adversarial networks is more unstable, and there may be a mismatch in the learning rate between the generator and the discriminator, which makes it difficult for the model to converge to the desired state. Second, generative adversarial networks may experience pattern collapse when generating data.

3.2. DCGAN

Deep Convolutional Adversarial Generative Network (DCGAN) replaces the multilayer perceptron network in the generative and discriminative models of the original Adversarial Generative Adversarial Network with a convolutional neural network [18]. DCGAN has the following main improvement schemes.

In order to simplify the model structure, the pooling layer in the convolutional neural network is replaced in the generative model using fractional step convolution.

$$o' = s\left(i-1\right) + k - 2p \tag{5}$$

$$O = \frac{I - K + 2P}{S} + 1 \tag{6}$$

In the discriminative model, the pooling layer is replaced with an integer step size convolution. In contrast to the generative model the discriminative model uses downsampling to reduce the dimensionality of the multidimensional array and output the corresponding probability values.

For the selection of activation functions, the deep convolutional generative adversarial network model uses the scheme of Tanh function in the output layer and ReLU function in other layers.

3.3. WGAN and WGAN-GP

Due to the problems of unstable training of generative adversarial networks and the possibility of pattern collapse and thus poor quality of generated samples, Arjovsky et al. proposed Wasserstein GAN [19], abbreviated as WGAN, in 2017.The most significant improvement of WGAN is the adoption of the Wasserstein distance as a criterion for measuring the disparity between the generator and discriminator, which has better smoothness compared to the KL scatter and JS scatter, which has better smoothness and thus theoretically solves the problem of vanishing or exploding gradients, and also helps the network to generate higher quality samples. Usually, the Wasserstein distance is defined by the following formula:

$$W\left(P_r, P_g\right) = \inf_{\gamma \sim \prod\left(P_r, P_g\right)} E_{(x,y)\sim\gamma}\left[\|x - y\|\right] \tag{7}$$

where P_r represents the distribution of real samples, P_g represents the distribution of generator-generated samples, $\prod\left(P_r, P_g\right)$ represents the set of all possible joint distributions of P_r and P_g, and

for any of the possible joint distributions γ in this set, one can sample from it to get a real sample x and a generated sample y, and $\|x-y\|$ is the distance between these two samples, so that by sampling and calculating one can get the distance of the samples under the joint distribution with the expected value $E_{(x,y)\sim\gamma}\left[\|x-y\|\right]$ and the Wasserstein distance is the lower bound of that expected value for all possible joint distributions.

Since the Wasserstein distance cannot be solved directly, in WGAN the authors define it in the following form:

$$W\left(P_r,P_g\right)=\frac{1}{K}\sup_{\|f\|_L\le K}E_{x\sim P_r}\left[f\left(x\right)\right]-E_{x-p_g}\left[f\left(x\right)\right] \tag{8}$$

In this equation, Arjovsky et al. introduced the Lipschitz continuity condition. This condition requires that there exists a constant greater than or equal to 0 such that any two elements x_1 and x_2 of the function f in the domain of definition satisfy the following equation, where K is called the Lipschitz constant of the function f.

$$\left|f\left(x_1\right)-f\left(x_2\right)\right|\le K\left|x_1-x_2\right| \tag{9}$$

In WGAN, the Wasserstein distance is the value of taking an upper bound of $E_{x\sim p_t}\left[f\left(x\right)\right]-E_{x\sim p_g}\left[f\left(x\right)\right]$ and dividing by K for all functions f that satisfy the condition in the $\|f\|_L\le K$ case where the Lipschitz constant of the function f is required. If a set of parameters ω is used to represent these functions f_ω that may satisfy the condition, the approximate solution is expressed as follows:

$$K\cdot W\left(P_t,P_g\right)\approx\max_{W:\|f_\omega\|\le K}E_{x\sim P_r}\left[f_\omega\left(x\right)\right]-E_{x\sim p_g}\left[f_\omega\left(x\right)\right] \tag{10}$$

However, in the above equation, there is also a restriction of $\|f_\omega\|_L\le K$. To solve this problem, WGAN introduces a parameter pruning strategy. Such an operation enables the derivatives $\frac{\partial f_s}{\partial x}$ of the input samples x to be restricted to a certain range as well, thus ensuring that there exists a constant K such that $\|f_\omega\|_L\le K$. At this point, the task of minimizing the Wasserstein distance transforms into maximizing $E_{x\sim p_t}\left[f_\omega\left(x\right)\right]-E_{x-p_g}\left[f_\omega\left(x\right)\right]$.

Although WGAN improves the training stability by introducing the Wasserstein distance and reduces the occurrence of problems such as pattern collapse. However, the weight pruning strategy proposed in WGAN has some defects: on the one hand, it leads to parameter centralization, i.e., the weights learned by the network converge to two extremes, which will lead to a decrease in the discriminator's generalization ability and judgment ability. On the other hand, when the network used by the discriminator is deep, WGAN may still suffer from gradient explosion or gradient vanishing. For this reason, Ishaan Gulrajani et al. proposed WGAN-GP.

