Multi-Track Music Generation Based on the AC Algorithm and Global Value Return Network

Wei Guo
College of Music and Dance, Huaqiao University, Xiamen, 361021, China

***Abstract*—In the current field of deep learning and music information retrieval, automated music generation has become a hot research topic. This study addresses the issues of low clarity and musicality in current multi-track music generation by combining the Actor-Critic algorithm and the Global Value Return Network to create a novel multi-track music generation model. The study first utilizes the Actor-Critic algorithm to generate single-track music rhythm and melody models. Building upon this foundation, the study further optimizes the single-track models using the Global Value Return Network and proposes the multi-track music model. The results demonstrate that the harmonization accuracy of the final multi-track music generation model ranges from 0.90 to 0.98, with a maximum value of 0.98. Additionally, the audience satisfaction and expert satisfaction of the model are 0.96 and 0.97, respectively, indicating that the model has a high musical appreciation value. Overall, the multi-track music generation model designed in this study addresses the limitations of single-track music generation and produces more rhythmically diverse multi-track music.**

***Keywords*—AC; global value; return network; track; music model; rhythm; melody**

I. Introduction

With the advancement of artificial intelligence technology, particularly in the field of reinforcement learning, researchers are exploring the use of advanced algorithms to simulate and reproduce complex music composition processes [1-2]. Multi-track music generation involves simultaneously creating melodies and harmonies for multiple instruments, making it a particularly challenging research direction. It requires not only considering the melody generation for individual tracks but also coordinating and synchronizing across tracks. Although existing researches have achieved certain results in the field of single-track music generation, the field of multi-track music generation remains an urgent problem to be solved in terms of how to effectively coordinate the generation process of each track, and how to comprehensively consider the global music structure and the long-term value return in the composition [3-4]. Among the many attempts, nature-inspired algorithms have received particular attention due to their effectiveness in optimisation and search problems. For example, Genetic Algorithms and Particle Swarm Optimisation algorithms have been used as new ways of exploring music composition by simulating natural selection or the flight behaviour of flocks of birds to generate harmonious melodies. These algorithms show potential for generating melodies by iteratively searching the solution space, especially in terms of following the rules of a particular music theory and composing simple melodies. However, despite the progress made by nature-inspired algorithms in music generation, they face a number of challenges when dealing with complex music composition tasks [5-6]. Firstly, these algorithms often rely on predefined rules or objective functions which limit their application in creative music composition, which not only has to follow theoretical rules but also has to be artistic and emotionally expressive. Secondly, nature-inspired algorithms do not perform well in terms of global structure and long-term value maximisation which is particularly important in multi-track music generation, as it requires both harmony between different tracks and overall expression of a unified musical style and emotion. Facing these challenges, this study utilizes the Actor-Critic (AC) algorithm from reinforcement learning and establishes a global value return network to capture the long-term value of music and ensure that the generated music has high quality and artistry in terms of global structure. This research is divided into six sections, Section I is a brief introduction to the full text, Section II is a review of the related literature, Section III is the construction of the mono-track multi-track model, and Section IV is the testing of the model performance. Discussion and conclusion is given in Section V and Section VI respectively.

II. Related Works

AC is a reinforcement learning technique that combines value functions with policy gradients. The advantage of this algorithm is that it combines the strengths of value functions and policy gradients, allowing for effective handling of continuous action spaces and complex policy problems. Many researchers have conducted studies on the application of AC algorithms. Zare et al. employed asynchronous advantage AC to address the service placement problem in fog computing environments. The paper proposed placing services in the local fog domain and leveraging neighboring fog domains when necessary to improve resource utilization. Additionally, a time-distributed resource allocation technique was considered to handle future requests more effectively. Simulation results demonstrated that this mechanism significantly improved cost efficiency and response speed compared to other methods [7]. Scorsoglio et al. proposed a feedback-guided algorithm for near-ground lunar operations based on AC reinforcement learning. The algorithm had the advantages of being lightweight, closed-loop, and capable of considering path constraints. Test results showed excellent performance of the designed algorithm in path constraint problems across various restrictive scenarios [8]. To address the limited data storage capacity of Earth observation satellites in dense observation scenarios, Wen et al. proposed a time-continuous model that jointly considered data transmission and observation tasks. To handle this problem more efficiently, a hybrid AC reinforcement learning approach was employed in the paper. Experimental results showed that this hybrid approach exhibited high efficiency and good performance in solving large-scale problems, which was of practical significance for the data management and scheduling of Earth observation satellites [9].

To create works that are both musically sound and emotionally impactful, and further explore the possibilities of artificial intelligence in artistic creation, many experts have built a series of multi-track music generation models using various deep learning techniques. Liu researched and developed an improved multi-track music generative adversarial network model, which was validated by generating five different instrument tracks. The research results showed that the music snippets generated by the proposed model had better artistic aesthetics. In the end, 62.8% of the listeners had difficulty distinguishing between the generated melodies and real melodies, demonstrating the high authenticity and effectiveness of the model in music generation [10]. Wang et al. proposed a Transformer-based multi-track music generative adversarial network, aimed at adhering to music rules to generate works with higher musicality. The model utilized the Transformer decoding component and a cross-track Transformer improved based on Transformer to separately learn information between single tracks and multiple tracks. The training of the generative network was guided by combining music rules and cross-entropy loss, and a well-designed target loss function was optimized when training the discriminative network. Experimental results demonstrated that the constructed model, on piano, guitar, and bass tracks, exhibited higher track prediction accuracy compared to other multi-instrument music generation models, effectively enhancing the overall quality of music [11]. In the face of the challenge of integrating independent melodies in polyphonic music composition, Huang et al. proposed an innovative multi-voice music composition model. That model integrated the concepts of Markov decision processes and Monte Carlo tree search and improved upon Wasserstein generative adversarial network theory. Through the zero-sum game and conditional constraints between the generator and discriminator, the model achieved music creation closer to unconstrained conditions, and the growth of music sequences did not affect their coherence. Experimental results indicated that the algorithm proposed in the study outperformed the latest methods in multi-voice music generation, demonstrating significant advantages [12].

