Automatic Translation of Music-to-Dance for In-Game Characters

Yinglin Duan[1*], **Tianyang Shi**[1*], **Zhipeng Hu**[1,2], **Zhengxia Zou**[3], **Changjie Fan**[1], **Yi Yuan**[1†], **Xi Li**[2]
[1]NetEase Fuxi AI Lab,
[2]Zhejiang University,
[3]University of Michigan, Ann Arbor
{duanyinglin, shitianyang, huzhipeng}@corp.netease.com, zzhengxi@umich.edu,
{fanchangjie, yuanyi}@corp.netease.com, xilizju@zju.edu.cn

Abstract

Music-to-dance translation is an emerging and powerful feature in recent role-playing games. Previous works of this topic consider music-to-dance as a supervised motion generation problem based on time-series data. However, these methods require a large amount of training data pairs and may suffer from the degradation of movements. This paper provides a new solution to this task where we re-formulate the translation as a piece-wise dance phrase retrieval problem based on the choreography theory. With such a design, players are allowed to optionally edit the dance movements on top of our generation while other regression-based methods ignore such user interactivity. Considering that the dance motion capture is expensive that requires the assistance of professional dancers, we train our method under a semi-supervised learning fashion with a large unlabeled music dataset (20x than our labeled one) and also introduce self-supervised pre-training to improve the training stability and generalization performance. Experimental results suggest that our method not only generalizes well over various styles of music but also succeeds in choreography for game players. Our project including the large-scale dataset and supplemental materials is available at https://github.com/FuxiCV/music-to-dance.

1 Introduction

The music-dance is a very popular feature in many Role-Playing Games (RPGs), where players can control their characters to dance with the music (e.g. "Just Dance[1]" and "FINAL FANTASY XIV[2]"). Recent games like "Heaven mobile [3]" further enriched this feature, where various instruments and pre-defined dance movements are provided. Players can edit vivid music-dance videos and can even share them on social networks. However, the editing and customization of music and dance require a lot of expertise. For those players without experience in such area, choreography for game characters would be a very difficult task. Even for a very experienced team in music-dance, from the early capture of dance movements to the final software synthesis, it usually takes several days to complete the entire production. In this paper, we investigate an interesting problem called "Music-to-dance translation" which aims to automatically generate dance movements for game characters according to the player-uploaded music.

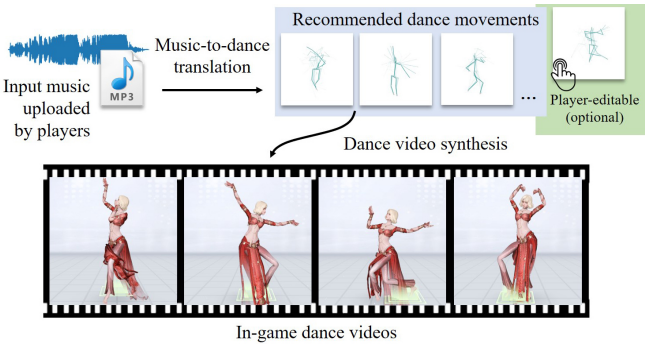

Figure 1: We propose a new method for in-game music-to-dance translation that can be applied in RPGs and generate high-quality and player-friendly dance videos according to the uploaded music.

Recently, music-to-dance translation has drawn increasing research attention due to its wide applications in the game industry and virtual reality. Deep learning based methods have shown great potential in this task [Alemi *et al.*, 2017; Tang *et al.*, 2018; Lee *et al.*, 2019; Ren *et al.*, 2020]. However, these methods are difficult to apply to in-game music-to-dance applications. The reason is threefold: 1. *Quality*. Previous regression-based models may suffer from a degradation of movements, which makes them difficult to generate Mocap-level dances and difficult to apply to recent RPGs. 2. *Generalization*. Players may upload various music, while previous methods trained under a fully supervised manner are difficult to generalize to unseen music styles. 3. *Interactivity*. Rich interactivity is crucial for RPGs. However, the dance movements generated by regression-based models are not editable, which seriously limits their interactivity.

[*]These authors contributed equally to this work.
[†]Corresponding author.
[1]https://www.ubisoft.com/en-us/game/just-dance-2020/
[2]https://www.finalfantasyxiv.com/
[3]http://tym.163.com/

To solve the above problems, we propose a novel method for generating high quality and player-friendly music-dances for RPGs. We symbolize the dance movements and re-formulate the music-to-dance translation as a phrase-wise dance phrase retrieval problem. Different from the dance generative models that directly generate the dance movements from the input music, we consider the dance movements as a set of semantic fragments, and then arrange these phrases according to the choreography theory [Humphrey, 1959]. To map music phrases to dance movements, we build an encoder-decoder network that takes in the Mel Spectrogram of a music phrase and then predicts the index-code of the dance phrase. As a temporal prediction problem, we introduce "transition priors" of the dance phrases based on a first-order Markov model to integrate context reasoning, where the transition matrices are used to re-scale the probability of predicted results and so that to produce a smoother and more consistent generation result.

