ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

A Dynamic Interaction-oriented Reinforcement Learning Framework for AI-Assisted Piano Improvisation Accompaniment - From Melodic Adaptation to Stylistic Evolution

Yilin Wang[1,*]

[1] Conservatory of Music, Taizhou College of Nanjing Normal University, Taizhou, Jiangsu, 225300, China

SUMMARY: *Piano improvisation accompaniment ability, as an important part of piano education, lacks scientific and effective teaching means and cannot be well adapted to the intelligent era. In this paper, we focus on melodic adaptation and stylistic evolution to build a dynamic interactive intelligent piano accompaniment generation framework. The framework is based on the extraction of piano music features, based on the Mido library to realize the preliminary analysis of MIDI files, combined with the event information of different tracks to obtain the note feature matrix. Then, oriented to classical reinforcement learning RLTUNER baseline model, LSTM gates are used as inputs for optimization to generate piano accompaniment melodies. On the other hand, in this paper, rhythmic features are used as the stylistic representation of piano music, and Actor and Critic are used as the generative and discriminative networks to construct the rhythm generation mechanism, respectively. It is found that the generative framework of this paper has high accuracy and strong generalization ability in multi-track processing, with an accuracy rate of 89.78% and an error rate of only 0.047% in the test set. Meanwhile, the accompaniment music generation results effectively balance originality and recognizability, and are recognized by the audience.*

KEYWORDS: *piano improvisation accompaniment; reinforcement learning; RLTUNER baseline model; LSTM; melody generation*

1 Introduction

Piano improvisation accompaniment is a form of musical expression that requires a high degree of on-the-spot creative ability from the performer. It requires the performer to complete the understanding of the melody, the configuration of harmony, the weaving of rhythm and the shaping of acoustic effects in an instant, and to instantly transform the artistic conception into a fluent musical language [1, 2]. This process combines technical proficiency, knowledge of music theory, aesthetic intuition and improvisational inspiration, and its complexity and creativity constitute a unique artistic charm, but also poses a serious challenge to the performer [3]. Traditional improvisation accompaniment is highly dependent on the player's personal experience accumulation, instantaneous reaction and creativity play, and there are limitations in style expansion, efficiency enhancement, and artistic expression breakthrough [4, 5]. The rapid development of artificial intelligence technology, especially the breakthroughs in the fields of music information retrieval, pattern recognition, deep learning and real-time audio processing, has opened up new possibilities for music creation and performance. For piano

*wangyilin706@126.com
https://doi.org/10.65102/is2026377
Published: 30 April, 2026

improvisation accompaniment, AI technology shows a powerful empowering potential, which can not only deeply analyze massive musical vocabulary and provide rich stylistic references and emotional guidance for accompaniment, but its generative model can also creatively integrate musical elements and provide novel accompaniment ideas [6-9]. The real-time interactive capability of AI is expected to reshape the live experience of accompaniment and realize the collaborative co-creation of man and machine.

AI-assisted piano improvisation accompaniment technology refers to an automated accompaniment system based on AI algorithms that dynamically generates musical expressiveness and structural fit based on instantly parsing the multidimensional signal flow of human piano performance (including note sequences, rhythmic rhythms, intensity levels, and pedal control parameters) [10-12]. Its core technical characteristics are characterized by a threefold dimension. Timing dependence, the system needs to complete the whole process of performance signal capture→semantic parsing→response generation within milliseconds delay, which constitutes a limit constraint on computational efficiency and model lightweight design [13]. Bidirectional interaction, the technical framework is a closed-loop feedback system, the improvisational changes of the player will trigger the adaptive adjustment of the AI, while the harmonic weave and rhythmic patterns generated by the AI will also guide the player's subsequent decision-making, and construct a bidirectional coupling of creative synergistic mechanisms [14, 15]. The expressive decoupling ability, beyond the traditional automatic accompaniment of mechanical rules mapping, deep neural networks through the implicit layer to capture the performance of the difficult to quantify the emotional intent, to achieve from the physical signals to the musical semantics of the cross-modal migration [16, 17].

The so-called piano improvisational accompaniment refers to the use of the piano to quickly and effectively make accompaniment arrangements for music without accompanying voices, only with the help of melodic voices, in order to play the role of complementing and setting off the song [18-20]. With the development of information technology, scholars and experts have proposed many improvement directions for the limitations of traditional improvisational accompaniment. For example, the automated accompaniment system proposed by Xia, G and Dannenberg, R utilizes features such as temporal control and modification complexity by learning example performances, and subjective evaluations through surveys have shown that accompaniment generated by models for specific measurements is more musical, interactive, and naturalistic [21]. Raphael, C utilized Bayesian belief network technology to construct an efficient accompaniment model designed to enhance the intelligence of musical accompaniment, which adjusts the tempo and performance of the accompaniment through continuous monitoring of the performance data to make the overall performance more coherent and expressive [22]. Chacón, C et al. proposed an expressive piano accompaniment system for MIDI input that tracks the soloist's playing position and automatically adjusts the tempo, enabling expressive control of intensity, tempo, and playing details [23]. Papakostas, M et al. proposed an adaptive improvisational accompaniment system based on differential evolution and genetic algorithms, which is capable of providing real-time automated accompaniment based on the player's playing style, creating an unrestricted improvisational environment without prior knowledge of his or her intentions [24]. Li, C integrated the concept of “Internet+” into the teaching of piano improvisation accompaniment course, pointing out that the core of the teaching of improvisation accompaniment is to cultivate students' musical expressiveness, improvisation response ability, as well as the ability to understand and apply different musical styles [25]. Liu, H et al. proposed an improvisational piano accompaniment system based on BP (back-propagation) neural network.The core of improvisational accompaniment lies in its high degree of flexibility and creativity, which is able to adjust the music content in real time according to the changes of the dance movements and enhance the artistic effect of the overall

performance [26].

Deep learning algorithms constitute the core driving engine of the AI real-time accompaniment system, and their technical implementation focuses on solving the bi-directional mapping problem between performance intent decoding and dynamic response generation [27]. Niu, H pointed out that the combination of deep learning (LSTM) algorithms and genetic algorithms in the field of music composition and automatic accompaniment generation could provide new solutions for realizing efficient and intelligent music accompaniment [28]. Kritsis, K et al. used Recurrent Neural Networks (RNN) to implement an improvisational accompaniment system, which aims to analyze the soloist's intention to play and automatically generate the corresponding chordal accompaniment, thus enriching the expressiveness of the performance as well as providing a new technological tool for music composition and performance [29]. Castro, P. S 'research found that most deep learning models are still in "offline" processing mode, which means they require a relatively long computing time when generating accompaniment and are not suitable for improvisations. Therefore, two challenges need to be addressed: "real-time requirements" and "synchronization of rhythm/harmony" [30]. Kitani, K. M and Koike, H proposed an online generation algorithm called "ImprovGenerator", which utilizes a hybrid model combining random context-free grammar (SCFG) and transition probability model to generate accompaniment patterns. Experimental results show that this system can capture the theme melody. And respond according to the current performance rhythm sequence [31].Jiang, N. et al. proposed deep reinforcement learning to generate music accompaniment in real time in online interactive human-computer duet improvisation. This technique can create both melodies and harmonies, and in subjective evaluation studies, it is shown that the output provided by this technique is higher in quality than that produced by the benchmark techniques. [32].