To address the shortcomings of WGAN, WGAN-GP uses a gradient penalty instead of the weight pruning strategy in WGAN. The formula for its gradient penalty is as follows:

$$\begin{aligned}GP\_Loss=\lambda\cdot E_{x\sim p_{data}(x),\varepsilon\sim U_{[0,1]}}\\\left[\left(\left\|\nabla_x D\left(\varepsilon\cdot x+\left(1-\varepsilon\right)\cdot G\left(z\right)\right)\right\|_2-1\right)^2\right]\end{aligned} \tag{11}$$

where λ is the coefficient of the gradient penalty, which is used to control the weight of the gradient penalty in the total loss, x is a sample taken from the real data distribution $p_{data}\left(x\right)$, ε is a random value taken from a uniform distribution $U\left(0,1\right)$, z represents the random noise, $G\left(z\right)$ represents the generator's generated samples, and $\nabla_x D\left(\varepsilon\cdot x+\left(1-\varepsilon\right)\cdot G\left(z\right)\right)$ is the gradient of the discriminator with respect to its inputs.

By adding a gradient penalty to the loss function, the training stability of WGAN is further improved and higher quality samples can be generated. In the model presented in Chapter 3 of this paper, WGAN-GP is used to improve the training stability and generate higher quality music samples.

4. Pop Music Generation Model

4.1. Generating Pop Music Generation Models for Adversarial Networks

One after another, the idea of generating pop music based on generative adversarial networks has become more and more solid. The goal of this paper is to generate contextually relevant multi-track pop music, and a classical multi-track pop music model, MuseGAN, is introduced here. The model adopts DCGAN as the base framework, and combines with the optimization algorithm of the WGAN-GP model to optimize the model training.

The authors summarize the respective advantages and disadvantages of Jamming model and Composer model, and propose a third generative model - Hybrid model.

The Hybrid model combines the respective advantages of the two models. The Hybrid model kind of generative model has two inputs, a random vector Z_I for each track and a global input vector Z_J, which maintains the individual track characteristics of the multi-track music samples as well as the harmony between the tracks.

The Muse GAN model also proposes two temporal generation models, where the temporal model deals with the temporal connections between the independent bars of the generated music.

$$G(z)=\left\{G_{bar}\left(G_{temp}(z)^{T}\right)\right\}_{t=1}^{T} \tag{12}$$

The first is the Generation from scratch timing model. The functional expression of the timing model is shown in Equation (13):

$$G^{o}(z,x)=\left\{G_{bar}^{o}\left(z^{(t)},E\left(x^{(t)}\right)\right)\right\}_{t=1}^{T} \tag{13}$$

The second temporal model is Track-conditional generation, which can be used to assist in the musical accompaniment of an instrument, and assist human composers in effectively expanding from single-instrument music to multi-instrument music.

In terms of generated music, most of the music generation models based on generative adversarial networks, including Muse GAN, generate music phrase samples with poor contextual correlation and large tempo jumps in the phrases. The generated music samples do not articulate smoothly between the notes of the music and other problems, and the resultant music samples need to be further processed manually.

4.2. Recurrent Feature Generation Adversarial Networks

4.2.1. Cyclic mode of the generator. To address the problem that the music phrases generated by the model are not contextually relevant. We propose to convert the unidirectional structure of the original GAN generator into a cyclic structure.

The network state information of the previous moment will act on the network state of the next moment. The similarity of the recurrent generation model proposed in this paper is that the output of the previous round of training will have an effect on the output of the next round of training. Both the connection between each generated music sample is strengthened and the randomness of the samples generated in each round of training is maintained because of the presence of random noise. The innovativeness of the pop music samples generated in each round is guaranteed.

4.2.2. Timing structure of the model. Based on this model, this paper proposes a new timing generation pattern. In the first round of training a single track of music is generated as the main melody from the

Generation from scratch timing pattern of the generative model in the Musegan model. Only in the first round the Generation from scratch timing pattern of the original model is used to generate the single-track main melody as a condition, and in the other rounds of training only the results of the previous round of training are used as the main melody, and then the feature vector is spliced with random noise after the feature extractor has extracted the features of the popular music information that produces the sample results. With this as a priori condition injected into the Track-conditonal generation timing pattern of the Musegan model. Music containing 5 tracks is generated. This makes the generative model produce pop music samples in series.

$$G\left(z_i, y\right)=\left\{G^o(\mathrm{E}(\mathrm{y}), z)\right\}^T \tag{14}$$

$$y=G_{bar}\left(G_{temp}(z)^T\right), s=1 \tag{15}$$

$$y=G_{s-1}(z, y), s>1 \tag{16}$$

The temporal pattern expression is shown above, Equation (14) is the temporal pattern expression, and Equation (15) and (16) are the different data represented by y for different number of training rounds, respectively. Equation (15) is the y function for the first round of training, and equation (16) is the y function for the subsequent training.

