Application of Deep Learning in Multi-Style Dance and Music Matching Choreography

Jue Rong[1*]

[1]Minxi Vocational and Technical College,Longyan City,Fujian Province,China

[*]Corresponding author:Jue Rong   a_stcld@163.com

Abstract: In recent years, with the development of the digital era, music to dance generation research has received extensive attention from industry and academia, and has become one of the basic tasks in the cross-modal field, which can be widely used in entertainment, education, virtual and other fields, and has a good application prospect. Based on deep learning, the choreography system is proposed in this paper. In this paper, a novel automatic music choreography system is proposed, aiming to realize efficient matching and generation of music and dance movements through deep learning models. In the study, a bidirectional cyclic gating unit is used as the core network structure, and multi-temporal modeling is carried out for the timing characteristics of the dance movements. Two standard motion capture datasets, Laban-16 and Laban-48, were used in the experiments to verify the effectiveness of the proposed algorithm. The accuracy of the model in continuous motion recognition is significantly improved on the Laban-16 dataset, with a recognition rate of 72.79%, demonstrating strong generative ability and innovation, and the recognition rate on the Laban-48 dataset is 68.92%. In addition, the experimental results show that the multi-temporal-based algorithm outperforms the traditional joint features in motion recognition performance, verifying the effectiveness of feature selection.

Keywords: Deep Learning; Bi-Gru; Multi-Temporal; Choreography

1. Introduction

Dance is a kind of performing art that shows various ornamental movements, usually accompanied by music, with rhythmic movements as the main form of expression. To design rhythmic and artistic dance movements requires professional training and constant practice [1], so only experienced choreographers are competent, which is a time-consuming and expensive task. Therefore, AI choreography came into being, i.e., AI-based music-to-dance generation [2]. AI-based music-to-dance generation refers to inputting a piece of music and generating a series of coherent and naturalistic dance movements through an AI model.

In recent years, with the rapid development of artificial intelligence, deep learning models have been successfully used to solve problems in various fields of intelligent generation, and researchers have also achieved certain research results in the field of music-to-dance generation based on deep learning models [3]. However, music-to-dance generation research not only needs to generate long continuous movements with high complexity, but also needs to capture the nonlinear relationship between movements and actions, and generate dance movements that match the music, making music-to-dance generation research still a very challenging problem [4].

In 3D dance generation, in 2020, Ahn et al [5] proposed a music to 3D dance action sequence generation framework, which consists of three parts: feature extractor, action generator and classifier. Firstly, a piece of music is input into the feature extractor to get the music features, then the music features are input into the classifier to classify the music according to its type to get the type of the music, and finally, the music features and the music type are input into the action generator to generate the dance action sequence. Zhuang et al [6] proposed a time-convolutional LSTM based dance generation model, which uses time-convolutional LSTM to generate the dance movements, in addition to introducing a control signal, i.e., the dance melody line, to improve controllability. Zhuang et al [7] also proposed an autoregressive generative model to generate 3D dance movements using the style, rhythm and melody of the music as control signals, and to improve the performance of the model multiple synchronized music and dances from professional dancers were also captured. In 2023, He Yayun et al [8] proposed a two-step approach to generate 3D dance: the first step uses music features to train a deep model that can generate corresponding key movements from music clips, the second step uses a VQ-VAE-2 network to encode and quantize the dance key movements, and finally decodes them into a coherent sequence of dance movements.

Aiming at the problems of noise in the joint data of dance movements in the dataset and the local and global dependencies in the timing of music sequences and dance movement sequences, this paper proposes a choreography system based on deep learning.

2. Dance score generation algorithm based on multi-temporal modeling

2.1 Bi-directional recirculating door control unit

Recurrent gating unit (GRU), belongs to one of the recurrent neural network (RNN) variants. Bi-GRU is to add GRU neural unit on the basis of bi-directional RNN, which can effectively solve the problem of the inability to realize the long-term memory of the effective information in the time-series data, as well as the gradient message and gradient explosion in the process of back propagation [9]. Compared with the use of LSTM neural units, GRU units can not only achieve comparable or even better performance, but also contain fewer parameters and easier training, which can effectively reduce the training time of neural networks, so in most cases GRU neural units will be chosen to build and train the network. The enter and output shape of GRU is the equal as that of RNN. The enter and output shape of GRU is the equal as that of RNN. If the contemporary enter statistics is Xt, the hidden layer kingdom handed from the preceding neuron node to the modern-day node containing the relevant statistics of the preceding node is ht-1. Combining the consequences of Xt and ht-1 on the cutting-edge state, the GRU neuron can calculate the output yt of the cutting-edge hidden node of the community as properly as the hidden layer nation ht to be surpassed to the subsequent node. Fig 1 suggests the shape of the GRU.