Nature-inspired algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), etc., are a class of optimisation algorithms that draw on phenomena of nature, such as biological evolution, flight of birds, ant colony ant colony foraging, etc [13-14]. These algorithms usually mimic certain processes in nature to solve optimisation problems, and are particularly good at dealing with large-scale and complex search space problems. Some scholars have also conducted a series of music-related studies using nature-inspired algorithms. Majidi M and Toroghi R M propose a method for generating polyphonic music works based on multi-objective genetic algorithms. This method takes into account both the accuracy of music theory and the satisfaction of both expert and general listeners. The results show that the method is able to produce pleasing pieces that meet the desired style and length, and follow grammatical rules to produce harmonies [15]. Tian R et al. proposed a music emotion classification model combining convolutional neural networks and random forests. The model first converts audio data into Mel spectrum for feature extraction, then uses random forest algorithm for initial emotion classification, and finally achieves 97% accuracy in emotion classification, which is 1.2% and 1.6% higher than that of traditional particle swarm optimization and genetic algorithm [16]. Cao H proposed a system architecture based on the combination of edge computing and cloud computing to optimize the scheduling strategy of music education resources. Compared with the traditional genetic algorithm and ant colony algorithm, this method can improve the system efficiency by 23% [17].

In summary, despite the progress of AC algorithms in various fields, their application in multi-track music generation is still in its early stages. Existing music generation models need improvement in creativity and track harmony. Against this background, this study proposes a multi-track music generation model that combines AC algorithms with a global value return network. The aim is to address this challenge and experimentally verify its effectiveness in improving the quality of music generation.

III. Multi-track Music Generation Combining AC Algorithm and Global Value-based Network

In order to address the rhythm generation, melody generation, and multi-track generation issues in the current music generation problem, this research first utilizes the AC algorithm to construct separate models for melody generation and rhythm generation. Based on this, a global value-based network is designed to integrate multiple agents, resulting in the development of a multi-track music generation model.

A. *Construction of Music Rhythm and Melody Generation Models based on AC Algorithm*

In the field of reinforcement learning, let's assume that the note sequence and rhythm sequence in the music generation problem are represented as $S_n = \{n_1, n_2, \cdots n_L\}$ and $S_r = \{r_1, r_2, \cdots r_L\}$, respectively. L L represents the sequence length. $\{n_1, n_2, \cdots n_L\}$ and $\{r_1, r_2, \cdots r_L\}$ represent various types of notes and rhythm in the note sequence and rhythm sequence, respectively. The specific representation of note types is shown in Eq. (1) [18].

$$n_i = \{s_1, s_2, \cdots, s_N\} \quad i = 1, 2, \cdots L \tag{1}$$

In Eq. (1), n_i represents a specific note type, and N represents the number of notes included in that note type. When N = 1, it indicates a monophonic output, and when N > 1, it indicates a polyphonic output. By encoding the note sequence and rhythm sequence, the encoded input data sequences in Eq. (2) are obtained.

$$\begin{cases} S_n^E = Encoder(S_n) = MultiHot(S_n) = \{n_1^{mh}, n_2^{mh}, \cdots n_L^{mh}\} \\ S_r^E = Encoder(S_r) = OneHot(S_r) = \{r_1^{oh}, r_2^{oh}, \cdots r_L^{oh}\} \end{cases} \quad (2)$$

In Eq. (2), S_n^E and S_r^E represent the note sequence and rhythm sequence after $MultiHot$ encoding and $OneHot$ encoding, respectively. L represents the length of the sequence. $\{n_1^{mh}, n_2^{mh}, \cdots n_L^{mh}\}$ and $\{r_1^{oh}, r_2^{oh}, \cdots r_L^{oh}\}$ represent the encoded note type and rhythm type, respectively. By inputting the encoded note sequence and rhythm sequence into the model, we can obtain the output note sequence and rhythm sequence, as shown in Eq. (3).

$$\begin{cases} S_n^g = \{n_1^g, n_2^g, \cdots n_L^g\} \\ S_r^g = \{r_1^g, r_2^g, \cdots r_L^g\} \end{cases} \quad (3)$$

In Eq. (3), S_n^g and S_r^g represent the output note sequence and rhythm sequence, respectively. $\{n_1^g, n_2^g, \cdots n_L^g\}$ and $\{r_1^g, r_2^g, \cdots r_L^g\}$ represent the output note type and rhythm type, respectively. Based on Eq. (1) to (3), a complete music melody can be obtained, as shown in Eq. (4).

$$S_m^g = \{\{r_1^g, n_1^g\}, \{r_2^g, n_2^g\}, \cdots, \{r_L^g, n_L^g\}\} \quad (4)$$

In Eq. (4), S_m^g represents the complete music melody. Based on the definitions of music concepts in Eq. (1) to (4), this research combines the AC algorithm to construct the music rhythm and melody generation model, referred to as the Actor-Critic Melodic Rhythm Generation Model (ACMRGM). The specific framework structure of ACMRGM is shown in Fig. 1.

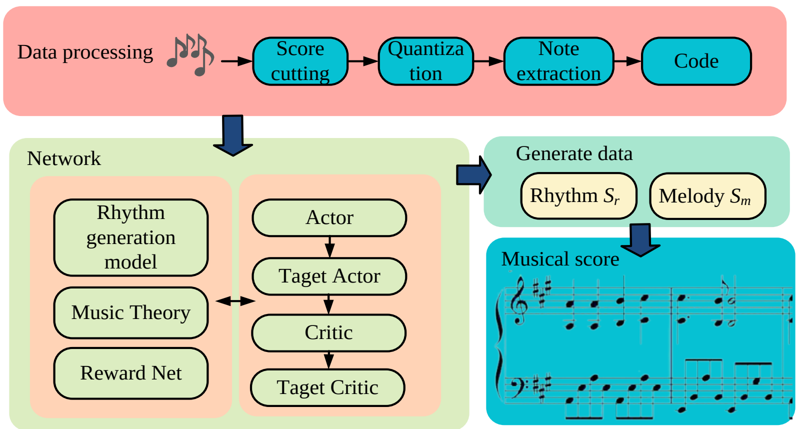


Fig. 1. ACMRGM model structure diagram.

