

## Research Article

# Construction of an Intelligent Arrangement Model for Vocal Music Based on HPC Cluster Programming

Tianzhuo Gong <sup>1</sup> and Chunli Han <sup>2</sup>

<sup>1</sup>Academy of Music, Capital Normal University, Beijing 100040, China

<sup>2</sup>Academy of Music, Harbin Normal University, Harbin, Heilongjiang 150080, China

Correspondence should be addressed to Chunli Han; hsd2022@hrbnu.edu.cn

Received 11 March 2022; Revised 27 April 2022; Accepted 5 May 2022; Published 26 May 2022

Academic Editor: Sheng Bin

Copyright © 2022 Tianzhuo Gong and Chunli Han. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We model in this paper the HPC cluster programming structure and perform a multidimensional computational analysis to design a vocal intelligent arrangement model based on HPC cluster programming using brilliant vocal arrangement. This paper proposes the construction scheme of the HPC interconnection simulation environment based on the full investigation of the research related to the construction of a supercomputer simulation environment at home and abroad. The rapid development of HPC technology has promoted the study of automatic music generation technology, which effectively avoids the traditional machine learning method of music. The rapid growth of HPC technology has facilitated the research of automated music generation techniques, which effectively avoids the overreliance of conventional machine learning methods on music rules and artificial design features. It achieves better results in complex music generation tasks. The key features of intelligent music arrangement are designed and implemented by summarizing the principles of converting paintings to music and then creating the rules of converting images to music and the corresponding relationships based on the theoretical basis. After the conversion rules are clarified, the specific design is carried out according to the computing process of the genetic algorithm. At the same time, the framework of the arrangement is combined with children's aesthetic psychology; the evolutionary adaptation function is developed in combination with music theory, which finally ensures the motility of the generated music.

## 1. Introduction

Music is an art that goes deep into life, and creating musical works is not an easy task. A good music piece usually requires a composer to compose carefully based on personal inspiration and professional knowledge, and it is difficult for an untrained layperson to compose music [1]. With the development of artificial intelligence and its related technologies, computerized automatic music generation system has gradually become a novel research direction, which emerged to address the needs of people in nonmusic fields to create musical products, can significantly reduce the technical threshold and save production costs, and has a wide range of application prospects. The purpose of automatic computer arranging or music generation is to enable computers to automate music composition by learning

about human music composition. The implementation of this process usually requires the representation of music in the form of data that can be processed by the computer, followed by learning about music composition and, finally, the creation of new music [2]. Therefore, the methods can be distinguished by the form of music data to be processed. They can be divided into the generation of waveform music data and the generation of symbolic music.

With the technological development of hardware and software devices, the ability and superiority of artificial intelligence are now rapidly penetrating all areas of life. Yet, it has also become the hottest word in music technology now. We know that the promotion of music is based on technological change. Without the invention of tools, there would be no musical instruments; without the emergence of electronic technology, there would be no electronic music

[3]. Thus, the significant technological occurrence of artificial intelligence is one of the motivations for writing this paper on how much music can be changed. For the final generative model to have the ability to influence the output preferences based on the input contour control labels, it is necessary to match the training samples with their corresponding sequences of contour labels during the training process. Artificial intelligence arranger generation is an interdisciplinary field that requires researchers to master many interdisciplinary disciplines, including music production, music technology, artificial intelligence, and automatic accompaniment. Since it is an emerging field and thus the domestic research is still relatively poor, we hope to contribute to the area by combining our professional hobby of arranging music, driven by the times.

Building an HPC simulation environment is not an easy task. First, a distributed parallel computing cluster system needs to be made, and functional modules are provided for the cluster system to support its parallel computing tasks. There are two issues that the question needs to focus on solving: the clock synchronization problem of the HPC interconnection simulation environment and the rate adaptation problem of the HPC interconnection simulation environment [4]. To improve the accuracy of the simulation environment, we must solve the problem of clock synchronization between the nodes, which makes the task calculation results more inaccurate and thus affects the final simulation results. It is necessary to provide a highly consistent clock for each node of the simulation environment to ensure clock synchronization between nodes. Device emulation, especially network and CPU device emulation, is a critical element of the virtual simulation platform. When network rate emulation and CPU rate emulation are performed in the simulation environment and parallel computing tasks are executed, the problem of abnormal exit of similar computing tasks may occur due to a mismatch of network rate and CPU rate. The network rate simulation and CPU rate simulation are the critical elements of the virtual simulation platform. When the network rate and CPU rate simulations are performed in the simulation environment and the parallel computing tasks are executed, the network rate and CPU rate may not be adapted and similar computing tasks may exit abnormally [5]. Studying the rate adaptation problem in the simulation environment is necessary to adjust the CPU rate of nodes and the network rate to ensure that the simulation tasks can be executed correctly in the simulation environment.

## 2. Related Works

Abualigah proposed the speech sequence technique in the early twentieth century; Diabat and his successors sequenced this; in this series of courses, musical features are extracted and controlled by pitch, pitch length, and timbre [6]. A range of possible values can be chosen from each parameter to form the sound sequence. The parameter values may be arranged next in the resulting sound sequence or in the reverse order of the sound sequence. In the early 1950s, Wahengbam used a stochastic process to generate music

fragments by hand [7]. By this time, computers had been created and were becoming a tool for music composition, and people began to use them to produce music. They began to be used as tools in the music composition process by Lejaren in 1956 with the publication of the first fully computer-generated musical work, the string quartet ILLIAC Suite. The latest research and development of intelligent arranging technology focuses on Sony Google. Zhang has designed an online interactive piano that allows users to play a small number of notes and automatically play matching music based on the consistency of the music [8]. Google has also developed an application that integrates with Google Glass, connected to Android, to help blind people determine the obstructing objects and their surroundings through echolocation technology. This application first uses the Android device's camera to perform image recognition, match it with the corresponding voice prompts, and then pass it through stereo headphones to convert image to sound. Google Glass frees the user's hands and works independently, but its biggest drawback for blind users is that it transmits sound to the user's inner ear through bone conduction, thus lacking the stereo effect. Further improvement of this application will require developing a new system for converting from image to sound from the perspective of the blind, using Android mobile devices to serve the blind better.

Marsal-Llacuna et al. investigated the application of RNN-based music generation and compared the performance of different RNNs [9]. Scherer et al. implemented a model oriented to polyphonic music generation using char RNN with one LSTM to learn the modeling of all tracks [10]. Borges et al. implemented polyphonic music generation using a multilayer stacked LSTM network; they encoded random variables as melodies through one layer of LSTM [11]. On top of that, they used multiple stacked LSTMs to achieve the generation of music for drums, BASS. Dichev et al. proposed a large-scale polyphonic music generation model consisting of a melodic and rhythmic cross-generation model based on harmonic progressions and a multi-instrument track accompaniment model [12]. This model consists of a musical and rhythmic crossover generation model based on harmonic progressions and a multi-instrument track-based accompaniment generation model with attentional mechanisms. The former consists of two intersecting GRUs that form a melodic encoder for learning to generate the notes and rhythms of a melody based on a given melodic and harmonic progression, while the latter is responsible for learning to create an accompaniment consisting of multiple instrument tracks for an existing song by introducing an attention mechanism to encourage the model to learn how to maintain the harmony between instrument tracks. Their approach is multitasking learning and achieves good generative results. Also focusing on polyphonic music modeling, Mittal et al. applied the neurolinguistic model Transformer to piano music generation, which consists entirely of attentional units and has better long-term dependency modeling capabilities than RNN methods LSTM learning associations at

different scales in sequences. Their model can generate very long piano polyphony music and sound more thematic [13].