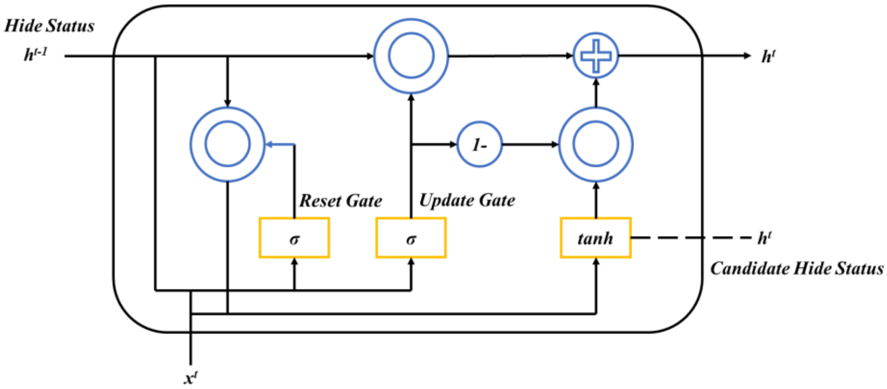

Figure 1 GRU structure

Firstly, the hidden layer state passed from the previous neural node and the input data of the current neural node are spliced, and then the data are mapped to the data in the range of [0, 1] by Sigmoid activation function to obtain the gating state of the reset gate and update gate [10]. As shown in Equation (1) and Equation (2)：

$$r = \sigma\left(W^{r\,{X^t \atop h^{t-1}}}\right) \qquad (1)$$

$$z=\sigma\left(W^{r\,X^{t}}_{\quad h^{t-1}}\right) \qquad (2)$$

Gating r in Equation (1) is used to indicate the control of resetting data, and gating z is used to indicate the control of updating data. After obtaining the signal strengths of the reset gating and update gating, the data after “reset” is first obtained through the reset gating r, as shown in Equation (3)：

$$h^{t-1'}=h^{t-1}\odot r \qquad (3)$$

The state of the hidden layer at the current moment, as shown in the following equation (4) h' is the hidden layer state at the current moment. Here h' mainly contains the current neural network input data X^t. The reset h' can be selectively added to the current hidden state, whereupon it can be assumed that the network has a new memory of the state at the current moment [11].

$$h^{'}=\tanh\left(W^{X^{t}}_{h^{t-1'}}\right) \qquad (4)$$

After the GRU resets the hidden state, it enters a new phase: the “memory update” phase. In this phase, the network selectively memorizes part of the information and forgets part of the information at the same time. This process uses the update gating z calculated previously in equation (2). the update expression is：

$$h^{t}=z\odot h'+(1-z)\odot h^{t-1} \qquad (5)$$

The value of the gating signal z, which controls the data update, ranges from 0 to 1. When the value of z is closer to the value of 1, it means that the hidden state “remembers” more useful data information, while when z is closer to the value of 0, it means that the “forgotten” data is more. The closer z is to 0, the more data is “forgotten”. GRU is very efficient in that it can selectively memorize useful information and forget useless information at the same time by using the same gating unit z to control data updating, while LSTM has to use multiple gating units [12]. Compared to LSTM, GRU has one less “gate” and fewer parameters than LSTM, but it is able to achieve the same functionality as LSTM.

The Bi-GRU network has two types of linkages, a forward temporal linkage which helps to learn from previous representations and a backward temporal linkage which helps to learn from future representations. Forward propagation is divided into two steps：

(1) Moving from left to right, calculating from the initial time step until the last time step is reached.

(2) Moving from right to left, the calculation starts from the last time step until the initial time step is reached. Thus the combination of Bi-GRU modules can utilize both historical information and information from future moments, which can significantly improve performance.