As a subfield of machine learning, reinforcement learning (RL) has been a hot topic in the research of improvisational accompaniment systems in recent years, and its basic principle is to let the model learn by interacting with the environment [33]. The models in RL are also called intelligences because they judge the actions to be performed according to the current environment like human beings, and optimize their own action strategies through continuous trial-and-error-feedback loops [34-36]. Regarding the research on RL in improvisational accompaniment, Jiang and J proposed a song accompaniment generation method that combines audio analysis and symbolic music generation. This method adopts the RL model to extract key musical concepts from the audio and uses neural networks to evaluate the quality of the works and determine the quality of the pieces. The experimental results show that the accompaniment arrangement generated by this algorithm is extremely similar in style and structure to the works of human composers, demonstrating a high degree of musical creativity and diversity, and is capable of adapting to a variety of different musical styles [37]. Wu, Y et al. proposed an online generative model (ReaLchords) for generating improvised chordal accompaniments for user melodies.The core architecture of the model was pre-trained based on Maximum Likelihood Estimation (MLE), and during the fine-tuning phase, RL techniques were introduced to further optimize the model performance [38]. Collins, N has adopted a novel symbolic interactive music system called "Improvagent", aiming to enhance the management and application of dynamic state-action case libraries in MIDI piano improvisation. By integrating RL algorithms, this system can perform autonomous learning and optimization in a real-time performance environment. This enhances the expressiveness and interactivity of the performance [39]. The improvisational accompaniment system proposed by Assayag, G et al. uses a hybrid architecture that combines the use of Max and OpenMusic for real-time processing, employs sequence modeling and statistical learning to generate musical sequences, and employs RL techniques with rewards and policy updating to improve accompaniment generation and

interaction [40].

In this paper, we first build the Mido library for piano improvisation music accompaniment, load the MIDI music file in Python environment, and read the feature information of track header and block. Calculate the note time interval and extract the musical features such as pitch and pitch length. Then, RLTUNER is selected as the baseline model for music generation, and LSTM is used to replace the original input gates, and the internal structure of LSTM applicable to music features is redesigned. Reinforcement learning sequence decision is used to replace the generation problem, and the melody generation adaptation module for piano improvisation accompaniment is constructed. Finally, the stylistic evolution problem is materialized as a rhythmic generation requirement, and the baseline model RLTUNER is again adapted. The Actor and Critic networks are connected in series in the RewardNet return network to compute the rhythmic sequence that has been initialized, and the probability distribution of rhythmic time values is generated using sequence learning. This can be done in such a manner that the appropriate rhythm-time values are selected at random according to that distribution. The research has achieved the AI-assisted piano accompaniment task of melodic adaptation and style evolution at this point.

2 Technical framework for AI-assisted piano improvisation accompaniment

Reinforcement learning has a powerful dynamic learning capability that can facilitate AI's simulation of melodic and rhythmic matching for piano playing. In piano improvisation accompaniment, both melodic adaptation and stylistic evolution can be translated into corresponding reinforcement-demanded learning tasks.

2.1 Piano music feature extraction

2.1.1 MIDO library

MIDI objects, or Mido, is a Python library that processes MIDI messages and MIDI ports. It is meant to be brief and convenient and since it was written in Python, it can be seen as a useful set of tools to parse MIDI files in this language.

The Mido library can be installed by typing and running the following command in a terminal: pip install mido. mido's message contains a number of different parameters, each of which represents information about the musical characteristics of a different piece of music.

“program_change” is a parameter indicating the instrument timbre of a different channel, in the format: Message(‘program_change’, channel, program, time=0). Where channel is a value from 0 to 23, corresponding to each of the 24 channels provided by the MIDI file, which selects the channel number of the instrument; program corresponds to a different instrument number.

“note_on” indicates the beginning of a note, its format is: Message (‘note_on’, note, velocity, time, channel). Where note is a number in the range 0~255, indicating the pitch of the note; velocity indicates the intensity of the note; time is a time variable indicating the time since the end of the previous note; channel represents the number of the channel.

The “note_off” parameter, representing the end of the note, is generally followed immediately after the note_on message, and its format and the meaning of the parameter are the same as those of “note_on”.

2.1.2 Note feature matrix extraction

The note feature matrix is a matrix tabular representation of the note features of a piece of music in chronological order, which presents a sequence of common note information features of a piece of music in an intuitive form, including pitch, acoustics and duration. The Mido library can be used to extract the relevant musical features in MIDI files, and the steps for extracting and generating note feature matrices from MIDI files are shown in Figure 1.

First, load the MIDI music file in the Python environment, and realize the preliminary parsing of the MIDI file based on the Mido library to get the basic information of the track header and block. Secondly, determine whether the number of tracks is 1 or not, if the number of tracks in the file is greater than 1, then only read the first track and parse its Meta event information. If the number of tracks in the file is equal to 1, then read this track directly and parse its Meta event information. After that, the track header is judged, and the note time interval delta-time is calculated to determine the start and end moments of each note. For each block of a MIDI file that has recorded note events, it is parsed and basic musical features such as pitch, intensity and duration are extracted. Finally, the note event information is selected and arranged into a matrix in chronological order, so that the note feature matrix is obtained.

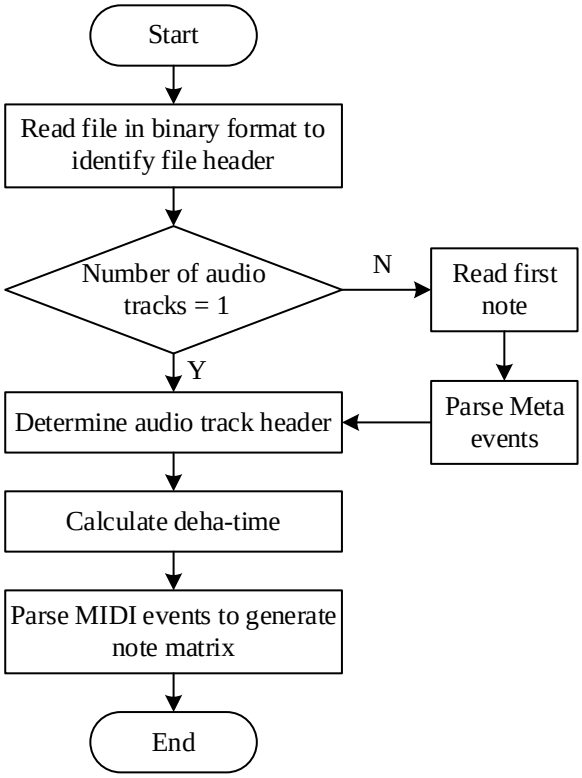


Figure 1: Extraction of the feature matrix of musical notes

To summarize, the implementation process of extracting and generating note feature matrices from MIDI files consists of steps including:

(1) Load the MIDI music file in Python environment, realize the preliminary parsing of the MIDI file based on Mido library, and get the basic information of track header and block.

