Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis

Taoran Tang
Tsinghua University
Beijing, China
463618845@qq.com

Jia Jia*
Tsinghua University
Beijing, China
jjia@mail.tsinghua.edu.cn

Hanyang Mao
Tsinghua University
Beijing, China
maohanyang789@163.com

ABSTRACT

Dance is greatly influenced by music. Studies on how to synthesize music-oriented dance choreography can promote research in many fields, such as dance teaching and human behavior research. Although considerable effort has been directed toward investigating the relationship between music and dance, the synthesis of appropriate dance choreography based on music remains an open problem.

There are two main challenges: 1) how to choose appropriate dance figures, i.e., groups of steps that are named and specified in technical dance manuals, in accordance with music and 2) how to artistically enhance choreography in accordance with music. To solve these problems, in this paper, we propose a music-oriented dance choreography synthesis method using a long short-term memory (LSTM)-autoencoder model to extract a mapping between acoustic and motion features. Moreover, we improve our model with temporal indexes and a masking method to achieve better performance. Because of the lack of data available for model training, we constructed a music-dance dataset containing choreographies for four types of dance, totaling 907,200 frames of 3D dance motions and accompanying music, and extracted multidimensional features for model training. We employed this dataset to train and optimize the proposed models and conducted several qualitative and quantitative experiments to select the best-fitted model. Finally, our model proved to be effective and efficient in synthesizing valid choreographies that are also capable of musical expression.

KEYWORDS

Motion synthesis, LSTM, autoencoder, music-dance dataset, 3D motion capture

ACM Reference format:
Taoran Tang, Jia Jia, and Hanyang Mao. 2018. Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis. In *Proceedings of ACM Woodstock conference, Seoul, Korea, July 2018 (MM'18),* 9 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author

MM'18, July 2018, Seoul, Korea

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

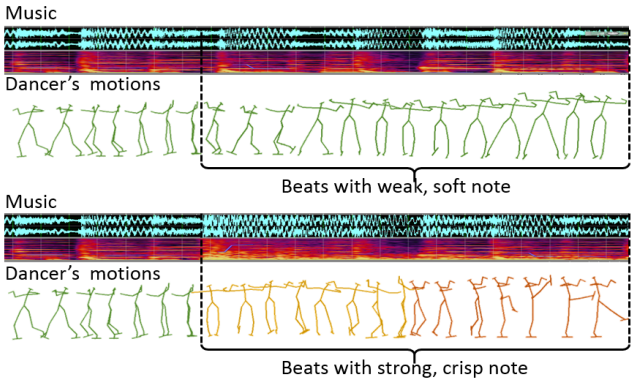

Figure 1: Choreography with dance figure adjustments in accordance with the length, rhythm, and emotion changes of the music.

1 INTRODUCTION

Dance harmoniously engages the auditory, motor, and visual senses, thereby promoting both learning ability and brain development [22]. Throughout the history of human development, since music and dance always appear simultaneously, dance choreography has been strongly influenced by music. People often dance to music as a form of ritual or etiquette at social occasions or festivals [17, 18]. Therefore, a successful dance synthesis algorithm could be beneficial in fields such as music-aided dance teaching [29], character motion generation for audio games [11] and research on human behavior [23].

Traditional studies on motion learning have mainly focused on recognizing actions from RGB videos recorded by 2D cameras [26]. However, capturing human motions in the full 3D space in which they are performed can provide more comprehensive information. Zhu et al. [30] successfully applied long short-term memory (LSTM) networks for motion recognition evaluated over several 3D-space motion datasets, including the SBU Kinect interaction dataset [28], the HDM05 dataset [21], and the CMU dataset [8]; however, the motions in these datasets are mainly limited to common motions such as "jumping" or "running", and data for dance motion learning are still lacking. Regarding dance motion synthesis, Cardle et al. [7] presented a general framework for the local modification of motions using perceptual cues extracted from music. Shiratori et al. [24] proposed an approach for synthesizing dance motions using fixed motion frames to perform each figure. In [2], the authors combined motion features from two databases to generate new dance motions. However, it has been demonstrated that the inputs

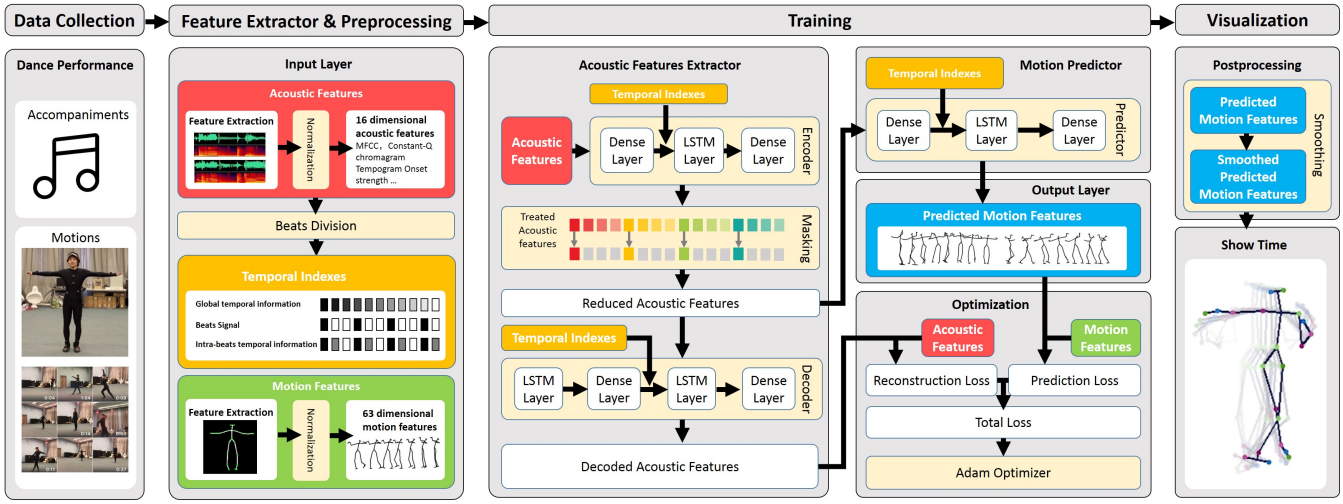

Figure 2: Workflow of our framework.

