IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 10, 2019, accepted November 6, 2019, date of publication November 15, 2019, date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953698

Music-Driven Dance Generation

YU QI, YAZHOU LIU, AND QUANSEN SUN
School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding author: Yazhou Liu (yazhouliu@njust.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61672286 and Grant 61673220.

ABSTRACT In this paper, a novel model for synthesizing dance movements from music/audio sequence is proposed, which has variety of potential applications, e.g. virtual reality. For a given unheard song, in order to generate musically meaningful and natural dance movements, the following criteria should be met: 1) the rhythm between the dance action and music beat should be harmonious; 2) the generated dance movements should have notable and natural variations. Specifically, a sequence to sequence (Seq2Seq) learning architecture that leverages Long Short-Term Memory (LSTM) and Self-Attention mechanism (SA) is proposed for dance generation. The work in this article is interesting in the following aspects: 1) A cross-domain Seq2Seq learning framework is proposed for realistic dance generation; 2) A set of evaluation criterion is proposed for synthetization evaluation which do not have source for reference; 3) A dance dataset that including both music and corresponding dance motions collected, and very competitive results have been obtained against the-state-of-the-arts.

INDEX TERMS Music, dance, movement, music-driven dance generation.

I. INTRODUCTION

There are many applications for sequence analysis based deep learning [1], [2], including language processing [3], video tracking [4], cross-domain analysis [5], [6], and semantic features based sentiment analysis [7]. For sequence analysis, cross-domain sequence analysis is one of the important branches. Cross-domain sequence analysis refers to finding the correspondence between two different types of sequences. There are many related applications, such as, translating between different languages [5], [8]–[10], using natural language to synthesize real images [11].

Audio to video analysis is a special case of cross-domain sequence analysis [12], [13]. Comparing to the other topics, the research on audio-video analysis is relatively few. The main reason is that for conventional video, the correlation between audio and video is not very strong. For example, for a particular video scene, there may be multiple audio sequences corresponding to it; for a particular audio sequence, it can also be used as background audio for multiple video scenes.

However, the correlation between music and dance movements is relatively significant compared to general audio and video sequences. Although there is no one-to-one correspondence between dance movements and music, the correlation between the beats of dance movements and music beats is relatively strong. This relatively strong correlation provides a possibility for cross-domain analysis of music and video. For example, the harmonious of the dance movement beat can be analysed according to a specific music sequence; or the appropriate background music can be selected according to the dance movement. Specifically, the target of this work is to generate dance movement according to the music sequence.

There are several attempts to analyse the connection between audio and dance movements. Alemi *et al.* [14] use GrooveNet to learn the relationships between low-level audio features and dance movements. Chan *et al.* [15] propose a model for achieving movement style migration between different human subjects. Cai *et al.* [16] attempts to synthesize human motion video from noise. The limitations of the above attempts are either not finding a strong correlation between music and video or simply focusing on synthesizing human motion while ignoring the association between music and video.

In addressing the above problems, this paper proposes a new Seq2Seq [17]–[19] framework, which is referred to as Long Short-Term Memory Self-Attention (LSTM-SA), to learn the correlation between music and dance movements. Specifically, the proposed method firstly extracts features of music and dance sequence, then uses an Encoder-Decoder [5], [20] network to learn the correlation between music and dance sequence. Finally, the synthetic dance sequence was evaluated. The overall processing flow of the method is shown in Fig. 1.

The associate editor coordinating the review of this manuscript and approving it for publication was Omar Sultan Al-Kadi.

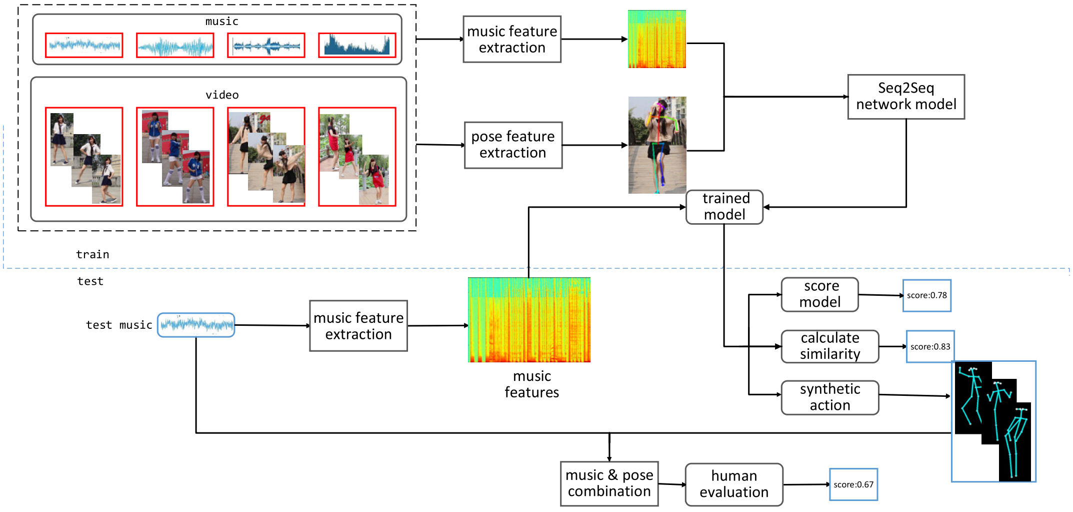

FIGURE 1. **The overall structure of the proposed music-driven dance generation.**

II. RELATED WORK

Music-driven dance generation is an important research topic in the field of computer vision because it has many potential applications. For example, synthetic video can be used for animation generation, choreography, virtual reality, virtual characters and gaming. In this section, we are going to briefly review the advances of the related works from three aspects: dance movement generation, cross-domain sequence analysis and Seq2Seq learning networks.

A. DANCE MOVEMENT GENERATION

A variety of deep learning models are used to learn and generate human motion from motion capture data. Hidden Markov Models (HMMs) [21] and its extensions have been applied to the synthesis of dance movements. Linear Dynamical System [22] relies on dynamic system models to learn and synthesize dance movements. The system automatically learns motion, allows for real-time synthesis, and provides a range of methods such as key-framing and noise-driven generation for synthesizing movements.

Recently, artificial neural networks have made great progress in synthesizing dance movements. Donahue *et al.* [23] focus on generating choreographies for the Dance Dance Revolution game, and used the LSTM [24], [25] network to synthesize a new step chart. However, their approach is limited to generating discrete sequences of step indicators rather than continuous movements. Crnkovic-Friis [26] uses the Long Short-Term Memory Recurrent Neural Networks to learn how to synthesize choreography. But this approach did not provide any methods of controlling the generation and did not accompany any music.