The simulation and prototyping of application scenarios in different industries often rely on high-performance computing technology, which further enhances the value of HPC applications in various fields and has a more urgent need for high-performance computing [14]. Verma et al. optimized CFD (Computational Fluid Dynamics) applications based on heterogeneous systems by building functional performance models, balancing workloads, and optimizing communication overheads based on domain decomposition [15]. Sebastian et al. successfully migrated the Relion application to GPU heterogeneous system [16]. They effectively improved the program's parallelism through various adaptive parallelism frameworks, data layout optimization, loop expansion, and so on. Ao et al. performed HPCG (High-Performance Conjugate Gradient) by multicoloring, data mapping, and other methods. Humayun et al. tested the performance and power consumption of Gromacs application under heterogeneous platforms based on different environment configurations [17]. Bui et al. accelerated the implementation of NAMD (Nanoscale Molecular Dynamics) applications by optimizing the data layout and accelerating the operational efficiency with the help of the Charm++ parallel interface [18].

### 3. Design of an Intelligent Arrangement Model for Vocal Music with HPC Cluster Programming

**3.1. HPC Cluster Programming System Construction.** The HPC cluster programming environment is a parallel computing cluster environment built using virtualization technology to create more virtual nodes on a limited number of physical machines to meet the node size requirements of the cluster environment. The cluster's parallel computing environment is installed to provide a runtime software environment for similar computing tasks. This section introduces the essential technologies in building the HPC interconnection simulation environment to lay the theoretical foundation for the subsequent research. Host virtualization technology is the primary technology for creating an HPC interconnection simulation environment cluster. It tests and predicts the performance of computer systems and reveals the strengths and weaknesses of different architecture machines in specific aspects. Virtualization technology builds a larger-scale cluster system with limited physical machine resources and enables easy cluster node expansion. In the X86 platform, the virtualization layer is abstracted as a virtual machine monitor or hypervisor, which runs in the operating system kernel space of the physical machine (host). The virtual machines created using virtualization technology run in the physical machine kernel space. The virtual machines created using virtualization technology run in the user space of the physical machine operating system. A virtual machine node is often called a guest, and

the virtual machine node can run the operating system just like a physical machine.

There are many virtualization solutions, and the following is an introduction to the more popular virtualization technologies. The X86 architecture, which occupies a significant share of the server market in the HPC field, has the following characteristics: first, it adopts a core-heavy architecture, which advocates a "performance-first" design philosophy; second, it uses the CISC (Complex Instruction Set Computers) instruction set, which is mainly designed to provide a high level of performance. Secondly, CISC (complex instruction set computers) instruction set is used, which mainly designs instructions with variable lengths to deal with complex computing tasks. Thirdly, the closed hardware architecture is adopted. Fourth, the ecological maturity and processor are multifunctional. Compared with the ARM architecture, X86 has several shortcomings: first, it is characterized by poor technology and slow performance improvement; second, the small size of the general-purpose registers limits the access performance of the CPU and affects the overall system execution speed; third, relying on the complex instruction set, the device needs to be micro-decoded into multiple simple instructions when decoding complex instructions, which reduces the instruction execution speed, and the structure is relatively tricky due to the varying instruction lengths. The system of the HPC application is shown in Figure 1. Host virtualization technology is the primary technology for building HPC interconnection simulation environment clusters. The use of virtualization technology creates a larger-scale cluster system with limited physical machine resources and enables easy cluster node expansion.

One of the most common network topologies used for clustered networks is the multistage Clos topology, which consists of multiple-stage switches interconnecting each computing device's input and output links. A small-scale three-stage Clos network topology is shown in Figure 2, which consists of three stages of switches connecting the input and output links of nine computing devices. An example of packet routing in a multistage network is presented next. The packet routing problem is modeled as a control problem for a discrete-time queueing system in a three-stage network. At a particular moment  $t$ , data packets generated by a computing device arrive at a switch in the input stage. An input or output switch is usually connected to multiple computing devices. The computing devices connected to the input and output switches are not presented for brevity. The number of arriving packets destined for the first  $\lambda_{ij}$  output switch obeys a Poisson distribution with an arrival rate  $\lambda_{ij}$  at the first input switch. These arriving packets are queued in the first  $j$  input switch's first queue. The network state is the number of queued packets in that network, including newly arrived packets and packets already in the queue. At each moment, each link between the different phase switches is responsible for transporting the boxes from the upstream switch to a downstream switch with a unit capacity. The routing algorithm needs to select a downstream link for each queue-headed packet to transport it. The routing of all queue-head packets can be considered a global routing

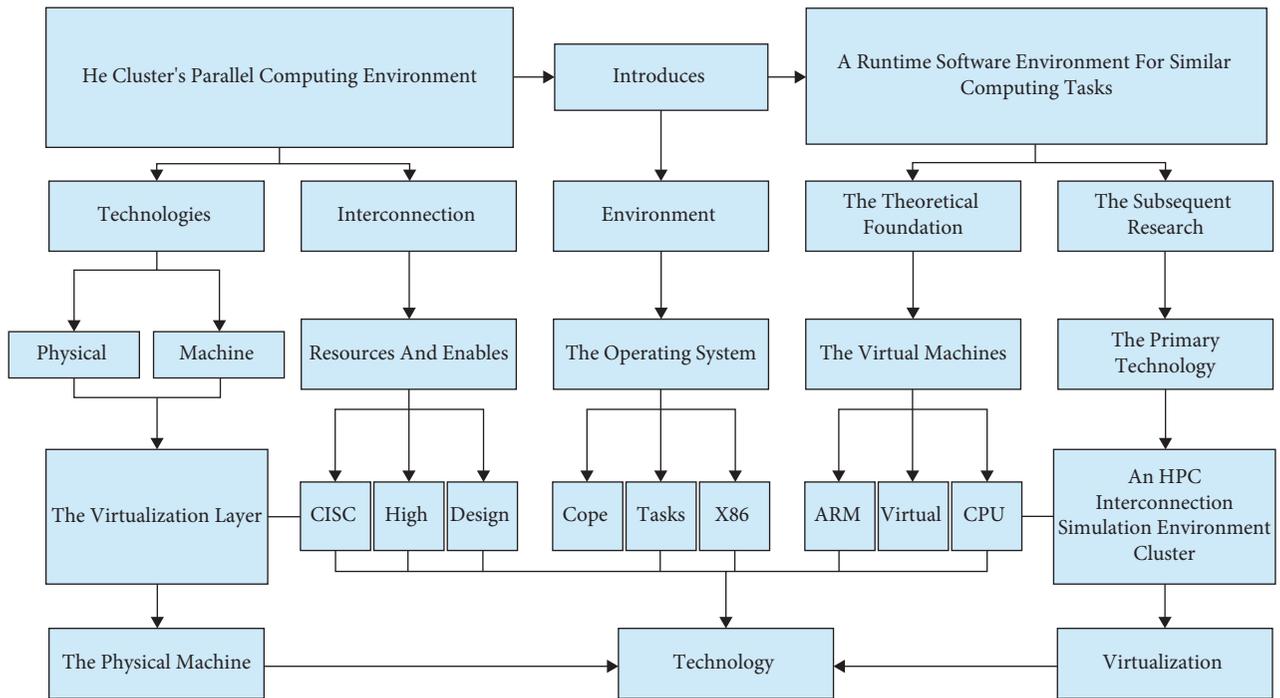


FIGURE 1: HPC application structure diagram.