and outputs of the dance synthesis process in the abovementioned methods are incomplete. Each segment of dance figures is randomly selected from the database, leading to synthesized choreographies with little linguistic or emotional meaning. As shown in Figure 1 , a subtle change in melody may not cause dancers' choices of dance figures to change, but their local joint postures or rhythm of motion can be adjusted to suit the emotion of the music.

In our study, we summarize several challenges of music-dance synthesis: 1) the lack of available training data, 2) how to choose appropriate figures in accordance with music, and 3) how to create a model in which local joint postures and rhythm can be adjusted to suit musical emotion. To enrich the existing datasets, we constructed a music-dance dataset containing choreographies for four types of dance, totaling 907,200 frames of music and 3D motions, and extracted multidimensional features for model training. To further study the relationship between music and dance, we employed our dataset for model training and developed an LSTM-autoencoder model for motion synthesis. Moreover, several quantitative and qualitative experiments were conducted to optimize the model, and we selected the best-fitted model for music-oriented dance synthesis. Figure 2 illustrates the workflow of our study.

Our contributions can be summarized as follows:

- We constructed a music-dance dataset that contains 40 complete dance choreographies for four types of dance, totaling 907,200 frames collected with optical motion capture equipment (Vicon). We also recorded the music with which the choreographies were performed, making the collected data especially useful for music-oriented dance synthesis. We extracted comprehensive features, including 63-dimensional motion features, 16-dimensional acoustic features and 3-dimensional temporal indexes. Thus, we obtained pertinent, accurate, and complete features for the training of neural networks. To our knowledge, this is the largest music-dance dataset currently in existence. We have made our dataset openly available to facilitate related research.[1]
- We developed an LSTM-autoencoder model for music-oriented dance synthesis to better understand the harmonious relations between music and dance motions. Specifically, the model is designed to extract mappings between acoustic and motion features, such that the emotions of the music will be reflected by the synthesized dance. Thus, the model can learn how dancers adjust their local joint postures and rhythm of motion to express changes in musical emotions and the rules for choosing motions in choreography.
- We conducted several qualitative and quantitative experiments to quantify the performance of our model. As dance is a kind of artistic creation, we considered user evaluations in addition to the common Euclidean loss function to evaluate the performance of the model. The experimental results indicate that compared to several baselines, our model successfully extracts acoustic features related to dance figure choices and performs well in choosing dance figures that match the length, rhythm, and emotion of a piece of music. These experiments facilitate the understanding of the relationship between music and dance.

The remainder of the paper is organized as follows: Section 2 focuses on related work, Section 3 formulates the main problem, Section 4 presents the dataset, Section 5 presents the methodologies, Section 6 presents the experimental procedures and results, and Section 7 concludes the paper.

2 RELATED WORK

3D motion dataset. In previous studies, such as the one conducted by Fragkiadaki et al. [10], motion features have been extracted from videos. However, these motion features were based on 2D capture. Biological observations suggest that humans can recognize actions from just the motions of a few light displays attached to the human body [14]. The main existing datasets of this type are the SBU Kinect interaction dataset [28], which contains 230 sequences in 8 classes and 6,614 frames in total; the HDM05 dataset [21], which contains 2,337 skeleton sequences and 184,046 frames in total; and

[1]https://github.com/Music-to-dance-motion-synthesis/dataset

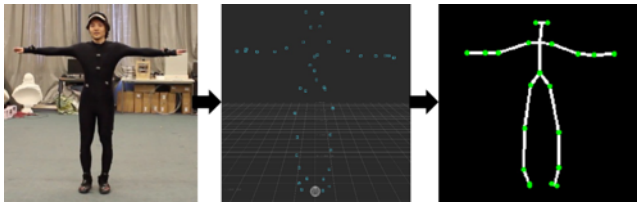

Figure 3: Motion capture, correction, and simplified visualization. The 20 simplified joint points that are indicated in this figure represent the human posture well.

the CMU dataset [8], which contains 2,235 sequences and 987,341 frames in total and is the largest skeleton-based human action dataset collected to date. However, the CMU dataset contains only 64,300 frames of pure dance motions, and these dance motions are discontinuous and not accompanied by music.

Therefore, regarding research on the relationship between music and dance motions, data for model training are still lacking.

Dance motion synthesis. Berman [5] extracted motion features within dance frames to build an action map to evaluate how motions could be harmoniously connected. Takano [25] produced a human motion generation system considering motion labels. Katerina [10] proposed an *Encoder-Recurrent-Decoder (ERD)* model for the recognition and prediction of human body poses. Although these methods analyze the relations between motions, they ignore the fact that music often strongly affects human dance motions.

Some works have focused on establishing mappings between acoustic and motion features. Kim [15] labeled music with joint positions and angles, and Shiratori [24] added gravity and beats as supplementary information for predicting dance motions. Recently, Manfrè [19] showed that a *hidden Markov model (HMM)* improved the fluency of motion cohesion according to human experience. However, since the motions used for mapping were fixed frames, whether the generated dance animation was vivid enough was entirely dependent on the number of motions present in the library.

Recently, Yaota [27] used several classic deep learning models for dance synthesis. Alemi [1] proposed methods of predicting subsequent frames from previous frames. However, these studies did not consider the beat information of the music, which proved to be very important.