There are several models related to music-driven dance movement generation. Ofli *et al.* [6] introduce a music-driven dance avatar based on the motion synthesis method HMMs. For training, their approach requires movement to be manually annotated into specific patterns synchronized with the beats. For generation, the music is segmented using beat detection. Yet, one of the main limitations of this approach is that it relies on the categories of input music patterns, so there is very little opportunity to generate new motion patterns. Alemi *et al.* [14] propose a GrooveNet model that could solve the problem of relying on classification or segmentation of the music signal. The basic principle behind GrooveNet is to allow the model to learn continuous cross-modal mapping from music information to motion data in an unsupervised manner. However, the model could not generalize well to music tracks beyond the training data.

B. CROSS-DOMAIN SEQUENCE ANALYSIS

At present, there are many interesting applications for cross-domain sequence analysis. Dong *et al.* [11] propose a way to semantically manipulate images by text descriptions. Using the adversarial learning technique, a synthetic realistic image generation model is trained with the given source image and target text description. The GAN based encoder-decoder architecture is able to disentangle the semantics contained in both images and text descriptions, while keeping other image features that are irrelevant to the text descriptions. Psychological studies have provided evidence that human emotions can be aroused by visual content, e.g. images [27]–[29]. Based on these findings, recently computer scientists also started to delve into this research topic and make progress in [7], [30].

Cheng *et al.* [31] introduce an innovative idea for generating an artistic poem from an image. Given an image, the first is to extract a few keywords representing objections and sentiments perceived from the image. These keywords are then expanded to related ones based on their associations in human written poems. Finally, verses and generated gradually from the keywords using recurrent neural networks trained on existing poems. Nallapati *et al.* [32] apply the attention encoder-decoder for the task of abstractive summarization,

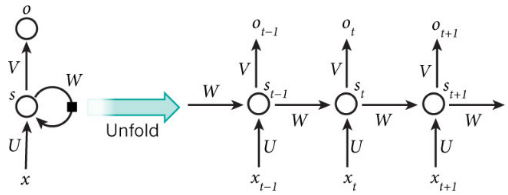

FIGURE 2. Structure of RNN cell.

and the model has very promising results of handling the problem of abstract text summarization, which is the task of generating a headline or a short summary consisting of a few sentences that captures the salient ideas of an article or a passage.

C. SEQ2SEQ LEARNING NETWORKS

The Seq2Seq framework is currently widely used for cross-domain sequence analysis processing [10], [11], [32]. Here, we will review some of the classic network models based on Seq2Seq learning. We have chosen four different types of Seq2Seq network models, which in turn are RNN [8], [33], Convolution Neural Networks (CNN) [9], [17], [34], Attention Model (AM) [35]–[37] and Self-Attention model (SA) [38]–[40].

One thing in common with the four models above is that each model has two components: an encoder [5], [20] and a decoder. These two parts are actually two different neural networks, which are finally combined into a huge network by combination. In summary, the task of the encoder network is to understand the input sequence and create a smaller dimensional representation of it. The low-dimensional representation of the encoder is then forwarded to the decoder network, and the desired output sequence is generated by the decoder network.

Each model has its own special part. RNN works well when dealing with problems with short-term dependencies. RNN is a neural network used to process sequential data, such as text, DNA sequence, music sequence, handwriting, speech and stocks. Compared with traditional neural networks, it takes time and sequence into account and has a temporal dimension. For example, when people need to analyse a word based on previous words and context, traditional neural networks would have bad performance because it has difficulty in remembering previous data, whereas RNN can address this problem well, as it uses a loop to remember information. As shown in Fig. 2, x_t is an input at time t, s_t is the hidden state at time t which will be passed to the next time step $t+1$ as part of the input. It is used to remember the previous states. s_t is computed based on previous state s_{t-1} and current input x_t. o_t is the output at time step t.

However, sometimes we only need to look at recent information to perform the present task and do not want to remember old data. Also, due to the vanishing gradient problem, RNN cannot effectively deal with long-term dependencies. LSTM is a network model based on an RNN that has proven successful at extracting time series of features. Compared with an RNN, an LSTM network contains additional three gates (I.e. forget gate, input gate and output gate, as shown in Fig. 3) to control the cell state, which help it to perform better in Seq2Seq mapping. Therefore, people usually use LSTM instead of RNN when dealing with long sequence problems. One disadvantage of LSTM model is that it compresses the entire input sequence into a fixed representation. Compared to LSTM, CNN model creates a fixed-size context representation. Moreover, the network allows each element in the sequence to be parallelized because the convolutional network does not rely on the calculation of the previous time step. This parallelization can greatly reduce the training time of the network. However, the effective context size of the network can easily be made larger by stacking several layers on top with each other.

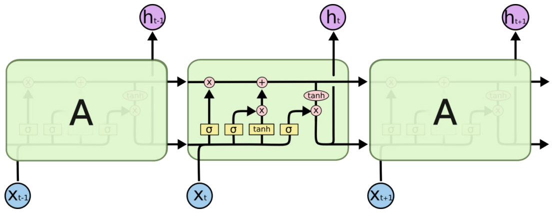

FIGURE 3. Structure of LSTM cell.

The attention mechanism proposed by Parikh can solve the problem of compressing the entire input sequence into a fixed representation. In an AM network, it holds onto all states from the encoder and gives the decoder a weighted average of the encoder states for each element of the decoder sequence. Note that the shortcoming of the AM model is that parallelization is not possible, so the training time will increase. And it not only the order of elements in input sequence, but ignores the connection in output sequence. Self-Attention, sometimes referred to as internal attention, is a mechanism of attention that correlates the different positions of a single sequence to compute a sequence representation. The SA model takes into account the connections between the elements inside the input sequence, which results in a more reasonable output sequence.

III. MUSIC-DRIVEN DANCE GENERATION

This paper aims to learn the mapping between music and dance movements, and synthesize musically meaningful and natural dance movements driven by music. Firstly, we will specifically describe how to extract features from music and video. Then, the architecture of LSTM-SA is detailed.

A. MUSIC FEATURE EXTRACTION

Music has many features [41], [42], such as low-level features (Bark bands, RMS level), spectral features (spectral spread, spectral crest, spectral complexity), and melody features (pitch, pitch salience and confidence, dissonance). This paper chooses Mel Frequency Ceptral Coefficients (MFCC) [43]–[45] as music features. In sound processing, MFCC is the cepstrum parameter extracted in the Mel scale frequency domain, which is a feature widely used in automatic speech and speaker recognition [44], [45].