In Fig. 1, the constructed music rhythm and melody generation model consists of four main parts: data processing, network construction, data generation, and sheet music output. The data processing part converts the initial music score data into a format suitable for inputting into the model and can also convert the model output back to a music score file. In the rhythm network, assuming that the (Long and Short-Term Memory) LSTM network's hidden state and cell state are represented as h and c, respectively, and the output is O_{lstm}, the calculation formula for obtaining the output is shown in Eq. (5).

$$O_{lstm} = h_t^2 \quad (5)$$

In Eq. (5), h_t^2 represents the second-layer output of the rhythm network. The loss function calculation formula for the rhythm network is shown in Eq. (6).

$$loss_1 = soft\max\_cross\_entropy(O_{linear}) \quad (6)$$

In Eq. (6), O_{linear} represents the linear transformation tensor of O_{lstm} after passing through the Linear layer. $soft\max$ represents the activation function. $cross\_entropy$ represents cross-entropy. $loss_1$ represents the loss value of the rhythm network. The operational flowchart of the rhythm network model is shown in Fig. 2.

In Fig. 2, the trained model parameters are first read to initialize the rhythm network model. Then, an initial rhythm sequence is given as the initial duration data, and the length of the generated rhythm sequence is initialized. Next, the initial notes pass through the LSTM rhythm network to compute the network's output, O_{lstm}. The next step involves applying a linear transformation to O_{lstm} to obtain O_{linear}. O_{linear} then goes through an activation function and cross-entropy calculation to obtain the probability distribution values. Finally, the corresponding rhythm duration is randomly selected based on the calculated probability distribution values. The formula for rhythm generation is shown in Eq. (7) [19].

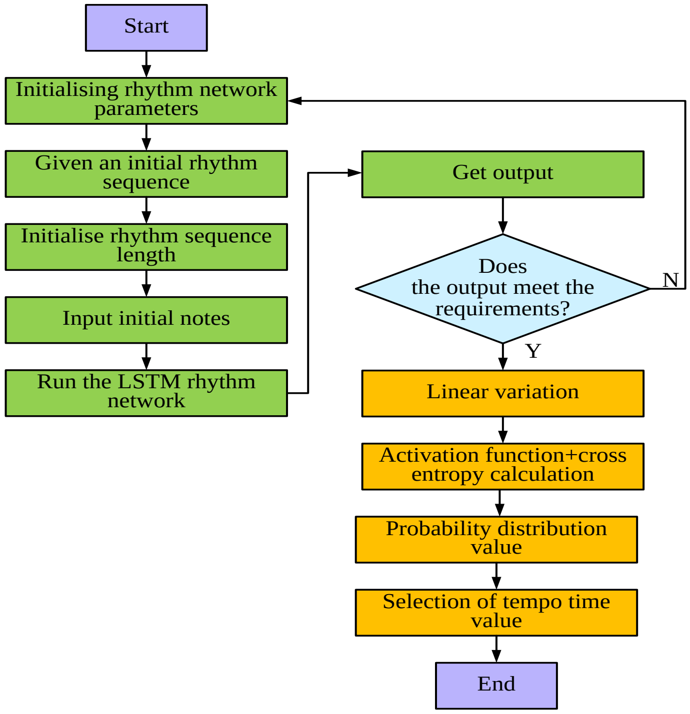


Fig. 2. Operation flow chart of the rhythm network model.

$$O_{linear} = O_{lstm} * w^T + b \tag{7}$$

w^T in Eq. (7) represents the weight matrix of the output gate in the LSTM network, while b represents the bias vector.

In the melody network, it is recognized that there is no direct mechanism for generating reward values in the music generation environment. Therefore, training an LSTM network is proposed to form a reward network and obtain the corresponding reward values. Compared to the rhythm network, the reward network adds an attention mechanism module, which further enhances the ACMRGM model's ability to learn important notes. Additionally, the activation function $soft\max$ is replaced with $sigmoid$ in the reward network to support the generation of polyphonic melodies. The formula for the loss function of the reward network is shown in Eq. (8).

$$loss_2 = sigmoid\_cross\_entropy(O_{linear}) \tag{8}$$

In Eq. (8), $loss_2$ represents the loss value of the reward network. Once the reward network is designed, an melody network model is built by combining the LSTM network with the Actor and Ctiric networks from the AC algorithm. Both the Actor and reward networks consist of LSTM, attention mechanism module, Linear layer, and Sigmoid module, while the Ctiric network consists of LSTM, attention mechanism module, and two Linear layers. Assuming the reward value based on music theory rules is r_m and the reward value obtained by the reward network is r_n, the formula for calculating the reward value of the ACMRGM model is shown in Eq. (9).

$$r_{mix} = k_m * r_m + k_n * r_n \tag{9}$$

In Eq. (9), r_{mix} represents the final reward value of the ACMRGM model. k_m and k_n represent the proportions of r_m and r_n, respectively. The workflow diagram of the melody network model is shown in Fig. 3.

In Fig. 3, the note parameters and melody length are initialized first. The initialized note parameters are input into the Actor network to obtain the probability distribution values of the next action. Then, an action is randomly selected based on the probability distribution values. The selected action is then transformed into the next state, which is also input into the Actor network to obtain the next action. The action transformation is repeated an equal number of times as the melody length, ultimately generating the corresponding note sequence. This note sequence is then combined with the rhythm sequence to obtain the complete musical composition.