Since these previous works have not successfully accomplished dance synthesis with rational figure sequences and artistic musical expression, the design of an appropriate music-oriented dance synthesis algorithm remains an open problem.

3 PROBLEM DEFINITION

The input dataset $F = \{A_i, M_i\}$ is a set of sequential features, consisting of acoustic features $A_i =< A_i^1, A_i^2, \ldots, A_i^{D_A} >$ and motion features $M_i =< M_i^1, M_i^2, \ldots, M_i^{D_M} >$. The acoustic features are extracted from the input audio files, and the motion features are extracted from the input dance capture data.

Our goal is to build a model $G(A \rightarrow M)$. First, our model is trained with the collected data $\{A_i, M_i\}$. Then, for any input music sequence A_i', our model can synthesize a new dance sequence M_i'.

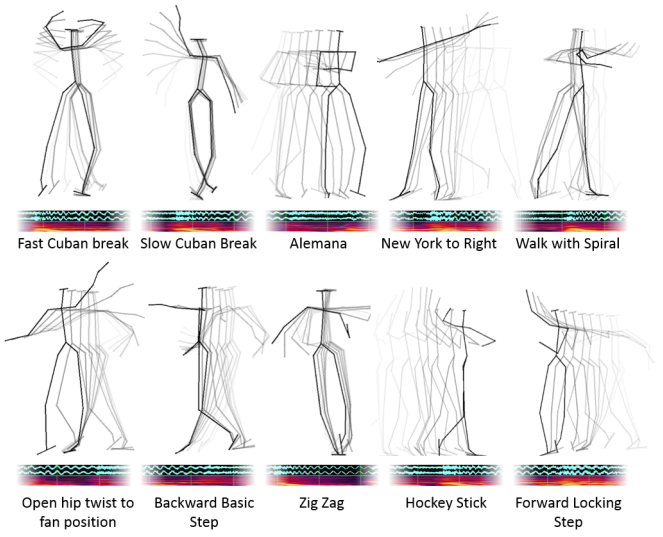

Figure 4: Our music-dance dataset.

4 DATASET

4.1 Data Acquisition

As mentioned in Section 2, although various motion datasets are available in the literature, there is still a lack of data on dance motions with music, which are necessary for studying the relationship between music and dance. A large set of motions for each type of dance is also needed to extract desirable features. To establish a dataset with sufficient and desirable data, we collected music-dance data as described below.

Motion data acquisition. We asked professional dancers to dance to music and captured their motions using *Vicon* optical motion capture equipment.[2] The captured data comprise four types of dance (waltz, tango, cha-cha, and rumba), totaling 94 minutes and 907,200 frames. We manually corrected the data and converted them into C3D format[3] via *Vicon Iq*. These data record the positions of 41 skeleton joints in 3D space in each frame. We used the Python c3d package to analyze the C3D files and reduced the 41 joint points to 21 joint points. A skeleton with a height of 1.62 meters in a neutral pose, as shown in Figure 3, was used for all experiments.

Index annotation. To find better mappings between music and motions, we asked professional dancers to tag all motions. According to the dancers, no matter how complicated a dance is, it is always possible to extract basic motions. The tagged motions were not used in model training but were useful in testing model performance.

Audio data acquisition. We also labeled the music with the dance figures performed at the corresponding time points.

[2] All dancers were female, with heights ranging from 1.68 meters to 1.72 meters.
[3] The C3D (Coordinate 3D) format provides a convenient and efficient means of storing 3D coordinates and analog data, together with all associated parameters, for a single measurement trial.

Feature	Sound Characteristic	Definition
MFCC	Pitch	$a_i^1, \ldots, a_i^3$
MFCC-delta	Pitch change	$a_i^4, \ldots, a_i^6$
Constant-Q chromagram	Pitch	$a_i^7, \ldots, a_i^{10}$
Tempogram	Strength	$a_i^{11}, \ldots, a_i^{15}$
Onset strength	Strength	a_i^{16}

Table 1: Acoustic features

Index	Definition
Arithmetic progression through the whole song	t_i^1
First frame of each beat	t_i^2
Arithmetic progression repeated within beats	t_i^3

Table 2: Temporal indexes

4.2 Feature Extraction

Our dataset provides extraction methods and extracted features for future use.

Three types of features were extracted from the original data.

- Acoustic features, labeled as $A_i =< a_i^1, a_i^2, \ldots, a_i^{16} >$.
- Motion features, labeled as $M_i =< m_i^1, m_i^2, \ldots, m_i^{63} >$.
- Temporal indexes, labeled as $T_i =< t_i^1, t_i^2, t_i^3 >$.

All features were normalized to ensure that a mean value of zero and a standard deviation of one for each sequence.

Acoustic features. An audio analysis library named *librosa* was used for music information retrieval, as proposed by McFee [20]. Librosa provides an easy means of extracting the spectral and rhythm features of audio data. Specifically, we chose the *mel frequency cepstral coefficients (MFCC)*, *constant-Q chromagram*, *tempogram* [12], and *onset strength* [6] as the acoustic features in our dataset.

In most cases, users are free to choose which features to use and how many dimensions their feature vector has. The pre-extracted acoustic features are shown in Table 1.

Temporal Indexes. The use of temporal indexes was inspired by the network training process. The temporal indexes were extracted from the acoustic features.

Tempo or beat information[4] is crucial for our task because all dance music has a fixed tempo (number of beats/bars per minute). When the recurrent layers in a neural network are fed beat information as input, it is easier for them to understand the sequence as a whole. Although the acoustic features contain the beat information, this information will gradually fade with increasing layer depth because the acoustic features are fed into the model only at the very beginning. Thus, it is unwise to use the beat information in the same way that we use other acoustic features. In addition to the beat information, there are some other features that are also simple and useful to all recurrent layers in our model. We selected those features as temporal indexes to be fed directly into all recurrent layers. These temporal indexes, listed in Table 2, can be regarded as the control signal for our model. The arithmetic progression is used to identify each frame's location. It specifies the location of each frame within a period of time. We built an arithmetic progression ranging from zero at the beginning of the music to one at the end to allow our model to easily identify the location of each frame. A similar method was used to represent the locations of frames within one beat.