The steps of MFCC feature extraction are: 1) frame the signal into short frames; 2) take the Fourier transform (FFT); 3) map the powers of the spectrum obtained above onto the male scale, using triangular overlapping windows; 4) take the logs of the powers at each of the mel frequencies; 5) take the discrete cosine transform of the corresponding music are obtained.

One point to note when performing MFCC feature extraction is that the number of frames of music and video is not a one-to-one mapping. Because time aligned music-video pairs are needed for training, we adjust the hop size (h) when extracting the mel-spectrogram so that the music data has the same frame rate as that of video. In this work, the value of the number of video frames (S) is known. It is the total number of images obtained through OpenPose [46]. In order to obtain a certain length to correspond to the dance motions, when using Fourier transform (FFT) to calculate the number of sample intervals (h) between adjacent frames, it is necessary to set the size of h to be the result of the following formula:

$$h = wsize - ((S + 1) * wsize - M) / S \qquad (1)$$

where $wsize$ represents the size of the FFT window, and the default value is 2048; M is the total number of samples of the music. Through this processing, time aligned music MFCC features can be obtained.

B. VIDEO FEATURE EXTRACTION

In this work, it is necessary to use the obtained dance action information as a label for the corresponding music. Each dance movement is actually presented in a combined pose of different parts of the body. Therefore, as long as the estimation result of the human body posture of each frame image is obtained, the corresponding dance motion information can be obtained. In this paper, the OpenPose [46] system was chosen to obtain the results of human pose estimation.

OpenPose system takes an image of $w \times h$w × h as input and produces the 2D locations of anatomical keypoints for each person in the image. The specific processing flow is as follows. First, a feedforward network predicts a set of 2D confidence maps S of body part locations and a set of 2D vector fields L of part affinities; then, the confidence maps S and the affinity fields L are parsed by greedy inference to output 2D keypoints for all people in the image.

After getting the output of OpenPose system, the following processing needs to be done: 1) only retain the posture information of 18 key parts of the human body, they are nose, neck, right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, right eye, left eye, right ear, left ear; 2) mark those undetected points as special character, otherwise, the value of these abnormal points will affect the accuracy of the training; 3) filter out the viewer and normalize all detected points.

C. LSTM-SA NETWORK ARCHITECTURE

Mapping high-dimensional sequences such as motion is a challenging task for deep neural networks (DNN) because such sequences are not constrained to a fixed size. Besides, to generate motion from music, the model under consideration must map highly non-linear representation between music and dance movements. As a preliminary attempt, basic methods of Seq2Seq implemented with LSTM layers displayed remarkable performance and stable training. And considering that music-driven dance generation is a long sequence analysis problem we used an LSTM network as the basis of our model, which is a representative and popular cross-domain sequence analysis model. The LSTM structure has a memory channel c_t that stores useful information from previous outputs as it passes them to subsequent cells. During the training process, the network not only maintains the memory information, but also concentrates on the most important features. Therefore, we chose the encoder-decoder based LSTM network as the basic model.

There are two problems with the LSTM network. One is that the LSTM network compresses the entire input sequence into a fixed vector, so the semantic code c_t corresponding to each frame of the output is the same. The other problem is to ignore the interrelationships of the elements in the music sequence, which can result in less harmonious and natural dance sequences. In order to solve the above mentioned problems, we introduced the idea of attention mechanism. As mentioned before, attention mechanism refers to the process of focusing on importing information while filtering out unnecessary data. After integrating the attention mechanism the network will retain all states from the encoder and assign a weighted average to the encoder state of each element in the decoder sequence. Thus the semantic code c_t corresponding to each frame of the output is different, so that the problem of compressing the entire input sequence into a fixed vector can be solved. At the same time, the interrelationships of the elements in the sequences can be obtained by adding attention mechanism when processing the sequence, which referred as self-attention.

The network architecture is shown in Fig. 4. It contains three major modules. LSTM and Dense module are designed to process input and output sequence and the attention mechanism is applied to change the decoding process. In the decoding process, the state of the decoder network is combined with the state of the encoder and passed to the feedforward network. The feedforward network returns the weight of each encoder state. The encoder inputs are then multiplied by these weights and then the weighted average of the encoder states is calculated. The resulting context is then passed to the decoder network. Thus the decoder network can use different parts of the encoder sequence as context in processing the decoder sequence instead of using a single fixed representation of the input sequence. This allows the network to focus on the most important part of the input sequence, not the entire input sequence.

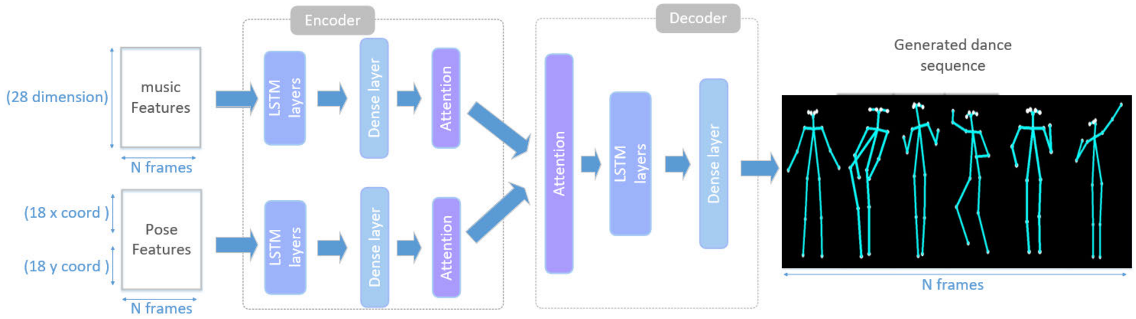

FIGURE 4. Outline of the proposed LSTM-SA for music-driven dance framework.

The input music features and pose features firstly fed into a three-cell LSTM layers with 128 units each. Then an attention mechanism is applied to complete the structure of the encoder. The Decoder network has a structure similar to that of the encoder network. It comprised of an attention layer, three LSTM layers, and a dense layer, in order of the data flow. The LSTM layers with 128 units each and a dense layer with 36 dimensions. The output of the network is compared with the ground truth dance sequence by use MSEM loss as a cost function. We used the Adam optimizer for training and set the learning rate to 0.001.