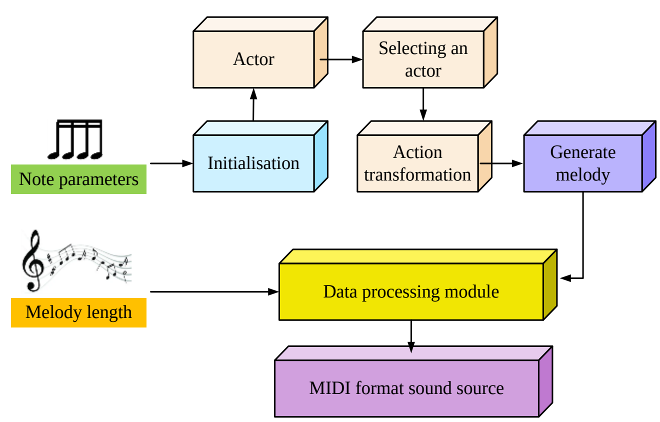


Fig. 3. Operation flow chart of the melody network model.

B. Construction of a Multi-AC Melodic Rhythm Generation Model by Integrating AC Algorithm and Global Value-Return Networks

To generate polyphonic music with coordinated consistency, this study extends the single Actor and Critic modules from Section II by increasing their quantity to handle multiple musical tracks. Additionally, to ensure coordination among different tracks, a centralized Global Reward Network is constructed. This network imposes constraints on the note relationships between different tracks, ensuring overall harmony and consistency [15]. The resulting multi-track music generation model is referred to as the Multi-Actor-Critic Melodic Rhythm Generation Model (MACMRGM), as illustrated in Fig. 4.

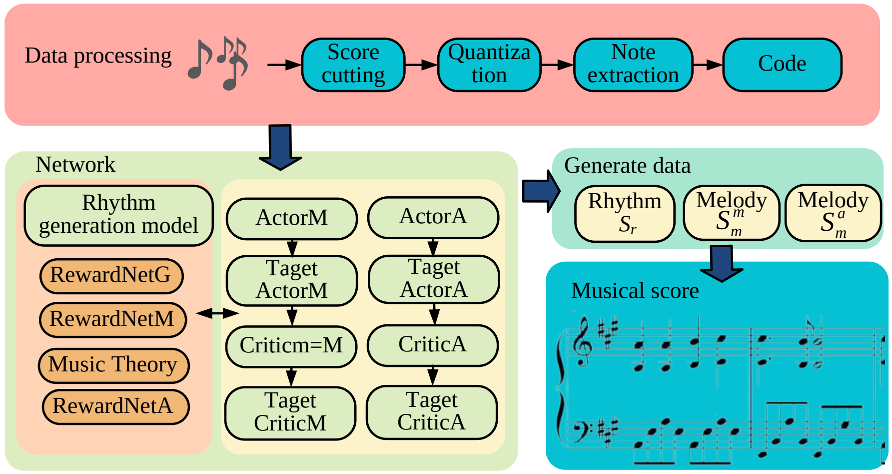


Fig. 4. MACMRGM model structure diagram.

In Fig. 4, the MACMRGM model is primarily divided into four parts: Data Processing, Network Model, Music Generation, and Score Output. ActorM, CriticM, and their corresponding target networks are responsible for generating the main melody track, while ActorA, CriticA, and their target networks handle the accompaniment track. RewardNetM and RewardNetA train on the main melody and accompaniment tracks, respectively, while RewardNetG, as the global reward network, focuses on training the processed track data from the data processing module, aiming to ensure coordination between tracks. Additionally, the music theory reward module and rhythm generation model further enhance the theoretical accuracy and rhythmic sense of the music. The combination of the output rhythms and melodies from the network model section yields the final output score. When processing multi-track music data, the workflow is slightly more complex compared to single-track music. The processing flow for multi-track music data is depicted in Fig. 5.

In Fig. 5, the processing flow for multi-track music includes steps such as inputting audio data sets, dividing audio data sets, cutting scores, quantizing, transposing, extracting notes, encoding, and outputting audio. Firstly, multiple tracks from the score are extracted and divided into audio training and testing sets. Next, the divided dataset is segmented into smaller sections. If there are changes in tempo within a score, the score is cut at those points. The segmented music sections are stored as TFRecord-format files, and the music segments in these files undergo quantization. After quantization, the transposition module is applied. Following the key conversion, the main melody, accompaniment track, and synthesized track of the score are extracted. These track data are then encoded into a multi-hot format and stored as TFRecord-format data for training purposes. The synthesized track combines synchronized notes from the main melody and accompaniment tracks to form harmony. All tracks are combined with the rhythm sequence to generate a complete score, which is then converted into a MIDI file format.

During the training of the MACMRGM model, the first step involves pre-training three reward networks in the model [20-21]. RewardNetM and RewardNetA are trained using the main melody track and accompaniment track, respectively, while RewardNetG is trained using a synthetic track. The one-dimensional array calculation formulas for RewardNetM, RewardNetA, and RewardNetG are given by Eq. (10).

$$\begin{cases} O_{linear}^{m} = O_{lstm}^{m} * \left(w^{m}\right)^{T} + b^{m} \\ O_{linear}^{a} = O_{lstm}^{a} * \left(w^{a}\right)^{T} + b^{a} \\ O_{linear}^{g} = O_{lstm}^{g} * \left(w^{g}\right)^{T} + b^{g} \end{cases} \tag{10}$$

In Eq. (10), O_{linear}^{m}, O_{linear}^{a}, and O_{linear}^{g} represent the one-dimensional arrays of RewardNetM, RewardNetA, and RewardNetG, respectively. O_{lstm}^{m}, O_{lstm}^{a}, O_{lstm}^{g} denote the output values of the LSTM networks in the three reward networks. $\left(w^{m}\right)^{T}$, $\left(w^{a}\right)^{T}$, $\left(w^{g}\right)^{T}$ represent three weight matrices, and b^{m}, b^{a}, b^{g} represent three bias vectors. In each of the three reward networks, the calculation process for extracting action values from the reward value array is shown in Eq. (11) [22-23].

$$\begin{cases} R_{n}^{m} = O_{linear}^{m}\left[a^{m}\right] \\ R_{n}^{a} = O_{linear}^{a}\left[a^{a}\right] \\ R_{n}^{g} = O_{linear}^{g}\left[a^{g}\right] \end{cases} \tag{11}$$

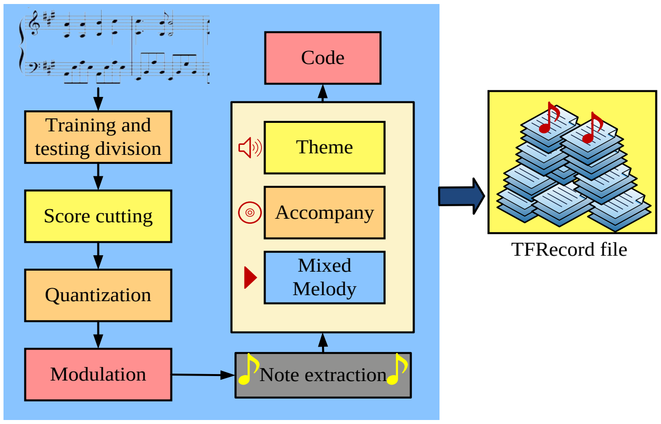


Fig. 5. Multi-track music processing flow chart.