(2) Determine the track header according to the different number of tracks, calculate the time interval delta-time, and determine the start and end moments of each note.

(3) Analyze the MIDI event information of all the track blocks, select the required part of the note event information and arrange it into the form of matrix table in chronological order to

get the note feature matrix.

At this point, the extraction of the note feature matrix is realized. The note feature matrix contains every note information of the music, including pitch, intensity, duration and other features, which can be used to extract the required musical features directly.

2.2 Melody Generation Model for Piano Accompaniment

2.2.1 RLTUNER baseline modeling

Traditional supervised learning methods often suffer from the problem of focusing on learning the current information without sufficient consideration of historical information, and thus are not applicable to music, a long time sequence generation task. Reinforcement learning is applied to the sequential decision problem by designing the reward function, setting the goal of learning to maximize the long-term gain, and can learn the intrinsic laws of music without any a priori knowledge. Because of this, reinforcement learning is very suitable for music generation scenarios. In RLTUNER, the music generation problem is treated as a sequential decision problem using the classical DQN algorithm. Its structure is shown in Fig. 2 and contains an already trained NoteRNN, a pair of Q-networks and a reward function module.

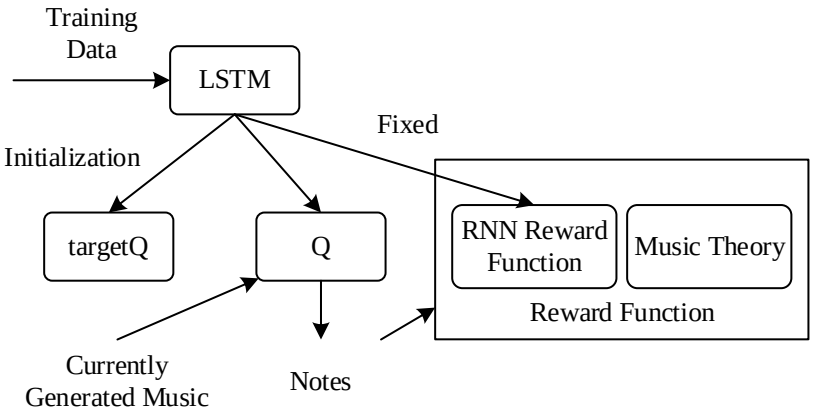


Figure 2: RLTUNER model structure

2.2.2 Improvements to RLTUNER

LSTM is a classical sequence task algorithm, similar to Character RNN, in music generation, the LSTM model makes a prediction each time based on the information memorized earlier in combination with the input of the present moment and saves this time information in combination with the previous information for the prediction of the next note.

The inputs to the LSTM gates are the time-step data at the current time-step as well as the hidden state in the previous time-step. The signals go through three fully connected layers and a sigmoid activation function to compute the gate values and output gate activation. The NoteRNN implemented in this research is the same as the melodyRNN of Magenta, which involves one LSTM layer and one fully connected layer. Fig. 3 shows the LSTM architecture used in this work. In reinforcement learning, this study defines the environment at moment t as the currently generated melody $env = x_0, x_1, \cdots, x_{t-1}$, and since the LSTM can take into account the antecedent information every time it generates, the state of the interior of the LSTM is considered as the environment at moment t, which converts the whole music generation problem into a question asking the reinforcement learning sequential decision-making problem.

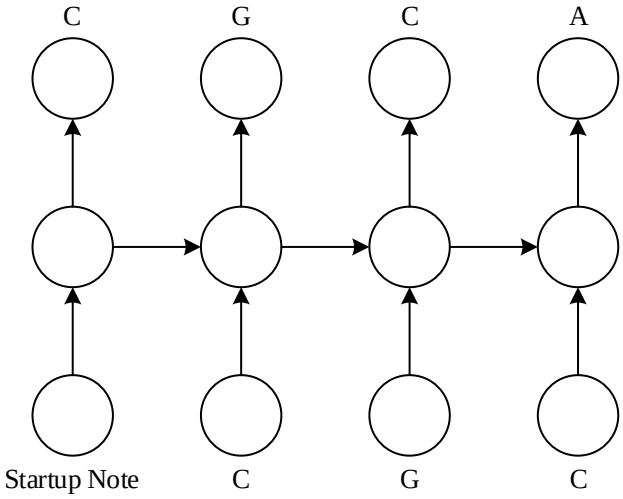


Figure 3: Schematic diagram of music generation based on LSTM model

In reinforcement learning, an agent interacts with its environment. Given an environmental state s_t at time t, the agent selects an action a_t according to the policy $\pi(a_t \mid s_t)$ and then obtains a reward $r(s_t, a_t)$. The environment subsequently moves to a new state s_{t+1}. The agent executes a sequence of decisions to maximize cumulative reward, where γ denotes the discount factor applied to future returns. The optimal policy π^* should satisfy the Bellman equation:

$$Q(s_t, a_t; \pi^*) = r(s_t, a_t) + \gamma \mathbb{E}_{p(s_{t+1}|s_t, a_t)} \left[\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \pi^*) \right] \tag{1}$$

where $Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi} \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_t, a_t) \right]$ is a function of Q in the strategy π. The Q-learning method minimizes the Bellman residuals by iteration. The optimization strategy is implemented by $\pi^*(a \mid s) = \text{argmax}_a Q(s, a)$. Deep Q-learning uses a deep neural network Q network to estimate $Q(s, a; \theta)$. This network parameter θ is updated by stochastic gradient descent (SGD) with the following loss function:

$$L(\theta) = \mathbb{E}_{\beta} \left[\left(r(s, a) + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \tag{2}$$

where β is the exploration strategy and θ^- is a parameter of the targetQ network, which is fixed in the gradient computation and periodically updated with θ during training, the exploration strategy usually uses random sampling or Boltzmann sampling. Additional standard techniques such as playback pooling and DDQN are used to stabilize and enhance learning. In the baseline model, the reward of the reward function is designed as follows:

$$r_t = \ln p(a \mid s) + r_{MT}(a, s) / c \tag{3}$$

where r_t denotes the reward at the moment of t, $\ln p(a \mid s)$ denotes the reward for choosing the action a when the state is s, r_{MT} denotes the reward of the currently generated sequence based on the manual music rules, and c is a constant controlling the reward of the music rules.