IV. EVALUATION METHODS

There is no clear reference mapping between music and dance movements, so it is necessary to find some criteria to evaluate whether the generated dance sequence was natural and whether it was produced in accordance with the music. Considering that there are few studies on dance sequences, this paper sets some new rules for judging the advantages and disadvantages of the generated model. In order to ensure the comprehensiveness of the evaluation, this paper is evaluated from two aspects: subjective evaluation and objective evaluation. Specifically, this paper proposes a manual scoring, creates a new learning based scoring model, and calculates the correlation coefficient between the original dance sequence and the generated dance sequence. Three evaluation methods will be specifically described below.

A. HUMAN EVALUATION

In this section, we conduct a human evaluation to compare the baseline methods with ours. First, we generated five dance sequences with each of the five models (CNN, LSTM, AM, SA, LSTM-SA). After generating the sequences, we mixed all the videos in a random order. Then all of the synthesized music&pose combinations were presented to the subjects without telling them which method was used for generation so that the subjects were blind to the method they evaluated. And next ten subjects were recruited and asked to score the music&pose combination generated by different methods. Half of the 10 subjects were professional dancers which called expert group, and the rest were ordinary people without professional dance knowledge which called ordinary group.

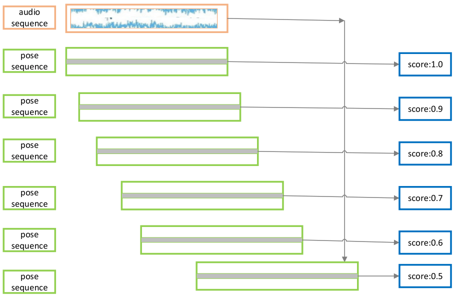

FIGURE 5. Method of dividing original sequence for Learning-based scoring model.

The subjects were required to score the combinations (0.4 for worst, 1.0 for best) based on following criteria.

For training: 1) whether the synthesized actions keep the original pose of the music; 2) whether the synthesized actions are natural; 3) whether the generated sequence fit well with music.

For testing: 1) whether the synthesized actions are natural; 2) whether the generated sequence fit well with music; 3) whether the synthesized actions have notable variations. The specific human evaluation experimental results are shown in Section 5.

B. LEARNING BASED SCORING MODEL

In this section, we have designed a new learning based scoring model that automatically scores the generated dance movements. The model includes two important parts: source data preprocessing and training model design. Next we will describe the implementation details.

1) SOURCE DATA PREPROCESSING

The raw data is a set of video data consisting of the corresponding music sequence and dance sequence. In order to get a better score model, we divide the grades interval in [0.5, 1]. The closer the score is to 1, the higher the match between music and dance. Fig. 5 shows how to divide the original sequence. If the original music and dance are one-to-one, we set the score for the corresponding dance sequence to 1.0. If they are misplaced for a few frames (set this value to 50 frames in our experiment.), we set the score to 0.9, and so on, until the score is 0.5.

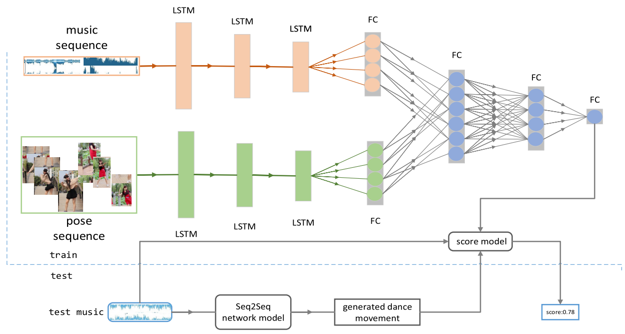

FIGURE 6. The network architecture of learning based scoring model.

2) TRAINING MODEL DESIGN

The network architecture of scoring model is shown in Fig. 6. This network consists of two sub-networks. One is used to process music sequences and the other is used to process dance sequences. The subnet is first processed through three layers of LSTM and then through a fully connected layer. And then the two sub-networks are connected through the full connection layer for joint training.

In theory, the score of the generated dance sequence can be obtained by inputting the music and the corresponding dance sequence into the scoring model. The higher the score means that the better the performance of the corresponding model. In order to ensure the accuracy of the experiment, we have verified the generated scoring model, and the specific results are shown in the Section 5.

C. COSINE-BASED SIMILARITY EVALUATION

Another processing strategy is to find the correlation between the original dance sequence and the generated dance sequence. This paper chooses cosine similarity [47], [48] as the final score of the generated dance sequence. Below we will first describe the relevant content of cosine similarity, and then describe how we get the final score.

1) COSINE SIMILARITY

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any angle in the interval $(0, \pi]$ radians. It could give a useful measure of how similar two documents are likely to be in terms of their subject matter. In fact, two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of the magnitude.

Given two vectors of attributes, A and B, the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as:

$$\text{similarity} = \cos(\theta) = \frac{A.B}{\|A\|\,\|B\|} \tag{2}$$

The value of similarity ranges from -1 means exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality or decorrelation, while in-between values indicate intermediate similarity or dissimilarity.

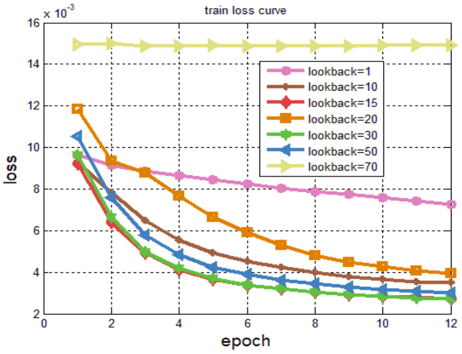

FIGURE 7. Comparisons of different values of lookback in our method on train datasets.

2) SIMILARITY CALCULATION

Before performing the cosine similarity calculation, we used two processing methods to extract the features of the original dance sequence and the generated dance action sequence. The reason for feature extraction is that this can eliminate the interference of individual data, so the experiment can pay more attention to the overall change of the sequence. One method (*M1*) is to calculate the average distance value from each point to the center point. Another method (*M2*) is to calculate the positional change of each point in two consecutive frames.

M1 indicates the extent to which the overall sequence deviates from the center point. *M2* recorded some special movement changes, such as repeating kicks, spinning, twisting the hips, and often moving the arm up and down. In experiment, the two methods are combined in different proportions to extract the features of the sequence. After extracting the features, the cosine similarity between them is taken as the score of the generated dance sequence.

