Audio-driven Human Dance Motion Synthesis: Deep Learning Auto-choreography Model with RNN Freestyle Dancer vs. Choreographer

Yanqiong Zhang

Abstract

This proposal proposes the research in robotic creativity which mainly focus on dance motion synthesis based on a piece of given music, which could contribute to online dance self-learning, dance game improving, human-machine competition etc. To implement auto-choreography in terms of two circumstances(freestyle dancer and choreographer), two Recurrent Neural Network(RNN) models are proposed: a two-abstract-layer RNN model and a Long Short Term Memory(LSTM) RNN with attention model. Similar to human dancers, a freestyle dancer model is supposed to generate dance motion sequences in real time, simultaneously while reading the audio input, whereas a choreographer model should generate the complete choreography art work after reading and processing the whole piece of music.

1 INTRODUCTION

As AlphaGo, a program developed by DeepMind team of Google, defeated a number of human Go master players in 2016, robotic creativity and machine intelligence is obtaining an sharp increasing amount of attention[14]. From music generation[16][15], art style transfer[10] to intellectual dialogue[20], auto-driving[27][22], programs are accomplishing more and more complex and creative tasks. In this proposal, an intellectual art generation problem will be addressed: Can a program dance to the music?

The dancing style in this proposal will be street dance. Similar to human street dancers, the program need to solve two occasions: freestyle with a piece of random unknown music and bring out a choreography art work with a certain music. Simulating a human freestyle dancer performing a random song, the former circumstance requires the program to generate dance motion sequences simultaneously while hearing the song. Every output dance move is generated based on the preceding music and dance motion, without looking at the later unknown part of the music. The later occasion allows the program to look at the whole piece of music and generate the choreography accordingly. With the complete scope of the audio input, the generated choreography is expected with higher integrity and coherence.

1.1. Importance

Dance online self-learning. Dance, as an artistic form that shows a significant dependence on physical expression, relies on face-to-face teaching sessions. Under the background of Coronavirus in 2020, the limitation of offline gathering drives the dance studios and dance departments of

universities to arrange online teaching sessions. However, the inconvenience was revealed. Due to the limited sight, the students are only able to watch the teacher's demonstration from a single angle. Thus many details and useful information can be lost. In addition, as the choreography is created by the teacher manually, the content and music of the choreography are determined for students. An 3D auto-choreography program can help with this circumstance commendably. With this model, students will be able to learn novel choreography routines generated with any music they provide, from a 3D figure.

Inspiration. In addition, for the human choreographers, the generated dance routines can also be used as referential art works for inspiration.

Dancing game improvement. Further more, an auto-choreography program can promote the ability of dancing games(like Just Dance) to interact with human players. For instance, it allows the players to upload or just play their own music directly to generate a tailored choreography. At the same time, there could be a virtual freestyle dancer as the opponent of the player in a freestyle battle mode.

human-machine competition. Inspired by AlphaGo, if the model was physically implemented as a dancing robot in further research, it can even participate in street dance freestyle battle competitions, battling against human dancers.

1.2. Barriers

Human motion synthesis. Due to the continuity, high dimensionality and the complexity in both temporal and spatial aspects of human movements, human motion synthesis seems to be one of the top laborious tasks in computational modelling research. Performing even just typical movements in daily life like walking or picking up a cup calls for sophisticated and refined models that is able to extract and handle the large amount of joint coordination data.[2][29]

Dance motion simulation. Unlike normal task-driven movements(like aforementioned walking and picking up a cup), dancing lies more on the execution aspect instead of the functional aspect[2]. As a highly creative art work, dancing requires superior coordination, motor control, flexibility and equilibrium.[5] Instead of just accomplishing a certain task, dance simulation focus more on the naturalness, continuity and aesthetic quality of the generated motion sequences.[1]. However, the physiological and biochemical factors of human dance motion are both hard to be modelled mathematically, which could lead to rigid output movement sequences.[30]

Dance to the music. Adding music to the task makes it even more complex, because it then involves the features, genres, dynamics, structures of both the motion capture data and the audio data. In order to generate choreography that admirably fits with the music, with accurate timing partition, fine rhythmic synchronization and harmony style pattern, the mapping relationship between the music and the movements needs to be captured precisely. However, the only relationship that could be observed directly is the temporal synchronization. As for the motion features and musical features, there is no previous well-established mapping relationship between them[6][24]. Thus, the mapping between the music audio and the dance movements is time-dependent and highly non-linear.[1]

2 RELATED WORK

2.1. Dance motion synthesis with machine learning

Machine learning has been used in a number of approaches to generate continuous dance motion automatically.