[4]The librosa library also provides tools for beat extraction [9].

Index	Abbreviation	Definition
Head-left	Avg(LFHD LBHD)	m_i^1, m_i^2, m_i^3
Head-right	Avg(RFHD RBHD)	m_i^4, m_i^5, m_i^6
Waist	Avg(T10 STRN)	m_i^7, m_i^8, m_i^9
Shoulder-left	LSHO	$m_i^{10}, m_i^{11}, m_i^{12}$
Elbow-left	LELB	$m_i^{13}, m_i^{14}, m_i^{15}$
Wrist-left	LFRM	$m_i^{16}, m_i^{17}, m_i^{18}$
Hand-left	LFIN	$m_i^{19}, m_i^{20}, m_i^{21}$
Waist-left	Avg(LFWT LBWT)	$m_i^{22}, m_i^{23}, m_i^{24}$
Knee-left	LKNE	$m_i^{25}, m_i^{26}, m_i^{27}$
Ankle-left	LANK	$m_i^{28}, m_i^{29}, m_i^{30}$
Heel-left	LHEE	$m_i^{31}, m_i^{32}, m_i^{33}$
Toe-left	LTOE	$m_i^{34}, m_i^{35}, m_i^{36}$
Shoulder-right	RSHO	$m_i^{37}, m_i^{38}, m_i^{39}$
Elbow-right	RELB	$m_i^{40}, m_i^{41}, m_i^{42}$
Wrist-right	RFRM	$m_i^{43}, m_i^{44}, m_i^{45}$
Hand-right	RFIN	$m_i^{46}, m_i^{47}, m_i^{48}$
Waist-right	Avg(RFWT RBWT)	$m_i^{49}, m_i^{50}, m_i^{51}$
Knee-right	RKNE	$m_i^{52}, m_i^{53}, m_i^{54}$
Ankle-right	RANK	$m_i^{55}, m_i^{56}, m_i^{57}$
Heel-right	RHEE	$m_i^{58}, m_i^{59}, m_i^{60}$
Toe-right	RTOE	$m_i^{61}, m_i^{62}, m_i^{63}$

Table 3: Motion features

Motion features. The original data contained 41 joints in 3D space, from which we manually selected the 21 joints listed in Table 3 to represent the dancers' motions.

In the original data, the absolute position was used to locate each joint. We calculated the center of mass of each skeleton and re-expressed the position of each joint relative to the center of mass to ensure that identical movements performed in different locations would be regarded as the same.

In our model, the Euclidean distance is used to quantify the similarity of two frames. To ensure that the directions the dancers were facing would not influence the performance of our model, for each frame, we rotated the skeletons[5] to face the audience.

The input motion data contained several spurs. For spur removal, we applied a sliding window (4 frames wide) to the frame sequence. If the Euclidean distance[6] between the previous and next frames was larger than a certain threshold, the original motion was replaced with the linear interpolation between the two frames.

The acoustic and motion features were extracted with two different methods, so it was possible for them to have different sampling rates. Thus, we needed to align them with each other.[7] Acoustic

[5]The angle of rotation was determined from the dancer's orientation, which could be best identified from the waist joints (left-waist and right-waist).

[6]The Euclidean distance between two frames is defined as the sum of the Euclidean distances between corresponding joints.

[7]Specifically, the acoustic features were sampled at 25 frames per second, whereas the motion features were sampled at 40 frames per second.

features are difficult to interpolate, so we applied a *linear interpolation algorithm* to the motion features instead. The final acoustic and motion features have all been scaled to a speed of 25 frames per second (fps).

5 METHODOLOGIES

To extract the mappings between the acoustic and motion features, we developed an *LSTM-autoencoder* model. The model contains several basic layers, such as LSTM layers and dense layers.[8]

5.1 Fundamental Models

Here, we introduce the two fundamental models employed in our synthesis approach.

LSTM. *LSTM* [13] is a network model based on a *recurrent neural network (RNN)* that has proven successful at extracting time series of features. Compared with an RNN, an LSTM network contains additional forget gates and memory blocks, which help it to perform better in sequence-to-sequence mapping.

Autoencoder. An *autoencoder* [3] [4] is a machine-learning tool used to reduce feature dimensions. An autoencoder can compress a high-dimensional vector into a reduced-dimensional vector without losing too much essential information.

An autoencoder consists of two neural networks. The first network is called the *encoder*. When the original high-dimensional vector x is fed into the encoder layers, it will be translated into a reduced-dimensional feature vector z.

$$z = Encoder(x) \tag{1}$$

The second network is called the *decoder*. The decoder reads the feature vector z and predicts the original feature vector.

$$x' = Decoder(z) \tag{2}$$

The loss function of the model is defined as the mean Euclidean distance between the predicted vector x' and the real vector x.

$$minimize \sqrt{\|x' - x\|^2} \tag{3}$$

To reduce loss, the encoder tends to preserve as much information as possible, while the decoder ensures that the feature vector z contains sufficient information to represent the original vector.

5.2 Approach Description

LSTM Approach. As a preliminary attempt, we used an LSTM network as the main basis of our model. In this model, the acoustic features and temporal indexes of each frame are transformed into hidden features via a dense layer, introducing additional nonlinearity into the model. Then, this series of hidden features is fed into a three-cell LSTM layer. The LSTM structure has a memory channel C_t that stores useful information from previous outputs as it passes them to subsequent cells. Finally, the outputs of the LSTM layer are transformed by a dense layer to predict a series of motion features.