In order to prove the feasibility of calculating the cosine similarity between the original dance sequence and the generated dance sequence, we verified the experimental results. We calculated the cosine similarity between all music sequences and all generated dance sequences. *CosA* is used to represent the sum of cosine similarities between the original dance sequence based on the same music and the generated dance sequence. *CosB* is used to represent the sum of cosine similarities between the original dance sequence based on different music and the generated dance sequence. Finally, we use scoreRatio to represent the ratio of *cosA* to *cosB*. The value of scoreRatio indicates the performance of the corresponding model. The experimental results are shown in Section 5.

V. EXPERIMENTS

In this section, we evaluate the proposed LSTM-SA on our private dataset. Firstly, we introduce the implementation details. Then, we analyse performance of evaluation methods

TABLE 1. Average Human evaluation results of private dataset obtained by CNN, LSTM, AM, SA, and the proposed method LSTM-SA. The red and blue markers indicate the highest score given by the expert group and the ordinary group respectively under the corresponding criteria.

		CNN		LSTM		AM		SA		**LSTM-SA**	
		expert	ordinary	expert	ordinary	expert	ordinary	expert	ordinary	expert	ordinary
train	similarity	0.62	0.64	0.86	0.89	0.49	0.54	0.42	0.43	0.90	0.93
	natural	0.68	0.71	0.87	0.92	0.50	0.54	0.40	0.44	0.89	0.95
	fit	0.52	0.56	0.86	0.90	0.48	0.52	0.43	0.45	0.91	0.96
test	richness	0.46	0.47	0.68	0.71	0.50	0.65	0.40	0.41	0.69	0.76
	natural	0.42	0.43	0.62	0.64	0.43	0.42	0.41	0.41	0.75	0.71
	fit	0.40	0.44	0.65	0.63	0.46	0.48	0.41	0.45	0.74	0.78

proposed in this paper. Finally, to show processed LSTM-SA synthesis scoring results, we compare it with four baselines.

A. IMPLEMENT DETAILS

1) DATASET PREPARATION

Few motion capture datasets include music & dance movement data. To our knowledge, no dataset of synchronized music and motion capture data is currently available online. In order to realize the generation model of music-driven dance movements, we construct our own music and dance data sets. We have made our dataset openly available to facilitate related research. [1] The data set is a relatively high quality of dance video collected from the internet. The music-dance dataset contains videos for one type (jazz) of dance, totalling 120,000 frames of dance motions and accompanying music. These data record the positions of 18 skeleton joints in each frame and 28dimensional music features in each frame.

2) VALUE OF LOOKBACK

Lookback is the number of previous time steps used to predict the input variables for the next time period. The value of different lookbacks has a great influence on the synthetic dance movements. So we tested the values of multiple lookbacks to find the best value. As shown in Fig. 7, 30 or 15 is a good choice. Considering that the smaller the value, the shorter the training time, we choose 15 as the final lookback value.

3) LOSS FUNCTION

Undetected key points in the pose sequence should not be used to calculate training loss, otherwise the accuracy of the model will be affected. To solve this problem, we construct a new loss function, the mean square error mask (MSEM) function. The MSEM function is based on a mean squared error (MSE) function, which corresponds to the expected value of the squared error loss. Undetected points are marked with special characters when performing pose extraction. If the special character is encountered, the loss value is set to 0.

B. PERFORMANCE OF EVALUATION

In this section, we show the verification results of the three evaluation models proposed in this paper.

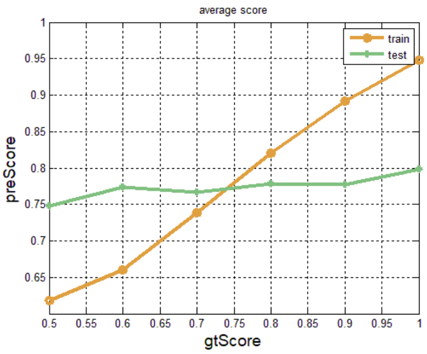

FIGURE 8. Scoring model verification of private dataset obtained by LSTM-SA.

We evaluate LSTM-SA on our private dataset and compare with other four state-of-the-art sequence generation methods.

1) HUMAN EVALUATION RESULTS

Table 1 presents the results of human evaluation. It shows that the scoring results of the two groups are basically the same, and each score of the expert group is basically lower than the ordinary group. This is understandable. The expert group will pay more attention to the details between dance and music, while the ordinary group pays more attention to visual effects. And it shows that our method outperformed baseline methods on following aspects:

a: KEEPING THE ORIGINAL MOVEMENTS

In the similarity column, our proposed method scores higher, indicating that our method is able to maintain the original pose better than baseline methods.

b: FITTING THE MUSIC

The results of user ranking indicate that our method can synthesize natural dance sequences fitting music better than baseline methods.

c: HAVING A RICH SET OF ACTIONS

Our expectation for the training model is that the generated dance movements should have notable variations. Obviously, our method performs better than baseline methods on this point.

[1] https://github.com/njustqiyu/Music-Driven-Dance-Generation-dataset

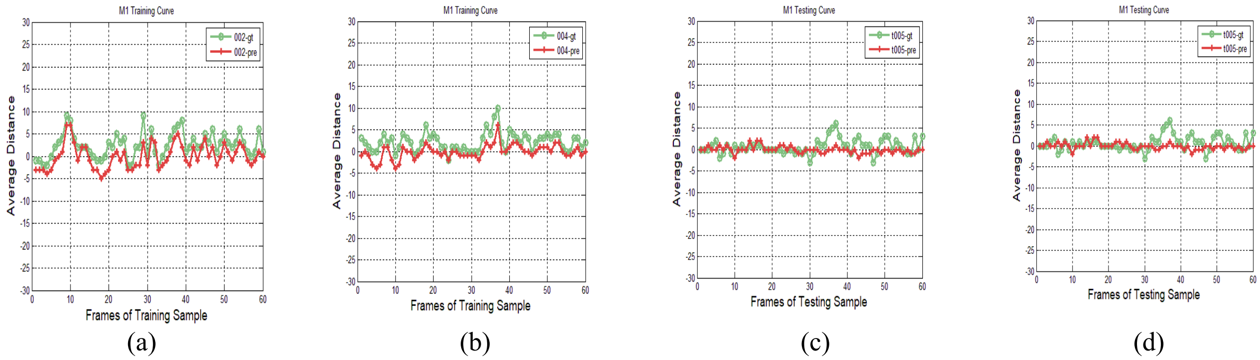

FIGURE 9. The trend of M1 method based on Cosine-based similarity evaluation. (a), (b) Average distance from the center point on training data. (c) and (d) Average distance from the center point on testing data (only the first 60 frames have been intercepted).