Li et al. proposed a two-level model for dance motion synthesis based on Hidden Markov Models(HMM)[21]. They defined the motion texture as a set of fundamental elements of the motion data(noted as motion textons) and their distribution. The two-level model they proposed uses Linear Dynamic Systems(LDS) to model motion textons(texton level) and a transition matrix to represent their distribution(distribution level). The motion texture will be firstly learnt from the dance motion capture data with a maximum likelihood algorithm and later be used to generate novel dance motion sequence with a two-step synthesis algorithm.

Another HMM-based approach has been proposed by Wang et al[30]. By training a Hierarchical Non-parametric Hidden Markov Model(NPHHMM), they simulated a set of dance motions including ballet walk, ballet roll, disco, and complex disco. Differ from the first-order HMM, NPHHMM is designed with non-parametric output densities and longer memory, which can help with avoiding the loss of explicitly compressing the motion frames and conveying the output with better accuracy.

Apart from the aforementioned approaches, Crnkovic-Friis et al. tried to generate meaningful, natural and continuous dance choreography with a deep Recurrent Neural Network(RNN) with Long Short Term Memory(LSTM)[8]. To better handle the continuous data and solve the problem of stagnating output, they added a mixture density network to the output of LSTM. Being trained for sufficient time(48h) with a large training set(5h of contemporary dance material), the model is able to capture the dancing style, syntax and basic semantics shown in the training corpus and generate admirable novel choreography.

The approaches above are mainly focusing on motion-data-driven choreography synthesis. The characteristics of the motion capture data itself are learnt by the models to generate novel dance move sequences, without any music input.

2.2. Music-driven dance motion synthesis

There are also several researches dedicate to music-driven dance motion synthesis.

Lee et al. introduced an approach of pre-defining a database of music-motion pairs and retrieve the pair whose music piece has the highest similarity with the given music segment[19]. By concatenating the motion pieces retrieved according to the given music segments, the output choreography can be generated. As the database was built from the choreography corpus, the mapping relationship from music to dance motion can be kept nicely. However, the dance motion sequence can show poor consistency and innovativeness because it is synthesized with a fixed set of discrete motion pieces. For the same piece of music, it will always output the same choreography. In addition, a pre-defined music-motion-pair database can be tremendous and memory consuming for a huge choreography corpus.

Ofli et al. proposed an HMM-based system of an audio-driven dancing avatar[25]. They capture

the multi-view motion data and manually segment it into semantic recurring patterns(also noted as dancing figures), each of which is modelled with an HMM. This dance pattern analysis process has been improved in their further research by introducing unsupervised learning [26]. For synthesis, the given music is firstly classified by genre based on Mel Frequency Cepstral Coefficients(MFCC) and HMM-based classification technique[3]. Then the corresponding motion pattern can be chosen accordingly to generate the output choreography. The main limitation of this model is that the possibility of generating novel dancing patterns is relatively low, due to its dependency on the classification of the audio[1].

Another model called GrooveNet has been introduced by Alemi et al., training the Factored Conditional Restricted Boltzemann MachinesFCRBM)[28] on a small 3D training set(about 23 minutes consists of four pieces of dancing performance)[1]. In an unsupervised manner, it learns a continuous mapping between audio data and dance motion data, without any classification or segmentation. According to their experiment, GrooveNet is able to generate decent output dance motions fit with the music appeared in the training set, but not with the unheard songs, which indicates that the model may suffer from overfitting.

Lee et al. proposed an Encoder-decoder Recurrent Neural Network deep learning model to generate novel k-pop dance movements in a 2D representation[18]. In each time step, the model encodes the audio input as well as the previous motion frame with Casual Dilated Highway Conv. Blocks and uses the encoded information in the decoding process to produce the output dance motion frames. The generated choreography shows a significant higher correlation with the input music than with a random piece of music, but still has a large gap with the ground truth choreography.

3 PROPOSED APPROACH

3.1. Data pre-process

As the model is intended to learn mapping relationships between two modalities(music audio and dance motion), it requires reasonable representations of the raw audio and motion data as input and output.