[8]Fully connected layers.

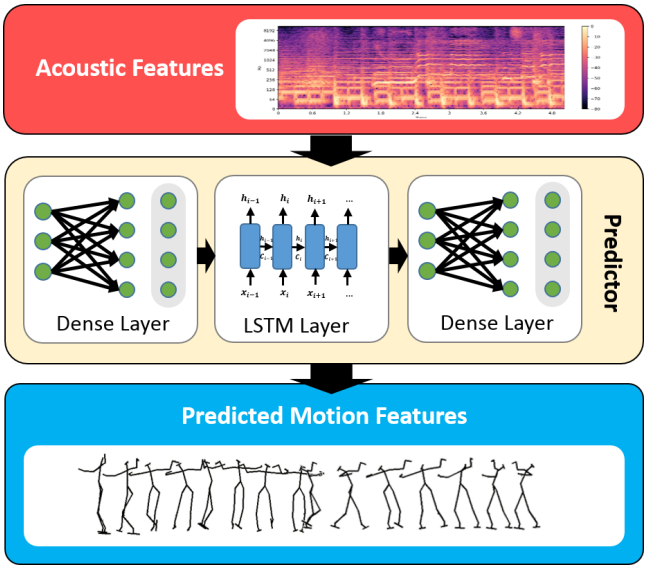

Figure 5: Structure of the LSTM approach.

The objective function of the model is the Euclidean distance between the predicted and ground-truth motion feature series. The stages of the model are illustrated in Figure 5.

This approach can learn the relationship between music and dance to some extent. However, the following problems arise:

- The acoustic features have a high dimensionality, and it is difficult for this model to summarize the mapping between music and motions. Consequently, this model has difficulty converging.
- It is difficult for this model to learn that motions associated with single beats cannot be divided. Motion integrity should instead be enforced by the structure of our model.

LSTM-autoencoder Approach. To address the difficulty of learning the acoustic features, we made some improvements to the LSTM approach. As a complement to the original *Motion Predictor* module, we added an *Acoustic Feature Extractor* module to our model.

The *Acoustic Feature Extractor* is a network designed to reduce the dimensionality of the acoustic features. In the original acoustic features, each frame has a feature vector. However, as mentioned above, the motion chosen for one beat should remain undivided; therefore, we should introduce some structure to compress the frame-indexed acoustic features into beat-indexed acoustic features.

An autoencoder design is used for the *Acoustic Feature Extractor*, which takes the acoustic features A_i and the temporal indexes T_i as input. We use an LSTM network to encode the acoustic features.

In general, the basic model structure is as follows:

$$Z'_i = Encoder(concat(A_i, T_i)) \tag{4}$$

$$Z_i = Masking(Z'_i) \tag{5}$$

$$A'_i = Decoder(concat(Z_i, T_i)) \tag{6}$$

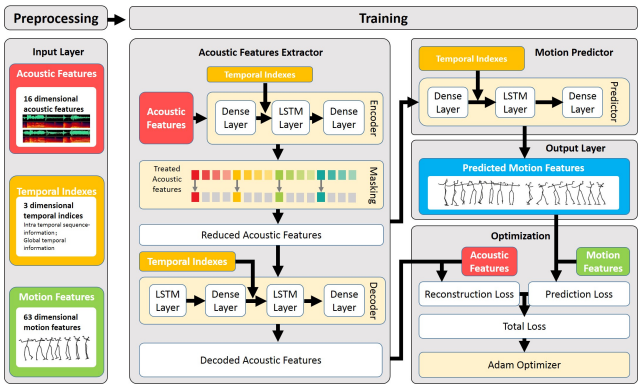

Figure 6: Structure of the LSTM-autoencoder model.

A_i is the original feature vector. Z_i is the reduced-dimensional feature vector extracted from A_i. T_i is the vector of the temporal indexes. *concat* is the vector-concatenating operation.

Encoder is a network with the following layers:

- A dense layer to transform the original features into hidden features.
- An LSTM layer to transform the hidden features and temporal indexes into a reduced-dimensional hidden feature vector.
- A second dense layer to perform the encoding process.

Masking is a layer introduced to transform the frame-indexed acoustic features into beat-indexed acoustic features, thus reducing their dimensionality. (This also reduces the likelihood of overfitting). The input vector Z'_i is a time series vector. The index i corresponds to one audio frame. Obviously, the feature vector Z'_i contains too much information. We use this layer to mask off extraneous information. In practice, we force the value of Z'_i to be zero unless i is the first frame of a beat. The prediction of the next motion is mostly based on this frame.

The *Decoder* network has a structure similar to that of the encoder network. It also consists of a dense layer, an LSTM layer, and another dense layer, in order of the data flow.

The *Motion Predictor* module uses the reduced-dimensional acoustic features Z_i to predict the motion features M_i. The network consists of an LSTM layer sandwiched between two dense layers. These layers convert the reduced acoustic features Z_i into the predicted motion features M'_i.

The final loss of our model is the combination of two losses.

The loss of the *Acoustic Feature Extractor* ($loss_{extr}$) is defined as the Euclidean distance between the original and predicted acoustic features:

$$loss_{extr} = \sqrt{\|A'_i - A_i\|^2} \tag{7}$$

The loss of the *Motion Predictor* ($loss_{pred}$) is defined as the Euclidean distance between the predicted and ground-truth motion features:

$$loss_{pred} = \sqrt{\|M'_i - M_i\|^2} \tag{8}$$

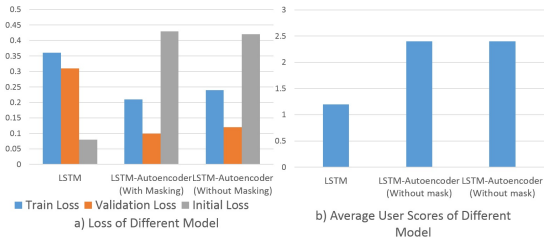

Figure 7: a) Quantitative losses of different models. b) Qualitative user scores of different models.