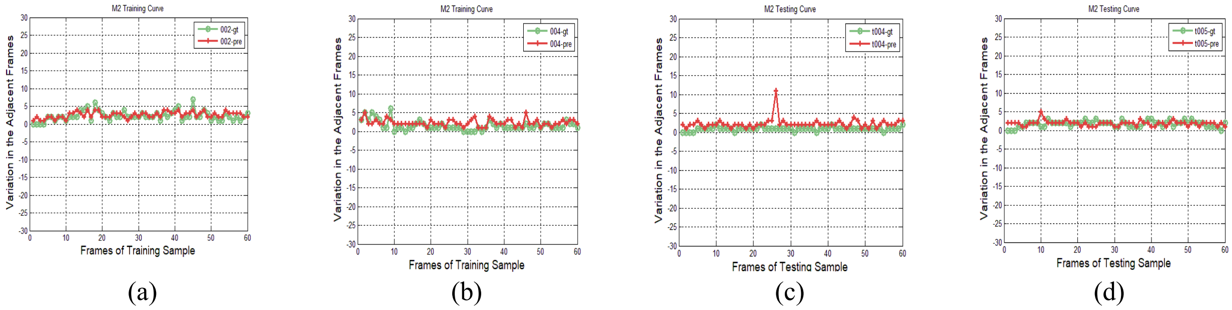

FIGURE 10. The trend of M2 method based on Cosine-based similarity evaluation. (a), (b) Variation in the adjacent frames on training data. (c) and (d) Variation in the adjacent frames on testing data (only the first 60 frames have been intercepted).

TABLE 2. Cosine similarity verification of private dataset obtained by CNN, LSTM, AM, SA, and the proposed method LSTM-SA. The number marked in red indicates the highest score for each item.

	train					test				
	M=α$M1$+(1-α)$M2$					M=α$M1$+(1-α)$M2$				
Method	*M2*	*M1*	α=0.5	α=0.7	α=0.9	*M2*	*M1*	α=0.5	α=0.7	α=0.9
CNN	1.0016	1.2938	1.1228	1.1837	1.2542	1.0025	0.9911	0.9914	0.9982	0.9979
LSTM	**1.1632**	1.5570	1.3311	1.4129	1.5058	**1.0052**	0.9785	0.9946	0.9890	0.9823
AM	1.1248	1.1444	1.1327	1.1368	1.1417	0.9985	0.9584	0.9830	0.9746	0.9644
SA	1.1232	1.1594	1.1423	1.1493	1.1561	0.9889	0.9422	0.9651	0.9559	0.9467
LSTM-SA	1.1610	**1.5618**	**1.3326**	**1.4159**	**1.5100**	1.0002	**0.9925**	**0.9987**	**0.9966**	**0.9940**

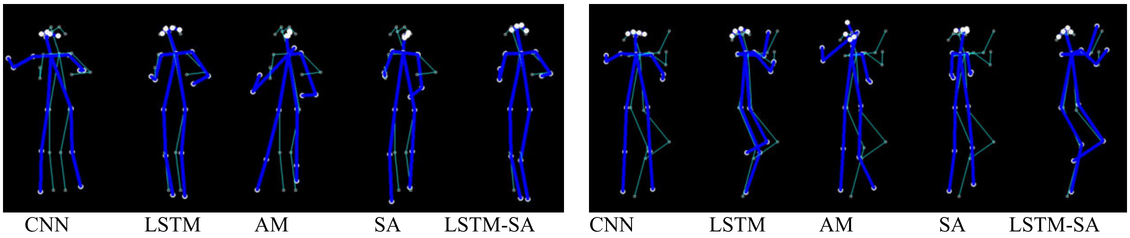

FIGURE 11. Synthesis dance movements of private train dataset obtained by CNN, LSTM, AM, SA, and the proposed method LSTM-SA.

2) LEARNING-BASED SCORING MODEL RESULTS

As described in Evaluation Methods, we create a new learning based scoring model that automatically scores the generated dance movements. When verifying this model, we find it can produce appropriate scores for the training data, but it currently falls short in generalizing beyond those