2.2 Dance score generation algorithm based on multi-temporal modeling

Considering that the motion capture data is a continuous action sequence and there is a strong correlation between different data frames in this chapter, this paper proposes a dance score generation algorithm based on multi-temporal modeling. The dance score generation algorithm is mainly realized based on Bi-GRU. In the feature engineering part, the fused spatial features proposed in Chapter 3 are used, which can fully utilize the spatial information in the action data [13]. In the action recognition part is built based on Bi-GRU neural units, which can better deal with long-time temporal sequences and can fully utilize the contextual information of the temporal sequences.

The input to this network is a time series of 40 frames (after frame extraction in this paper) of motion capture data after feature extraction: 20 of the data characterize the Joint feature sequence, and the other 20 frames characterize the Lime feature sequence. Equations (6), (7) show the relationship between the hidden layer state at moment t, the hidden layer state at moment (t-1) and the inputs at the current moment [14]. The inputs to the network are X$_1$ and X$_2$, where X$_1$ is the extracted Joint feature that characterizes the dynamic change of human posture and X$_2$ is the extracted Line feature that carries information about the direction of motion.

$$h_t^1 = W_{X_1 h} X_t^1 + W_{hh}^1 h_{t-1}^1 + b_h \qquad (6)$$

$$h_t^2 = W_{X_1 h} X_t^2 + W_{hh}^2 h_{t-1}^2 + b_h \qquad (7)$$

The network of dance spectrum generation algorithm based on multi-temporal modeling is constructed based on Bi-GRU. It is primarily made up of three distinct layers in the network: the initial two layers are Bi-GRU layers, while the final layer is a fully connected layer. The first two layers focus on capturing the temporal relationships across various frames. The ultimate layer is the fully connected one, utilizing the Softmax function, often referred to as the Softmax layer. The output generated by this network represents the predicted classification outcome.

The Joint feature denotes the positional alterations of human skeletal joints in a three-dimensional space as movement occurs, whereas the Line feature indicates how adjacent joints work together to ascertain the direction of movement during physical activity, and these features are combined as input for the network. The Bi-GRU network subsequently relates the spatial information dynamically for each time instance along the time axis. Consequently, the network merges the spatial location details, orientation data, and lengthy time series information across different frames, enabling the integration of spatio-temporal characteristics, enhancing the feature representation, and, in turn, theoretically boosting the accuracy of recognition.

2.3 Experimental analysis

Two standard continuous motion capture datasets labeled by Laban symbols, Laban-16 and Laban-48, are used for the experimental validation of this algorithm. Among them, Laban-16 has a smaller sample size, and this paper first conducts ablation experiments on it to verify the effectiveness of network construction and feature selection, and then conducts comparison experiments on the two datasets afterwards. In this paper, we divide the training set and test set with a ratio of 1:1, and sufficiently randomly disperse the samples during model training and validation.

For motion capture data processing, since convolutional neural networks require all input sequences to have the same length (same number of frames), in this paper, each continuous motion sample is uniformly down sampled to a fixed length T of input sequence. Another reason for down sampling is that this is a fine-grained recognition task, and too many input frames provide richer spatio-temporal information while at the same time affecting the CTC-based symbol transcription. The recommended value for the number of down sampled frames is 3 to 4 times the maximum number of elemental actions in the dataset (8 in this case), so here in this paper we choose T = 30. The selection of the number of down sampled frames will be analyzed experimentally in detail in Section 5.3.2.

In terms of network model construction, this paper uses the TensorFlow framework to write the whole network model. Among them, in order to prevent model overfitting, this paper adds Dropout layer after 1D-CNN network and Bi-GRU network respectively, and the neuron dropout rate is set to 0.2. Hyperparameters related to the model structure are optimally selected by the grid search method. The whole network is trained by Adam optimizer for 240 iterations with a training batch size of 80.