In Eq. (11), a^m, a^a, a^g represent the predicted actions of RewardNetM, RewardNetA, and RewardNetG, respectively. R_n^m, R_n^a, R_n^g represent the reward value arrays of RewardNetM, RewardNetA, and RewardNetG. The final calculation formula for the MACMRGM model's overall reward value is presented in Eq. (12).

$$r'_{mix} = k_1 * r^m + k_2 * r^a + k_3 * r^g \tag{12}$$

In Eq. (12), r'_{mix} represents the model's ultimate reward value. r^m, r^a, r^g represent the reward values of RewardNetM, RewardNetA, and RewardNetG, respectively. k_1, k_2, k_3 denote the proportions of the reward values for the three networks. After training the reward networks, the process involves combining other modules [24-25]. In the MACMRGM model, the network structures of ActorM and ActorA are consistent with the reward networks, composed of LSTM, attention mechanism module, Linear layer, and Sigmoid module. The structures of CriticM and CriticA continue to consist of LSTM, attention mechanism module, and two Linear layers. The training of Actor and Critic networks is carried out in an alternating manner, where the networks are trained every certain number of steps until the specified step limit is reached. The final multi-track music generation process is illustrated in Fig. 6.

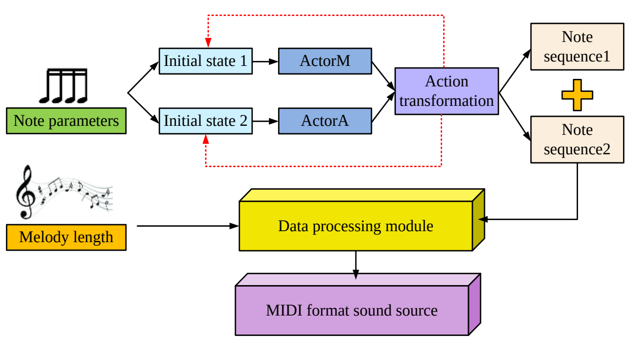


Fig. 6. Multi-track music generation flow chart

In Fig. 6, the initialization of note 1 and note 2 in the model, along with the configuration of the melody length, is the initial step. These notes are set as the initial states 1 and 2. Initial states 1 and 2 are input into ActorM and ActorA to obtain probability distribution values for the next actions. Actions 1 and 2 are then randomly selected based on the probability distribution values, converted into states 1 and 2, and input into ActorM and ActorA to obtain the next actions. This process continues until the model performs actions updates equal to the length of the melody, resulting in the generated note sequences 1 and 2. Finally, the two obtained note sequences are combined with the rhythm sequence to output a complete multi-track score.

IV. Performance Testing and Application Analysis of Different Track Music Generation Models Based on AC Algorithm

To demonstrate the performance of the single-track music rhythm and melody generation model ACMRGM and the multi-track music generation model MACMRGM, a comparative experiment was conducted using the publicly available dataset MAESTRO. The final research results indicate that ACMRGM has better music melody and rhythm compared to traditional LSTM, Transformer, and Generative Adversarial Network (GAN). MACMRGM can generate multi-track music with better listening experience compared to Bi-Long Short-Term Memory (Bi-LSTM), Bidirectional Encoder Representations from Transformers (BERT), and Deep Convolutional Generative Adversarial Network (DCGAN).

A. *Performance Testing and Application Analysis of Single-Track Music Generation Model*

The MAESTRO dataset is a high-quality music performance dataset provided by Google's Magenta project. The dataset consists of approximately 2000 different types of music performances, with all performances stored in MIDI format scores and corresponding audio forms. The selected 2000 music performances were divided into training and testing sets in an 8:2 ratio. Since the music types in this dataset cover a wide range of styles from classical to modern and include both single-track and multi-track performances, it is suitable for various music-related machine learning research projects. To ensure the consistency of note durations, the tempo of the scores was set to 120 BPM. In order to ensure the uniqueness of the research results, all experiments were conducted on the same computer device. The experimental setup and initial network parameters are shown in Table I.

TABLE I. Experimental Environment and Network Parameter Configuration Table

Experimental equipment	Value
CPU	Intel Core i9-10900K
GPU	NVIDIA GeForce RTX 3080
Memory	11GB
Operating system	Ubuntu 20.04 LTS
Python version	Python 3.8
Deep learning framework	TensorFlow 2.4 and PyTorch 1.7
Network training optimizer	Adam
Batch size	32
Epochs	5000
learning rate	0.001

Table I provides the environmental settings and initial network parameter values for this experiment. In order to evaluate the performance of the single-track music generation model, this study selected two metrics, Melodic Harmony (MH) and Music Clarity (MC), for testing. Fig. 7 compares the MH values of the LSTM, Transformer, GAN, and ACMRGM models on the training and testing sets.

In Fig. 7, the MH values of four models, namely LSTM, Transformer, GAN, and ACMRGM, are presented in both the training and testing sets. As indicated in Fig. 7(a), when testing with any randomly selected five monophonic sources from the training set, the maximum MH values for LSTM, Transformer, GAN, and ACMRGM models were 0.85, 0.85, 0.93, and 0.98, respectively. Fig. 7(b) shows that when testing with any randomly selected five monophonic sources from the testing set, the maximum MH values for LSTM, Transformer, GAN, and ACMRGM were 0.83, 0.87, 0.93, and 0.99, respectively. Overall, based on Fig. 7, it can be observed that ACMRGM model exhibits better stability, while the MH values for the other three models fluctuate across different monophonic sources, indicating the higher stability of ACMRGM model.