One of the proposed musical representation in this approach is mel-spectrogram, which can be extracted from the raw audio sound track, with a feasible window size matching the temporal density of the motion frame series. Alternatively, similar to image encoding and word embedding, the music notes can be used to produce a musical feature embedding representation temporally align with the motion frames, using the WaveNet-style autoencoder proposed and trained by Engel et al.[9]

As for the motion capture data, the human figure will be modelled with N_j joints as well as a global position, with $3 * N_j$ rotation angles and 3 displacement global translations involved. Exponential maps[12] will be used for 3D rotations to represent the joint rotation angles.

3.2. Freestyle dancer model

As mentioned in previous sections, the functional requirements of the freestyle dancer model is generating dance move sequences simultaneously while processing the music input data. This means while reading the audio input, at each time step t, the new audio input A_t will be fed into the model and the output dance motion $\hat{M}_t$ needs to be generated based on the new music input unit as well as the information from the memory of the previous time steps 1 to $t-1$.

3.2.1 Model intuition

Referring to the process of a human street dancer dancing to a random song, there would usually be a pre-listening stage and a freestyle stage. Before actually dancing, the dancer would typically listen to the starting part of the music to get a sense of the genre, tempo and fundamental structure of the music, which is called the pre-listening stage here. Stepping into the freestyle stage, the dancer then expresses the music with dance moves. In this proposal we regard the mapping relationship between the music and the dance moves as the "pattern" of expressing the music, which depends on the features of music and the dancer's personal performing characteristics.

Although a large proportion of street dance music shows fundamentally repetitiveness and structural periodicity, a freestyle dancer would not express two similar music sections in the same way. Observing the freestyle performance from a temporal aspect, the pattern a dancer expresses the music is variational and dynamic, which could also be regarded as a kind of sequence.

Thus, in the proposed model, the audio and motion data will be segmented into several sections(a section sequence) and two abstract layers will be used to respectively capture the pattern of each section and the dynamics of patterns over the section sequence. This mechanics will be explained in detail in the next subsection.

3.2.2 Learning

1. Overview of the mechanics
   As illustrated in Figure 1, after pre-processing, the audio data and motion data are of the same frame density, with T time steps in total. Firstly, the data will be divided into pre-listening stage and freestyle stage. The freestyle stage is then going to be segmented into N_s sections using beat detection[23][25], with $N_{slength}$ eight-beats per section, and eventually be fed into the model.

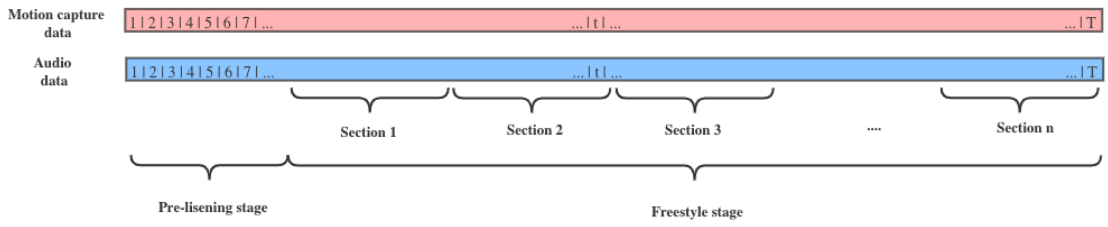

Figure 1: Data segmentation

The proposed model is composed of two abstract layers: pattern capturing and pattern sequence generating. As shown in Figure 2, the scope of the first layer is a single section s, the aim of which is capturing the pattern of s, noted as P_s. Using a many-to-many Recurrent Neural Network with $T_x = T_y = T_s$, the first layer takes the T_s audio frames $[A_1, A_2, ..., A_{T_s}]$ as input and outputs the T_s motion frames $[\hat{M}_1, \hat{M}_2, ..., \hat{M}_{T_s}]$ (noted as M1_pre etc. in the figure). Those predictions will then be paired with the ground truth motion frames to compute the cross-entropy loss L_1. Noticeably, this RNN is only for section s and will be trained with only one piece of training data. After training, the set of parameters in the RNN, noted as $Para_s$, is here regarded as a representation of the pattern of section s. In other words, the way the dancer expresses the music in section s is represented by the set of parameters which is used to map the audio data to the motion data in this certain section. By training N_s RNNs for the n sections $[Section_1, Section_2, ..., Section_n]$, N_s sets of parameters can be acquired and can form a temporal pattern sequence which will be fed into the second abstract layer.