2.3 Piano Accompaniment Rhythm Generation Modeling

2.3.1 Problem definition

Music generation task definition: Let the note sequence be $S_n = \{n_1, n_2, ..., n_L\}$ and the rhythm sequence be $S_r = \{r_1, r_2, ..., r_L\}$ where $n_i = \{s_1, s_2, ..., s_N\}, (i = 1, 2, ...L)$. In this case, L is the length of a note sequence, and N is the number of notes sounding at a particular time in that sequence. If $N = 1$, the output is one note and if , the output is one chord note. First, the note sequence S_n and S_r rhythm sequence are encoded and then translated into model-input data as follows:

$$S_n^E = Encoder(S_n) = MultiHot(S_n) = \left\{n_1^{mh}, n_2^{mh}, ..., n_L^{mh}\right\} \tag{4}$$

$$S_r^E = Encoder(S_r) = OneHot(S_r) = \left\{r_1^{oh}, r_2^{oh}, ..., r_L^{oh}\right\} \tag{5}$$

Since it is necessary to generate polyphonic melodies, the note sequence S_n is encoded with multi-hot, while the rhythm sequence needs only one duration information at the same moment, so the rhythm sequence S_r is encoded with one-hot. Let the final note sequence generated by the model be $S_n^g = \left\{n_1^g, n_2^g, ..., n_L^g\right\}$, the rhythmic sequence is $S_r^g = \left\{r_1^g, r_2^g, ..., r_L^g\right\}$, and finally the rhythmic sequence S_r^g and the note sequence S_n^g are combined to obtain the complete melody $S_m^g = \left\{\left\{r_1^g, n_1^g\right\}, \left\{r_2^g, n_2^g\right\}, ..., \left\{r_L^g, n_L^g\right\}\right\}$.

2.3.2 Rhythm models

The label of the current duration value in the training set used to define the rhythm model is the next duration value in the rhythm sequence. Figure 4 illustrates the training process of the rhythm model, where $x = S_r = \{r_1, r_2, ..., r_L\}$ is the rhythmic sequence in the dataset and and and h and c are the hidden state and cell state of the LSTM respectively. Suppose that the output of the LSTM is O_{lstm}, thus $O_{lstm} = h_t^2$, where h_t^2 is the output of the second LSTM layer. The D_n is constructed as a tensor of , where O_{lstm} is the number of neurons in the LSTM network and $batch\_size * D_r$ is linearly transformed into a tensor of , O_{linear} , where D_r is the rhythm species number.

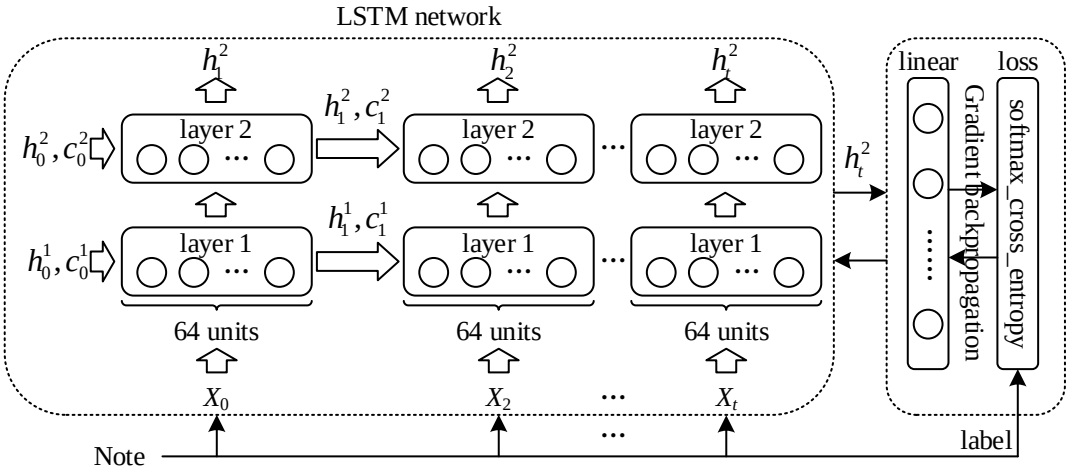


Figure 4: Rhythm training model

$$O_{linear} = O_{lstm} * w^T + b,\ w \in \mathbb{R}^{D_r * D_n},\ b \in \mathbb{R}^{D_r} \tag{6}$$

O_{lstm} is softmaxed and then cross entropy is calculated as the loss function of the model.

$$\begin{aligned} loss &= soft\max\_cross\_entropy\left(O_{linear}\right) \\ &= cross\_entropy\left(soft\max\left(O_{linear}\right)\right) \\ &= -\sum_{i=0}^{D_r} y_i \log\left(\frac{\exp\left(O_{linear}^{i}\right)}{\sum_{j=0}^{D_r}\exp\left(O_{linear}^{j}\right)}\right) \end{aligned} \tag{7}$$

where y_i is the label corresponding to input x_i.

At the beginning, the trained parameter file is loaded to initialize the model, and either an initial duration value r_{init}, an initial rhythm sequence $S_r = \{r_1, r_2, ..., r_N\}$, or a randomly chosen duration value r_{random} is used as the starting duration information for rhythmic sequence generation; meanwhile, the target length L of the generated rhythm sequence is specified. The initial note r_{init} is processed by the LSTM network to produce O_{lstm}. After that, O_{lstm} is mapped by a linear layer to obtain O_{linear}, and O_{linear} is further fed into the softmax module to yield the probability distribution over rhythmic time values. The corresponding rhythmic duration is then sampled according to that probability distribution. The computational flow of rhythm generation is as follows:

$$O_{linear} = O_{lstm} * w^T + b,\ w \in \mathbb{R}^{D_r \times D_n},\ b \in \mathbb{R}^{D_r} \tag{8}$$

$$O_{soft\max} = soft\max\left(O_{linear}\right) \tag{9}$$

After L iterations, the model outputs a rhythmic sequence $S_r^g = \{r_1, r_2, ..., r_L\}$, and the final processing module transforms S_r^g into a MIDI music file for output.

Until the Actor and Critic networks are trained, the reward network RewardNet needs to be pre-trained. RewardNet has an architecture similar to the rhythm generation model; however,

because it is necessary to focus more on important notes of the note sequence and capture them more efficiently, an Attention module is added to the RewardNet. In addition, in order to allow polyphonic melody generation, the activation function in the rhythm generation model is modified to a sigmoid and the loss function is replaced by sigmoid cross-entropy loss. The rhythmic information used to train the payoff network are the rhythmic data obtained based on the original music files and the parameters are saved locally after the training. The training process of the payoff network is as follows.

$$\begin{aligned} O_{linear} &= O_{lstm} * w^T + b,\ w \in \mathbb{R}^{D_m \times D_n},\ b \in \mathbb{R}^{D_m} \\ loss &= sigmoid\_cross\_entropy\left(O_{linear}\right) \\ &= cross\_entropy\left(sigmoid\left(O_{linear}\right)\right) \end{aligned} \tag{10}$$

where D_m represents the number of note classes, and the sigmoid function can be written as $f(x) = \frac{1}{1+\exp(-x)}$.

