International Journal for Housing Science and Its Applications
Publish August 4, 2025. Volume 46, Issue 3 Pages 3080-3090

https://doi.org/10.70517/ijhsa463258

Using Artificial Intelligence to Generate Models to Promote Diversity in Popular Music Composition Forms

Fan Wang[1,*]
[1] Art College, Zhengzhou Shengda University, Zhengzhou, Henan, 451191, China
Corresponding authors: (e-mail: 13598000435@163.com).

Abstract With the rapid development of artificial intelligence technology, generative modeling is increasingly used in the field of artistic creation. In this paper, we design a Note Rank Transformer pop music generative model based on improved Transformer-XL. The model combines lyrics embedding with memory embedding module, and adopts masking array to improve the mechanism of multi-head attention to realize the effective modeling of music sequences. Based on the self-constructed dataset, the Note Rank Transformer model generates samples with a mean value of 90.46% for scale consistency, which is closest to real music (90.22%) in terms of statistical significance, and most of the values of the generating samples are slightly higher than the mean value of real samples for the three metrics of polyphony, note span, and note uniqueness, and for the repetitiveness and high quality notes than the Note Rank Transformer converges faster than Transformer-XL, and the training process is more stable, the interval range of the parameter distributions obtained from the music generation experiments based on Note Rank Transformer is in the range of (-0.2,0.2), which is significantly smaller than that obtained from the experiments using The interval range of parameter distribution obtained from Transformer-XL experiments is (-0.5.0.5), which proves that the improvement strategy in this paper is effective.

Index Terms music generation model, Transformer architecture, Note Rank Transformer model, multi-head self-attention mechanism

I. Introduction

In the past, music composition was considered to be a unique creative expression of human beings, a crystallization of emotions, experiences and inspirations [1], [2]. However, with the continuous development of science and technology, the application of artificial intelligence has become more and more extensive, and it has been able to simulate human creative thinking and process to a certain extent [3], [4]. In the popular music industry, AI generative modeling to promote the diversification of popular music creation forms is gradually becoming a new trend [5], [6].

AI generative models, which can mimic the style and works of creators, generate similar musical works [7]. By training the model, the AI generative model can analyze the features, chords, melodies and other elements of the music and create brand new musical works accordingly [8]-[10]. The application of this technology can greatly save the time of music creation and provide more inspiration for creators [11], [12]. The involvement of artificial intelligence in music creation firstly changes the way of music creation [13]. Traditional music creation often relies on the composer's inspiration, experience, and skills, and requires a long period of study and practice [14]. In contrast, AI is able to quickly generate various musical elements, such as melody, rhythm, and harmony, by analyzing and learning from a large amount of music data [15], [16]. It can create a large number of music clips in a short period of time, providing creators with rich inspirational materials [17], [18]. It is undoubtedly a powerful tool for musicians who face creative bottlenecks or need to get ideas quickly [19]. The application of artificial intelligence in music creation also provides more possibilities for the innovation of music styles, which can break the boundaries of traditional music styles, integrate the characteristics of different genres of music, and create brand new music styles [20]-[22]. This innovative attempt not only enriches the diversity of music, but also satisfies the increasingly diverse musical tastes of listeners [23], [24].

In this paper, we first introduce the basic music theory and explain the structure of the Transformer model. Based on the Transformer-XL architecture, the Note Rank Transformer music generation model is proposed. The synergistic mechanism of lyrics embedding and memory embedding is designed to optimize the multi-head attention mechanism. The popular music generation results of the model are evaluated based on the twelve mean laws and objective metrics comparison. Analyze in terms of stability, convergence speed and overfitting to test the effectiveness of the model improvement.

II. Transformer-based music generation model design

II. A.Basic knowledge of music theory

Music is a form of language expression, from the initial people by hitting the object to send out a simple rhythm through the evolution of history, to the current variety of music styles and genres blossomed in the blooming of music, music has formed a set of exclusive structure and representation. In this section, we will introduce the basic knowledge of music theory to establish the foundation for the subsequent understanding of the music generation model.

Music uses sound as a medium, combining different sounds according to certain rules to form a moving melody. To understand the basic theory of music, one first needs to understand a few essential key concepts in computer music: notes, intervals, chords, and rhythm, which are briefly described next.

(1) Note. The original meaning of a note is “a symbol that records the different lengths of a tone progression”, which is used to indicate the duration of a tone in a pentatonic score. In computer music, notes are given a broader meaning, and are used to represent a basic unit of music, containing not only the duration of the tone, but also the pitch and strength of the tone. Pitch denotes the height of a tone, and the essence of different pitches is the different frequencies of the tones. Pitch is usually represented by a tone name, such as C, D, E. Each of the seven white keys in a piano is a set of pitch cycles CDEFGAB, with the pitch of the latter set being twice the frequency of the former. A common form of a tone name plus a number, such as C3, indicates the C note in the 3rd group of the cycle. the frequency of C3 is twice that of C2. Duration for the duration of the note, there are two forms of expression in the computer, one is expressed in absolute time, such as 3:14-3:16, indicating that the note lasts from 3 minutes and 14 seconds to 3 minutes and 16 seconds for two seconds; the second is expressed in relative time, which is also expressed in the score, such as the two-minute note, the quarter note and so on. Tone intensity indicates the strength of the tone, which means the amplitude of the vibration. Similarly, there are also two kinds of representation of Wanfa, one is a continuous quantitative description, that is, decibels (dB): the second is the use of instrumental dispersion of quantitative description, such as a strong beat, a strong beat, a weak beat, a weak beat, a weak beat, and so on.