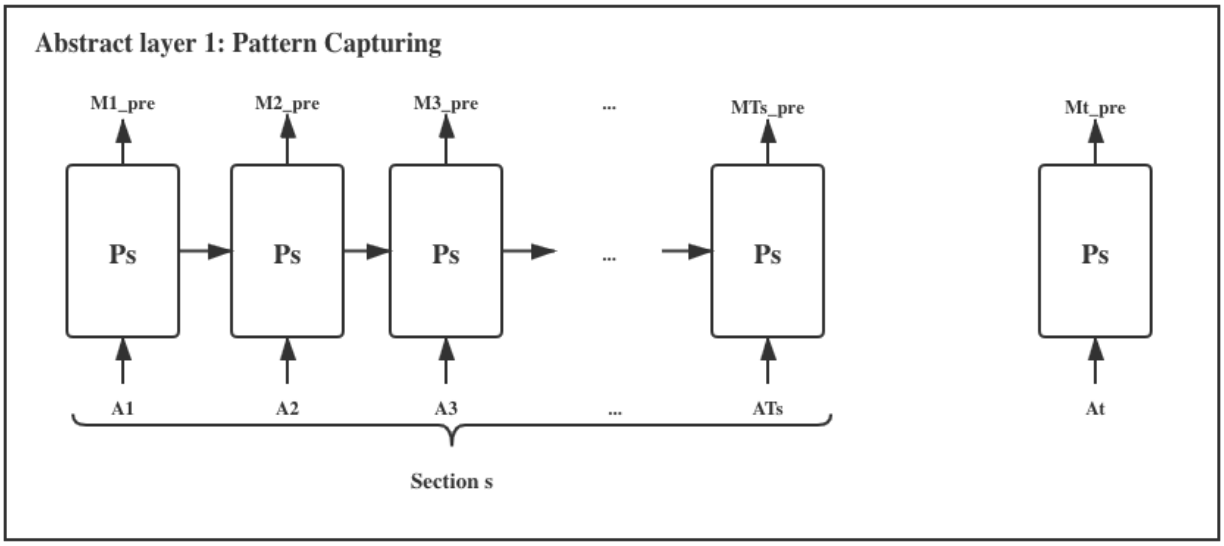

Figure 2: The first abstract layer of the model: pattern capturing

The second layer trains a one-to-many Recurrent Neural Network with the parameter set sequence produced by abstract layer 1, as shown in Figure 3. Before training, each parameter set will be concatenated and flattened into a single vector and an initial activation state a_0 will be produced with the audio data in the pre-listening stage(see Figure 1). Subsequently, in any single time step n in $[1, 2, ..., N_s]$, $Para_{n-1}$ will be fed into the network to generate the prediction $\hat{Para}_n$(noted as Paran_pre in the figure). The cross-entropy loss L_2 will then be computed with $\hat{Para}_n$ and $Para_n$. Being trained with m pattern sequences computed from m training examples(m freestyle dancing videos), the network in abstract layer 2 is expected with the ability of generating a novel parameter sequence with a given initial activation state a_0.

2. Computing details
   This subsection explains the details of computing units in the proposed model, which are indicated by the rectangles in Figure 2 and Figure 3.