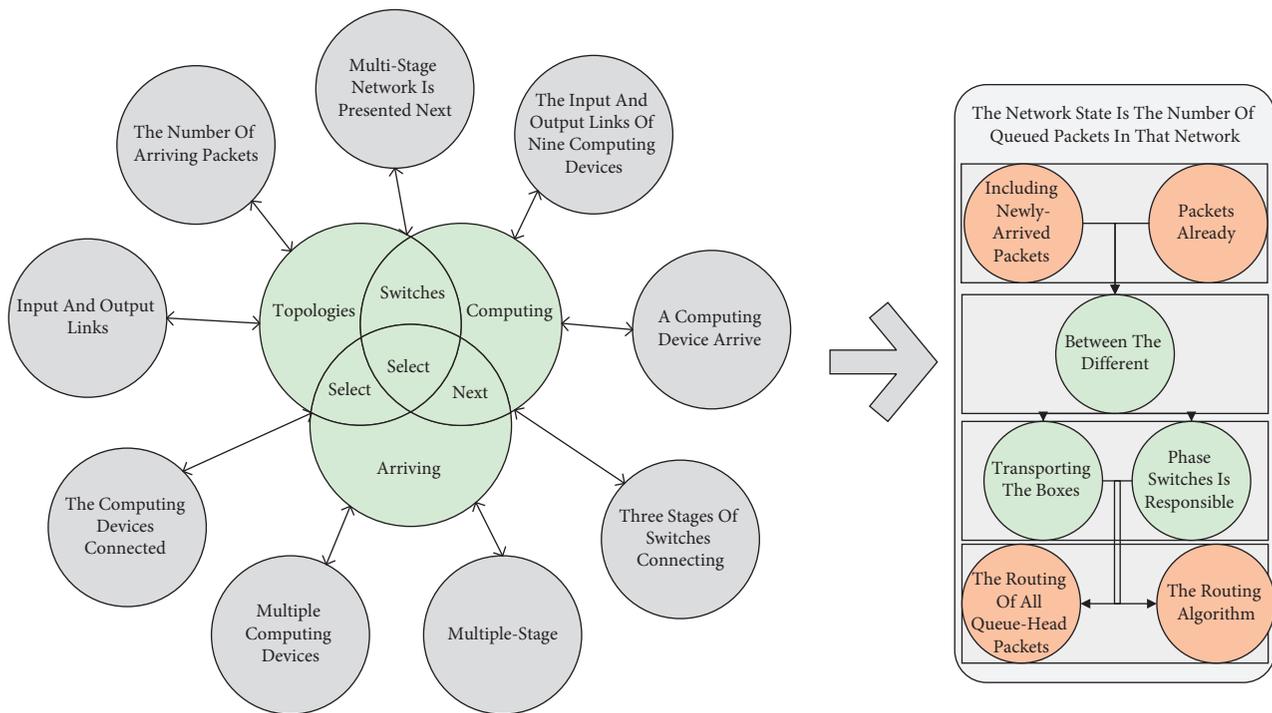


FIGURE 2: Cluster network with Clos topology.

action. The head-of-queue packet is transmitted on the selected link following this routing action. Parcels in transit simultaneously arrive at the corresponding queue at the downstream switch. Packages arriving at the output switch are delivered directly to the computing device. After all, packets arrive at the downstream regulator, and a new round of packets arriving at the input stage switch follows the same

Poisson distribution. Then similar routing actions and packet transfers are repeated.

With the rapid growth of CPU central frequency and CPU cores in recent years, new requirements have been put forward for software technology development. The use of parallel computing technology can make full use of multi-core processor resources, adapt to the rapid growth of

hardware devices, accelerate the computation rate of tasks, and improve resource utilization. The leading parallel computing technologies are OpenMP based on shared memory and MPI based on message passing, and the two similar programming models are described in the following.

**3.1.1. OpenMP Parallel Programming Model.** OpenMP is a shared memory-oriented programming model that supports distributed shared storage and multithreaded applications. When a parallel computation task is needed, the main thread spawns a child thread to handle a similar computation task. The mainline and the child thread run in parallel during the execution of the parallel task. OpenMP is designed for single-host multicore processors and multiprocessor parallel computing. The shared storage between lines makes it highly efficient on multicore processors, with less memory overhead and more straightforward parallel programming. The performance of OpenMP is more advantageous in single-computer shared-storage parallel computing, while its performance in distributed parallel computing is often unsatisfactory.

**3.1.2. MPI Parallel Programming Model.** MPI is a message-passing-based parallel programming model. Message passing means that each thread has an entire memory space and code segment during parallel execution, and similar tasks run independently. MPI is a parallel programming standard with different manufacturers' implementations. Its vital portability ensures that code written in one implementation can be easily ported to another without significant changes. MPI provides functions for allocating computational resources according to a logical interconnection topology, and MPI ensures optimal message delivery using the underlying physical network topology. MPI provides rich communication functions and robust capabilities for point-to-point and aggregate communication. In addition, MPI supports multithreaded and concurrent library application development by introducing communication subs.

The cluster network packet routing problem is modeled as a Markov decision process specified by a quadruplet  $(S, A, c, p)$ .  $S$  is the state null Ho,  $A$  is the action space,  $C$  is the cost function, and  $P$  is the state transfer probability. Condition: the state of the MDP now  $t$ , denoted as  $s_t$ , is a 3-dimensional matrix  $\{n_{sij}^t\}$  that represents the number of groups in the first queue on the first  $j$  switch in the  $s$ -th phase. Action: assuming that there are  $M$  queue-head groups in the network, the movement of the MDP at a time  $r$  is the set  $\{a_1, a_2, \dots, a_M\}$  of links selected for each of the queue-head groups. The order of action generation starts from the head group in the lowest index queue on the most down index input switch in the most upstream stage and gradually polls to the head group in the higher index queue on the same input switch, then to each head group on the higher index input switch in the same stage, and finally to each head group on the other downstream button [19]. The link selected for the first  $m$  head group is free downstream links that other head groups do not select. When some queues on a switch do not have a head packet, nonhead packets in

different queues on that switch may also receive a link assignment after the head packet selects a link to utilize the link fully. Melody, consisting of a series of notes and corresponding note durations, is a fundamental part of popular music, yet generating a harmonious melody still presents many challenges.

$$n_{sij}^t = \frac{n_{sij}^{t-1} - 1}{\sqrt{d_{sij}^t - e_{sij}}} \quad (1)$$

The cost of this MDP now  $r$  is the total number of queued packets in the network, denoted as  $c(s_t) = \sum_i^s j n_{sij}^t$ . By Ritter's theorem, the average total number of queued packets in the network is proportional to the average transmission delay of the boxes as they pass through this network. Thus, reducing the average number of queued packets in the network reduces the average packet transmission delay. State transfer equation: since the transfer probability matrix of this problem is infinite-dimensional and its expression is cumbersome, this paper uses a more intuitive and equivalent state transfer equation to express the MDP. For the noninput phase of the switch, the following random difference equation gives the state transfer. Evaluating the performance of a computer system, including the use of specific evaluation methods and the selection of representative applications, is the issue that needs to be considered when building an HPC cluster.

$$n_{sij}^{(t-1)} = \frac{n_{sij}^{(t-1)}}{\sqrt{d_{sij}^t + h_{ij}^{(t-1)}}} \quad (2)$$

The techniques used to build the HPC cluster programming environment and implement clock synchronization and rate adaptation in the simulation environment are presented. Firstly, the leading technologies required to create an HPC interconnection simulation environment are introduced, such as virtualization, parallel computing cluster construction, and network emulation; the mainstream virtualization and similar computing technologies are highlighted, and their respective advantages and disadvantages are analyzed. Then, the main clock synchronization protocols currently used to achieve network clock synchronization are introduced. Finally, the relevant technologies to achieve CPU rate adaptation are raised from the hardware and software levels, respectively, laying the foundation for realizing the rate adaptation of the HPC cluster programming environment.