The sigmoid cross-entropy loss with binary classification is formulated as:

$$\begin{aligned} loss &= -\left[y * \log\left(\frac{1}{1+\exp(x)}\right) + (1-y) * \log\left(1 - \frac{1}{1+\exp(x)}\right)\right] \\ &= y * \log\left(1+\exp(x)\right) + (1-y) * \log\left(x + \log\left(1+\exp(-x)\right)\right) \\ &= (1-y) * x + \log\left(1+\exp(-x)\right) \\ &= x - x * y + \log\left(1+\exp(-x)\right) \end{aligned} \tag{11}$$

where x is the output of the model prior to the application of the activation function and y is the matching label. If $x < 0$ to make sure that x does not become too small leading to an $\exp(-x)$ overflow, Eq. (11) can be written in the following equivalent form:

$$\begin{aligned} loss &= x - x * y + \log\left(1+\exp(-x)\right) \\ &= -x * y + \log\left(1+\exp(x)\right) \end{aligned} \tag{12}$$

When it comes to practice, to ensure the stability of training as well as prevent overflow, the following equivalent form is usually taken up:

$$loss = \max\left(x, 0\right) - x * y + \log\left(1+\exp\left(-abs\left(x\right)\right)\right) \tag{13}$$

In multi-label classification, every label is a separate binary classification task and thus, calculating the sigmoid cross-entropy loss in the multi-label context only involves substituting the scalar values of x and y in the binary-classification equation with vectors.

The Actor network structure of the model is consistent with the structure of the return network and consists of an LSTM network, an Attention module, a Linear layer, and a Sigmoid module. Consisting of LSTM network, Attention module, and two Linear layers, the final Linear layer outputs the Q value corresponding to the action.The Actor and Critic networks of the ACMG model are trained by randomly sampling quaternion data $\left(s_t, a_t, r_{t+1}, s_{t+1}\right)$ from

a pool of empirical playbacks, so empirical collection is required first. First, set the initial note s_{init} as the initial state s_0 of the model. The Actor network outputs action a_0 based on the initial state s_0, and inputs a_0 into the RewardNet network to obtain the value return r_m^1 of a_0. The music theory return module outputs the music theory return r_n^1 corresponding to a_0 based on the music theory rules, and calculates the final return r_{mix}^1 through the formula $r_{mix}^1 = k_m * r_m^1 + k_n * r_n^1$. Then the next state s_1 is obtained based on a_0 and the quaternion $\left(s_0, a_0, r_{mix}^1, s_1\right)$ is placed into the experience playback pool, and so on, until the data in the experience playback reaches the SeedSteps bar.

3 Matching effects generated by AI piano improvisation accompaniment

3.1 Piano music feature extraction results

3.1.1 Piano tone reconstruction test

A piano tone has a fundamental frequency plus additional higher-frequency components that are integer multiples of the fundamental frequency. Pitch is perceived by humans primarily by its fundamental frequency. Through investigation of the measured performance data and waveform characteristics, it is observable that the energy of the fundamental to the fifth harmonic and the waveform behavior in that small frequency range are both representations of the pitch, timbre and energy of the played tone in the frequency domain. In this paper, a function will be designed to simulate the waveform of the corresponding tone in the fundamental-frequency region to the fifth-harmonic region. Given that the fundamental frequency of the provided tone and the amplitude proportional relationship of the first and fifth harmonic frequency peaks are known, the piano timbre is successfully reconstructed. The results of reconstructing the piano timbre are shown in Fig. 5. It can be observed that within the interval $(\omega_i - \alpha_i, \omega_i + \alpha_i)$, the waveforms resemble certain specific functions. Other similar functions that can be experimentally selected are Cauchy function, Gaussian function, sinc function, and Cauchy function multiplied by sine and cosine.

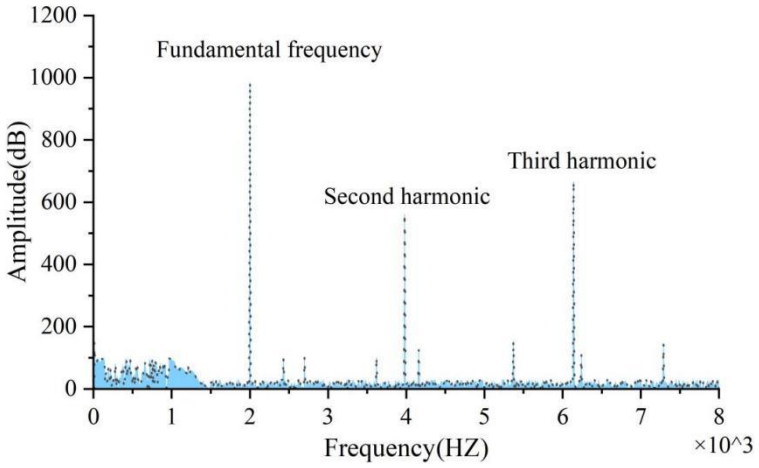


Figure 5: Reconstruction of piano timbre

3.1.2 Audio Feature Extraction Recognition Results

The findings of monophonic pitch tensor extraction in MIDI file extraction are presented in Table 1. The experimental sample consisted of 120 musical excerpts of Chinese and foreign classical music as well as modern styles. The fragments were played by piano students and none of them was longer than 20 bars which means less than one minute long. Selected pieces had a difficulty level of introductory level to amateur piano grade 10 but monophonic melodic lines were used only.

The note frequencies, pitch names and timing values based on the endpoint detection and note recognition are shown in Table 2. The system perfectly recognizes the names of all notes in the musical tone. As to the time-value element, the real goal is to define the delay of every note, specifically, to differentiate between whole notes, half notes, quarter notes and other similar lengths. The length of a quarter note is equal to that of the whole note in the chosen timing conversion, where the quarter-note value is 545.555 ms and the eighth-note value is 289.99 ms. Hence, given the error provided in table 2, as long as the tolerance interval is defined at (26.5%,-26.5), all time values can be corrected. All notes are recognized properly and with the corrected timing values the composition process may be finalized.

Table 1: The extraction results of the pitch and duration of single tones

Musical symbols	Standard frequency (Hz)	Identification frequency (Hz)	Error rate (%)
a	437.375	435.092	0.522
b	494.821	491.551	0.6608
c	261.303	260.7935	0.195
d	293.345	290.7491	0.8849
e	331.523	329.2246	0.6933
f	347.026	350.9806	1.1396
g	390.114	389.4743	0.164