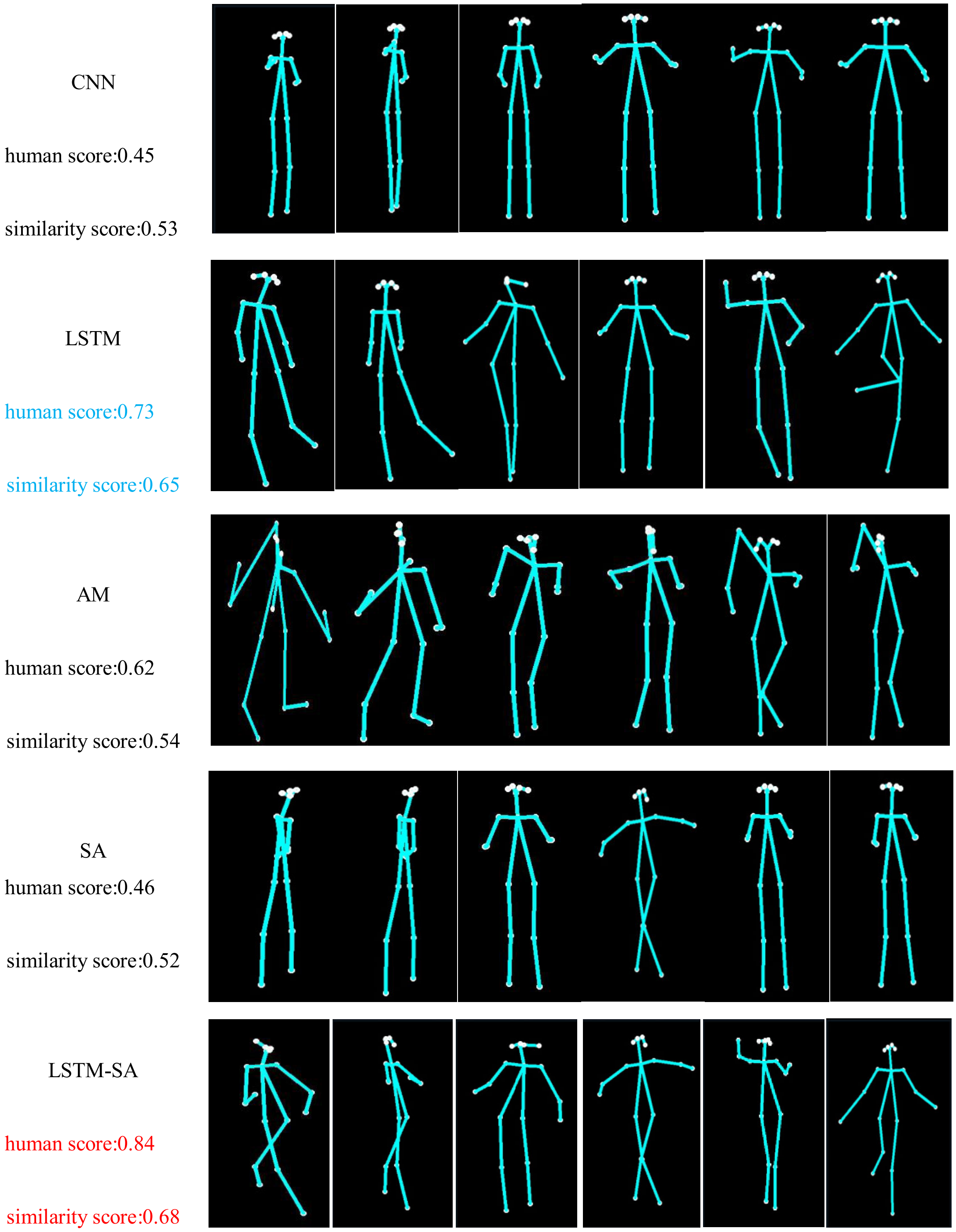

FIGURE 12. Synthesis dance movements of private test dataset obtained by CNN, LSTM, AM, SA, and the proposed method LSTM-SA and corresponding scoring results.

training samples. As shown in Fig. 8, for the training set, the score trend of the original data is basically consistent with the predicted score. But for the test set, no matter how the score of the original data changes, the predicted score is basically around 0.76. This shows that our scoring model has been over-fitting, and we suspect that this may be due to the small training data set. Since the model is somewhat over-fitting, the predicted scores are not shown here.

3) COSINED-BASED SIMILARITY RESULTS

In Section IV, we propose two methods (*M1* and *M2*) to extract the features of the original dance sequence and the generated dance action sequence. In order to prove the rationality of this, we have statistically calculated the results. Fig. 9 plots trend curve of *M1* method based on Cosine-based similarity evaluation. It represents average distance values from each point to the center point. Fig. 10 plots trend curve of *M2* method based on Cosine-based similarity evaluation. It represents positional change of each point in two consecutive frames. It can be clearly seen that both the predicted value and the ground truth value fluctuate within a certain range. And the curve of the original dance sequence and the synthetic dance sequence are basically the same.

We have calculated the value of scoreRatio for different methods to verify feasibility of calculating the cosine similarity between the original dance sequence and the generated dance sequence. Table 2 shows scoreRatio results on different methods. We mark the best methods in bold-red and runner-ups in bold-blue. Although the scoreRatio of the test set is less than 1, the main reason may be that different dance sequences can match multiple pieces of music. It can be observed that our method gets better ratio values than baseline methods.

C. QUALITATIVE RESULTS

In this section, we show the synthesis dance sequence scoring results of different models on private dataset. Fig. 11 shows the results of the train dataset dance movements synthesized by five different generation methods. The blue stickman indicates the action synthesized by the music, and the green indicates the original action of the corresponding music. From left to right, their corresponding models are in turn CNN, LSTM, AM, SA, LSTM-SA. It can be seen from the training results that the two methods of LSTM and LSTM-SA have the highest coincidence with the original action.

Fig.12 shows examples of scoring evaluation results on test dataset of two different evaluation methods on five models. From top to bottom, their corresponding models are in turn CNN, LSTM, AM, SA, LSTM-SA. Each model shows the generated dance movements and two scoring results. The test result in Fig. 11 shows that the generated dance movements of LSTM-SA has notable variations and has higher ratings.

It can be seen from the above evaluation results that LSTM-SA gets comparable results compared with baseline methods. Moreover, the results of the models based on different evaluation indicators are basically the same. The order of performance from high to low is: LSTM-SA, LSTM, AM, CNN, SA. Human evaluation is mainly subjective visual evaluation, and there are many limitations: personal preference, professional or not, and time consumption. The similarity evaluation is an objective evaluation, but it does not represent the human criteria.

VI. CONCLUSION

This paper proposed a new method for synthesizing dance movements from music sequence. The proposed LSTM-SA adopts a Seq2Seq network to train the generation model, which is capable of synthesizing natural and richness dance sequences that are in harmony with corresponding music. In addition, in order to evaluate the generated model, three new evaluation criterions were proposed in this paper. The evaluation results demonstrated that our approach achieved the desired requirements and outperforms the baseline methods. Our future work will focus on three major objectives: 1) acquiring more different type data to continuously enhance our dataset, 2) finding more criterions to evaluate the generated models, and 3) leveraging our model to build various applications.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

[2] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.

[4] A. M. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 2015.

[5] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” Sep. 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[6] F. Ofli, Y. Demir, Y. Yemez, E. Erzin, A. M. Tekalp, K. Balci, I. Kizoğlu, L. Akarun, C. Canton-Ferrer, J. Tilmanne, E. Bozkurt, and A. T. Erdem, “An audio-driven dancing avatar,” *J. Multimodal User Inter.*, vol. 2, no. 2, pp. 93–103, Sep. 2008.

[7] S. Zhao, Y. Gao, X. Jiang, H. Yao, T.-S. Chua, and X. Sun, “Exploring principles-of-art features for image emotion recognition,” in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 47–56.

[8] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” Jun. 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078

[9] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, “A convolutional encoder model for neural machine translation,” Nov. 2016, *arXiv:1611.02344*. [Online]. Available: https://arxiv.org/abs/1611.02344

[10] Y. Wu, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” Sep. 2016, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144

[11] H. Dong, S. Yu, C. Wu, and Y. Guo, “Semantic image synthesis via adversarial learning,” Jul. 2017, *arXiv:1707.06873*. [Online]. Available: https://arxiv.org/abs/1707.06873

[12] R. Fan, S. Xu, and W. Geng, “Example-based automatic music-driven conventional dance motion synthesis,” *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 3, pp. 501–515, Mar. 2011.

[13] N. Yalta, K. Nakadai, and T. Ogata, “Delayed skip connections for music content driven motion generation,” in *Proc. Int. Conf. Learn. Representations*, Vancouver, BC, Canada, vol. 4, 2018.