4.2.3. Feature Extractor. A feature extractor is added to the front end of the generator with a loop structure. The information extracted by the feature extractor is fed into the generator as a condition. Depending on the number of layers of the feature extraction network, different information is thus extracted for the features. Different pop music feature information is passed to the next round of training of the generative model. The feature extractor learns the temporal information of the generated results and the pitch feature information of the music. So that this information is passed to the next round of training.

4.2.4. Average pooling layer. The use of an average pooling layer preserves the overall characteristics of the data. The average pooling layer reduces the jumps between values in the regions of the resultant sample matrix, and further organizes and filters the generated data so as to smooth the data of the generated pop music results, and the transitions and articulations of the music notes are smoother when converted to MIDI format. The average pooling layer plays a role in smoothing the sample results here.

4.2.5. Total Network Model Structure for RFGAN. In summary, a new multi-track generation model is formed by combining the above improved schemes, the Cyclic Feature Generation Adversarial Network (RFGAN), the structure of this model is shown in Figure 1, which can be used to generate context-sensitive musical phrases, the first part input is the main melody generation part as the prior condition of the generative model, in the first round of training of the model, a single track music sample is generated by the T1 mode, which is fed into the decoder E as the main theme, and in the subsequent training, the result sample of the previous round of training is used $F\left(G_{s-1}\left(z, E\left(y\right)\right)\right)$. After extracting the feature information $F\left(G_{s-1}\left(z, E\left(y\right)\right)\right)$ of the result sample through the feature extractor TE, the feature information is decoded by the decoder E and input into the convolutional layer of different generating models respectively (the shape of the input array matrix is the same), so as to influence the generative model to produce pop music results.

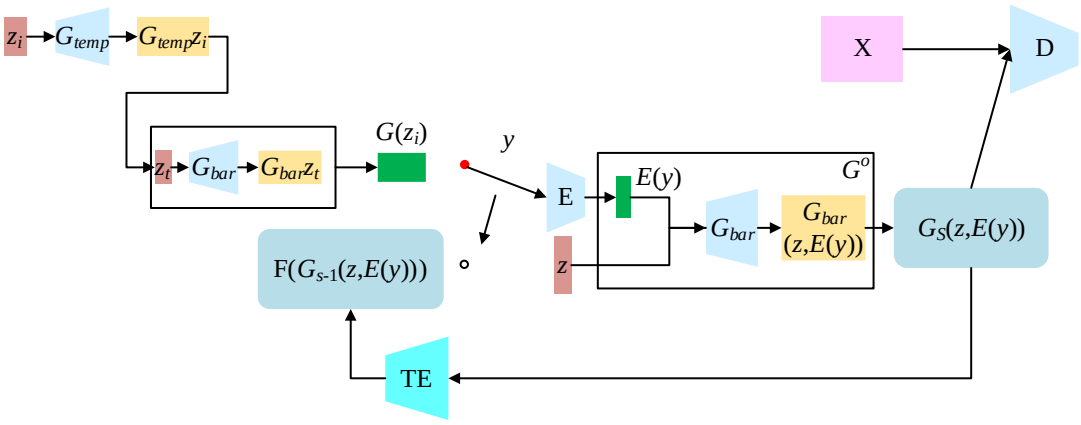

Fig. 1. Rfgan model summarizes the composition

The result samples $G_s\left(z,E\left(y\right)\right)$ obtained from this round of training will not only be used as inputs for discriminative model D together with the real training set x to complete the training of discriminative model D, but will also be used in the next round of generative model training. Same as the Muse GAN model training strategy, according to every 5 updates of the discriminative model, the generative model is updated once. The training is alternated in turn.

The input random noise Z in the T2 timing model is composed of four parts respectively four random noises within and between tracks. One part is the shared weights input to the five-track generation model, random noise with time series z_t and random noise without time series z, which are used to constrain the inter-track music generation. One part is the random noise with and without time series z_t and z within the tracks, which is used to influence the popular music generation within the tracks.

$$y = G_{bar}\left(G_{temp}\left(z_i\right)^T\right), s = 1 \tag{17}$$

$$y = G_{s-1}(z, y), s > 1 \tag{18}$$

$$G(z, y) = \left\{G^o(E\left(f\left(G_{s-1}(z, y)\right), \tilde{z}\right)\right\}^T \tag{19}$$

Equation (17) is the expression for the total model generating model, and (18), (19) are the y - expression functions for the case of different number of training rounds, respectively.