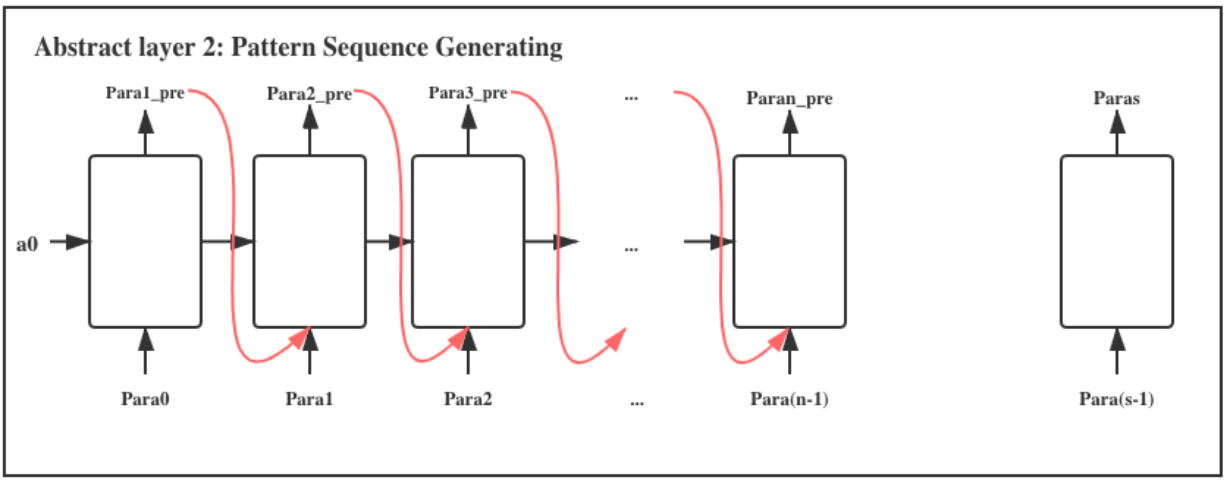

Figure 3: The second layer of the model: pattern sequence generating

(a) Layer 1
In layer 1, there will be N_s RNNs trained for N_s sections, each of which takes only one training example. The object of better capturing the mapping relationship between the music data and the motion data could drive us to chose computing units with more complex structures and, as the result, larger sets of parameters. However, as there will be $(N_s * m)$ RNNs being trained for m training examples, sophisticated computing units can bring enormous computing cost in abstract layer 1. In addition, as the sets of parameters in layer 1 RNNs will be fed into layer 2 RNN, a large size of input and output can also lead to problems like computationally expensive and unsatisfactory accuracy. Thus, the choice of computing unit is supposed to be based on the trade-off between them.

There are four proposed computing unit options for abstract layer 1, from which the best one could be picked according to the actual performances in the experiments.

- Factor Conditional Restricted Boltzmann Machines.[28][1]
- Casual Dilated Highway Conv. Block for encoding and decoding. [18]
- Long Short Term Memory unit + softmax. [13]
- Simplified Gated Recurrent Unit + softmax. [7]

(b) Layer 2
Similar to nature language modelling and music generating, the aim of abstract layer 2 is to generate a sequence with a given initial state. Thus, refer to the published approaches in jazz music generation[17][11], the combination of Long Short Term Memory unit and softmax is proposed for layer 2[13].

3.2.3 Inference

After training, the model should be able to generate a novel dance motion sequence fits with the given music in real time. As illustrated in Figure 4, firstly, the beginning part of the music will be read as the pre-listening stage and be processed to calculate the initial activation state a_0 which

will be provided to abstract layer 2. Subsequently, the RNN in layer 2 will execute one time step to generate the first set of parameters $\hat{P}_1$(noted as P1_pre in the figure), which is going to be used in abstract layer 1 as the parameters for prediction. Until now, the abstract layer 1 is fully equipped to take the next T_s audio input vectors one by one and output the corresponding motion vectors. Finished the first T_s time steps($Section_1$), the parameter set used in abstract layer 1 is supposed to be updated. Abstract layer 2 will then execute the next time step to produce $\hat{P}_2$ to update abstract layer 1, after which abstract layer 1 will be able to move T_s more steps($Section_2$). Moving forward like so, the model can synthesize dance motion simultaneously while reading the audio.