(2) Intervals. The meaning of interval is the pitch distance between two notes, its unit is degree, such as C and D between the interval of 2 degrees, C3 and C2 between the interval of 8 degrees. According to the different intervals brought about by different auditory experience, intervals can be divided into concordant intervals and discordant intervals, intervals are concordant or not depends on the ratio between the tone frequency. Pure one degree and pure octave for the very fully consonant intervals; pure four, five degrees for the fully consonant intervals; not fully consonant intervals are the size of three, six degrees.

(3) Chords. A chord is a group of sounds with a certain interval relationship, specifically three or more notes played at the same time. The notes in a chord need to fulfill the conditions of consonant intervals in order to bring about a positive superimposition of experience; otherwise, a chord made up of dissonant intervals will only feel jarring.

(4) Rhythm. Rhythm, as a concrete description of musical rhythm, takes the relationship between the strength of the beat and the length of the tone as a carrier, and constitutes the cornerstone of musical expression of emotion. In computerized music, the rhythm of a piece of music is not usually recorded in a display, but depends on the pitch and duration of the notes.

II. B.Transformer model

Music is a temporal art, which has a strong back-and-forth correlation on the time scale. Due to the superiority of recurrent neural networks in processing temporal data, many scholars have adopted them as the basic model for music generation. However, due to its recursive structure, the recurrent neural network can only pass the information to the next moment after processing the current information, which causes the recurrent neural network can not be computed in parallel and the training efficiency is low. Moreover, as the length of the sequence grows, the information may be lost in the passing of the overgrowth, which causes the recurrent neural network cannot capture the long-run off-dependency relationship. The Transformer model uses an attention mechanism to model the data, which solves the problems of recurrent neural networks not being able to compute in parallel and gradient vanishing, and achieves superior performance in tasks such as natural language processing, music generation, and machine translation.

Transformer is a sequence-to-sequence model, which consists of an encoder and a decoder.Transformen encoder consists of a stack of multi-head self-attention mechanisms and feed-forward neural networks, with residual connectivity and normalization operations performed at each layer of the multi-head self-attention mechanism and feed-forward neural network. In contrast to the Transformer encoder, the multi-head self-attention mechanism of the Transformer decoder performs a masking operation in order to prevent each position from participating in the attention computation. In addition, the decoder has an extra multi-head cross-attention mechanism between the multi-head self-attention mechanism and the feed-forward neural network, which can be used to receive the information passed by the encoder.

II. B. 1) Attention mechanisms

When processing information, human beings are usually unable to scrutinize everything they see, but are quicker to focus on a particular part. After receiving information, the human brain can quickly focus on the relatively important part of the information, so as to make the correct judgment. This mechanism of human information processing inspired researchers to propose the famous attention mechanism, Transformer is based on the attention mechanism of the model, it was released in the field of machine translation and other natural language processing has achieved great success.

The attention mechanism in Transformer is the self-attention mechanism. For the self-attention mechanism, the Query, Key and Value input to the model all come from the same input, i.e., the model decides which part of the important information to focus on according to the input data itself. If the input vector is $A=(a_1,a_2,...,a_n)^T$, the self-attention mechanism firstly maps the input vector into Q, K, and V according to a fully connected layer, which is calculated as in Eq. (1), Eq. (2), and Eq. (3):

$$Q = AW^Q = (q_1,q_2,...,q_N) \tag{1}$$

$$K = HW^K = (k_1,k_2,...,k_N) \tag{2}$$

$$V = HW^V = (v_1,v_2,...,v_N) \tag{3}$$

Then, Q and K are dot-producted to calculate the attention fraction and the results are weighted and summed. The calculation formula is shown in equation (4):

$$Attention(Q,K,V) = Softmax(\frac{QK^\bullet}{\sqrt{d_k}})V \tag{4}$$

where Q,K and V denote the query matrix, key matrix and value matrix, respectively; and d_k denotes the dimension of the query vector.

Conventional self-attention mechanisms may excessively focus their attention on their own position when modeling data. The multi-head self-attention mechanism receives and processes information from multiple subspaces, allowing the model to learn different attentional patterns in different subspaces and improving the modeling ability of the model. The multi-head self-attention mechanism performs an h-group operation on the input sequences and joins them together, and finally performs a linear transformation to obtain the output. The formulas are calculated as in Eq. (5) and Eq. (6):

$$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^o \tag{5}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{6}$$

where W^o, W_i^Q, W_i^K, W_i^V are trainable parameter matrices, and Q,K and V denote the query matrix, key matrix and value matrix, respectively.