Table 2: Extraction of the pitch duration of the entire musical piece

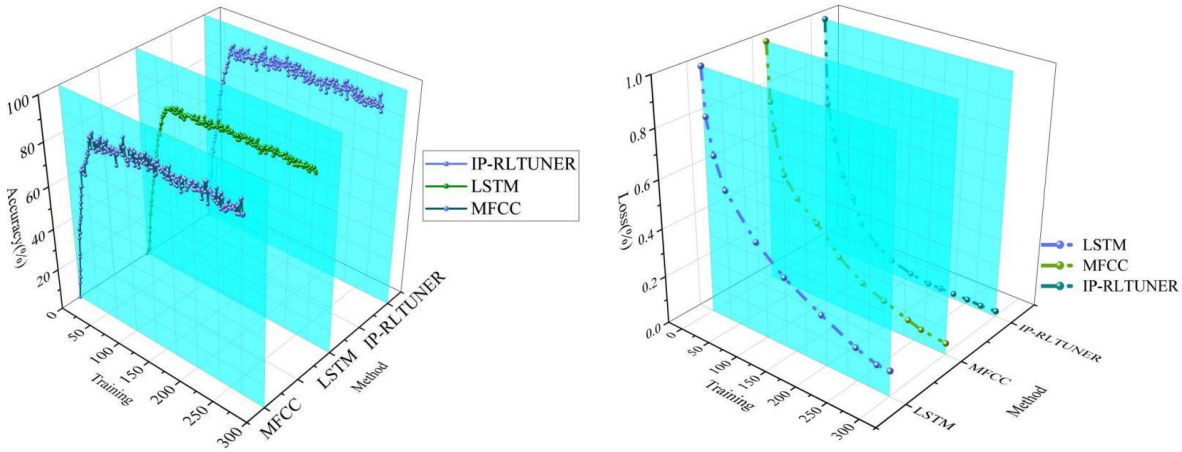
Serial Number	Frequency (Hz)	Musical symbols	Detection duration (ms)	Accurate duration (ms)	Value Error (%)
1	366.036	c	501.97	545.55	-7.99
2	380.069	e	670.29	545.55	22.86
3	315.331	a	530.62	545.55	-2.74
4	326.735	b	461.65	545.55	-15.38
5	294.876	d	440.96	545.55	-19.17
6	283.105	c	561.83	545.55	2.98
7	311.858	f	553.94	545.55	1.54
8	282.765	g	664.69	545.55	21.84
9	285.503	d	627.08	545.55	14.94
10	319.081	c	502.56	545.55	-7.88
11	310.643	f	235.21	289.99	-18.89
12	329.243	f	330.65	289.99	14.02
13	321.795	g	413.88	545.55	-24.14
14	385.694	g	523.52	545.55	-4.04
15	372.512	a	502.67	545.55	-7.86
16	293.023	a	352.25	289.99	21.47
17	339.409	e	549.58	545.55	0.74
18	375.873	c	644.52	545.55	18.14
19	325.602	d	649.65	545.55	19.08
20	359.217	b	483.96	545.55	-11.29

3.2 Piano accompaniment generation analysis

3.2.1 Performance Analysis of Deep Reinforcement Learning Music Generation

In order to validate the music matching and performance generated by the IP-RLTUNER generative model, the study utilizes the Lakh DIMI dataset for simulation experiments. The data was preprocessed before the simulation experiments started and only music with beat number 4/4 was retained. The learning rate of the model was set to 0.005 and the number of iterative training was 300. The Long Short-Term Memory Network (LSTM) and Mel Frequency Cepstrum Coefficient Algorithm (MFCC) were also compared with the generated model. Fig. 6 shows the comparison results of the accuracy and error rate of the three methods in dealing with the music generation process.

From Fig. 6(a), it can be seen that the data processing accuracy of the improved RLTUNER model in this paper is 89.78%, and the data processing accuracy of LSTM and MFCC are 79.34% and 82.68%, respectively. From Fig. 6(b), it can be seen that the smallest error rate is that of this paper's model with an error rate of 0.047%. the data processing error rates of MFCC and LSTM are 0.106% and 0.085%, respectively. This indicates that the generative model has higher robustness and accuracy in the processing of data in the multi-track generation process. In order to further validate the performance of the generative model in the multi-track generation process, the study compares the training loss value and the track matching degree as validation metrics.



(a) Accuracy (b) Error

Figure 6: The result of the music generation process

3.2.2 Evaluation of metrics for generating accompaniments

The results of the quantitative experiments on the piano accompaniment generation task are shown in Table 3. The reinforcement learning network outperforms all baselines in terms of rhythmic consistency, music quality, and generation stability. In the FilmDB dataset, the improved RLTUNER model achieved BCS = 67.54, IS = 4.05, and CSD = 18.63. It is worth noting that the LORIS method was assessed lower on two metrics, BHS and F1scores, compared to other methods. This is because it is designed for dance/movement music generation with a focus on modeling musical rhythms based on video poses/movements. It is not suitable for accompaniment generation tasks that require capturing changes in piano key. Nevertheless, in terms of overall music quality scores, the waveform-based LORIS method outperforms CDCD as well as D2M, and the spectrum-based Tango.This suggests that

generative modeling direct waveform generation better preserves musical details and ensures improved music quality. The model in this paper achieves significant improvements on both datasets compared to other methods.

Table 3: Quantitative experiments on piano accompaniment generation tasks

Data set	Method	Rhythm			Music quality		Generation stability	
		BCS↑	BHS↑	F1↑	IS↑	KL↓	CSD↓	HSD↓
FilmDB	D2M	61.18	58.44	62.43	1.49	5.71	23.92	26.8
	CDCD	57.09	60.96	61.44	1.24	10.11	21.48	24.48
	Tango	61.87	56.7	61.85	3.05	6.94	24.89	21.28
	LORIS	62.34	50.71	60.73	2.95	5.13	21.15	23.65
	This	**67.54**	**63.2**	**66.14**	**4.05**	**4.88**	**18.63**	**20.31**
EmoMV	D2M	60.13	63.01	66.34	1.21	5.43	21.62	21.7
	CDCD	59.91	61.53	62.75	1.77	7.1	19.74	20.22
	Tango	63.22	57.39	64.83	5.02	6.34	22.6	17.65
	LORIS	65.83	53.35	62.29	6.43	6.18	20.94	18.95
	This	**68.61**	**64.1**	**67.35**	**6.59**	**5.02**	**17.52**	**16.54**

Further, the study utilizes visualization methods to demonstrate the music generation effects. The results of this paper's improved reinforcement learning model compared to the baseline in terms of originality and recognizability are shown in Figure 7. Where the large data points represent a single model containing the average scores of originality and recognizability across all categories, while the surrounding smaller data points show the model's scores in each category. In contrast, D2M and CDCD are able to generate music that matches a production-specific style but is similar to the real thing (i.e., low originality but high recognizability). Although they excel at accurately learning the distribution of real data domains, which leads to their high recognizability, they do not do enough to create new pieces of music. This suggests that neither the diffusion model nor Gan can overcome the limitations imposed by Codebook. Waveform-based LORIS is able to continuously create music with distinctive features by learning from the most primitive musical representations. However, this results in generating music that deviates from the distribution of the real data domain, leading to lower recognizability. On a better note, however, IP-RLTUNER and Tango are able to effectively balance the relationship between originality and recognizability, demonstrating the effectiveness of the models in musical innovation and stylistic expression.