[14] O. Alemi, J. FranÃğoise, and P. Pasquier, “GrooveNet: Real-time music-driven dance movement generation using artificial neural networks,” *Network*, vol. 8, no. 17, p. 26, 2017.

[15] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, “Everybody dance now,” Aug. 2018, *arXiv:1808.07371*. [Online]. Available: https://arxiv.org/abs/1808.07371

[16] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang, “Deep video generation, prediction and completion of human action sequences,” in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2018, pp. 374–390.

[17] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” May 2017, *arXiv:1705.03122*. [Online]. Available: https://arxiv.org/abs/1705.03122

[18] D. L. W. Hall, D. Klein, D. Roth, L. Gillick, A. Maas, and S. Wegmann, “Sequence to sequence transformations for speech synthesis via recurrent neural networks,” ed: Google Patents, 2018.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[20] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” Sep. 2014, *arXiv:1409.1259*. [Online]. Available: https://arxiv.org/abs/1409.1259

[21] M. Brand and A. Hertzmann, “Style machines,” in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.*, 2000, pp. 183–192.

[22] Y. Li, T. Wang, and H.-Y. Shum, “Motion texture: A two-level statistical model for character motion synthesis,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 465–472, Jul. 2002.

[23] C. Donahue, Z. C. Lipton, and J. McAuley, "Dance dance convolution," Mar. 2017, *arXiv:1703.06891*. [Online]. Available: https://arxiv.org/abs/1703.06891

[24] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," Jan. 2016, *arXiv:1601.06733*. [Online]. Available: https://arxiv.org/abs/1601.06733

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] L. Crnkovic-Friis and L. Crnkovic-Friis, "Generative choreography using deep learning," May 2016, *arXiv:1605.06921*. [Online]. Available: https://arxiv.org/abs/1605.06921

[27] D. Joshi R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Z. Wang, J. Li, and J. Luo "Aesthetics and emotions in images," *IEEE Signal Process. Mag.*, vol. 28, no. 5, pp. 94–115, Sep. 2011.

[28] P. J. Lang, "A bio-informational theory of emotional imagery," *Psychophysiology*, vol. 16, no. 6, pp. 495–512, Nov. 1979.

[29] P. J. Lang, M. M. Bradley, and B. N. Cuthbert, "Emotion, motivation, and anxiety: Brain mechanisms and psychophysiology," *Biol. Psychiatry*, vol. 44, no. 12, pp. 1248–1263, Dec. 1998.

[30] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: The fine print and the benchmark," in *Proc. 13th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 308–314.

[31] W.-F. Cheng, C.-C. Wu, R. Song, J. Fu, X. Xie, and J.-Y. Nie, "Image inspired poetry generation in Xiaoice," Aug. 2018, *arXiv:1808.03090*. [Online]. Available: https://arxiv.org/abs/1808.03090

[32] R. Nallapati, B. Zhou, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," Feb. 2016, *arXiv:1602.06023*. [Online]. Available: https://arxiv.org/abs/1602.06023

[33] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork RNN," Feb. 2014, *arXiv:1402.3511*. [Online]. Available: https://arxiv.org/abs/1402.3511

[34] A. Graves, "Generating sequences with recurrent neural networks," Aug. 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[35] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," Jun. 2016, *arXiv:1606.01933*. [Online]. Available: https://arxiv.org/abs/1606.01933

[36] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[37] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," Aug. 2015, *arXiv:1508.04025*. [Online]. Available: https://arxiv.org/abs/1508.04025

[38] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," Mar. 2017, *arXiv:1703.03130*. [Online]. Available: https://arxiv.org/abs/1703.03130

[39] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," May 2018, *arXiv:1805.08318*. [Online]. Available: https://arxiv.org/abs/1805.08318

[40] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why self-attention? A targeted evaluation of neural machine translation architectures," Aug. 2018, *arXiv:1808.08946*. [Online]. Available: https://arxiv.org/abs/1808.08946

[41] E. Marchetto and G. Peeters, "A set of audio features for the morphological description of vocal imitations," in *Proc. 18th Int. Conf. Digit. Audio Effects*, 2015, pp. 207–214.

[42] A. Allik, G. Fazekas, and M. B. Sandler, "An ontology for audio features," in *Proc. ISMIR*, Aug. 2016, pp. 73–79.

[43] W. Wang, S. Li, J. Yang, Z. Liu, and W. Zhou, "Feature extraction of underwater target in auditory sensation area based on MFCC," in *Proc. China Ocean Acoustics (COA)*, Jan. 2016, pp. 1–6.

[44] K. S. Ahmad, A. S. Thosar, J. H. Nirmal, and V. S. Pande, "A unique approach in text independent speaker recognition using MFCC feature sets and probabilistic neural network," in *Proc. 8th Int. Conf. Adv. Pattern Recognit. (ICAPR)*, Jan. 2015, pp. 1–6.

[45] M. Bezoui, A. Elmoutaouakkil, and A. Beni-hssane, "Feature extraction of some Quranic recitation using mel-frequency cepstral coeficients (MFCC)," in *Proc. 5th Int. Conf. Multimedia Comput. Syst. (ICMCS)*, Sep./Oct. 2016, pp. 127–131.

[46] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," Dec. 2018, *arXiv:1812.08008*. [Online]. Available: https://arxiv.org/abs/1812.08008

[47] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, and P. Kenny, "Cosine similarity scoring without score normalization techniques," in *Proc. Odyssey*, Jun. 2010, p. 15.

[48] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proc. Asian Conf. Comput. Vis.* New Zealand, Oceania: Springer, 2010, pp. 709–720.

YU QI received the B.S. degree in computer science and engineering from the Zhengzhou University of Aeronautics, Zhengzhou, China, in 2017. She is currently pursuing the M.S. degree with the Department of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST). Her current research interests include human motion synthesis and object detection.

YAZHOU LIU received the B.S. degree in mechanical engineering from Harbin Engineering University, Harbin, China, in 2002, and the M.E. and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, in 2004 and 2009, respectively. From 2007 to 2009, he was an Engineer with Panasonic R&D Center Singapore. From 2009 to 2011, he was a Postdoctoral Research Fellow with the Machine Vision Group, Oulu University, Finland. Since 2011, he has been a Faculty Member with the School of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST).

QUANSEN SUN received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology (NJUST), China, in 2006. He visited the Department of Computer Science and Engineering, The Chinese University of Hong Kong, from 2004 to 2005. He is currently a Professor with the Department of Computer Science, NUST. His current research interests include pattern recognition, image processing, remote sensing information systems, and medicine image analysis.