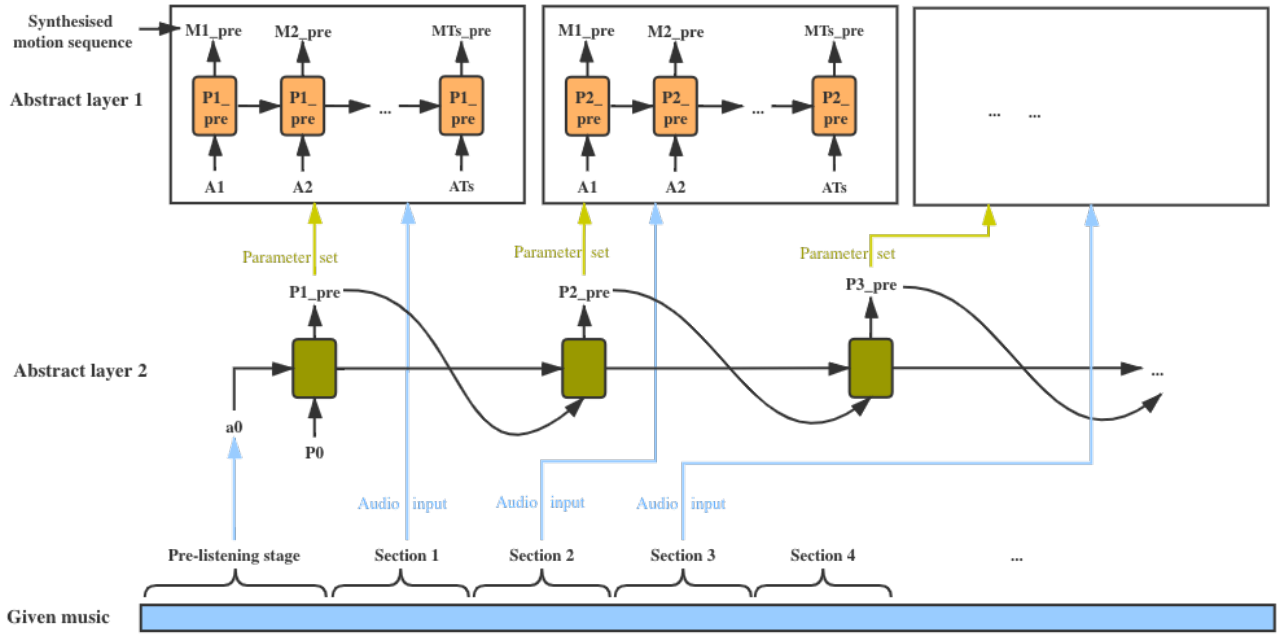

Figure 4: Synthesis the freestyle dance motion sequence with the given music

3.3. Choreographer model

Similar to a human choreographer, the choreographer model's job is reading and analysis the whole piece of music and output a complete dance choreography art work. This process is not in real time so it provides the model with the scope of the whole audio input before starting to generate the dance motion sequence.

3.3.1 Model intuition

For a choreography task, a human street dance choreographer would usually subtly analyze the characteristics and structure of the music. Subsequently, for every step of the generating process, the decision of the dance move will be made based on the information from the whole piece of music, not only the preceding part. As a result, the choreography can be more integrated and better-designed.

3.3.2 Learning

1. Overview of the mechanics
   Based on the intuition, the sequence-to-sequence Long Short Term Memory(LSTM) Recurrent Neural Network(RNN) with attention is proposed, which has been used in the task of machine translation[4][31].

   As shown in Figure 5, the model consists of three levels: pre-attention, context and post-attention. The pre-attention level plays a similar role as the process that a human choreographer analyzes the complete piece of music, which is implemented with a bidirectional RNN with LSTM. By reading the audio both forward and backward, RNN computes the activation state a_i for each time step i, which consists of the forward activation state $a_i^{\rightarrow}$ and the backward activation state $a_i^{\leftarrow}$(noted as ai_forward and ai_back in the figure). The activation state values will then be used to compute the context values in context level, with a set of parameter vectors $[\alpha_1, \alpha_2, ..., \alpha_T]$, each element in which is of the size T. The details of the computation will be explained in the next subsection. Moving forward to the post-attention level, the context values are fed into a typical RNN with LSTM, followed by a softmax layer to generate the final motion output. The cross-entropy loss will then be calculated with the ground truth motion frames and the predictions $[\hat{M}_1, \hat{M}_2, ..., \hat{M}_T]$(noted as M1_pre etc. in the figure).