**3.2. Vocal Intelligent Arrangement Model Design.** This paper proposes constructing a vocal brilliant arrangement model based on HPC cluster programming. The method introduces a simple sequence that is consistent with the actual melodic notes in pitch time-space and easy to be constructed directly by the user as the control label to achieve adequate control of the local contour characteristics of the melody; the automatic inference of the contour control label is achieved by setting up a differentiable objective function, thus avoiding the manual labeling work; the VAE variational inference is

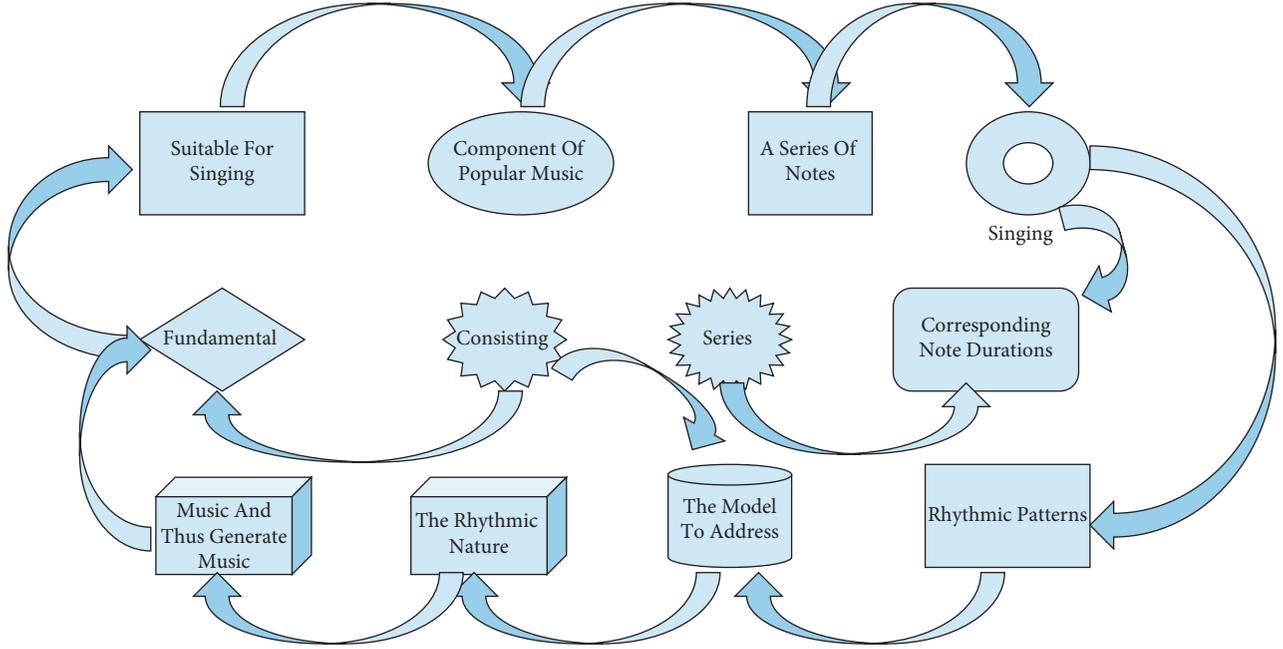


FIGURE 3: Rhythmic and melodic cross-generation model of chords.

used to accomplish the implicit encoding of the contour control label. The implicit encoding of melodic attributes other than contour control labels is achieved by using VAE variational inference. First, consider the connection between the actual data sample  $x$ , the explicit control condition  $c$ , and the latent variable  $z$ . Let the random variables  $c$   $z$  in the joint distribution  $P(x, c)$  be independent of each other. The melody samples generated by the model should have contour consistency with the user input contour label sequence; that is, the adjacent elements of the two sequences change in the same direction and have the same change trend.

$$P_{(x,c)} = \sum_x p(x, c), \quad (3)$$

$$\sum_x P(x, c) = \frac{\sqrt{P(x, c)}}{P(c) + 1}.$$

Melody, consisting of a series of notes and corresponding note durations, is a fundamental component of popular music, yet generating harmonious melodies still presents many challenges. Previous work on note-level-based generation methods has more randomness in rhythm and is not suitable for human singing. Therefore, in this paper, rhythmic patterns are incorporated into the model to address the rhythmic nature of the music and thus generate music suitable for singing. The CRMCG model architecture is given in Figure 3, which mainly consists of three parts: rhythm encoder-decoder, melody encoder-decoder, and chord encoder (Chord GRU). The encoder-decoder framework uses recurrent neural networks (RNN) to better handle music sequences with temporal relationships. The simulation and prototyping of application scenarios in different industries often rely on HPC technology support, which further enhances the value of HPC applications in

various fields and has a more urgent demand for high-performance computing.

The model mainly consists of an RNN encoder ( $E$ ) and an RNN decoder ( $D$ ) trained by a stochastic gradient descent algorithm. During training, the encoder network infers the contour control labels of the training samples and the latent variables. The decoder network reconstructs the melody samples from the label control sequences with the latent variables. The decoder network needs to be retained in the generation phase. The user inputs the contour label sequences and samples the latent variables from the prior distribution (high-dimensional standard Gaussian distribution). The global characteristics of the generated samples can be adjusted by changing the latent encoding for a given contour label. In the chord encoder, to better characterize the chord information, the model uses gated recursive units (GRUs) to map the chords to higher dimensions, represented as follows. In collaborating with each node to complete parallel tasks in the HPC interconnection simulation environment, the differences between devices cause the clocks between nodes to be unsynchronized, making the task calculation results have significant errors, which in turn affect the final simulation results.

$$h_i = \int_c \text{GRU}_c \left( \frac{h_{i+1}}{\sqrt{c_i - 1}} \right). \quad (4)$$

Since the generated rhythm needs to be in harmony with the existing musical part, the model considers the last part of the music. First, the previous rhythm  $R_{t-1}$  and melody are  $E_m$  multiplied  $M_{t-1}$  with the embedding matrix  $E_r$  and represented as a high-dimensional vector, and then the representation is obtained as follows. A complex HPC system requires high-performance processors, high-speed

interconnected network systems, storage systems, maintenance and monitoring systems, power supply and power systems, cooling systems, and structural assembly designs. When performing system performance evaluation, it is necessary to understand the concepts and methods related to the review to more accurately interpret the system's performance data and evaluate it.

$$M_t = \frac{\sqrt{E_m - m}}{m_{t+1} + E}. \quad (5)$$

For the final generative model to have the ability to influence the output preferences based on the input contour control labels, it is necessary to match the training samples with their corresponding contour label sequences during the training process. In the model of this paper, the input to the generator network at each time step contains a contour control label, and this training process can be viewed as a Monte Carlo method that aims to enable the generator network to establish a link between the contour control sequence and the melody, usually with supervised training. Automatic inference of contour control labels can be achieved by setting up additional differentiable optimization objectives, essentially transforming a supervised training problem into an unsupervised training problem using the nature of musical deconstruction as prior knowledge [20]. The model in this paper employs a bidirectional RNN (consisting of a forward RNN and a backward RNN) as the encoder network. It is shown empirically that the bidirectional RNN can learn better global sequences than the general unidirectional RNN. A single thermal vector sequence of melody samples is used  $x^n$  as the input of the encoder, and the output of the encoder at each time step is defined as a real scalar (usually, the output dimension of an RNN is the hidden layer dimension, and the scalar output can be obtained by adding a linear mapping layer with 1 output dimension). The output scalar sequence is noted as  $e^n$ . In addition, both the forward RNN and the reverse RNN in the encoder have an end hidden layer state, and the two are spliced into a vector noted as  $ste \in R$ , where  $D$  is the number of hidden layer units of the RNN in the decoder of two times. The purpose of automatic computer orchestration or music generation is to enable computers to automate music composition by learning about human music composition. The implementation of this process usually entails the representation of music into a form of data that the computer can process, followed by learning about music composition and, finally, the creation of new music.