5. Music generation experiments

For the experiments in this paper on pop music generation, MIDI files from the Lakh MIDI Dataset (LMD) were chosen to generate pop music containing the following five tracks: bass, drums, guitar, piano and strings. The LMD was used because it provides a rich and authentic collection of MIDI files and some associated metadata.The Lakh MIDI dataset is a collection of unique MIDI files, some of which have been matched and aligned with entries in the Million Song Dataset.The goal of the Lakh MIDI dataset is to facilitate large-scale retrieval of information about popular music, both notationally (using only MIDI files) and audio content-based (using information extracted from MIDI files as annotations for matching audio files).

In order to demonstrate the generative effect of the improved network model in this paper, experiments on music generation using the RFGAN proposed in this paper are conducted along with experiments on popular music generation using MuseGAN, and then, the generative results of the two methods are compared, so as to analyze the experimental effect of the RFGAN proposed in this paper. According to the experimental results, the generation results obtained from the experiments using RFGAN are compared with the generation results from the MuseGAN experiments as follows: firstly, the experiments obtained the training loss of the discriminator at different training iteration points, so the convergence curve of the

discriminator loss can be obtained, and thus compared with that of the MuseGAN, and the results are shown in Figure 2. From the two convergence curves of MuseGAN and RFGAN in Fig. 2, it can be found that compared with MuseGAN, the RFGAN in this paper converges faster and the training process is more stable, and it has already converged to stability when the number of iterations reaches 1500.

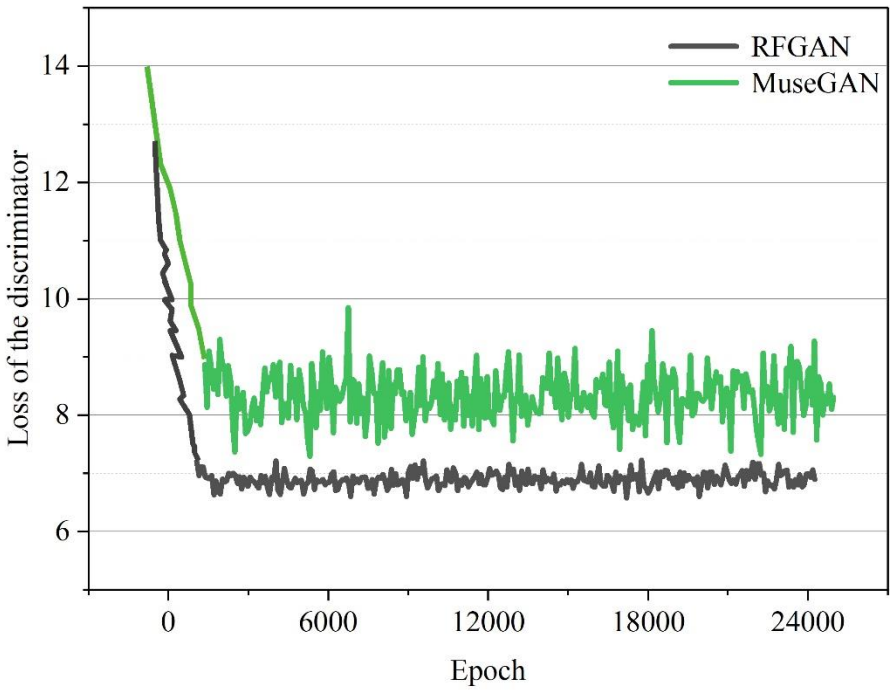

Fig. 2. The discriminator loss is compared to the convergence curve

And, after the experiments, the distributions of the discriminator parameters in the iteration process of the two experiments can be obtained separately, in which Fig. 3 shows the distributions of the discriminator parameters in the MuseGAN experiments, while Fig. 4 shows the distributions of the discriminator parameters obtained from the RFGAN experiments. From the comparison in the figure, it can be found that the interval range of the parameter distribution obtained from the music generation experiment based on RFGAN is [-0.2,0.2], which is obviously smaller than the interval range of the parameter distribution obtained from the experiment using MuseGAN, which indicates that the RFGAN proposed in this paper is less susceptible to the phenomenon of overfitting compared with MuseGAN. To summarize, after experimental comparison, RFGAN is better than MuseGAN in terms of stability, convergence speed and overfitting.