As can be seen from Fig. 2, adding a one-dimensional convolutional neural network before the recurrent neural network performs better than using the recurrent neural network alone, which proves that the spread Li group features are better aligned in the time domain after the one-dimensional convolution, allowing the subsequent recurrent neural layers to perform better in time-series modeling. In addition, stacking more convolutional neural layers does not benefit the computational power of the network, as models with too many layers have difficulty in accurately learning the values of more parameters from limited training data. In addition, the Bi-GRU network, compared to other variants of recurrent neural networks GRU, Bi-LSTM, and LSTM, demonstrates a better ability to model bi-directional long-term temporal relationships. Taken together, the multi-temporal network structure used in this algorithm is able to effectively perform effective spatio-temporal modeling and analysis of continuous human motion as represented by the Lie group feature, and achieves good continuous motion recognition performance.

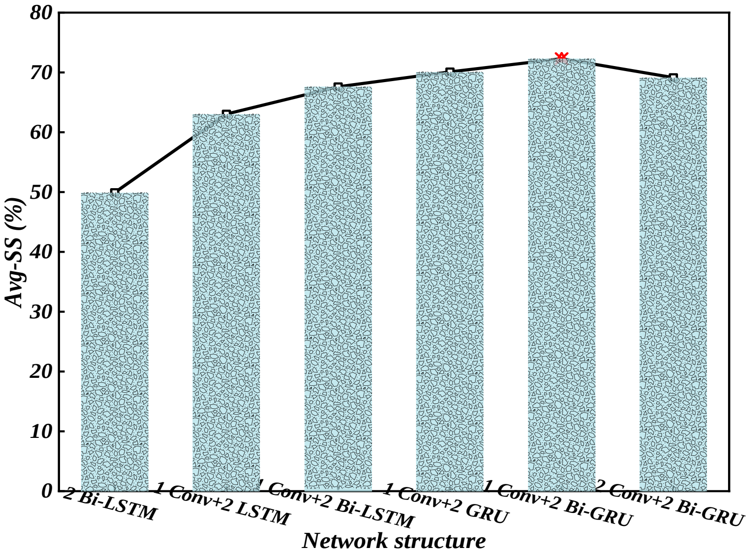

Figure 2 Ablation experiment of model structure

Secondly, this paper verifies the effectiveness of feature selection. In this paper, the optimal network structure is used to input three kinds of human skeleton features respectively for comparison, and the results are shown in Table 1. Unlike the 3D coordinate features of joints, which can only stack the spatial positions of joint points in a sequence form, and the bone vector features, which can only express the spatial relative positions between neighboring joint points, the Li group features can enhance the expression of the rotation information embedded in the human skeleton, and can describe the relative geometric relationships between neighboring joints and bones through the rotation matrix, which retains more topological information of the human skeleton. Therefore, the multi-temporal model with Lie group features as inputs exhibits better continuous motion recognition performance.

Table 1 Ablation experiments of model input features

Input feature	Network structure	Avg-SS (%)
Joint coordinate	1 Conv+2 Bi-GRU	52.13
Skeletal vector		71.93
Lie group feature		72.79

Finally, this paper compares the present algorithm with other algorithms for dance score generation, as shown in Fig. 3 to verify the efficiency of the proposed framework for continuous dance score generation.CRNN-CTC denotes the algorithm proposed in this chapter, which stands for the spatio-temporal serial model using the 1Conv+2Bi-GRU structure. The first three rows of Fig. 3 are the dance score generation algorithms in the traditional framework, which require pre-segmentation of continuous movements and then recognizing the elemental movement segments one by one. Therefore, in order to evaluate the traditional methods in a complete way, this paper additionally adds the action segmentation accuracy and single action segment recognition accuracy to measure the performance of the traditional methods in these two steps, respectively. Due to the lack of standard manual segmentation results on the Laban-48 dataset, it is not possible to calculate the action segmentation accuracy and single action segment recognition accuracy of the traditional method on this dataset. The dance score generation algorithm for continuous motion does not have the above two steps and is able to recognize continuous human motion end-to-end, using only the metric Ag-SS, which is a measure of global recognition accuracy.As can be seen in Fig. 3, the algorithms in the traditional framework are able to achieve a high single action segment recognition accuracy, but the global recognition of overall continuous motion is not as effective. This is due to the low accuracy of the previous action segmentation step, which affects the subsequent segment recognition, and the fact that the traditional framework recognizes a single action in a segmented way, which cannot take into account the long-time correlation across actions and is difficult to optimize globally.