In Fig. 8(a), 8(b), 8(c), and 8(d), the MC values for different monophonic music generation models are displayed. Utilizing 25 monophonic sources from the dataset as a baseline reference for standard pitch, it is evident from Fig. 8(a), 8(b), 8(c), and 8(d) that, except for the MC values generated by the ACMRGM model, which align with the baseline, the MC values generated by LSTM, Transformer, and GAN models exhibit significant deviations from the baseline. The clarity performance of the four models is ranked with ACMRGM model being the best, followed by Transformer model, and LSTM and GAN models showing comparatively poorer performance.

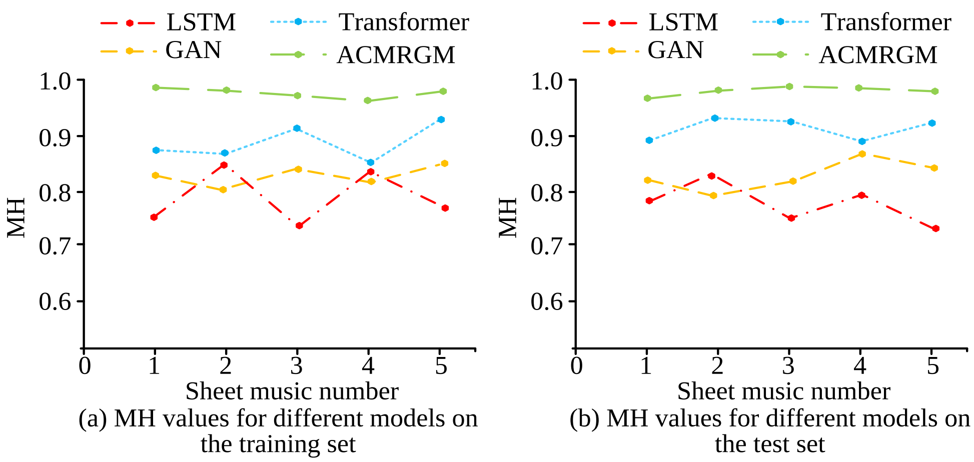


Fig. 7. MH values of different single-track music generation models.

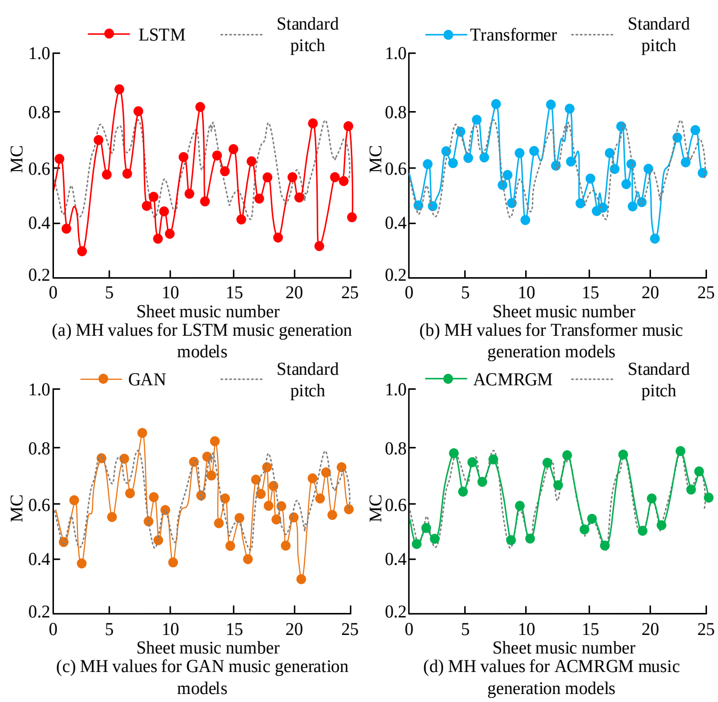
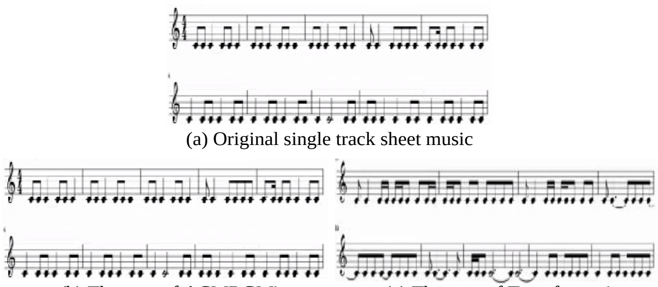


Fig. 8. MC values of different single-track music generation models.

Fig. 9 compares the performance of ACMRGM model and Transformer model in practical monophonic music generation problems. Choosing a segment of the original monophonic musical score as the reference source, as shown in Fig. 9(a), the monophonic generated musical scores by ACMRGM model and Transformer model are depicted in Fig. 9(b) and 9(c), respectively. Combining the information from Fig. 9 reveals that the musical score generated by ACMRGM model closely aligns with the original score, while the musical score generated by the Transformer model exhibits some differences from the original score.



(a) Original single track sheet music

(b) The case of ACMRGM's single-track score generation

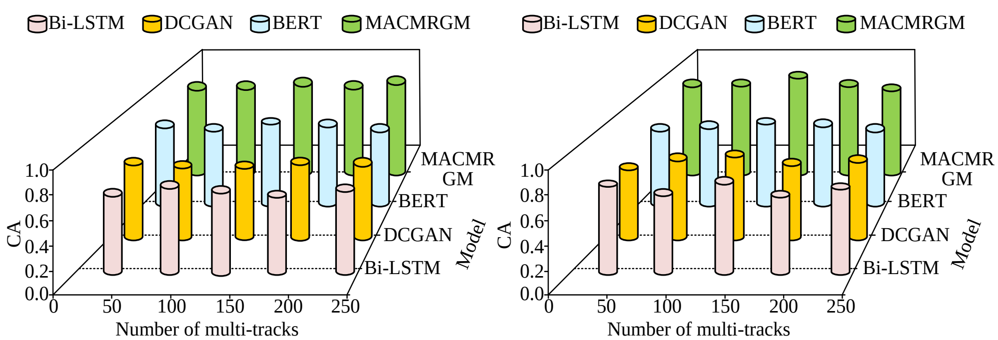
(c) The case of Transformer's single-track score generation

Fig. 9. Single-track music score generation using different single-track music generation models.