The final loss is the combination of these two losses. While $loss_{pred}$ should be as small as possible, $loss_{extr}$ should also be small to ensure that the extractor preserves the essential information from the original acoustic feature vector. However, sometimes the network will seek a smaller $loss_{extr}$ by allowing $loss_{pred}$ to increase, which is unacceptable. Hence, we define our objective loss function as follows:

$$loss = max(E_{th}, loss_{extr}) + loss_{pred} \tag{9}$$

The value E_{th} is a threshold value.[9] When $loss_{extr}$ is greater than E_{th}, it has a nonzero derivative. In this case, our model will choose to minimize $loss_{extr}$. When $loss_{extr}$ is less than E_{th}, optimizing $loss_{extr}$ will be as difficult as optimizing $loss_{pred}$, and thus, we do not attempt to optimize $loss_{extr}$.

Once the value of $loss$ has been computed, the network parameters are updated by the Adam update algorithm [16].

6 EXPERIMENTS

In this section, we report several qualitative and quantitative experiments conducted to optimize our model by evaluating the effects of synthesizing music-oriented choreography with different models and parameter settings.

6.1 Metrics

To evaluate the dance synthesis performance achieved using different models and parameter settings, we considered both the Euclidean losses for quantitative comparisons and user evaluations for qualitative comparisons.

$$metric\_loss = \sum_{t \in frames} \sqrt{\sum_{i \in frame[t]} \left(X_{t,i} - X'_{t,i}\right)^2} \tag{10}$$

In our quantitative experiment, we calculated the Euclidean losses between real and synthesized dances on a validation set.

The qualitative experiment was divided into 2 stages. We recruited 20 participants (10 male and 10 female) for our experiment.[10] In the first stage of the experiment, we asked the participants to score the dances synthesized by different models. In the second stage, we asked the participants to evaluate how well the dances we synthesized fit the corresponding music.

[9] E_{th} has a value of 0.45 in practice.
[10] Thirteen of the participants were professional dancers.

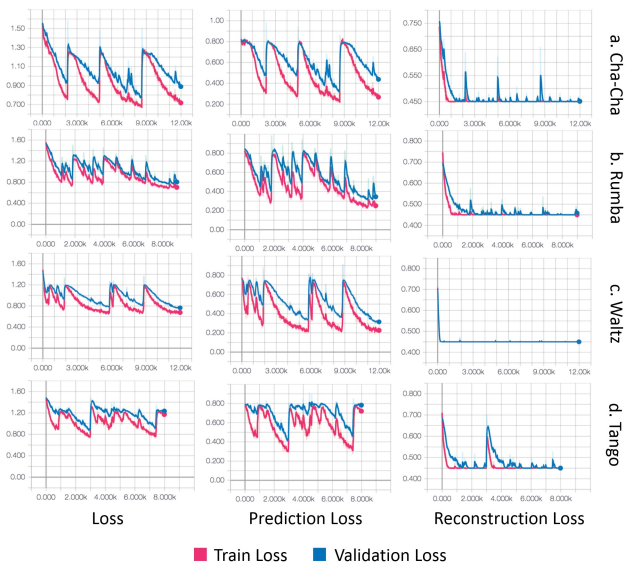

Figure 8: Losses of our model when fitting each type of dance (cha-cha, rumba, waltz, and tango). The x-axis represents the number of training iterations, and the y-axis represents the loss.

6.2 Methods for Comparison

As discussed in section 5.2, the proposed LSTM-autoencoder model is based on an initially developed LSTM model. Moreover, the LSTM-autoencoder model can either include the masking layer or not. Thus, we compared the three models listed below:

- The plain 3-layer LSTM model.
- The LSTM-autoencoder model without the masking layer.
- The LSTM-autoencoder model with the masking layer.

We compared these models in quantitative and qualitative experiments.

Quantitative Experiment. All three models shared the same set of training parameters.

- The model loss was defined as the minimum loss over 10,000 iterations.
- The learning rate was 10^{-4}.
- The batch size was 20.
- The data were split into 15% for validation and 85% for training.
- TensorFlow was used as the computational framework.
- The LSTM cells used in the *prediction layers* consisted of 3 sequential *basic LSTM cells*.

We measured the performance of the models in both quantitative and qualitative experiments.

In the quantitative experiment, we ran each model 10,000 times and recorded the minimum validation loss as the best performance.

Qualitative Experiment. In the first stage of the qualitative experiment, we generated four dances with each of the three models. We asked the participants to score the dances synthesized by the different models from 4, representing the best, to 1, representing the worst, and calculated the average user score for each model.

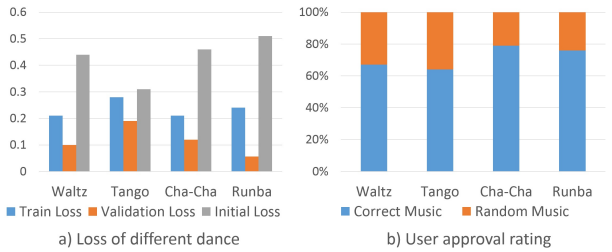

Figure 9: a) Losses for different type of dances. b) User approval ratings.

In the second stage, we evaluated how well the dances we synthesized fit the music. We selected 5 songs of the same tempo for each type of dance and synthesized a dance for each song. In the experiment, we showed the participants two dances accompanied by the same song; one was synthesized based on that song, while the other was randomly selected from among the other 4 dances synthesized for the other songs. We asked the participants to determine which of the two better expressed the emotions in the music and calculated the user approval ratings for the "matching dance" and the "non-matching dance" as the percentages of participants who indicated that the corresponding dance was superior.