II. B. 2) Location coding

Since the Transformer consists entirely of an attention mechanism, the model does not have access to the positional information of the data. That is, if the order of the input data is disrupted, the same attention result is obtained. Therefore, Transformer introduces positional coding to obtain the positional information of the data. The specific calculation method is shown in Eq. (7) and Eq. (8):

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_n}) \tag{7}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_m}) \tag{8}$$

where $PE_{(pos,i)}$ denotes the value of the i th element and d_m denotes the dimension.

II. B. 3) Feedforward Neural Networks

Transformer is connected to a feed-forward neural network after the attention mechanism, which can enhance the expression of the model by feature extraction of the output of the attention mechanism through linear transformation and nonlinear activation. The calculation formula is shown in equation (9):

$$FFN(Z)=ReLU(ZW_1+b_1)W_2+b_2 \tag{9}$$

where Z denotes the input vector, W_1,W_2 denotes the trainable parameter matrix, and b_1,b_2 denotes the bias vector.

II. B. 4) Residual linking and normalization

To model the deep layers, Transformer uses residual connectivity and layer normalization in each sub-layer. Residual connectivity is a deep learning technique that adds the inputs of the previous layer directly to the outputs of the current layer, creating a jump connection that makes it easier to pass information through the network, thus alleviating the problem of vanishing gradients in deep networks. Layer normalization is mainly used to stabilize the training of neural networks, which normalizes on all the features of each sample to make the inputs of different layers more consistent and speed up the convergence of the model. Taking a feedforward neural network sublayer of the Transformer encoder as an example, the residual connection is calculated as in equation (10):

$$Z=FFN(Z)+Z \tag{10}$$

The formulae for layer normalization can be expressed as Eq. (11), Eq. (12) and Eq. (13):

$$\mu^l=\frac{1}{N}\sum_{i=1}^{N}Z_i^l \tag{11}$$

$$\sigma^l=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(Z^l-\mu^l)^2} \tag{12}$$

$$LayerNorm(Z^l)=\alpha\odot\frac{Z^l-\mu^l}{\sqrt{\left(\sigma^l\right)^2+\epsilon}}+\beta \tag{13}$$

where l denotes the l th layer, N denotes the number of hidden nodes in the layer where it is located, ϵ prevents the denominator from being 0, and α,β are hyperparameters.

Due to the powerful modeling ability of Transformer's model and its excellent performance in maintaining long-term structural consistency, this paper proposes the Note Rank Transformer model based on Transformer-XL, and the specific structure of the model will be introduced in the following.

II. C. Music generation model based on Note Rank Transformer

In order to realize the intelligent generation of music data encoding, this paper designs a music generation model based on Note Rank Transformer on the basis of Transformer-XL, and the model structure is shown in Fig. 1.

Transformer-XL does not contain an encoder component, and its piecewise recursive mechanism allows the model to accumulate gradients of computed attention over long sequences, which removes the need for an encoder, since the attention mechanism can focus directly on the entire input sequence. For musical sequences, long-term dependencies are very important. In the short term, phrases often have similar, evolving chord progressions between them, and the cyclic period of this chord progression can be long, thus requiring attention to musical content across phrases, or even across sections. In the long term, phrases of the same type at different locations in the music (e.g., the first chorus versus the second chorus) have very similar content in the past, making it necessary to focus on the musical content several phrases earlier and to decide how to proceed with the next step in the generation process.The properties of Transformer-XL are well suited for tasks that require long term dependencies, which makes it the preferred choice for constructing the models in this thesis. .

As a music generation model that also uses Transformer-XL, the Note Rank Transformer encoded input sequence is fed into an embedding layer, which is then fed into Transformer-XL's decoder. After processing through multiple decoder layers, a list of probabilities for the next output is obtained through a linear layer and a Softmax layer. The hidden state outputs of each decoder layer are stored as memory embeddings and fed into the decoder along with the input embeddings at the next call. In this case, the input embedding is used to generate Q (query vector) to compute the attentional weight of each input, while the memory embedding, when spliced with the input embedding, is used to generate K (key vector) and V (value vector) to compute the context-sensitive representation of each input.