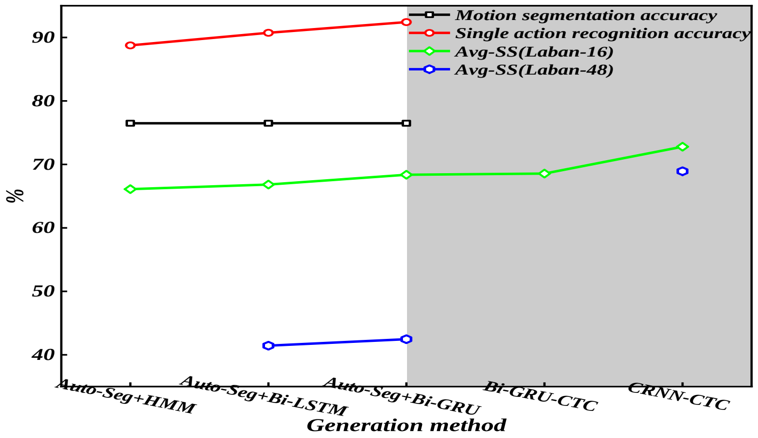

Figure 3 Comparison experiment

The continuous dance score generation algorithm based on multi-temporal permutation realizes the recognition of any number of movements through end-to-end automatic alignment, avoids the cumbersome manual segmentation and the poor flexibility and low accuracy of automatic segmentation algorithms, and is able to realize the unified training of the overall model, so that the model outputs expressive and discriminative fine-grained descriptions of each frame of gesture outputs, and flexibly recognizes the movements of varying durations. Therefore, the continuous dance score generation algorithm shows better continuous motion recognition performance compared with the traditional dance score generation algorithm, which provides a new direction for the research of automatic dance score generation.

3. Automatic music choreography system design

3.1 System function module design

The user's functional module design is divided into four modules, including account management module, resource management module, interactive display module and data processing module. Each module is described as follows:

(1) Account management module

In order to retain and facilitate the use of the system when the user's resource files, new users to use the system first need to register a system account, the user through the mailbox and the verification code to register and set the account password, after successful registration can log in and use the system's full range of functions, if you forget the account password can be through the mailbox to change the password verification code. Administrator users can cancel the account of ordinary users.

(2) Resource management module

Users can view the file resources saved in the account after recording, including audio files, input action files, model resource files and output action files. New users can directly view or download files that have been uploaded by the administrator and exist by default in the system. If necessary, users can upload new files other than the output action files for setting the display effect of the dance generation, in which the files uploaded by the administrator users cannot be operated by ordinary users. The uploaded files are kept in the resource library, i.e. stored in the server, and can be viewed, downloaded, deleted or renamed at will. In order to facilitate the user to quickly find the required resource files, in the search bar, enter the file name can be queried to match the file, in addition to the number of files for each type of resource are carried out statistics.

(3) Interactive display module

This module is the main module of the system and the intelligently generated dances are displayed on the home page of the system. Users can upload audio files, input action files and model files on the home page, or call the resource management module to select these three resources and background resources from the resource library. After the dance is generated, its first frame is displayed on the page, and the user can play and pause the dance, which is automatically played along with the music. The system generates three kinds of dance results according to the files selected by the user, and the user can switch and watch the effect of each kind of dance online and choose to download the corresponding dance data file in BVH format file.

(4) Generate model management module

This module can only be operated by the administrator. On the Generated Models page, the administrator can view all the files of the generated models on the server, and can download, delete, and rename each file. In addition, new files can be uploaded to the server. This module updates the network model files used to generate dances, which can be used to generate higher quality dance sequences.

3.2 Automated music choreography system framework

The issue of automated music choreography through computer systems has been examined by earlier scholars, leading to the development of related automated music choreography technologies. These systems can primarily be classified into two main types. The first type is exemplified by Shiratori [17], which depends on traditional, manually crafted features of music and movement along with feature matching techniques. This approach creates a movement database to choose segments that align with the desired music, allowing for the generation of dance routines; typically, this movement database is formed from motion capture information. The second type, illustrated by Alemi [18], leverages machine learning techniques to directly create a model that maps music to dance. This method establishes the relationship between musical elements and movement characteristics through model training, enabling the generation of dance sequences that correspond to the selected music.