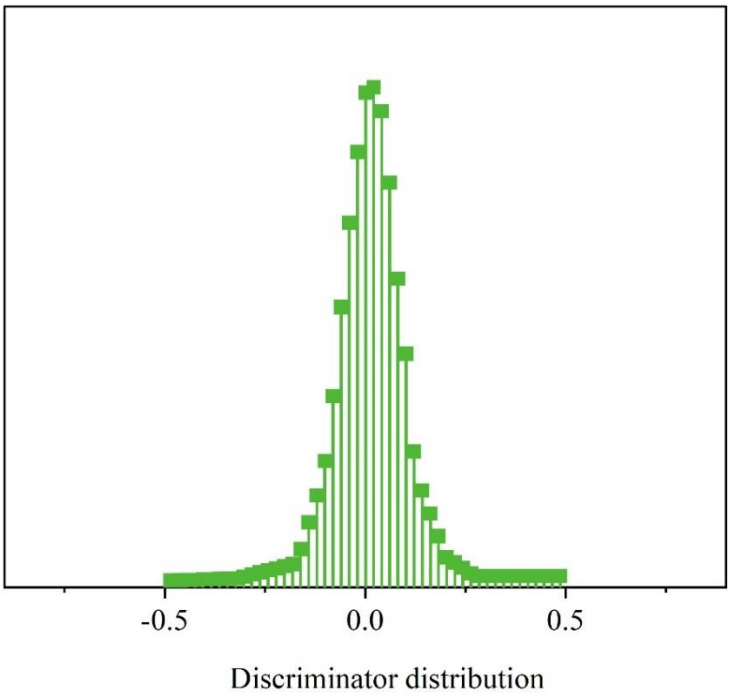

Fig. 3. The distribution histogram of the MuseGAN discriminator parameter

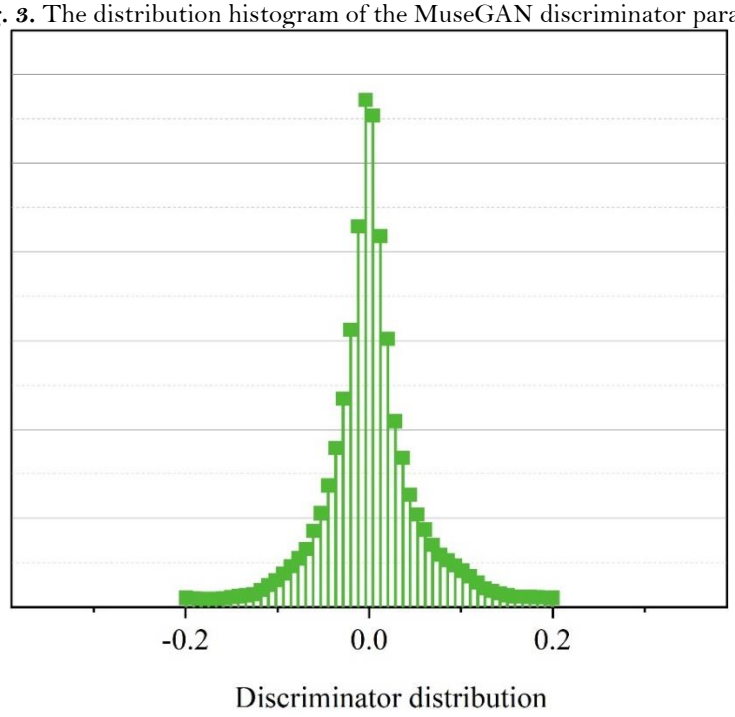

Fig. 4. The distribution histogram of the RFGAN discriminator parameter

This paper proposes several metrics to evaluate the generated results, so this method is used in this paper to compare and analyze MuseGAN with RFGAN. The following metrics are defined in MuseGAN, and the metrics are defined as shown in Table 1, where EB, UPC, and QN are intra-track evaluation metrics, and TD is inter-track evaluation metrics. All these metrics can be computed from the training (actual) data and the generated data to obtain the corresponding metric values, and the effect of the generator can be understood by comparing the values computed from the actual data and the generated data. Thus, the metrics evaluation results are in line with the idea that the distribution of real and fake data in GAN should be close.

Table 1. Model evaluation index

Index name	describe	Range of values
EB (Empty bars rate)	The ratio of the empty section (not the note section).	0%-100%
UPC (Number of used pitch classes)	Each section contains the number of levels of pitch.	1-12
QN (Qualified note rate)	The ratio of qualified notes: if a note is less than three times long (that is, a 32-note), it is considered a noise, or a note of an unstated note.	0%-100%
TD (Tonal distance)	It is also known as the pitch distance, which mainly measures the harmony and the degree of harmony between the sound and the sound tracks.	>0

Table 2 shows the calculation results of several metrics obtained based on the music data generated by RFGAN experiments, the music data generated by MuseGAN, and the real music data, where B stands for bass, G for guitar, S for strings, and P for piano. The row of T1 in Table 2 represents the calculated values of the metrics of the music data generated by the MuseGAN experiment using T1 as the temporal structure, T2 is the same, and T3 represents the calculated values of the metrics of the music data generated by the RFGAN experiment proposed in this paper using T3 as the temporal structure. From the comparison of the computed result values in the EB section, it can be found that the probability of generating blank bars in the generated pop music is relatively low for T3 compared to T2 and T1, and T3 on drums is 4.09, which is 0.49 smaller than that of T2. While from the UPC, it can be found that all the models tend to generate more than four notes at the same time, which suggests that the experiments may have generated some noise, and in the subsequent work we can focus on the solution of this problem. From the data of QN, it can be seen that all the QN values obtained by the method proposed in this paper are improved relative to those in MuseGAN. From the data of TD, it can be found that the performance of the generated pop music is better in terms of harmonicity, and in this respect RFGAN is improved relative to MuseGAN.