6.3 Results and Analyses

Quantitative Experiment. We used our proposed model to fit 4 types of dances. The loss functions[11] are shown in Figure 8.

The local minima of each plot in Figure 8 correspond to good model fits, indicating that the model is capable of synthesizing motions in accordance with the acoustic features. We conducted this quantitative experiment with each model discussed in section 6.2, and the results are shown in Figure 7. The autoencoder strategy has a clear influence on the loss of our model. The masking strategy may not have an observable influence on the loss, but it does reduce the dimensionality of the feature vectors fed into the *Motion Predictor* module discussed in section 5.2. The quantitative results indicate that the most suitable model for our task is the LSTM-autoencoder model with the masking strategy.

Qualitative Experiment. Our qualitative experiment consisted of two stages: model comparison and dance analysis. We then conducted a final experiment for performance verification.

In the first stage of the qualitative experiment, as shown in Figure 7, the average user scores for each model were 1.31 for the LSTM model, 2.33 for the LSTM-autoencoder model with masking, and 2.31 for the LSTM-autoencoder model without masking. Thus, the LSTM-autoencoder model with masking outperformed the others.

In the second stage, the matching dance synthesized based on the given song was more likely than the randomly selected dance to be chosen by the participants as the dance that better expressed the music's emotions. The approval ratings for the "matching dance" were 67% for waltzes, 65% for tangos, 76% for rumbas, and 79% for cha-chas. Notably, the cha-cha was the type of dance for which the "matching dance" had the highest probability of being selected,

[11] In our model, due to normalization, the *prediction loss* and *reconstruction loss* range in value from 0 to 1, while the *loss* ranges from 0 to 2.

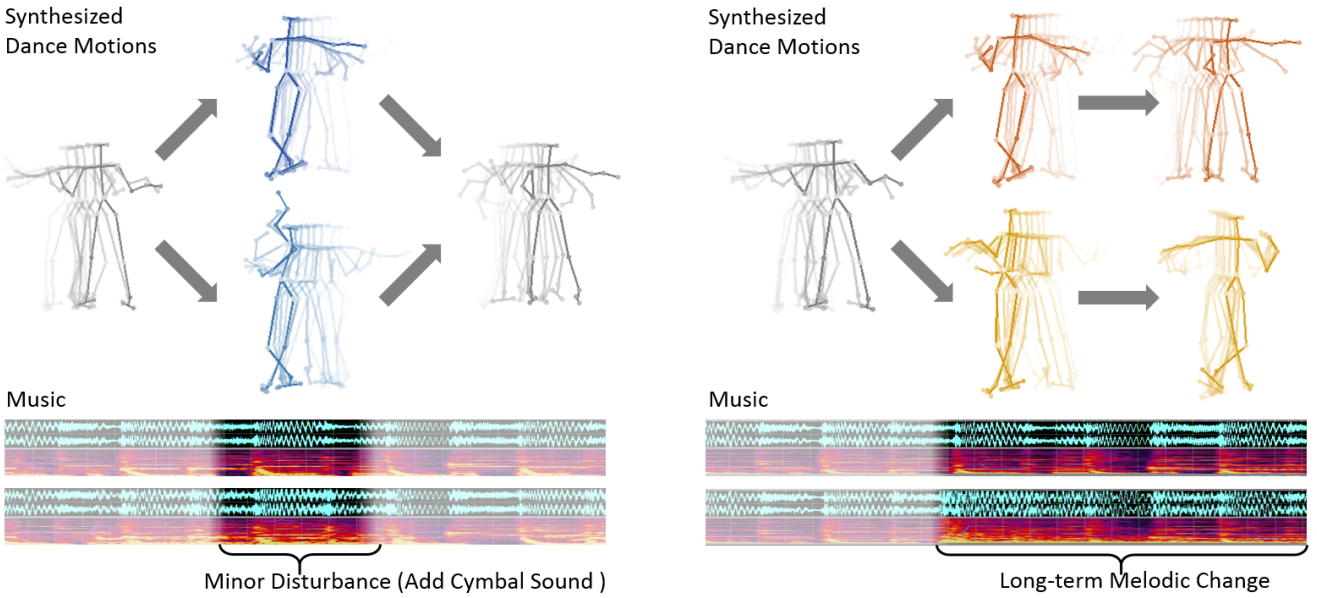

Figure 10: The dance motions synthesized by our model corresponding to different changes in the music. As shown in this figure, minor changes in the music will trigger only subtle adjustments in the dance motions, while marked changes in the music will lead to entirely different motions.

possibly because the cha-cha figures show more prominent emotional expression and the choreography is more consistent with the music. The results of this experiment are shown in Figure 9. This experiment also demonstrates that our model is capable of music-dance synthesis for multiple dance types.

Finally, we sought to verify that our model could overcome the challenges discussed in section 1. First, we added a short sequence of drumbeats into a melody. The synthesized dance changed in response to the drumbeats, and when the drumbeats ended, the dance went back to its original state. Then, we replaced the melody with a completely new one, and the entire corresponding segment of choreography also changed. The results are shown in Figure 10.