Considering the high cost of building large-scale dance movements datasets, we take advantages of the semi-supervised learning [Joachims, 1999; Bengio *et al.*, 2006; Oliver *et al.*, 2018], to improve the robustness and generalization ability of our method [Tarvainen and Valpola, 2017; Pereyra *et al.*, 2017]. We extend our method on a large unlabeled music dataset (20x larger than our labeled one). We first train our method on this unlabeled dataset with a set of self-supervised pretext tasks and enforce the network to reconstruct the music phrases as well as their melody and rhythm from the latent representations. The model can be thus pre-trained to learn a good representation of the music phrases from the pretext tasks we designed without human annotations. After the pre-training, we fine-tune the model on a labeled dataset. Since the transition matrices initially learned on the labeled data are half-baked, we propose a co-ascent mechanism to jointly refine the transition priors of movements and improve the accuracy of the prediction.

With the help of semi-supervised learning, our method can better generalize to in-the-wild music data. Such scalability is not considered and supported in previous methods.

Our contributions are summarized as follows:

- We propose a novel music-to-dance translation method that can be applied in game environments with high-quality, good-generalization and rich interactivity. Players can thus optionally and easily edit the dance moves on top of our translation results. Such interactivity was rarely considered in previous methods.
- We symbolize the dance movements based on choreography theory [Humphrey, 1959] and re-formulate the music-to-dance translation as a phrase-wise music-to-dance retrieval problem that prevents the motion degradation problem of previous methods.
- We extend our method to a large unlabeled music dataset and propose a self-supervised pre-training method that can greatly improve the accuracy of the downstream music-to-dance translation task. A co-ascent boosting method is also designed to further improve the accuracy.

2 Related Works

Music-to-dance translation, as a cross-modality generation problem, is an emerging research topic in recent years. Early methods of this field are mostly based on statistical models [Shiratori *et al.*, 2006; Ofli *et al.*, 2011; Lee *et al.*, 2013], while most recent methods are based on supervised deep learning models [Tang *et al.*, 2018; Lee *et al.*, 2019; Ren *et al.*, 2020] and show impressive results. The GrooveNet proposed by Alemi *et al*. is the first method that achieves real-time music-driven dance movements generation [Alemi *et al.*, 2017]. In this method, the Factored Conditional Restricted Boltzmann Machine (FCRBM) is used under a recurrent movements prediction framework that considers both current music features and historical states. Tang *et al*. further propose an LSTM based Auto-Encoder model named "Anidance" to predict motions from acoustic features [Tang *et al.*, 2018]. Lee *et al*. propose a decomposition-to-composition framework [Lee *et al.*, 2019], that uses a VAE and GAN model to represent and organize the dance units based on input music. Ren *et al*. use adversarial training to generate coherent dance sequences and then use pose-to-appearance mapping to generate human dance videos [Ren *et al.*, 2020]. Recently, transformer-based methods draw a lot of attentions [Huang *et al.*, 2020; Li *et al.*, 2021], e.g., Huang *et al*. introduce curriculum learning during transformer training for long-term dance generation [Huang *et al.*, 2020]. However, all the above methods directly generate the dance movements from music, which inevitably leads to a problem of motion degradation. In this paper, different from previous methods, we symbolize the dance movements and re-formulate the music-to-dance generation as a retrieval problem to avoid such a problem. The players can therefore obtain high-quality dance movements arranged by their input music and at the same time, the interactivity can be also preserved.

3 Methodology

We frame the music-to-dance translation as a retrieval problem. Our method consists of a music feature encoder, a dance phrase predictor, and several decoders. Fig. 2 shows an overview of our method. The encoder is a ResNet50-based [He *et al.*, 2016] convolutional network which is trained to encode the Mel Spectrogram of music phrases into music embeddings. The predictor is an attention-based fully connected network which takes in the embeddings and predicts the index-code of dance phrases. The decoders are designed for the pre-training task and will not be used during the inference stage.

3.1 Music Phrase Segmentation

Given a piece of music (e.g., a pop song), we first segment the music into several phrases. In music theory, music phrase is defined as a separate musical entity within the melodic line [Knösche *et al.*, 2005]. We thus define a music phrase as our basic processing unit in our retrieval model.