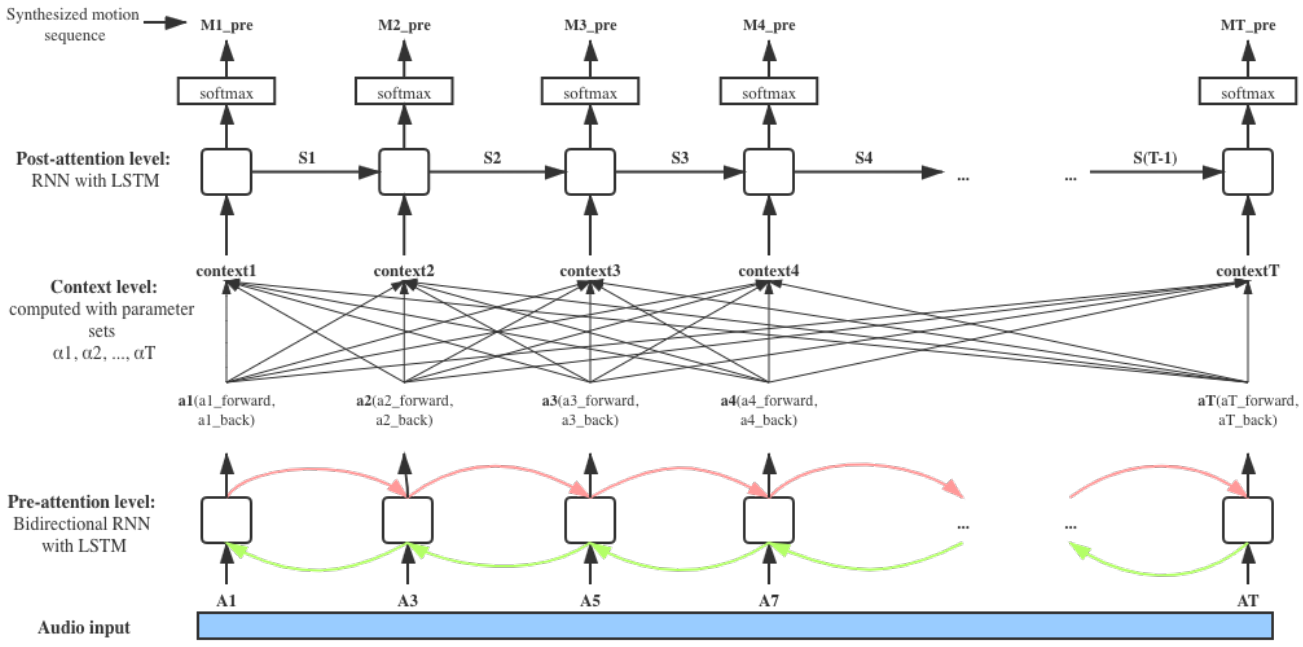

Figure 5: Sequence-to-sequence LSTM RNN model with attention

2. Computing details
   The context level plays a core role in the LSTM-RNN-with-attention model as it captures the "attentions". The calculations behind this level will be explained in detail in this subsection.

   We note $Context_t$ as an example context value at time step t in the T context values and

α_t is its corresponding parameter vector, with the elements $[\alpha_{t_1}, \alpha_{t_2}, ..., \alpha_{t_T}]$. Intuitively, α_{t_i} measures the amount of attention the model pays on a_i(the activation state value of time step i in the pre-attention RNN) to generate $Context_t$. Thus, the calculation of α_{t_i} should involve both of the states of a_i and $Context_t$. As $Context_t$ itself has not been computed yet, the preceding activation state s_{t-1} in the post-attention level RNN has been chosen to represent the state of $Context_t$. α_{t_i} is then computed by concatenating a_i with s_{t-1} and feeding them to a fully-connected neural network layer and a softmax layer.

$$\alpha_{t_i} = Softmax(Dense(Concatenate(a_i, s_{t-1}))) \tag{1}$$

Computed all elements in α_t, $Context_t$ can then be calculated as:

$$Context_t = \sum_{i=1}^{T} (\alpha_{t_i} * a_i) \tag{2}$$

3.3.3 Inference

As for prediction, the output dance motion sequence can be generated by the same process illustrated in Figure 5, with the trained parameters.

3.4. Evaluation

The evaluation of the model performance contains three parts: physical coordination, harmony with music and aesthetic quality.

Physical coordination mainly focus on the physical dance motion itself. The auto-correlation of the motion sequence is a good factor to reflect its properties in a periodic perspective.[18] Thus, a comparison of auto-correlations between the ground truth choreography and the produced choreography could be indicative.

As for the harmony with music, the correlation of the audio and motion sequences will be calculated for both the ground truth and the generated choreography and subsequently be compared in parallel.

The aesthetic quality can be measured with a user study session, by inviting a number of random people with diverse knowledge and background in music and dance to appraise the ground truth choreography and the generated choreography.

4 LIMITATIONS

In the proposed approach for freestyle dancer model, training $N_s * m$ RNNs in abstract layer 1 can be computationally expensive. Each RNN in this layer will be trained on only one dancing section example so its performance may suffer from lacking of training information. These problems require exploration and adjustment during the practical experiment

REFERENCES

[1] O. Alemi, J. Françoise, and P. Pasquier. “GrooveNet: Real-time music-driven dance movement generation using artificial neural networks”. In: *networks* 8.17 (2017), p. 26.

[2] O. Alemi, W. Li, and P. Pasquier. “Affect-expressive movement generation with factored conditional restricted boltzmann machines”. In: *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE. 2015, pp. 442–448.