6.4 Discussion

Based on the qualitative and quantitative experimental results, we find that using only Euclidean losses for dance synthesis evaluation is insufficient. Since dance is a kind of artistic creation, different dancers may create different dance motions for the same music in accordance with their personal styles, as shown in Figure 10. Moreover, within one dance, choreographers will try to avoid repeating exactly the same dance movements to enrich the expression. For example, in a tango, when executing a "basic reverse turn", dancers have a variety of choices, all starting with the man's left foot, such as the "five step" and the "contra check". However, the "contra check" may be the better choice if the next beat is a strong, crisp note, because this figure is a one-step quick pose showing the lady's line; the "five step", by contrast, is a moving figure that does not end in a pose until the fifth step, so it may be more suitable for a piece of melody that lasts into the second bar. Moreover, depending on the musical structure and expression, the "contra check" may last either 2 beats or 4 beats, and the "five step" may take the alternative form of the "extended five step", which has 7 steps and lasts two more beats, in order to match the punctuation of the music. Therefore, each dance movement in the real world is unique, and for an acoustic-motion mapping evaluation, it is meaningless to consider only the error between the output and the ground truth. In this study, we were committed to studying the rules of dance choreography as manifested in existing datasets and generating new dance choreographies according to these rules. Moreover, we considered that these synthesized choreographies should satisfy human aesthetics. By jointly considering user evaluations and Euclidean losses, we established a better evaluation mechanism as a basis for further optimizing our model.

7 CONCLUSIONS

In this study, we presented a model for music-oriented dance synthesis that uses acoustic features as input and outputs synthesized dance choreographies in accordance with the input music while achieving richer expression and better continuity. Our future work will focus on three major objectives: 1) acquiring more data to continuously enhance our dataset, 2) considering different user preferences with regard to the synthesized dances, and 3) leveraging our model to build various applications.

8 ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Plan (2016YFB1001200),the Innovation Method Fund of China (2016IM010200).

REFERENCES

[1] Omid Alemi, Jules FranÃğoise, and Philippe Pasquier. 2017. GrooveNet: Real-Time Music-Driven Dance Movement Generation using Artificial Neural Networks. In *Workshop on Machine Learning for Creativity, 23rd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[2] Wakana Asahina, Naoya Iwamoto, Hubert PH Shum, and Shigeo Morishima. 2016. Automatic dance generation system considering sign language information. In *ACM SIGGRAPH 2016 Posters*. ACM, 23.

[3] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*. 37–49.

[4] Yoshua Bengio et al. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.

[5] Alexander Berman and Valencia James. 2015. Kinetic Imaginations: Exploring the Possibilities of Combining AI and Dance.. In *IJCAI*. 2431.

[6] Sebastian Böck and Gerhard Widmer. 2013. Maximum filter vibrato suppression for onset detection. In *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx). Maynooth, Ireland (Sept 2013)*.

[7] Marc Cardle, Loic Barthe, Stephen Brooks, and Peter Robinson. 2002. Music-driven motion editing: Local motion transformations guided by music analysis. In *Eurographics UK Conference, 2002. Proceedings. The 20th*. IEEE, 38–44.

[8] CMU. 2003. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/.

[9] Daniel PW Ellis. 2007. Beat tracking by dynamic programming. *Journal of New Music Research* 36, 1 (2007), 51–60.

[10] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 4346–4354.

[11] S Gentry and E Feron. 2004. Modeling musically meaningful choreography. In *IEEE International Conference on Systems, Man and Cybernetics*. 3880–3885 vol.4.

[12] Peter Grosche, Meinard Müller, and Frank Kurth. 2010. Cyclic tempogramâĂŤa mid-level tempo representation for musicsignals. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 5522–5525.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Gunnar Johansson. 1973. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics* 14, 2 (1973), 201–211.

[15] Jae Woo Kim, Hesham Fouad, and James K Hahn. 2006. Making Them Dance. (2006).

[16] D Kinga and J Ba Adam. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[17] Allana C. Lindgren. 2011. Rethinking Automatist Interdisciplinarity: The Relationship between Dance and Music in the Early Choreographic Works of Jeanne Renaud and FranÃğoise Sullivan, 1948-1950. *Circuit Musiques Contemporaines* 21, 3 (2011), 39–53.

[18] Jiyue Luo. 2009. Looking on the Relationship between Music and Dance from the Angle of World Multicuturalism. *Explorations in Music* (2009).

[19] Adriano Manfrè, Ignazio Infantino, Filippo Vella, and Salvatore Gaglio. 2016. An automatic system for humanoid dance creation. *Biologically Inspired Cognitive Architectures* 15 (2016), 1–9.

[20] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*. 18–25.

[21] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. 2007. Documentation mocap database hdm05. (2007).

[22] Jean Piaget and W Mays. 1997. The principles of genetic epistemology. *Philosophical Quarterly* 24, 94 (1997), 87.

[23] F. H. Rauscher, G. L. Shaw, and K. N. Ky. 1995. Listening to Mozart enhances spatial-temporal reasoning: towards a neurophysiological basis. *Neuroscience Letters* 185, 1 (1995), 44–47.

[24] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. 2006. Dancing-to-Music Character Animation. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, 449–458.

[25] Wataru TAKANO, Katsu YAMANE, and Yoshihiko NAKAMURA. 2010. Retrieval and Generation of Human Motions Based on Associative Model between Motion Symbols and Motion Labels. *Journal of the Robotics Society of Japan* 28, 6 (2010), 723–734.

[26] Daniel Weinland, Remi Ronfard, and Edmond Boyer. 2011. A survey of vision-based methods for action representation, segmentation and recognition. *Computer vision and image understanding* 115, 2 (2011), 224–241.

[27] Nelson Yalta, Tetsuya Ogata, and Kazuhiro Nakadai. [n. d.]. Sequential Deep Learning for Dancing Motion Generation. ([n. d.]).

[28] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. 2012. Two-person interaction detection using body-pose features and multiple instance learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 28–35.

[29] Xiaoyan Zhang. 2013. *Impact of Music on Comprehensive Quality of Students in Sports Dance Teaching*. Springer London. 679–683 pages.

[30] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, Xiaohui Xie, et al. 2016. Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks.. In *AAAI*, Vol. 2. 8.