Table 2. The results of the index evaluation method

Index	EB				
Sound track	Bess	Drum	Guitar	Stringed instrument	Piano
Real	8.08	8.08	19.2	10.3	24.5
T1	2.09	29.5	11.9	6.12	17.5
T2	2.05	4.58	10.5	4.25	-
T3	2.02	4.09	10.2	4.23	5.4
Index	UPC				
Sound track	Bess	Guitar	Stringed instrument	Piano	
Real	1.78	3.15	3.25	3.19	
T1	2.25	4.79	5.28	5.38	
T2	2.75	4.48	4.36	-	
T3	2.41	4.52	4.39	5.09	
Index	QN				
Sound track	Bess	Guitar	Stringed instrument	Piano	
Real	90.2	81.8	87.5	85.4	
T1	44.5	43.1	53	45.9	
T2	43.6	55.9	68.1	-	
T3	45.9	57.9	69.8	54	
Index	TD				
Sound track	B-G	B-S	B-P	G-S	G-P	S-P
Real	1.54	1.54	1.59	1.13	1.13	1.14
T1	1.38	1.36	1.35	0.89	0.89	0.86
T2	1.42	1.38	1.39	0.97	0.96	0.94
T3	1.32	1.31	1.32	0.81	0.82	0.83

In addition, this paper also randomly selected a number of bars of the piano roll as a comparative demonstration, Figure 5 is the piano roll diagram using T3 as the temporal structure to generate the music, Figure 6 is the piano roll diagram using T1 as the temporal structure to generate the music, the bars circled by the red ellipse in the figure indicate the empty bars, from the comparison of the two diagrams can be seen, the use of T3 as the temporal structure to generate the blank bars only 5, which is less than using T1 to generate the blank bars. The number of blank nodes produced is less, thus indicating that the generation of RFGAN model using T3 as the temporal structure is improved compared with MuseGAN. To summarize, a comparison of the experimental results reveals that compared to the MuseGAN-based music generation experiments, the RFGAN proposed in this paper is improved in terms of the generation of blank bars, the quality of the generated notes, and the generation of the harmonicity of popular music.

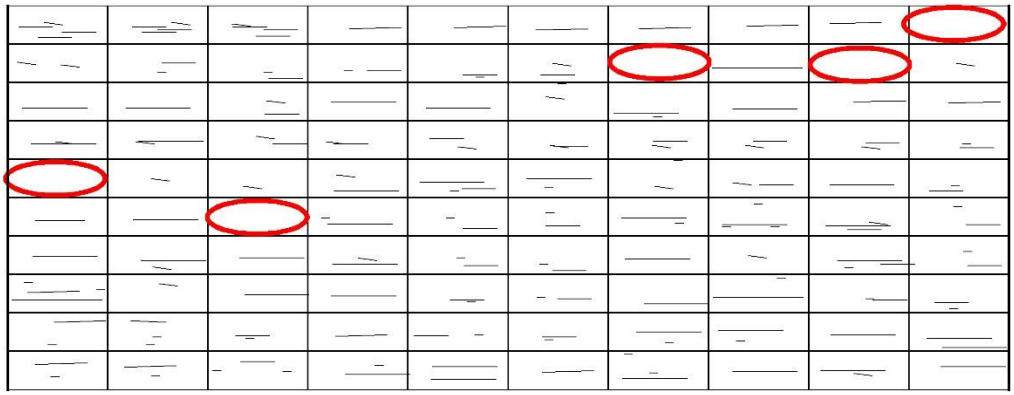

Fig. 5. Use T3 as a time structure to generate music piano rolls

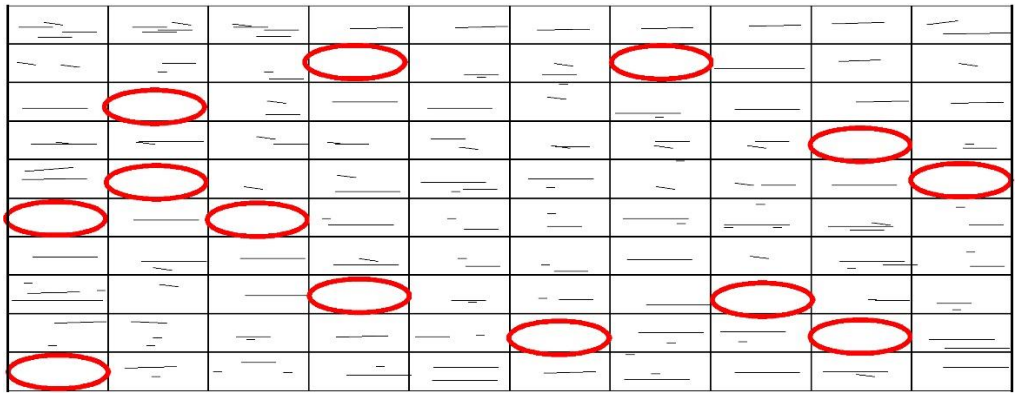

Fig. 6. Use T1 as a time structure to generate music piano rolls

6. Subjective assessment of generated music

This chapter describes the subjective assessment method of popular music, which consists of the main task: organizing live scoring for music majors and non-music majors. In this paper, 50 music majors and 50 non-music majors in our university were invited to conduct a live assessment of 30 popular music songs participating in the test. The specific steps of the assessment are as follows:

Introduce the 100 volunteers in the conference room to the matters related to this experiment, and record whether they have studied music and know about popular pop music.