[3] U. Bagcı and E. Erzin. “Automatic classification of musical genres using inter-genre similarity”. In: *IEEE Signal Processing Letters* 14.8 (2007), p. 521.

[4] D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[5] B. Bläsing et al. “Neurocognitive control in dance perception and performance”. In: *Acta psychologica* 139.2 (2012), pp. 300–308.

[6] B. Caramiaux et al. “Gestural Embodiment of Environmental Sounds: an Experimental Study.” In: *NIME*. Vol. 11. Citeseer. 2011.

[7] J. Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).

[8] L. Crnkovic-Friis and L. Crnkovic-Friis. “Generative choreography using deep learning”. In: *arXiv preprint arXiv:1605.06921* (2016).

[9] J. Engel et al. “Neural audio synthesis of musical notes with wavenet autoencoders”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1068–1077.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. “Image style transfer using convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[11] J. Gillick, K. Tang, and R. M. Keller. “Learning jazz grammars”. In: *Proceedings of the sound and music computing conference*. 2009, pp. 125–130.

[12] F. S. Grassia. “Practical parameterization of rotations using the exponential map”. In: *Journal of graphics tools* 3.3 (1998), pp. 29–48.

[13] S. Hochreiter and J. Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[14] S. D. Holcomb et al. “Overview on deepmind and its alphago zero ai”. In: *Proceedings of the 2018 international conference on big data and education*. 2018, pp. 67–71.

[15] A. Huang and R. Wu. “Deep learning for music”. In: *arXiv preprint arXiv:1606.04930* (2016).

[16] V. Kalingeri and S. Grandhe. “Music generation with deep learning”. In: *arXiv preprint arXiv:1612.04928* (2016).

[17] R. M. Keller and D. R. Morrison. “A grammatical approach to automatic improvisation”. In: *Proceedings, Fourth Sound and Music Conference, Lefkada, Greece, July.“Most of the soloists at Birdland had to wait for Parker’s next record in order to find out what to play next. What will they do now*. 2007.

[18] J. Lee, S. Kim, and K. Lee. “Automatic Choreography Generation with Convolutional Encoder-decoder Network.” In: *ISMIR*. 2019, pp. 894–899.

[19] M. Lee, K. Lee, and J. Park. “Music similarity-based approach to generating dance motion sequence”. In: *Multimedia tools and applications* 62.3 (2013), pp. 895–912.

[20] O. Lemon et al. “A multi-modal dialogue system for human-robot conversation”. In: *Proceedings of North American Association for Computational Linguistics (NAACL 2001)*. 2001.

[21] Y. Li, T. Wang, and H.-Y. Shum. “Motion texture: a two-level statistical model for character motion synthesis”. In: *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 2002, pp. 465–472.

[22] A. I. Maqueda et al. “Event-based vision meets deep learning on steering prediction for self-driving cars”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5419–5427.

[23] B. D. Miguel Alonso and G. Richard. “Tempo and beat estimation of musical signals”. In: *Proceedings of the International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain*. 2004.

[24] K. Nymoen et al. “Analyzing sound tracings: a multimodal approach to music information retrieval”. In: *Proceedings of the 1st international ACM workshop on Music information retrieval with user-centered and multimodal strategies*. 2011, pp. 39–44.

[25] F. Ofli et al. “An audio-driven dancing avatar”. In: *Journal on Multimodal User Interfaces* 2.2 (2008), pp. 93–103.

[26] F. Ofli et al. “Correction to" Learn2Dance: Learning Statistical Music-to-Dance Mappings for Choreography Synthesis"[Jun 12 747-759]”. In: *IEEE Transactions on Multimedia* 14.4 (2012), pp. 1376–1376.

[27] S. Ramos et al. “Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2017, pp. 1025–1032.

[28] G. W. Taylor and G. E. Hinton. “Factored conditional restricted Boltzmann machines for modeling motion style”. In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 1025–1032.

[29] J. Tilmanne and T. Dutoit. “Expressive gait synthesis using PCA and Gaussian modeling”. In: *International Conference on Motion in Games*. Springer. 2010, pp. 363–374.

[30] Y. Wang, Z.-Q. Liu, and L.-Z. Zhou. “Learning hierarchical non-parametric hidden markov model of human motion”. In: *2005 International Conference on Machine Learning and Cybernetics*. Vol. 6. IEEE. 2005, pp. 3315–3320.

[31] K. Xu et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *International conference on machine learning*. 2015, pp. 2048–2057.