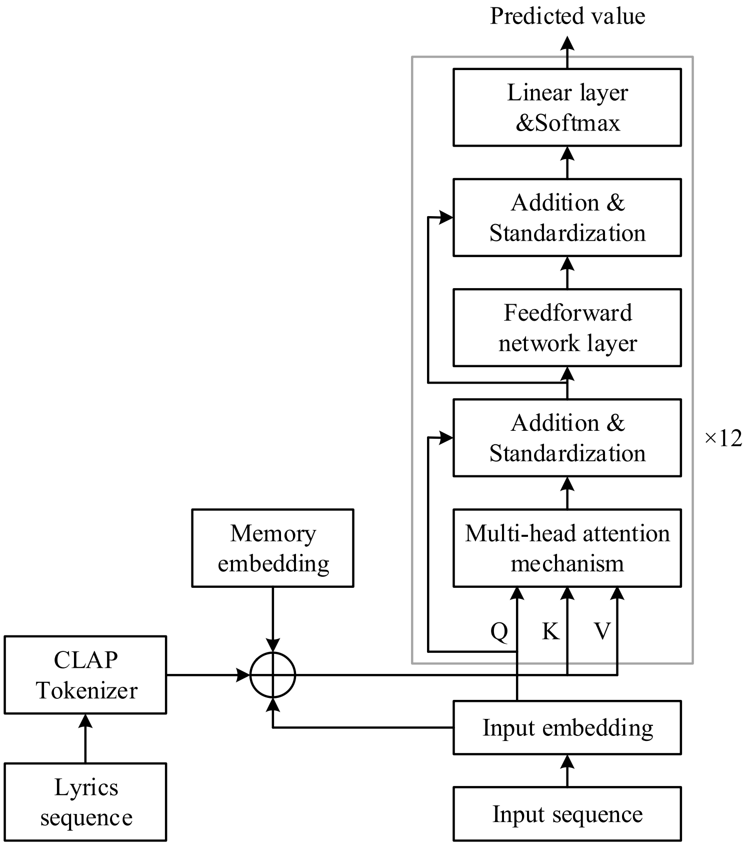


Figure 1: Structure Diagram of the Note Rank Transformer model

In the original Transformer-XL model, there is no module for conditional control, and content can only be generated from zero or continued based on input. In order to use lyric embeddings to condition the generation of music, in this paper, lyric embeddings are spliced together with memory embeddings and fed into the Note Rank Transformer together. Since the dimensionality of both lyric embeddings and memory embeddings is 512, they can be directly spliced together without dimensional transformation. The difference is that the memory embedding is updated after each training and backpropagation, while the lyrics embedding has the same input each time.

During training, the length of the input sequence is 512 codes each time. Among all the songs, the longest number of encodings is 8704, so a maximum of 17 groups of encodings per song are required. For groups at the end of the encoding and with an effective length of less than 512, a masking array is used to prevent multiple attention from learning parts other than the effective encoding. After obtaining the distribution of the predicted values through the Softmax layer, the model compares it with the true value that should be generated (i.e., the next encoding), and the loss function used is the cross-entropy loss function, as shown in Equation (14):

$$Loss = -\sum_{x} P(x)\log Q(x) \tag{14}$$

where x is the classification label, $P(x)$ denotes the probability of x in the true probability distribution, and $Q(x)$ denotes the probability of x in the probability distribution predicted by the model.

In the generation process, the input sequence is encoded with a length of 1, the previous output value. However, the length of the vectors and the number of parameters involved in their computation do not differ from the training due to the presence of memory embeddings and lyrics embeddings.Note Rank Transformer generates the next encoding of music data as shown in Eq. (15).

$$n_t, m_t = M(n_{t-1}, m_{t-1}, L, \theta_c) \tag{15}$$

where M denotes the present model, n_t denotes the music data encoding at the moment of t, m_t denotes the memory embedding at the moment of t, L denotes the lyrics embedding, and θ denotes the model parameters.

After obtaining the probability distribution of the next output, this paper uses a temperature parameter T to recalculate the probability distribution. This method readjusts the probability distribution, which can avoid model degradation and increase diversity, improving generalization ability. The specific way it works is shown in Equation (16):

$$p_{n,new} = \frac{e^{\frac{p_{n,old}}{T}}}{\sum_{i=0}^{N} e^{\frac{p_{i,old}}{T}}} \tag{16}$$

where p_n denotes the probability of the nth item, and N denotes the total number of items, in this case the dimension of the input coding.189 Finally, the sampling method chosen in this paper is kernel sampling. This sampling method sets a threshold ρ and sums all the selected items starting from the one with the highest probability until the sum of the probabilities exceeds ρ, and then randomly selects a value according to the weights among these summed items. This sampling method can enhance the diversity of generated content while ensuring that the model does not generate unexpected content with very low probability.

III. Experiments on music generation based on the Note Rank Transformer model

III. A. Data sets and pre-processing

The dataset used in this experiment is a total of 1000 single-track monophonic MIDI music samples collected on the web. The MIDI data were normalized and cut into fixed-length music samples, which were then encoded and converted into two-dimensional sequences for input into the model for training. In this experiment, we set each input and generated data as music samples of 5 bars in length, and normalized them with the following method: the beat is unified as 4/4 beat, the minimum note unit is sixteenth note, and the sampling rate is 0.125s, i.e., 8 minimum note units are sampled per second, and then all the music data are segmented into 5-bar music segments with a time step of 80.