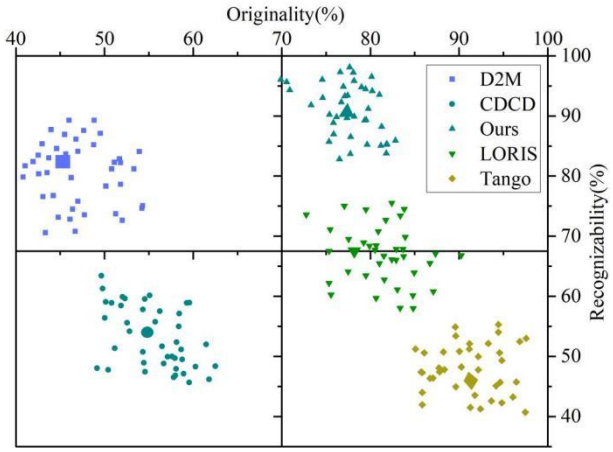
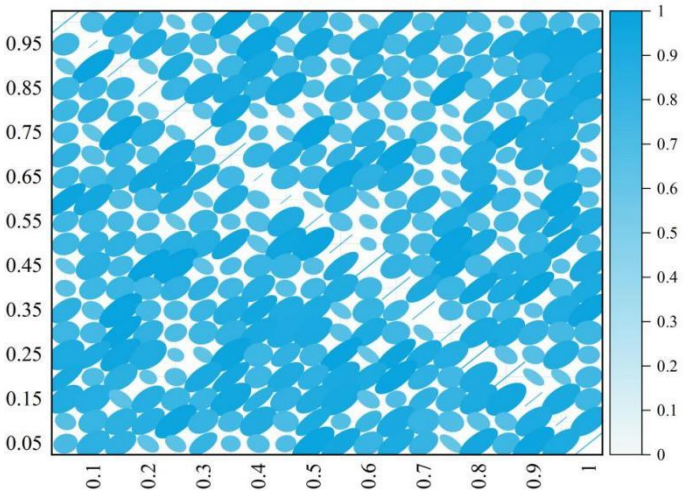


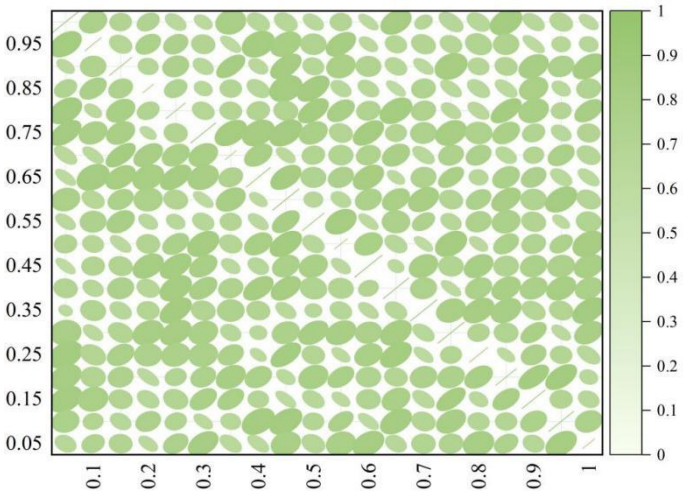
Figure 7: Visual comparison of originality and recognizability

3.3 Stylistic comparison of piano accompaniment generation

There are fewer studies on detailed music comparison, in this paper, we choose Park music comparison model to verify the stylistic similarity of piano accompaniment generated by reinforcement learning. The visualization results of the similarity comparison analysis are shown in Figure 8. Although there is no interpretation and description of the detailed similarity clips, the similarity to the original clips can be very intuitively seen for the generated melodies and rhythms, which achieved 85.52% and 80.14%, respectively. The dynamic interaction-based reinforcement learning framework for the simulation generation of piano music is verified to be matched and effective.



(a)Melody similarity



(b)Rhythm similarity

Figure 8: Visualization results of similarity comparison analysis

4 Conclusion

In this paper, we extend the melodic and rhythmic generation model on the RLTUNER baseline model to provide a practical path for AI-assisted piano improvisation accompaniment, and successfully apply the dynamic sequence evolution capability of reinforcement learning to

music generation. The research conclusions are as follows:

(1) When the error tolerance range is (-26.5%, +26.5%), it can be judged that all the tense values can be effectively adjusted and parsed based on the MIDO library. The method correctly recognizes all the notes and fully extracts the features of piano music.

(2) The accuracy of the reinforcement learning framework in generating melodies in the test set is 89.78%, which is 10.44% and 7.10% higher than the data processing accuracy of LSTM and MFCC, respectively. The error rate is also much smaller than that of the comparison models, reflecting extremely high learning performance and robustness.

(3) The improvement of the piano accompaniment rhythm generation model also achieves good results, with good rhythmic consistency (BCS=67.54), musical quality (IS=4.05) and generation stability (CSD=18.63) in the dataset FilmDB.

Funding

This work was supported by Music Production for the Corporate Promotional Video of Xialin Trading Co.

A Dynamic Interaction-oriented Reinforcement Learning Framework for AI-Assisted Piano Improvisation Accompaniment - From Melodic Adaptation to Stylistic Evolutio

Project Funded by: The First-Class Course "Piano Improvisation Accompaniment" of Taizhou College of Nanjing Normal University (2025), Project No. 141220162521

About the Author

Yilin Wang was born in Xuzhou City, Jiangsu Province, P.R. China in 1986. She holds a master's degree from Nanjing Normal University and currently works as an associate professor at the Conservatory of Music, Taizhou College of Nanjing Normal University, with her main research directions focusing on piano performance and piano improvisational accompaniment.

References

[1] Zeng, Z. (2025). The role of improvisation in piano performance: Cultivate creativity and musical performance. In SHS Web of Conferences (Vol. 222, p. 04018). EDP Sciences.

[2] Vechtomova, O., & Sahu, G. (2023, April). LyricJam sonic: a generative system for real-time composition and musical improvisation. In International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar) (pp. 292-307). Cham: Springer Nature Switzerland.

[3] McCormack, J., Gifford, T., Hutchings, P., Llano Rodriguez, M. T., Yee-King, M., & d'Inverno, M. (2019, May). In a silent way: Communication between ai and improvising musicians beyond sound. In Proceedings of the 2019 chi conference on human factors in computing systems (pp. 1-11).

[4] Wang, Z., & Xia, G. (2018). A framework for automated pop-song melody generation with piano accompaniment arrangement. arXiv preprint arXiv:1812.10906.

[5] Liang, Q., & Zeng, Y. (2021). Stylistic composition of melodies based on a brain-inspired spiking neural network. Frontiers in systems neuroscience, 15, 639484.

[6] Dewanto, A. D., & Sugiharto, A. (2025). Artificial Intelligence Applications in Piano Education: An Informatics-Based Literature Analysis. Polygon: Jurnal Ilmu Komputer dan Ilmu Pengetahuan Alam, 3(4), 91-100.

[7] Chen, H. (2021, September). Application of piano automatic accompaniment system based on artificial intelligence in piano enlightenment education. In 2021 4th International Conference on Information Systems and Computer Aided Education (pp. 1351-1355).