Although the above two music choreography system frameworks can fulfill the music choreography task to some extent, they both have obvious limitations in some aspects. For the first type of music choreography framework represented by Shiratori, although it can guarantee the authenticity and consistency of the dance, the novelty of the dance cannot be guaranteed because most of the synthesized movement sequences are obtained by cutting and splicing the movement samples that already exist in the original movement database, so it is very difficult to generate new movement patterns. For the second type of music choreography framework represented by Alemi, although the novelty of the dance can be guaranteed, the robustness of the model is not strong, and the dance synthesis effect is not ideal for the brand-new target music that the model has not seen before, and the coherence and authenticity of the movements need to be strengthened.

This document tackles the previously mentioned limitations and introduces an innovative framework for an automated dance choreography system. This new system aims to create dance motions that are not only unique and consistent but also aligned with the selected music, while also guaranteeing that the choreography system possesses adequate strength and adaptability. The framework is primarily divided into four components: assembling the initial data set, training the model and generating actions, choreography and synthesis, and visualizing the dance through 3D character animation. The pivotal processes in this framework include model training, action generation, and creating choreography that draws from both musical and action characteristics. Figure 4 illustrates the overall structure of the system:

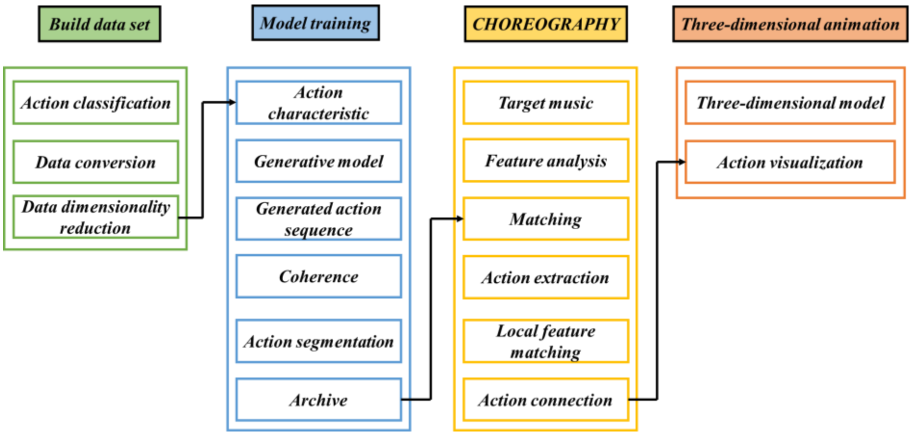

Figure 4 General framework of the music choreography system

(1) Building the action data-set module

The user-generated dance movement files in VMD format and the corresponding music files in WAV format obtained from the web are used to construct a movement-music dataset, and the movements are classified and labeled according to the style and tempo based on human experience. In this module, the system pre-processes the raw movement data with frame interpolation, angle-position conversion and dimensionality reduction to obtain 21*3-dimensional movement features in preparation for model training.

(2) Model training and action generation module

The system constructs a Bi-GRU-MDN network, which is trained according to the styles and speeds of the movements, obtains the movement generation models of different styles and speeds, and stores the model with the best effect. The system adopts the mean value method to generate action sequences according to the given initial state of the action, and selects the generated action sequences according to the coherence rule to construct a candidate action database, which prepares for the next step of choreography based on musical features.

During the development and training of the action generation model, various parameters must be established, such as: the count of hidden layers (D-layer), the neuron count within the hidden layers, the total number of Gaussian distributions utilized in the hybrid model, the feature dimensionality for inputs, the size of each batch, the length of the input sequence taken at one time, the duration of training, the rate of learning, and the distribution ratio of the training set compared to the validation set during the training phase, which is set at 9:1. The detailed values for these parameters are illustrated in Table 2. The detailed values for these parameters are illustrated in Table 2.

Table 2 Experimental parameters

Parameter name	Value	Parameter name	Value
D-layer	3	Batch size	100
D-dim	512	Sequence length	120
Mixture num	12	Epoch num	500
Input dim	63	Learning rate	1×10^{-3}

In action generation, a brand-new 120-frame action segment, i.e., 120*63-dimensional action features, is input using the trained action generation model of the system, and the action is generated according to the parameter control method proposed in this paper, which only considers the mean value. In order to ensure the quality of the action, 900 frames are generated each time. When screening the action coherence, according to the experience of many experiments, setting the intrusion value to 20 can get better results.