We design the following three steps for segmenting music phrases from a piece of music (Please refer to our *supp.* for the detailed algorithm):

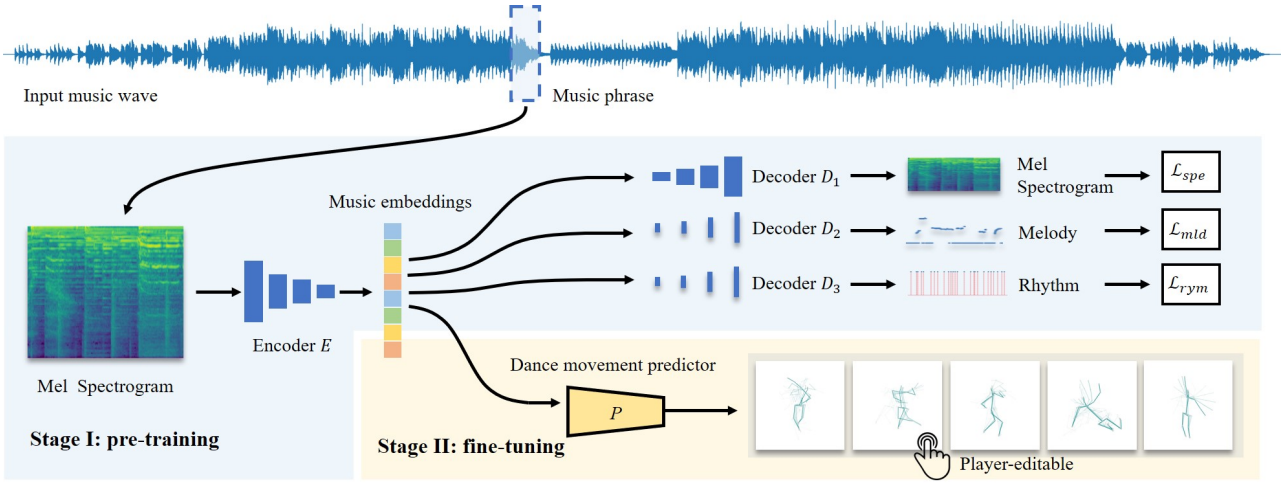

Figure 2: An overview of our method. Our method consists of a music encoder E and a dance phrase predictor P. We also introduce three decoders for self-supervised pre-training. In the pre-training stage, we train our encoder on a large unlabeled music dataset with three pretext losses - a spectrogram reconstruction loss $\mathcal{L}_{spe}$, a melody prediction loss $\mathcal{L}_{mld}$, and a rhythm prediction loss $\mathcal{L}_{rym}$. In the fine-tuning stage, we train the predictor P on a labeled dance-music dataset so that to translate the input music phrases to dance phrases. Finally, players can optionally replace the predicted phrases by preferred ones chosen from the dance library.

Step 1: Coarse segmentation. We analyze the music structure by using spectral clustering and segment music into long fragments [McFee *et al.*, 2015].

Step 2: Rhythm detection. We detect beats by librosa and extract main-melody by a deep learning method [Hsieh *et al.*, 2019].

Step 3: Phrase detection: For each phrase, we start from the end of the last one, merge at least 6 detected beats, and end at a breaking point of a long fragment (from step 1) or a music rest (from step 2) [Jehan, 2005].

3.2 Self-Supervised Pre-Training

The training of our method consists of two stages. In the first stage, we pre-train the encoder on a large unlabeled dataset (music without dance movements) with self-supervised pretext losses. In the second stage, we fix the encoder and fine-tune the predictor on a labeled dataset (music phrases and corresponding dance movements).

Considering that choreography requires the concordance of music-dance on rhythm and melody, we design three pretext tasks for the pre-training - a *spectrogram reconstruction* tasks, a *melody prediction* task, and a *rhythm prediction* task. The pre-training is performed solely on the music data without any human annotations.

Spectrogram Reconstruction We compute the Mel Spectrogram for an input music phrase and convert the 1d music signal to a 2D "image" by using librosa [McFee *et al.*, 2015]. We then feed the spectrogram to our ResNet encoder E to produce a set of low dimensional feature embeddings. Because we expect the embeddings containing all information of the input music phrase, we introduce a decoder D_1, to upsample the features and restore the spectrogram. We force the Mel Spectrogram before the encoder and after the decoder unchanged. We define the reconstruction loss as follows:

$$\mathcal{L}_{spe}(E, D_1) = \|D_1(E(\text{Mel}(x))) - \text{Mel}(x)\|_1, \quad (1)$$

where x is the music phrase and $\text{Mel}(x)$ is its Mel Spectrogram. The decoder D_1 has a similar structure as the generative network DCGAN [Radford *et al.*, 2015], with 8 transposed 2D-convolution layers.