[8] Xiao, P. (2025). Evolutionary algorithm-based system for real-time collaborative music creation and improvisation generation. Journal of Computational Methods in Sciences and Engineering, 14727978251391323.

[9] Wang, Q., Zhang, S., & Zhou, L. (2023, June). Emotion-guided music accompaniment generation based on variational autoencoder. In 2023 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[10] Wang, H., Zhang, X., & Iida, F. (2024). Human-robot cooperative piano playing with learning-based real-time music accompaniment. IEEE Transactions on Robotics.

[11] Yang, Y. (2022, February). Application of Deep Learning Piano Harmony Automatic Arrangement System in Piano Teaching. In 2022 IEEE 5th Eurasian Conference on Educational Innovation (ECEI) (pp. 90-93). IEEE.

[12] Benetatos, C., VanderStel, J., & Duan, Z. (2020, January). BachDuet: A deep learning system for human-machine counterpoint improvisation. In Proceedings of the International Conference on New Interfaces for Musical Expression.

[13] Dai, C. (2023, November). Design of an improvisational singing training system based on machine learning algorithms. In International conference on cognitive based information processing and applications (pp. 213-221). Singapore: Springer Nature Singapore.

[14] Lin, M., & Zhao, R. (2022). A Study of Piano-Assisted Automated Accompaniment System Based on Heuristic Dynamic Planning. Computational Intelligence and Neuroscience, 2022(1), 4999447.

[15] Deja, J. A., Štor, S., Pucihar, I., Weerasinghe, M., Balbin, R. M., Čopič Pucihar, K., & Kljun, M. (2025). ImproVisAR: designing augmented reality piano roll for teaching improvisation. Virtual Reality, 29(3), 140.

[16] Zhang, M. (2025). Advancing deep learning for expressive music composition and performance modeling. Scientific Reports, 15(1), 28007.

[17] Qian, L. (2025). Research and practice on instructional methods for piano improvization based on computer technology. International Journal of High Speed Electronics and Systems, 34(02), 2440080.

[18] Chang, H. (2025, March). Research on the Style Transfer Fusion Model of Piano Improvisation Accompaniment Driven by Artificial Intelligence. In 2025 IEEE International Conference on Electronics, Energy Systems and Power Engineering

(EESPE) (pp. 173-178). IEEE.

[19] Lv, J. (2014, June). Artistic aesthetics significance of piano improvising accompaniment. In 2014 International Conference on Education, Management and Computing Technology (ICEMCT-14) (pp. 107-110). Atlantis Press.

[20] Becker, N., LOUIE, R., THICKSTUN, J., & LIANG, P. (2024). Designing live human-ai collaboration for musical improvisation. In CHI Workshop on Generative AI and HCI (GenAICHI).

[21] Xia, G., & Dannenberg, R. (2017). Improvised duet interaction: learning improvisation techniques for automatic accompaniment. In Proceedings of the International Conference on New Interfaces for Musical Expression (pp. 110-114).

[22] Raphael, C. (2001). A probabilistic expert system for automatic musical accompaniment. Journal of Computational and Graphical Statistics, 10(3), 487-512.

[23] Cancino-Chacón, C., Bonev, M., Durand, A., Grachten, M., Arzt, A., Bishop, L., ... & Widmer, G. (2017). The ACCompanion v0. 1: an expressive accompaniment system. arXiv preprint arXiv:1711.02427.

[24] Kaliakatsos-Papakostas, M. A., Floros, A., & Vrahatis, M. N. (2012, November). Intelligent real-time music accompaniment for constraint-free improvisation. In 2012 IEEE 24th International Conference on Tools with Artificial Intelligence (Vol. 1, pp. 444-451). IEEE.

[25] Li, C. (2022). Innovative application of the teaching mode of piano impromptu accompaniment course under the perspective of" Internet+". Advances in Engineering Technology Research, 1(3), 161-161.

[26] Liu, H. (2022, November). Improvisational Dance Piano Accompaniment System Based on BP Neural Network. In 2022 International Conference on Computers and Artificial Intelligence Technologies (CAIT) (pp. 21-25). IEEE.

[27] Wang, X. (2025, August). CNN-Transformer architecture for piano performance style recognition and AI-based real-time music accompaniment. In Seventh International Conference on Image, Video Processing, and Artificial Intelligence (IVPAI 2025) (Vol. 13731, pp. 151-158). SPIE.

[28] Niu, H. (2023, April). Accompaniment Generation Based on Deep Learning and Genetic Algorithm. In 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT) (pp. 58-65). IEEE.

[29] Kritsis, K., Kylafi, T., Kaliakatsos-Papakostas, M., Pikrakis, A., & Katsouros, V. (2021). On the adaptability of recurrent neural networks for real-time jazz improvisation accompaniment. Frontiers in artificial intelligence, 3, 508727.

[30] Castro, P. S. (2019). Performing structured improvisations with pre-trained deep learning models. arXiv preprint arXiv:1904.13285.

[31] Kitani, K. M., & Koike, H. (2010, June). ImprovGenerator: Online Grammatical

Induction for On-the-Fly Improvisation Accompaniment. In NIME (pp. 469-472).

[32] Jiang, N., Jin, S., Duan, Z., & Zhang, C. (2020, April). Rl-duet: Online music accompaniment generation using deep reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 01, pp. 710-718).

[33] Guo, H. (2025). Piano harmony automatic adaptation system based on deep reinforcement learning. Entertainment Computing, 52, 100706.

[34] Smith, B. D., & Garnett, G. E. (2012, April). Reinforcement learning and the creative, automated music improviser. In International Conference on Evolutionary and Biologically Inspired Music and Art (pp. 223-234). Berlin, Heidelberg: Springer Berlin Heidelberg.

[35] Alrowais, F., Arasi, M. A., Alotaibi, S. S., Alonazi, M., Marzouk, R., & Salama, A. S. (2025). Deep gradient reinforcement learning for music improvisation in cloud computing framework. PeerJ Computer Science, 11, e2265.

[36] Hu, X. (2023). Reinforcement learning-based algorithms for music improvisation and arrangement in sensor networks for the Internet of Things. Scalable Computing: Practice and Experience, 24(3), 499-510.

[37] Jiang, J. (2023, March). DJ-agent: Music theory directed a cappella accompaniment generation using deep reinforcement learning. In Fifth International Conference on Computer Information Science and Artificial Intelligence (CISAI 2022) (Vol. 12566, pp. 971-979). SPIE.

[38] Wu, Y., Cooijmans, T., Kastner, K., Roberts, A., Simon, I., Scarlatos, A., ... & Huang, C. Z. A. (2025). Adaptive accompaniment with ReaLchords. arXiv preprint arXiv: 2506.14723.

[39] Collins, N. (2008, August). Reinforcement learning for live musical agents. In ICMC.

[40] Assayag, G., Bloch, G., Chemillier, M., Cont, A., & Dubnov, S. (2006, October). Omax brothers: a dynamic yopology of agents for improvization learning. In Proceedings of the 1st ACM workshop on Audio and music computing multimedia (pp. 125-132).