$$l = \sum_i \max \left( \frac{e_{i+1} - e_i}{\sqrt{x_{i-1} - x_i}} \right). \quad (6)$$

It is important to note that the neural network model based on the encoder-decoder architecture is also a self-encoder model. Without any regularization constraints, self-encoders can degenerate into meaningless index functions when network parameters are sufficiently large. In the extreme case, if the width of the bottleneck layer of the network is one such that the model can ultimately "remember" all the training samples and encode an identity for each training

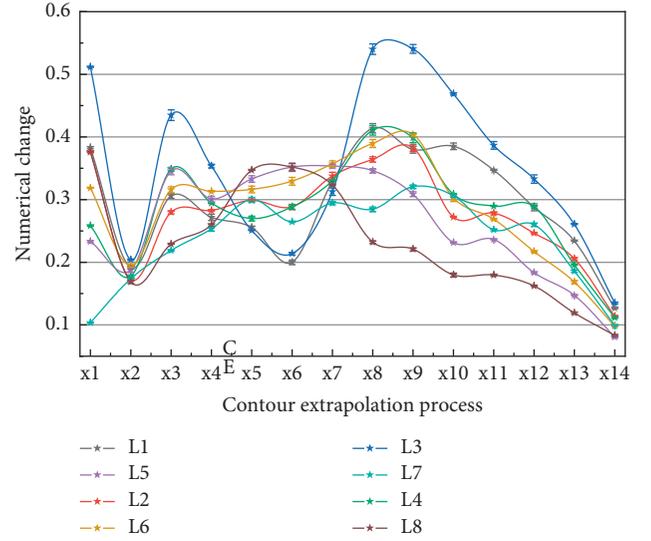


FIGURE 4: Numerical variation of contour extrapolation process.

sample if there is enough parameter capacity, such a model is meaningless. Artificial intelligence arranger generation is an interdisciplinary field that requires researchers to master much multidisciplinary knowledge, including music production, music technology, artificial intelligence, and automatic accompaniment. To avoid a similar situation, the number of contour quantization segments  $m$  described in this paper is much smaller than the actual note variation range. Suppose the existing note variation range is  $T$ , that is, 88 notes; when  $m = 88$  means the contour label has the same variation accuracy as the melody, the encoder will degenerate into a constant mapping from to input. The decoder network will also degenerate into a continuous mapping when the decoder network is trained to reconstruct the samples. The decoder network will completely ignore the latent variable encoding, and thus the posterior collapse will occur [21]. Only when the contour quantization segment  $m$  is incomplete relative to the range of note variation, the contour labels do not provide enough information to support the reconstruction of the melody, and the decoder is forced to establish the link between latent variables and the song. The numerical variation of the contour inference process is shown in Figure 4.

## 4. Analysis of Results

**4.1. Analysis of Vocal Intelligent Arrangement System with HPC Cluster Programming.** Performance evaluation of high-performance computer systems is an integral part of data centers or data centers of university research institutes when conducting procurement or project acceptance tests. Evaluating the performance of a computer system, including the use of specific evaluation methods and the selection of representative applications, is the issue that needs to be considered when building an HPC cluster. A complex HPC system requires high-performance processors, high-speed interconnected network systems, storage systems, maintenance and monitoring systems, power supply and power

systems, cooling systems, and structural assembly design. When conducting system performance evaluation, it is necessary to understand the concepts and methods related to the evaluation to interpret the system performance data and evaluate it more accurately. According to the different test purposes and test objects, the evaluation of ARM high-performance computing cluster systems is grouped into benchmark test objects and actual application objects. Building an HPC simulation environment is not an easy task. It requires building up a distributed parallel computing cluster system and providing functional modules for the cluster system to support its similar computing tasks. The corresponding test methods are the benchmark and the actual application tests. The benchmark program test method refers to the use of a variety of benchmark test programs developed by the industry to test the performance of specific aspects of high-performance computers, which belongs to the test method of partial hardware performance, such as the floating-point computing power of the processor, memory read and write speed, disk read and write speed, and network performance, with a certain degree of relevance.

On the one hand, it can test and predict the performance of computer systems, reveal the strengths and weaknesses of different architecture machines in specific aspects, and provide decision-making suggestions for computer selection and procurement in high-performance data centers such as enterprises and university research institutes [22]. On the other hand, benchmarking uses a more objective and unbiased evaluation of server performance to demonstrate system performance more directly in a way that users understand. When using benchmarking procedures for system performance evaluation, factors such as data set and problem size need to be considered. The real-world application testing method, which runs real, complex scientific computing applications in HPC systems, is essential for evaluating the performance of high-performance computer systems. The memory bandwidth performance of node-bound is shown in Figure 5.

The reconstruction of notes in an actual melody is defined as a multiclassification problem. The higher the probability that the model is assigned to the letters of the sampled song, the better the model reconstruction performance can be described by analogy with the definition of perplexity in language models. The cross-entropy function exponentiates the base of the natural logarithm to obtain the perplexity measure of the model. A model is considered more confident when it assigns a higher probability on the notes trained on the sample, and vice versa, indicating a more confused model. The harmony scores of the generated music are fine-tuned and improved, which means that the discriminator can guide the generator to a more harmonious state. Experimental results are shown for chord scores. The generator uses a multitask music generation model based on the above, using different multitask learning units, including attention and perception units. To better measure the reconstructive ability of the melody generation model, it is helpful to assume that the model can only guess with equal probability without learning any practical knowledge about melody construction. When the model fails to converge

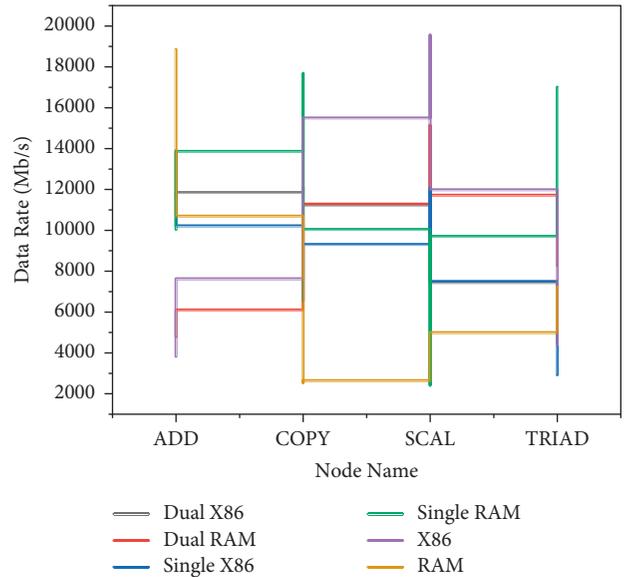


FIGURE 5: Memory bandwidth performance of node binding.

effectively in training, its average cross-entropy loss will be close to that value or even higher when the model behaves very confusingly. For models joining usually, a lower average cross-entropy loss indicates better reconstruction ability; that is, the model is more confident in its note prediction. Closely related to the reconstruction loss of the model is the correctness of the model in predicting notes, and theoretically, the lower the reconstruction loss, the higher the correctness of the model in predicting notes. An analysis of the arrangement generation is shown in Figure 6.