With the help of the MIDI data processing package Pretty-MIDI in Python, the segmented music segments were multi-hot encoded and converted into 2D note sequences, and finally a total of 9100 fixed-size 2D note sequences were obtained in the training dataset for model training.

III. B. Analysis of music generation effects

III. B. 1) Comparison of Twelve Mean Rhythms

In order to be able to analyze the model-generated data in comparison to the real data, this paper uses a twelve-mean law to count the note distributions of the generated samples and the real music data. The twelve-mean law is a generalized method of musical law that divides a set of octave intervals into twelve equal parts equally proportional to their frequency. By counting the twelve equal-tempered note distributions of real music and generated music, it is possible to measure whether the generative model can produce a score that matches the note distribution of real music. The note frequency distribution of the generated music and real music is shown in Fig. 2. The note frequency distribution of the music data generated by the Note Rank Transformer model is similar to that of the training music dataset, with D, A, and B notes appearing the most, while F#, G#, and C# notes all appear less frequently. notes all appear less frequently, indicating that the Note Rank Transformer model effectively learns the note distribution laws in the dataset, the note distributions basically match, and the generated music has a specific musical style.

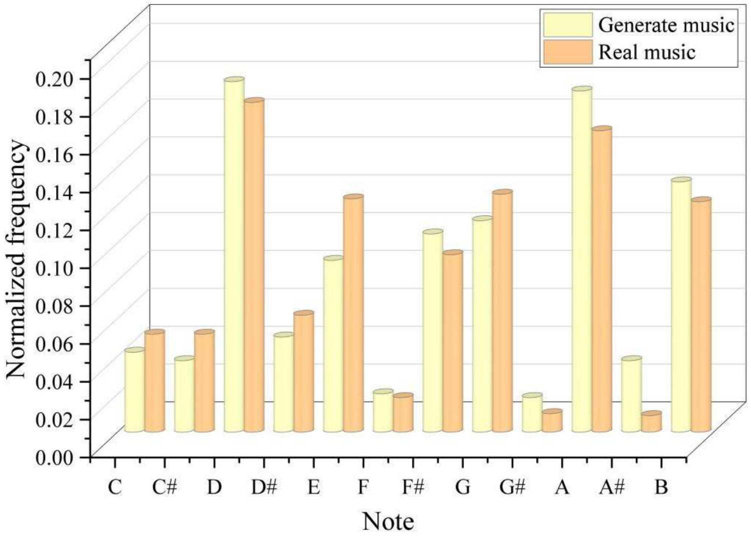


Figure 2: Comparison of note frequency distribution

III. B. 2) Comparison of objective indicators

In order to be able to more comprehensively verify the improvement of the present model in all aspects of enhancement, this paper uses six objective assessment metrics for comparison. Among them, the polyphony rate (PP) is defined and calculated as follows:

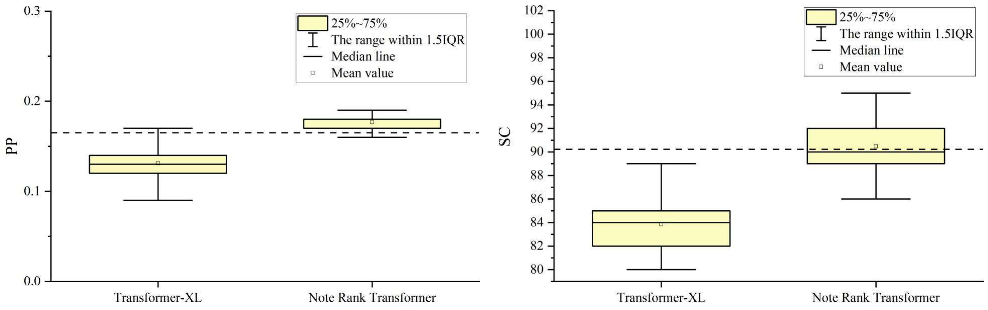
$$Polyphonicity = \sum_{i=0}^{T} \frac{P(n_i)}{T} \tag{17}$$

$$P(n_i) = \begin{cases} 1, if\ n_i \geq 2 \\ 0, if\ n_i < 2 \end{cases} \tag{18}$$