Randomly disrupt the 30 pop songs to be assessed, play them in sequence, and have the students score the melody, rhythm, and overall structure of each pop song.

The scored data was processed, the results of the 30 songs were tallied, and the results of the experiment were reported to the class.

Following the steps described above, the students were allowed to score the songs, after which the scoring results were processed to turn them into bar charts, so that the comparison of the scores of the 30 songs could be more intuitively seen. The scoring results are shown in Figure 7. When counting the scores, this experiment arranges the disrupted music in sequential intervals. The first of them is the music in the database, the second is the music generated with WGAN, the third is the music generated with the improved RFGAN, and so on after that the 30 songs are categorized into 10 groups, and the y-axis serial numbers 1-10 represent the groupings. On the graph is visualized by the color, the music in the database is represented by green, the music generated by WGAN is represented by blue, and the music generated by the improved RFGAN is represented by black and grey, and each type of music is 10 songs, and as can be seen from the graph, the highest score is for the pop music generated by the RFGAN of this paper located in the group 9, with a score of 99 points.

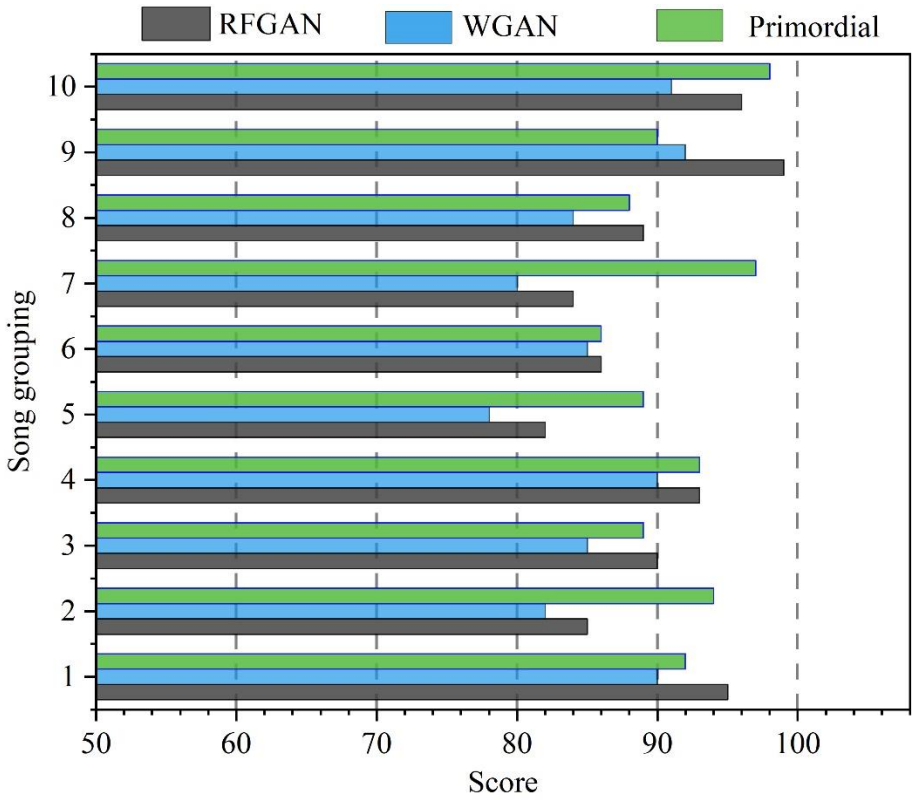

Fig. 7. Grading results

The ranking of the five music songs with the highest evaluation scores can be obtained from Figure 7, and the rankings are shown in Table 3. From Table 3, it can be seen that three of the five music songs with the highest scores were generated using the model proposed in this paper, and the scores they achieved are close to those of the songs in the database, which indicates that these three songs have reached the level of artificial pop music composition, which is recognized by both music majors and non-music majors, and verifies that the model proposed in this paper has the ability to generate high-quality pop music.

Table 3. Score list

Ranking	Song source	Sequence number	Song score
1	RFGAN	9	99
2	Primary database	10	98
3	Primary database	7	97
4	RFGAN	10	96.3
5	RFGAN	1	96

7. Conclusion

This paper presents an in-depth study of generative adversarial networks for the task of generating music on popular music and the application of music generation techniques in practical popular music generation experiments.