For an interactive melody generation model, the ability to flexibly control the feature preferences of the generated melodies through an interactive interface is one of the essential criteria to measure the model's performance. In this paper, the model only needs to keep the decoder network in the generation phase. The melody samples are generated by user input contour control tag sequence and latent variables. Among them, the contour control tag sequence is used to influence the local contour features of the melody generated by the model. With the development of artificial intelligence and its related technologies, computerized automatic music generation system has gradually become a novel research direction, which emerges to address the needs of people in nonmusic fields to create music products, can significantly reduce the technical threshold, can save production costs, and has a wide range of application prospects. Theoretically, the melody samples generated by the model should have contour consistency with the user input contour label sequence; that is, the adjacent elements of the two sequences change in the same direction and have the same change trend. The MICA model based on the perceptron unit achieves the highest score, which is a 24.4% improvement over the HRNN model, indicating that the MICA model can improve the harmony of multitask music by using helpful information from other tasks. Also, as the number of music tracks increases, the chord score decreases, which indicates that the theme with more tracks has a higher requirement for

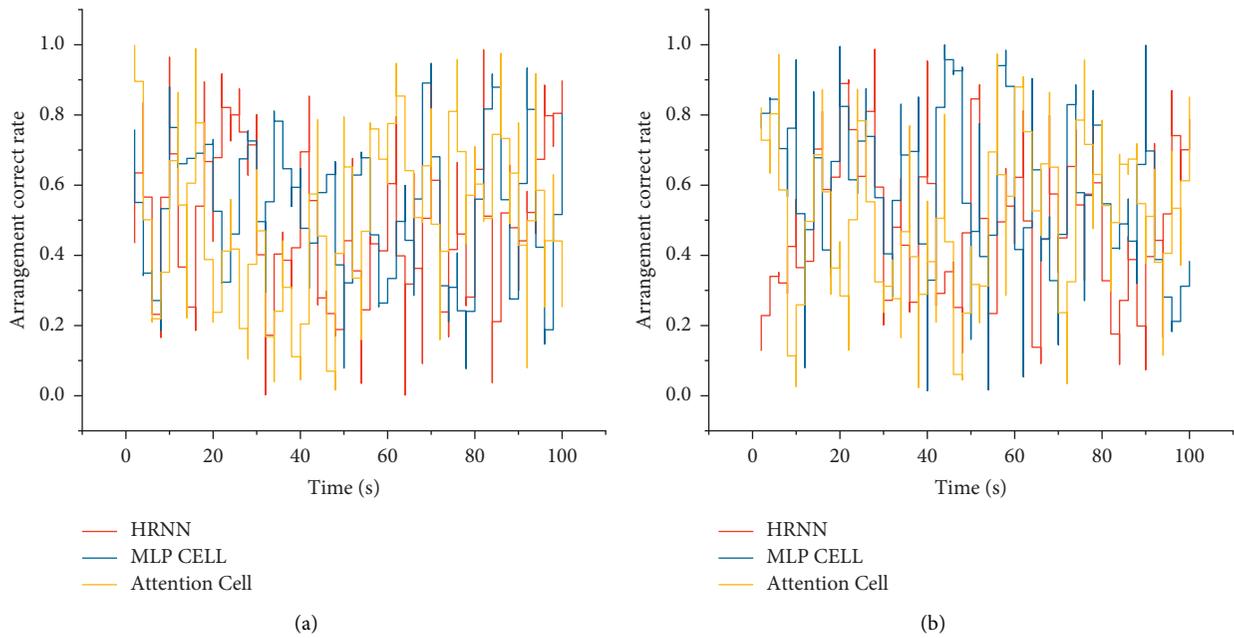


FIGURE 6: Analysis of the orchestration generation.

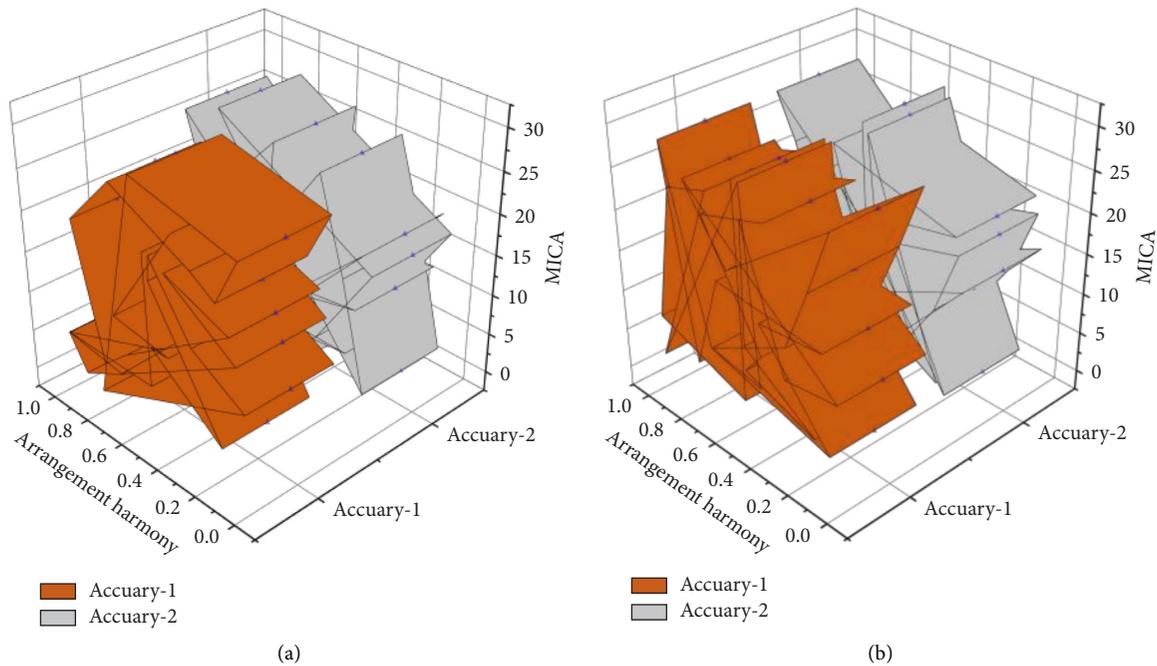


FIGURE 7: Harmony analysis of the arrangement.

harmony. The arrangement harmony analysis is shown in Figure 7.

A chord-based model for rhythmic and melodic cross-generation (CRMCG) is proposed to address the lack of musical domain knowledge in existing music generation models, the inability to guarantee interval relationships, and the difficulty in learning structural features of popular music. The chords ensure the interval relationship harmony, while the rhythmic patterns make the generated music more structured and thus enhance the melody effect. On the other

hand, the multitask learning-based multi-instrument joint arrangement model (MICA) is proposed to create multitrack music with harmonious coordination between multiple tracks and learn various instruments playing characteristics. Among them, two strategies for information interaction between numerous tasks, an attention unit and a perceptron unit, are used to ensure the harmony of multitrack music. Finally, extensive experiments on actual data are conducted to verify the model’s effectiveness in terms of manual evaluation and music theory metrics, respectively.