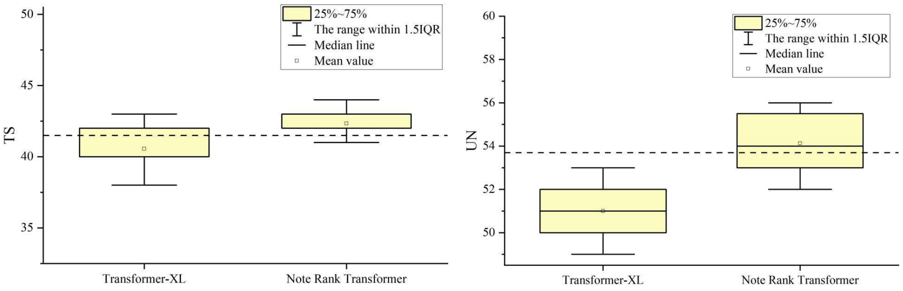
In Equation (17), T is the total number of time steps and n_i is the number of notes played simultaneously at the ith time step in the generated polyphonic music, which represents the ratio of the number of time steps in which the sample plays at least two or more pitches simultaneously to the total number of time steps. Scale Consistency (SC) is used to count the percentage of notes in the generated sample that can best match the standard scale. Tone Span (TS) counts the number of semitone steps between the lowest and highest notes in the sample. Note Uniqueness (UN) is used to count the percentage of all notes played uniquely once in all time steps. Repeatability (RP) is used to count the number of repetitions of two and more note combinations in music, where regular short sequences of repetitions over a long period of time reflect the unity of the thematic material and the smoothness of the melody. Quality notes (QN) counts the proportion of notes with a duration of more than 2 time steps in the music, and QN assesses whether the model is generating too many fragmented notes with shorter durations.

Using the above six evaluation indexes on the Note Rank Transformer model and Transformer-XL model to generate samples for experimentation, each model randomly generates 800 music samples of 15 bars in length and calculates the index value of each sample, and plots box plots and scatter plots for comparison. The comparison results of the six objective evaluation indexes of the generated samples are shown in Fig. 3(a)~(f), respectively, and the dotted line in the figure is the average value of the real samples.

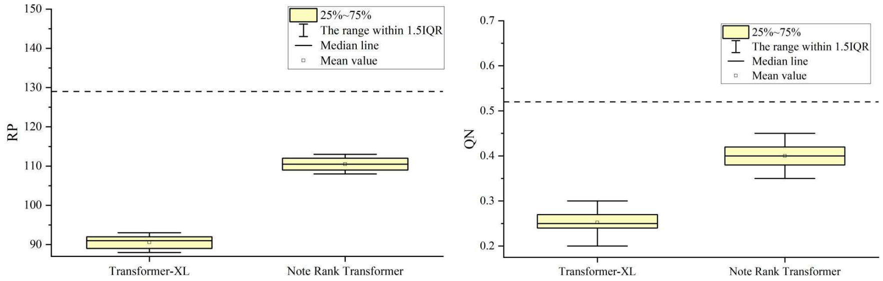
Most of the samples generated by the original Transformer-XL model are in the low range below the mean in terms of both polyphony rate and scale consistency, and the generation effect is the worst. On the other hand, the Note Rank Transformer model generates samples with a mean value of 90.46% for scale consistency, which is statistically the closest to real music (90.22%), indicating that the model learns the note distribution of real music well, and generates most of the notes in accordance with the natural scale. In the three indicators of polyphony rate, note span, and note uniqueness, most of the values of Note Rank Transformer generated samples are slightly higher than the average value of real samples, indicating that the generated samples of the model in this paper have higher diversity. As for repetition and high-quality notes, although all models fail to reach the standard of real samples, the Note Rank Transformer model shows better results, improving the repetition of short sequences of themes in the generated music and generating more notes with longer time values.



(a) PP (b) SC



(c) TS (d) UN

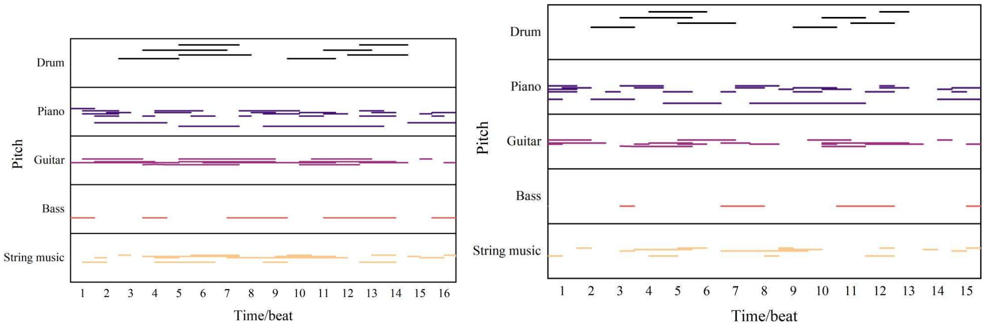


(e) RP (f) QN

Figure 3: Statistics of objective evaluation indicators

III. B. 3) Music generation results

The adjustment of the number of feature extractor layers is carried out according to the specific situation, and the comparison of each track before and after feature extraction is shown in Figure 4. After the adjustment, the key features in the data can be captured without being affected by the noise and details in the data, which helps to produce a more natural and realistic musical composition.



(a) Before feature extraction (b) After feature extraction

Figure 4: Comparison of each audio track before and after feature extraction

The generated results need to be digitized more finely, and in this paper, a set-value approach is used to process the resulting sample data. Specifically, any value in the array matrix that is greater than the value is considered a note and is labeled as a 1, indicating that it is active.9 Conversely, if a value is less than the threshold, then it is labeled as a 0, indicating that the note it represents is in a silent state.Examples of Note Rank Transformer generated samples are shown in Figure 5.