The range of parameter distributions obtained by RFGAN-based music generation technique is smaller than the range of parameter distribution intervals obtained by using MuseGAN technique, which illustrates that the RFGAN proposed in this paper is less prone to overfitting phenomenon. In addition to this, RFGAN is superior to MuseGAN in terms of stability, convergence speed, and overfitting in popular music generation, which shows that the improvement of music generation technique in this paper is effective.

There are 5 blank bars generated by T3 as a temporal structure, which is 8 less than those generated by using T1 as a temporal structure, indicating that the effect of pop music generation by RFGAN model using T3 as a temporal structure is improved than MuseGAN. Not only that, the RFGAN model has improved results in terms of generating note quality as well as generating pop music harmonization.

Subjectively assessed, 3 of the top 5 pop music in terms of score were generated using RFGAN model, which demonstrated that the pop music generated using RFGAN model has reached the level of artificial pop music composition, which is not only recognized by music majors and non-music majors, but also verified the efficiency of RFGAN model in generating high-quality pop music.

References

[1] Taylor, J., El Ardeliya, V., & Wolfson, J. (2024). Exploration of Artificial Intelligence in Creative Fields: Generative Art, Music, and Design. International Journal of Cyber and IT Service Management, 4(1), 39-45.

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

[3] Galuszka, P. (2024). The influence of generative AI on popular music: Fan productions and the reimagination of iconic voices. Media, Culture & Society, 01634437241282382.

[4] Bown, O. (2021). Sociocultural and design perspectives on AI-based music production: why do we make music and what changes if ai makes it for us?. Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity, 1-20.

[5] Avdeeff, M. (2019, October). Artificial intelligence & popular music: SKYGGE, flow machines, and the audio uncanny valley. In Arts (Vol. 8, No. 4, p. 130). MDPI.

[6] Deupa, A., Saha, G., Sharma, N. S., & Singh, V. P. (2024). Generative Adversarial Network Based Music Generation. Far Western Review, 2(1), 1-25.

[7] Lin, T. F., & Chen, L. B. (2024). Harmony and algorithm: Exploring the advancements and impacts of AI-generated music. IEEE Potentials.

[8] Suh, M., Youngblom, E., Terry, M., & Cai, C. J. (2021, May). AI as social glue: uncovering the roles of deep generative AI during social music composition. In Proceedings of the 2021 CHI conference on human factors in computing systems (pp. 1-11).

[9] Bryan-Kinns, N., Noel-Hirst, A., & Ford, C. (2024, June). Using Incongruous Genres to Explore Music Making with AI Generated Content. In Proceedings of the 16th Conference on Creativity & Cognition (pp. 229-240).

[10] Eric, T. G., Di Caro, L., & Rapp, A. (2024). Redefining the User in Human-Generative AI Collaboration: Insights from Music Composition. In CEUR WORKSHOP PROCEEDINGS (Vol. 3685, pp. 1-6). CEUR-WS.

[11] Deruty, E., Grachten, M., Lattner, S., Nistal, J., & Aouameur, C. (2022). On the development and practice of ai technology for contemporary popular music production. Transactions of the International Society for Music Information Retrieval, 5(1), 35-50.

[12] Nicholls, S., Cunningham, S., & Picking, R. (2018). Collaborative artificial intelligence in music production. In Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion (pp. 1-4).

[13] Louie, R., Coenen, A., Huang, C. Z., Terry, M., & Cai, C. J. (2020, April). Novice-AI music co-creation via AI-steering tools for deep generative models. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-13).

[14] Kaliakatsos-Papakostas, M., Floros, A., & Vrahatis, M. N. (2020). Artificial intelligence methods for music generation: a review and future perspectives. Nature-Inspired Computation and Swarm Intelligence, 217-245.

[15] Cádiz, R. F., Macaya, A., Cartagena, M., & Parra, D. (2021). Creativity in generative musical networks: evidence from two case studies. Frontiers in Robotics and AI, 8, 680586.

[16] Li, F. (2024). Chord-based music generation using long short-term memory neural networks in the context of artificial intelligence. The Journal of Supercomputing, 80(5), 6068-6092.

[17] Shilong Fan. (2019). Research on Deep Learning Energy Consumption Prediction Based on Generating Confrontation Network.. IEEE Access165143-165154.

[18] Jainy Sachdeva,Puneet Mishra & Deeksha Katoch. (2024). Diabetic retinopathy data augmentation and vessel segmentation through deep learning based three fully convolution neural networks. Image and Vision Computing105284-105284.

[19] Sungwoo Park,Jaeuk Moon & Eenjun Hwang. (2024). Data generation scheme for photovoltaic power forecasting using Wasserstein GAN with gradient penalty combined with autoencoder and regression models. Expert Systems With Applications125012-125012.