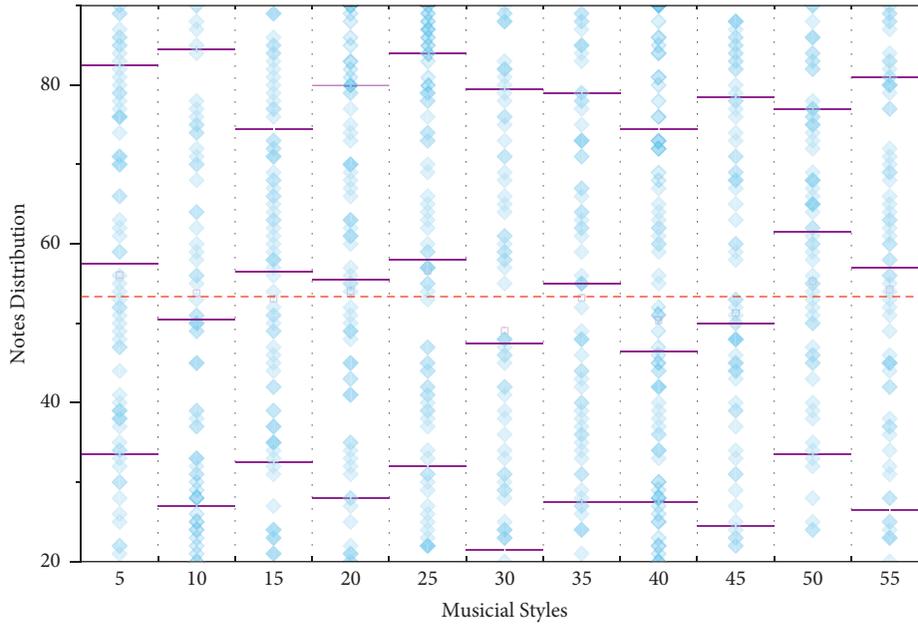


FIGURE 8: Instrumental song style analysis.

*4.2. Vocal Intelligent Arrangement System Construction Implementation.* The experiments fine-tune the generator so that the generated music keeps approaching a specific style of music, but this also changes the piece's characteristics, such as harmony, so the harmony discriminator needs to be used to ensure the balance of the music generated after fine-tuning. The theme of the music developed before and after fine-tuning with the harmony discriminator in multitrack music was experimentally compared to verify its performance. The harmony scores of the generated music improve after fine-tuning, which means that the discriminator can guide the generator to a more harmonious state. Experimental results are shown for the chord scores. The generator uses a multitrack music generation model based on the above by having different multitask learning units, including attention and perception units. From the experimental results, on the one hand, the fine-tuning of the discriminator improves the harmony of the generated music, proving the effectiveness of the adversarial training.

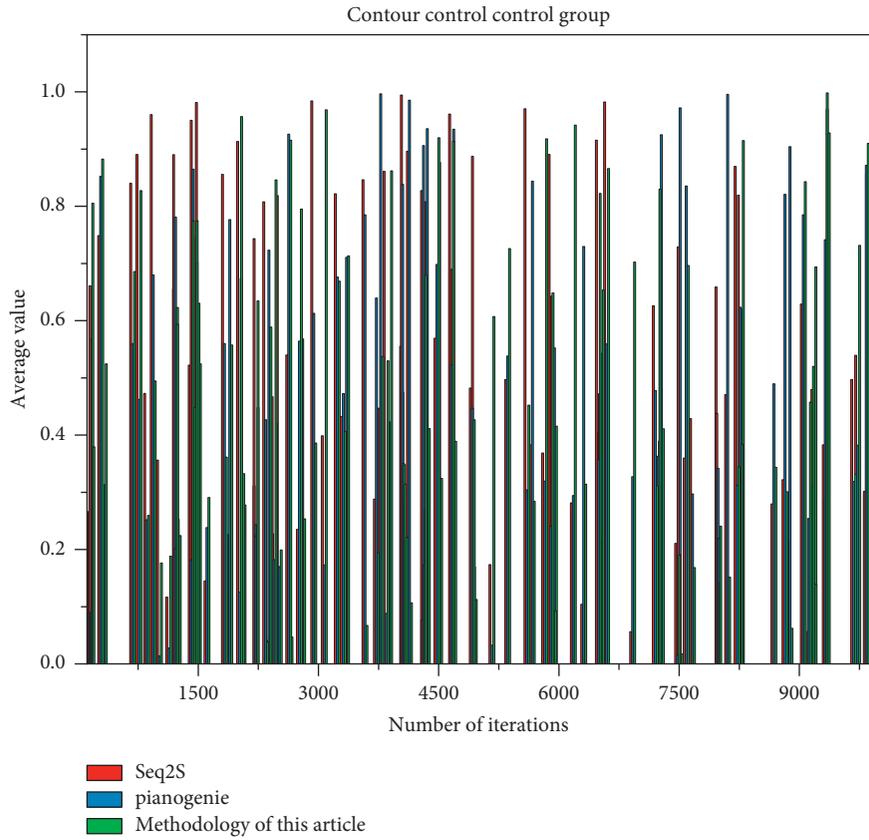
On the other hand, the generator based on the multitask learning approach has better harmonic performance, while the attention unit has the best harmonic score. In addition, the generator with multitask learning has better improvement than the generator without multitasking learning, which implies that the multitask learning approach can improve the performance of the multisequence generator. This paper conducts experiments on latent variable interpolation generation by interpolation rules. Theoretically, given a contour control sequence, points at different locations in the latent variable space represent melody samples with consistent contours but various global features. A series of models are generated by interpolating latent variables for two pieces with distinct global features. Suppose a gradual transition in the global characteristics of these samples can be observed. In that case, it is verified that the latent variables of this paper's model effectively encode the global properties

of melodies. In this paper, a melody sample of 64-note length is randomly selected from the test set, and the encoder network of our model computes the contour label sequence of this sample. Then 100a random Gaussian noise was sampled as a latent variable to generate 100 consistent contour samples. The instrumental melody analysis is shown in Figure 8.

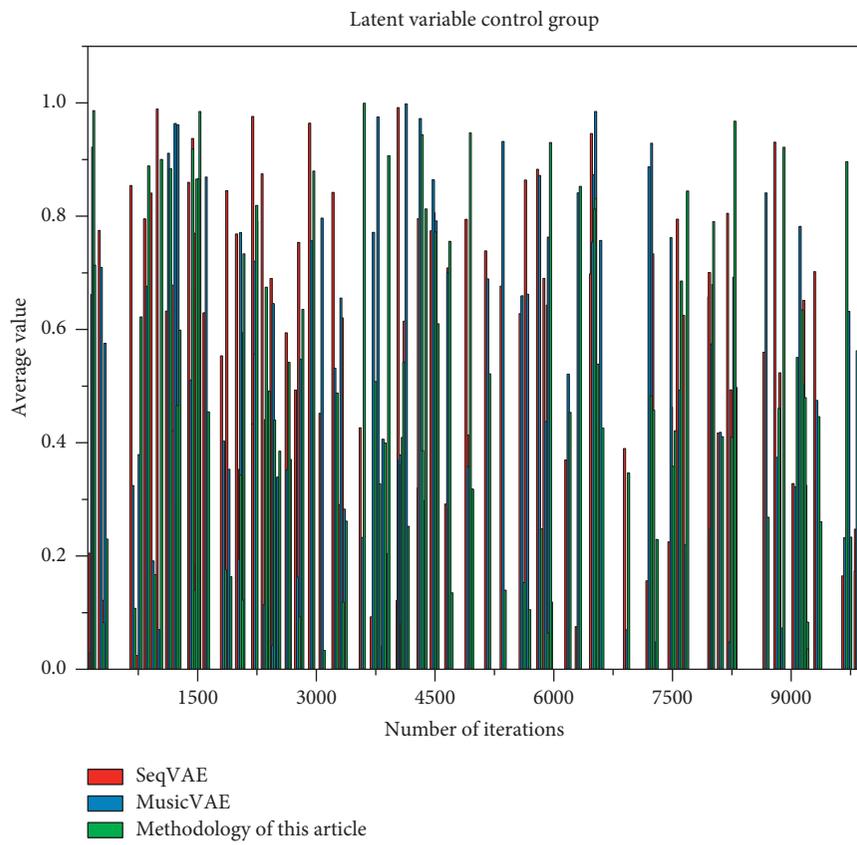
Choosing an appropriate value  $\lambda$  to maintain the music style and harmony is the problem to be solved. In this section, different parameters of A experimented with, and the effect of other generators on the strategy reward was evaluated by setting five different parameters of A, including 1.0, 0.8, 0.5, 0.2, and 0.0. The comparison of different generators on harmonic performance is shown in Figure 9, including HRNN and MICA under other parameters A. It is found that the harmony score works worst when the model has no harmony discriminator at  $\lambda = 1.0$  this time, which indicates that the harmony discriminator can improve the harmony of the music. In addition, the model with only harmonic discriminators did not get the best results.