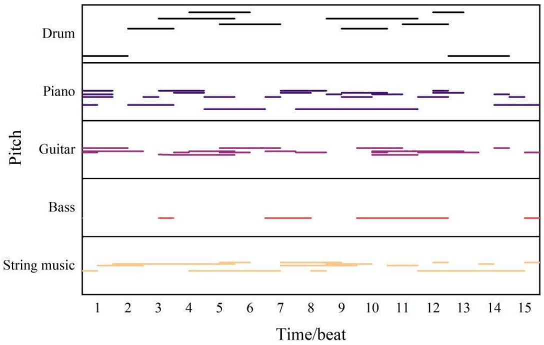


Figure 5: Generates sample examples

III. C. Model performance level assessment

In this paper, we make a comparison between the experiments of Note Rank Transformer and the experiments of Transformer-XL, so as to analyze the effect of improving the performance level of the improved Note Rank Transformer in this paper.

First of all, 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, so as to compare with the convergence curve of Transformer-XL, and the results of the comparison of the convergence curve of the discriminator loss of the two models are shown in Figure 6. From the two convergence curves of Transformer-XL and Note Rank Transformer in Fig. 6, it can be found that Note Rank Transformer converges faster and the training process is more stable, and it has already converged when the number of iterations is 529, whereas Transformer-XL starts to converge only when the number of iterations is 908.

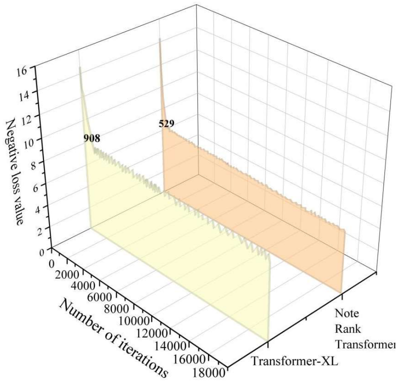


Figure 6: Comparison of convergence curves for discriminator loss

And, after the experiments, the distributions of the discriminator parameters in the iterative process of the two experiments can be obtained separately, and the results of the comparison of the parameter distributions are shown in Fig. 7. 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 Note Rank Transformer is (-0.2,0.2), which is obviously smaller than the interval range of the parameter distribution obtained from the experiment using Transformer-XL (-0.5.0.5), which indicates that the modified Note Rank Transformer model is less prone to overfitting.

To summarize, after experimental comparison, the Note Rank Transformer model is better than the Transformer-XL model in terms of stability, convergence speed, and overfitting.

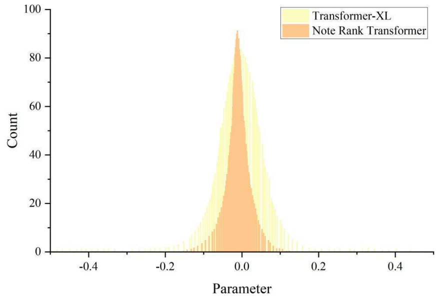


Figure 7: Comparison results of parameter distribution

The effect of the generator can be understood by comparing the values calculated from the real data and the generated data. The variation of the UPC metric value of the track where the stringed instrument is located during the experiment is shown in Fig. 8, where the blue straight line is the UPC metric value of the stringed instrument in the training set (i.e., real data). From the figure, it can be found that the model learns the metric at an iteration number of 1902, which further proves that the music generation method proposed in this paper is effective.

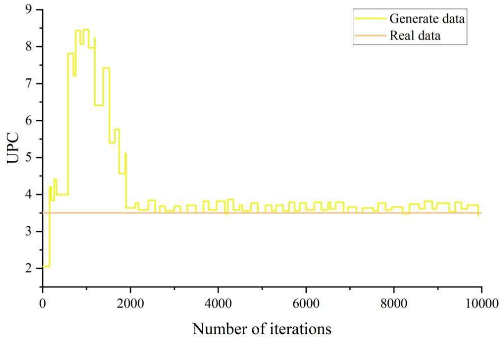


Figure 8: UPC index curve of stringed instruments

IV. Conclusion

In this paper, a music generation model based on Note Rank Transformer is designed, and its music generation effect and performance level are evaluated through experiments.

As for the music generation effect, the music data generated by the Note Rank Transformer model is similar to the note distribution of the training music dataset, which can effectively learn the note distribution law in the dataset. The average value of scale consistency of the generated samples is 90.46%, which is statistically the closest to the real music (90.22%), and most of the values of the generated samples are slightly higher than the average value of the real samples in the three indexes of polyphony rate, note span and note uniqueness, and for repetitive and high-quality notes, Note Rank Transformer shows better results than Transformer-XL. Note Rank Transformer shows better results than Transformer-XL for repetition and high quality notes.