B. *Performance Testing and Application Effect Analysis of Polyphonic Music Generation Models*

In addition to testing the performance of single-track music generation models, this study also conducted an analysis of the performance and application effects of polyphonic music generation models. Chorus Accuracy (CA) and Subjective Listening Satisfaction (SLS) were chosen as evaluation metrics. The CA values of four polyphonic music generation models—Bi-LSTM, DCGAN, BERT, and MACMRGM—were obtained as shown in Fig. 10.

Fig. 10(a) and Fig. 10(b) represent the CA values of different polyphonic music generation models in the training set and the test set, respectively. From Fig. 10(a), it is observed that as the training set size increases from 50 to 250, the CA values of the four models vary within the ranges of 0.72 to 0.83 (Bi-LSTM), 0.78 to 0.88 (DCGAN), 0.81 to 0.90 (BERT), and 0.90 to 0.98 (MACMRGM). Fig. 10(b) shows that with changes in the test set size, the CA values for Bi-LSTM, DCGAN, BERT, and MACMRGM range from 0.73 to 0.82, 0.79 to 0.86, 0.82 to 0.89, and 0.92 to 0.98, respectively.



(a) CA values of different multi-track generation models in the training set

(b) CA values for different multitrack generation models in the test set

Fig. 10. CA values of different multi-track music generation models.

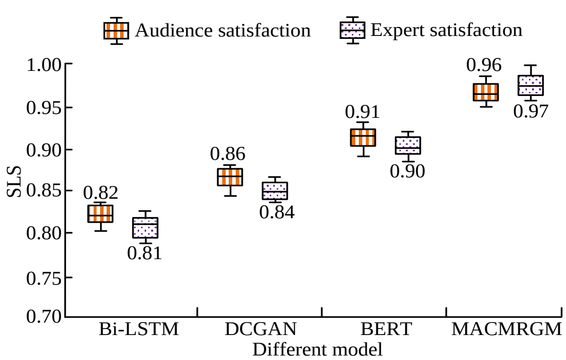
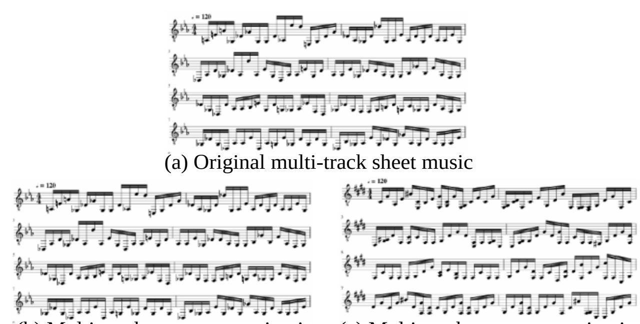


Fig. 11. SLS of different multi-track music generation models.

Fig. 11 illustrates the satisfaction values of both listeners and experts for the four polyphonic music generation models, represented by the SLS metric. Assuming scores from 0 to 1 indicate dissatisfaction to satisfaction, it can be inferred from Fig. 11 that listeners gave SLS scores of 0.82, 0.86, 0.91, and 0.96 for Bi-LSTM, DCGAN, BERT, and MACMRGM, respectively. Experts' SLS scores were 0.81, 0.84, 0.90, and 0.97 for Bi-LSTM, DCGAN, BERT, and MACMRGM, respectively. In conclusion, the MACMRGM model achieved higher satisfaction from both listeners and experts, indicating that the music it generated is more enjoyable.



(a) Original multi-track sheet music

(b) Multi-track score generation in ACMRGM

(c) Multi-track score generation in BERT

Fig. 12. Multi-track music score generation using different multi-track music generation models.

Fig. 12(a), 12(b), and 12(c) respectively depict an original polyphonic music score, a polyphonic music score generated by the MACMRGM model, and a polyphonic music score generated by the BERT model. By comparing these figures, it can be observed that the MACMRGM model is capable of faithfully reproducing the multi-track music template, whereas the BERT model may exhibit variations in rhythm and melody, deviating from the original music.

V. Discussion

The multi-track music generation model combining the Actor-Critic algorithm and the Global Value Return Network proposed in this research aims to solve the problems of insufficient track coordination and global music structure optimisation in multi-track music generation. By introducing the Actor-Critic algorithm, this study first builds a separate music rhythm generation model and a melody generation model, which is notated as ACMRGM. based on this, the single Actor and Critic modules are extended to increase the number of the two modules to deal with multiple tracks, and then the constraints are imposed on the note relationships among different tracks by combining with the global value-returns network, which ensures the In MACMRGM, the Actor-Critic algorithm enables the constructed multi-track generation model MACMRGM to effectively balance the contradiction between exploration and exploitation, while the global value return network helps MACMRGM to capture the long term value and global structure of the music to achieve the best results in terms of harmony, accuracy and listener satisfaction. Accuracy and listener satisfaction is important to achieve significant improvements. The MH and MC values were selected as performance test metrics and performance comparisons were made with other models. The results show that ACMRGM has better performance. Compared with the existing literature, the models in this study not only achieved significant improvements in technical performance, but also demonstrated advantages in musical artistry and listener acceptance. For example, although the model based on generative adversarial networks proposed by Liu et al. has made progress in terms of diversity and novelty of music generation, it is still deficient in terms of harmonic accuracy and coherence of music structure. The model in this study effectively overcomes these limitations by integrating global musical structure and long-term value returns, providing a new approach to generating multi-track music that is both richly diverse and harmonically coherent.