In contrast, the model with multiple discriminators had better results because the model with various discriminators had more training data of different music styles, which had commonality in harmony. The experimental results show that the generators of the multitask learning approach have better performance in calculating the policy reward. It is verified that the multitask learning approach helps share the tip among different tasks.

The validity of the methods in this paper was verified experimentally. The best performance was achieved by comparing two conditional autoregressive generative models with two VAE latent variable models under the same experimental conditions. The empirical evidence shows that the models in this paper are easy to train and optimize and have good generalization ability. In the interactivity method of the generative model, the contour violation rate of the



(a)



(b)

FIGURE 9: Comparison of different generators in terms of harmonic performance.

generated samples by the statistical model and the manually constructed label sequence generation tests verify that the model in this paper can effectively control the contour features of the generated samples; through the latent variable spherical interpolation generation experiments in space, obvious transitions of the melodic global feature attributes are observed, and the changes of these attributes have good independence relative to the contour features. Compared with existing conditional autoregressive generation models and latent variable generation models, this model achieves a more flexible and effective interactive melody generation. In terms of the quality of the generated samples, a comparison with existing noninteractive generation methods proves that this model can generate good music, and the reliability of the generated samples is demonstrated by analyzing the music rationality of many generated pieces.

## 5. Conclusion

In today's pop music production and creation, pop music arranging, which is responsible for creating all musical elements of a pop song beyond the vocal melody and lyrics, has become a crucial part of artistic creation. The professional group that performs this creative act, that is, the pop music arrangers, has also become an essential group of creators in the industry and the specific practice of music creation. This paper focuses on HPC cluster programming-based vocal intelligent arranging, that is, HPC cluster programming theory and methods to implement symbolic vocal astute arranging. Compared with traditional machine learning methods for automatic arrangement algorithms, the HPC cluster programming approach does not need to rely on specialized domain knowledge and manual design features. This paper proposes a VAE melody generation model based on contour control in melody generation. The model defines a sufficient explicit control condition for regulating the local contour features of the generated melody and uses VAE latent variable inference to achieve implicit encoding of all attributes other than sample contour attributes, compensating for the former's shortcoming of incomplete encoding of melodic information. The model in this paper finally provides the user with two control interfaces, melody contour labels and latent variables. The characteristic attributes of the model-generated melodies can be flexibly and effectively controlled.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Academy of Music, Capital Normal University.

## References

- [1] B. Ai, A. F. Molisch, M. Rupp, and Z. D. Zhong, "5G key technologies for smart railways," *Proceedings of the IEEE*, vol. 108, no. 6, pp. 856–893, 2020.
- [2] Z. Yuxiao, Y. Yujie, and W. Zijun, "Design and research of interactive intelligent melody arrangement system through deep learning and artificial neural networks[C]," in *Proceedings of the 2021 IEEE international conference on emergency science and information technology (ICESIT)*, pp. 567–570, IEEE, Chongqing, China, November 2021.
- [3] L. Wang, Z. Liu, A. Liu, and F. Tao, "Artificial intelligence in product lifecycle management," *International Journal of Advanced Manufacturing Technology*, vol. 114, no. 3-4, pp. 771–796, 2021.
- [4] T. J. Saleem and M. A. Chishti, "Data analytics in the internet of things: a survey," *Scalable Computing: Practice and Experience*, vol. 20, no. 4, pp. 607–630, 2019.
- [5] L. Chao, X. Peng, Z. Xu, and L. Zhang, "Ecosystem of things: hardware, software, and architecture," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1563–1583, 2019.
- [6] L. Abualigah, A. Diabat, and M. A. Elaziz, "Intelligent workflow scheduling for Big Data applications in IoT cloud computing environments," *Cluster Computing*, vol. 24, no. 4, pp. 2957–2976, 2021.
- [7] K. Wahengbam, M. P. Singh, K. Nongmeikapam, and A. D. Singh, "A group decision optimization analogy-based deep learning architecture for multiclass pathology classification in a voice signal," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8100–8116, 2021.
- [8] K. Zhang, H. Ying, H. N. Dai et al., "Compacting deep neural networks for internet of things: methods and applications," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11935–11959, 2021.
- [9] M. L. Marsal-Llacuna, "How to succeed in implementing (smart) sustainable urban Agendas keep cities smart, make communities intelligent," *Environment, Development and Sustainability*, vol. 21, no. 4, pp. 1977–1998, 2019.
- [10] K. R. Scherer, "Theory convergence in emotion science is timely and realistic," *Cognition & Emotion*, vol. 36, no. 2, pp. 154–170, 2022.
- [11] R. M. Borges, S. M. Colby, S. Das et al., "Quantum chemistry calculations for metabolomics," *Chemical Reviews*, vol. 121, no. 10, pp. 5633–5670, 2021.
- [12] K. Dichev, D. De Sensi, D. S. Nikolopoulos, K. W. Cameron, and I. Spence, "Power Log'n'Roll: power-efficient localized rollback for MPI applications using message logging protocols," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 6, pp. 1276–1288, 2022.
- [13] R. Mittal, S. Arora, V. Bansal, and M. P. S. Bhatia, "An extensive study on deep learning: techniques, applications," *Archives of Computational Methods in Engineering*, vol. 28, no. 7, pp. 4471–4485, 2021.
- [14] S. R. Jordan, "Challenges of artificial intelligence review in a soft law environment," *IEEE Technology and Society Magazine*, vol. 40, no. 4, pp. 57–67, 2021.
- [15] S. Verma and A. Bala, "Auto-scaling techniques for IoT-based cloud applications: a review," *Cluster Computing*, vol. 24, no. 3, pp. 2425–2459, 2021.
- [16] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.

- [17] M. Humayun, "Industry 4.0 and cyber security issues and challenges[J]," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 2957–2971, 2021.
- [18] T. Bui, D. Cooper, J. Collomosse et al., "Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain," *IEEE Transactions on Multimedia*, vol. 22, no. 11, pp. 2858–2872, 2020.
- [19] S. Y. Joshi and S. A. Deshmukh, "A review of advancements in coarse-grained molecular dynamics simulations," *Molecular Simulation*, vol. 47, no. 10-11, pp. 786–803, 2021.
- [20] D. Pérez, I. Gasulla, P. Das Mahapatra, and J Capmany, "Principles, fundamentals, and applications of programmable integrated photonics," *Advances in Optics and Photonics*, vol. 12, no. 3, pp. 709–786, 2020.
- [21] D. Fennema Galparsoro, X. Zhou, A. Jaaloul, F. Piccirilli, V. Vetri, and V Fodera, "Conformational transitions upon maturation rule surface and pH-responsiveness of  $\alpha$ -lactalbumin microparticulates," *ACS Applied Bio Materials*, vol. 4, no. 2, pp. 1876–1887, 2021.
- [22] F. Belviso, V. E. P. Claerbout, A. Comas-Vives et al., "Viewpoint: atomic-scale design protocols toward energy, electronic, catalysis, and sensing applications," *Inorganic Chemistry*, vol. 58, no. 22, pp. 14939–14980, 2019.