Note Rank Transformer converges faster and the training process is more stable, converging at 529 iterations, while Transformer-XL starts converging at 908 iterations. The interval range of the parameter distribution obtained from the music generation experiments based on Note Rank Transformer is (-0.2,0.2), which is significantly smaller than that obtained from the experiments using Transformer-XL (-0.5.0.5), proving that the Note Rank Transformer model is more effective in terms of stability, convergence speed and overfitting are better than the Transformer-XL model.

References

[1] Frid, E., Gomes, C., & Jin, Z. (2020, April). Music creation by example. In Proceedings of the 2020 CHI conference on human factors in computing systems (pp. 1-13).

[2] Horger, D. (2023). Artificial Music: Authorship and Copyrightability of Artificial Intelligence Musical Works. AIPLA QJ, 51, 77

[3] Laidlow, R. (2024). Generative AI and music composition. In Music, Technology, Innovation (pp. 253-273). Routledge.

[4] Li, H. (2024, September). Application of artificial intelligence technology in AI music creation. In AIP Conference Proceedings (Vol. 3131, No. 1). AIP Publishing.

[5] Robert-Constantin, I., & Trăușan-Matu, S. (2023). A quantitative aesthetic analysis of artificial intelligence generated music. Proceedings or RoCHI, 63-8.

[6] Cai, L., & Cai, Q. (2019). Music creation and emotional recognition using neural network analysis. Journal of Ambient Intelligence and Humanized Computing, 1-10.

[7] Zhao, Y., Yang, M., Lin, Y., Zhang, X., Shi, F., Wang, Z., ... & Ning, H. (2025). AI-Enabled Text-to-Music Generation: A Comprehensive Review of Methods, Frameworks, and Future Directions. Electronics, 14(6).

[8] Boosa, S. (2019). AI and Creativity: How Writing, Music, and Art Are Increasingly Revolutionized by Machine Learning. EPH-International Journal of Science And Engineering, 5(4), 37-47.

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

[10] Zulić, H. (2019). How AI can change/improve/influence music composition, performance and education: three case studies. INSAM Journal of Contemporary Music, Art and Technology, (2), 100-114.

[11] Qiao, X. (2024). REDEFINING CREATIVITY: THE EFFECTS OF ARTIFICIAL INTELLIGENCE ON HUMAN MUSICAL INNOVATION. Journal of Dharma, 49(03).

[12] ŞUTEU, L. C. (2024). Artificial Intelligence in Music: The Digital Revolution of Sound Creativity. ICT in Muzical Field/Tehnologii Informatice si de Comunicatie in Domeniul Muzical, 15(2).

[13] Xu, L. (2024). A Study on the Fair Use Principles of Artificial Intelligence Generated Music. Lecture Notes in Education Psychology and Public Media, 34, 228-235.

[14] Babu, C. S., Karuppuswamy, S., Rijairaj, R., & Vignesh, A. (2025). Harnessing Artificial Intelligence in Music Creation: Exploring Genre Musical Features and Their Reflection in the Pedagogical Process. In Enhancing Music Education With Innovative Tools and Techniques (pp. 51-86). IGI Global Scientific Publishing.

[15] Seneadza, J. S., Boateng, S. L., Marfo, J. S., Boateng, R., & Budu, J. (2025). Transformative Impacts of Artificial Intelligence on the Music Industry: A Narrative Review. AI and the Music Industry, 34-58.

[16] Koempel, F. (2020). From the gut? Questions on Artificial Intelligence and music. Queen Mary Journal of Intellectual Property, 10(4), 503-513.

[17] Verma, S. (2021). Artificial intelligence and music: History and the future perceptive. International Journal of Applied Research, 7(2), 272-275.

[18] Fang, X., & Wei, G. (2024). Research on entertainment creation robot based on artificial intelligence speech recognition in the process of music style analysis. Entertainment Computing, 51, 100739.

[19] Zhou, X. (2023). Analysis of Evaluation in Artificial Intelligence Music. Journal of Artificial Intelligence Practice, 6(8), 6-11.

[20] Bonjack, S., & Trujillo, N. (2024). Artificial intelligence and music discovery. Music Reference Services Quarterly, 27(1), 1-9.

[21] Ma, H., Zhang, Y., Shan, X., & Hu, X. (2025). Exploring the Impact of Artificial Intelligence on the Creativity Perception of Music Practitioners. Journal of Intelligence, 13(4), 47.

[22] Kumar, L., Goyal, P., & Kumar, R. (2020). Creativity in machines: music composition using artificial intelligence. Asian Journal For Convergence In Technology (AJCT) ISSN-2350-1146, 6(2), 36-40.

[23] Siphocly, N. N. J., El-Horbaty, E. S. M., & Salem, A. B. M. (2021). Top 10 artificial intelligence algorithms in computer music composition. International Journal of Computing and Digital Systems, 10(01), 373-394.

[24] Dahlstedt, P. (2021). Musicking with algorithms: Thoughts on artificial intelligence, creativity, and agency. Handbook of artificial intelligence for music: Foundations, advanced approaches, and developments for creativity, 873-914.