(3) Choreography and synthesis module

The framework implements a tiered algorithm for matching music and action features to conduct correlation assessments. During the phase involving the overall feature matching algorithm centered on music, the system offers a user interface that presents two choices: one option utilizes the built-in parameters for matching, while the other allows users to manually configure the local skeletal tempo thresholds and the spatial values of the action segments, both of which can influence the outcomes of the matching process. In the section that focuses on the algorithm for matching based on local rhythm and intensity characteristics, the system identifies the rhythm and intensity traits from both the musical and action clips, performs the necessary feature matching, and integrates the outcomes from connectivity analysis to derive an action sequence that aligns with the specified music. Afterwards, it interpolates and connects the neighboring action segments in the action sequence to obtain the final choreography result and complete the computerized automatic music choreography task.

3.3 Test analysis

In this paper, five objective evaluation metrics are used in all experiments: FID, BeatCoverage, HitRate, Diversity and Multimodality. However, since the goal of this paper is to study the multiplayer dance movement generation task, it is not meaningful to study the quality of single-person dance movement generation too much, so this paper only adopts subjective evaluation metrics in the comparison experiments of different methods on the DDVM dataset. In this paper, we firstly conduct a comparison experiment between multi-person dance movement generation and some single-person dance movement generation tasks, which is used to illustrate the usability of the DDVM dataset and the feasibility of the proposed method in this paper. The experimental results are shown in Figure 5.

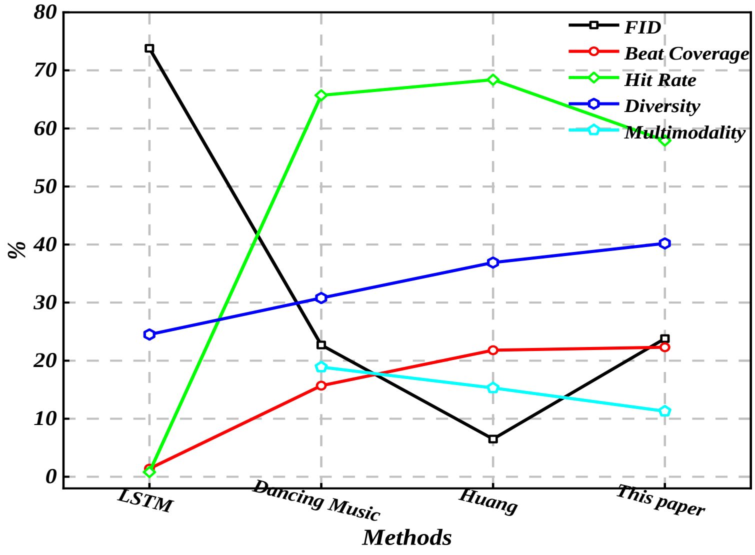

Figure 5 Comparison of dance movement generation results

From the metrics in Fig. 5, it can be seen that the multiplayer dance movement generation method in this paper is able to obtain better metrics on the DDVM dataset, with an improvement of 0.5 in beat coverage and 3.3 in diversity metrics compared to the other best methods. The beat hit rate was 10.5 lower than the best method and the multimodality metric was 7.6 lower. Only the FID metric has a large gap with the results of the single dance generation task, which is 17.3 higher than the best method, which is due to the fact that there are more frames with missing keypoints due to occlusion in the multiplayer dance video, which are made up by the linear interpolation algorithm, which is a bit different from the real samples. From the objective index, the method of this paper has obvious advantages and is feasible.

4. Conclusion

The automatic music choreography system studied in this paper successfully realizes the automatic matching and generation of music and dance movements by means of a deep learning-based multi-temporal modeling algorithm. The experimental results show that the system performs well in the continuous dance score generation task, especially in the beat coverage and diversity indexes, which have achieved significant improvement. By introducing the Bi-GRU network, the system effectively captures the temporal relationship between movements, optimizing the coherence and realism of dance generation.