VI. Conclusion

To ensure that the melodies and rhythms in polyphonic music generation models harmonize effectively, thereby creating music compositions of greater aesthetic value, this research integrated the AC algorithm with a Global Value Return Network to develop a novel polyphonic music generation model, MACMRGM. Initially, the performance of single-track music generation models was assessed. The findings indicated that the highest MH value achieved by the single-track music generation model, ACMRGM, was 0.99. Furthermore, the music generated by this model closely aligned with the pitch accuracy of the baseline audio source, thereby confirming its capability to produce commendable musical rhythms and melodies. In the evaluation of polyphonic music generation models, the maximum CA values for the four models—Bi-LSTM, DCGAN, BERT, and MACMRGM—were 0.83, 0.88, 0.90, and 0.98, respectively. The satisfaction ratings from listeners were 0.82, 0.86, 0.91, and 0.96 for the aforementioned models, while expert satisfaction ratings stood at 0.81, 0.84, 0.90, and 0.97, respectively. When provided with a musical score from an actual audio source, it was observed that MACMRGM generated a more compliant score compared to BERT. In summary, both polyphonic models designed in this study demonstrated commendable performance and exhibited practical applicability. However, given that polyphonic music involves various combinations of instruments, future research could delve deeper into assessing the performance of the proposed models across more intricate combinations of tracks.

References

[1] Majidi M, Toroghi R M. A combination of multi-objective genetic algorithm and deep learning for music harmony generation. Multimedia Tools and Applications, 2023, 82(2): 2419-2435.

[2] Peng Y, Jiang A, Lu Q. Automated music making with recurrent neural network. Computer Science & Information Technology, 2019, 19(13): 183-188.

[3] Siphocly N N, Salem A B M, El-Horabty E S M. Applications of computational intelligence in computer music composition. International Journal of Intelligent Computing and Information Sciences, 2021, 21(1): 59-67.

[4] Yu Y, Zhang Z, Duan W, Srivastava A, Shah R, Ren Y. Conditional hybrid GAN for melody generation from lyrics. Neural Computing and Applications, 2023, 35(4): 3191-3202.

[5] Hong M, Wai H T, Wang Z, Yang Z. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. SIAM Journal on Optimization, 2023, 33(1): 147-180.

[6] Sun Q, Si Y W. Supervised actor-critic reinforcement learning with action feedback for algorithmic trading. Applied Intelligence, 2023, 53(13): 16875-16892.

[7] Zare M, Sola Y E, Hasanpour H. Towards distributed and autonomous IoT service placement in fog computing using asynchronous advantage actor-critic algorithm. Journal of King Saud University-Computer and Information Sciences, 2023, 35(1): 368-381.

[8] Scorsoglio A, Furfaro R, Linares R, Massari M. Relative motion guidance for near-rectilinear lunar orbits with path constraints via actor-critic reinforcement learning. Advances in Space Research, 2023, 71(1): 316-335.

[9] Wen Z, Li L, Song J, Zhang S, Hu H. Scheduling single-satellite observation and transmission tasks by using hybrid Actor-Critic reinforcement learning. Advances in Space Research, 2023, 71(9): 3883-3896.

[10] Liu W. Literature survey of multi-track music generation model based on generative confrontation network in intelligent composition. The Journal of Supercomputing, 2023, 79(6): 6560-6582.

[11] Wang T, Jin C, Li X, Tie Y, Qi L. Multi-track music generative adversarial network based on Transformer. Journal of Computer Applications, 2021, 41(12): 3585-3589.

[12] Huang W, Xue Y, Xu Z, Peng G, Wu Y. Polyphonic music generation generative adversarial network with Markov decision process. Multimedia Tools and Applications, 2022, 81(21): 29865-29885.

[13] Gabhane J P, Pathak S, Thakare N M. A novel hybrid multi-resource load balancing approach using ant colony optimization with Tabu search for cloud computing. Innovations in Systems and Software Engineering, 2023, 19(1): 81-90.

[14] Li N, Cai J. The Aesthetic Analysis of Music Generation Algorithms Based on Artificial Intelligence Technologies. Humanities and Social Sciences, 2023, 11(6): 223-230.

[15] Majidi M, Toroghi R M. A combination of multi-objective genetic algorithm and deep learning for music harmony generation. Multimedia Tools and Applications, 2023, 82(2): 2419-2435.

[16] Tian R, Yin R, Gan F. Music sentiment classification based on an optimized CNN-RF-QPSO model. Data Technologies and Applications, 2023, 57(5): 719-733.

[17] Cao H. The analysis of edge computing combined with cloud computing in strategy optimization of music educational resource scheduling. International Journal of System Assurance Engineering and Management, 2023, 14(1): 165-175.

[18] Jin C, Wang T, Li X, et al. A transformer generative adversarial network for multi-track music generation. CAAI Transactions on Intelligence Technology, 2022, 7(3): 369-380.

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

[20] Keerti G, Vaishnavi A N, Mukherjee P, Vidya A S, Sreenithya G S, Nayab D. Attentional networks for music generation. Multimedia Tools and Applications, 2022, 81(4): 5179-5189.

[21] Amin S N, Shivakumara P, Jun T X, Chong K Y, Zan D L L, Rahavendra R. An Augmented Reality-Based Approach for Designing Interactive Food Menu of Restaurant Using Android, Artificial Intelligence and Applications. 2023, 1(1): 26-34.

[22] Liu W. Literature survey of multi-track music generation model based on generative confrontation network in intelligent composition. The Journal of Supercomputing, 2023, 79(6): 6560-6582.

[23] Ding F, Cui Y. MuseFlow: music accompaniment generation based on flow. Applied Intelligence, 2023, 53(20): 23029-23038.

[24] Ji S, Yang X, Luo J. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. ACM Computing Surveys, 2023, 56(1): 1-39.

[25] Huang W, Xue Y, Xu Z, Peng G, Wu Y. Polyphonic music generation generative adversarial network with Markov decision process. Multimedia Tools and Applications, 2022, 81(21): 29865-29885.