1. Adding the structure of a one-dimensional convolutional neural network before the recurrent neural network makes the model perform better in time domain alignment, which further improves the time series modeling performance of the subsequent recurrent neural layer. By comparing different network structures (e.g., Bi-GRU, GRU, LSTM, etc.), Bi-GRU demonstrates superior bidirectional long-term temporal relationship modeling capability. On the Laban-48 dataset, the proposed multi-temporal network structure performs well in terms of global recognition accuracy (68.92%) for continuous motion recognition, which avoids the tediousness of manual segmentation and improves the flexibility and accuracy of the recognition compared to the traditional dance score generation algorithm.

2. The multiplayer dance movement generation method in this paper is able to obtain better metrics on the DDVM dataset, with an improvement of 0.5 in beat coverage and 3.3 in diversity metrics compared to the best other methods. beat hit rate is 10.5 lower than the best method, and multimodality metrics are 7.6 lower.

References

[1] Zhang S. Recent advances in the application of deep learning to choreography[C]//2020 International Conference on Computing and Data Science (CDS). IEEE, 2020: 88-91.

[2] Sun G, Wong Y, Cheng Z, et al. Deepdance: music-to-dance motion choreography with adversarial learning[J]. IEEE Transactions on Multimedia, 2020, 23: 497-509.

[3] Zhou Q, Tong Y, Si H, et al. [Retracted] Optimization of Choreography Teaching with Deep Learning and Neural Networks[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 7242637.

[4] Broadwell P, Tangherlini T R. Comparative K-Pop Choreography Analysis through Deep-Learning Pose Estimation across a Large Video Corpus[J]. DHQ: Digital Humanities Quarterly, 2021, 15(1).

[5] Ahn H, Kim J, Kim K, et al. Generative autoregressive networks for 3d dancing move synthesis from music[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 3501-3508.

[6] Zhuang W, Wang Y, Robinson J, et al. Towards 3d dance motion synthesis and control[J]. arXiv preprint arXiv:2006.05743, 2020.

[7] Zhuang W, Wang C, Chai J, et al. Music2dance: Dancenet for music-driven dance generation[J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2022, 18(2): 1-21.

[8] He Yayun, Peng Junqing, Wang Jianzong, et al. Rhythm Dancer: a 3D dance generation method based on Key Movement transformation Diagram and conditional pose interpolation network [J]. Big Data, 2023, 9(1): 23-37.

[9] Chen K, Tan Z, Lei J, et al. Choreomaster: choreography-oriented music-driven dance synthesis[J]. ACM Transactions on Graphics (TOG), 2021, 40(4): 1-13.

[10] Xue J. RETRACTED ARTICLE: Human motion tracking and system design for dance choreography teaching based on deep learning[J]. Soft Computing, 2024, 28(Suppl 2): 587-587.

[11] Dalmazzo D, Waddell G, Ramírez R. Applying deep learning techniques to estimate patterns of musical gesture[J]. Frontiers in psychology, 2021, 11: 575971.

[12] Feng H, Zhao X, Zhang X. Automatic Arrangement of Sports Dance Movement Based on Deep Learning[J]. Computational Intelligence and Neuroscience, 2022, 2022(1): 9722558.

[13] Le N, Pham T, Do T, et al. Music-driven group choreography[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 8673-8682.

[14] Huang Y F, Liu W D. Choreography cGAN: generating dances with music beats using conditional generative adversarial networks[J]. Neural Computing and Applications, 2021, 33: 9817-9833.

[15] Ferreira J P, Coutinho T M, Gomes T L, et al. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio[J]. Computers & Graphics, 2021, 94: 11-21.

[16] Cang J, Huang Y, Huang Y. [Retracted] Research on the Application of Intelligent Choreography for Musical Theater Based on Mixture Density Network Algorithm[J]. Computational Intelligence and Neuroscience, 2021, 2021(1): 4337398.

[17] Shiratori T, Nakazawa A, Ikeuchi K. Dancing-to-music character animation[C]//Computer Graphics Forum. Oxford, UK and Boston, USA: Blackwell Publishing, Inc, 2006, 25(3): 449-458.

[18] Alemi O, Françoise J, Pasquier P. Groovenet: Real-time music-driven dance movement generation using artificial neural networks[J]. networks, 2017, 8(17): 26.