Melody Prediction Main-melody defines the pitch contours of polyphonic music. Different from the previous method [Tang *et al.*, 2018] that uses vanilla melody, we use the Main-Melody extracted by deep learning method [Hsieh *et al.*, 2019] to improve the robustness. We define the prediction loss as follows:

$$\mathcal{L}_{mld}(E, D_2) = \|D_2(E(\text{Mel}(x))) - \text{Melody}(x)\|_1, \quad (2)$$

where D_2 is a decoder with 5 transposed 1D-convolution layers for regressing the melody from the embeddings. $\text{Melody}(x)$ is the pre-computed target melody from the music phrase x.

Rhythm We define another prediction head to predict the rhythm from the music embeddings. The prediction loss is defined as follows:

$$\mathcal{L}_{rym}(E, D_3) = BCELoss(D_3(E(\text{Mel}(x))), \text{Rythm}(x)) \quad (3)$$

where BCELoss denotes the Binary-Cross-Entropy-Loss, D_3 is a rhythm decoder which has a similar structure as D_2 but produces binary output, and $\text{Rythm}(x)$ is the target rhythm from the music phrase x, which is pre-computed based on librosa [McFee *et al.*, 2015] and main-melody.

Final Pre-Training Loss By combining the loss term (1), (2) and (3), we define the final pre-training loss as follows:

$$\begin{aligned}&\mathcal{L}_{pre-tr}(E, D_1, D_2, D_3)\\ =&\beta_1\mathcal{L}_{spe} + \beta_2\mathcal{L}_{mld} + \beta_3\mathcal{L}_{rym},\end{aligned} \quad (4)$$

where β_1, β_2, and β_3 are the weights to balance the loss terms. We train the encoder E and the decoders (D_1, D_2, D_3) to minimize the above loss function. After the pre-training, we remove the decoders and only keep the weights of the encoder for a further fine-tuning on music-dance data pairs.

3.3 Dance Phrase Prediction

We build an attention-based multilayer perceptron as our dance phrase predictor P. The P consists of three residual attention blocks and two Fully Connected (FC) layers. In each of the block, we make a simple modification of the squeeze and excitation block in SENet [Hu *et al.*, 2018] to apply it to an FC layer (the global pooling layer thus is removed).

The P is trained to predicts the index of a proper dance phrase. For each music phrase, we define the prediction loss as the cross-entropy loss between the predicted probability distribution and the K possible dance phrases captured in the dance library:

$$\mathcal{L}_{pred} = -\sum_{i=1}^{K} \hat{y}_p^{(i)} \log(F_{pred}(\boldsymbol{u})^{(i)}), \tag{5}$$

where $[\hat{y}_p^{(1)}, ..., \hat{y}_p^{(K)}]$ represent the one-hot ground truth vector of the prediction. $F_{pred}(\boldsymbol{u})^{(i)}$ represents the predicted probability for the ith kind of dance phrase. $\boldsymbol{u} = E(\text{Mel}(x))$ is the music embedding from the encoder E. We train the encoder and predictor from the self-supervised pre-trained initialization. During the training, we fix the encoder E and only update the predictor P for a faster convergence.

3.4 Co-Ascent Learning

Semi-supervised learning forms a challenging but important foundation of machine learning methods [Bengio *et al.*, 2006] that combines a small amount of labeled data with a large amount of unlabeled one during training to improve the prediction, and now it has been widely used in various tasks in the multimedia field [Song *et al.*, 2007; Poria *et al.*, 2013; Li *et al.*, 2019]. Considering that building a large scale dance phrase dataset is very expensive, we introduce the co-ascent learning mechanism to migrate our learning process to unlabeled data. This method also improves the prediction with context reasoning.

Transition Matrix Inspired by the N-gram [Brown *et al.*, 1992] that has been widely used in the field of Natural Language Processing, we introduce a dance phrase transition matrix $\mathbf{M} \in \mathbb{R}^{K \times K}$ to capture the probability transition between the two adjacent dance phrases. This matrix can be seen as having a similar meaning to the probability transition matrix in the first-order Markov process. During the inference stage, we use this matrix to re-scale the prediction results of the current phrase (based on the history predictions). The re-scale of the predicted class probability can be written as follows:

$$\begin{aligned} P(d_t|\boldsymbol{u}_t, d_{t-1}) &= P(d_t|\boldsymbol{u}_t)P(d_t|d_{t-1}) \\ &= F_{pred}(\boldsymbol{u}_t)\mathbf{M}(d_{t-1} \rightarrow d_t), \end{aligned} \tag{6}$$

where d_t is the dance phrase at the time step t, $P(d_t|\boldsymbol{u}_t, d_{t-1})$ is the re-scaling results, $F_{pred}(\boldsymbol{u}_t)$ is the raw prediction results of the prediction head F_{pred}, and $M(d_{t-1} \rightarrow d_t)$ is the transition probability between two dance phrases from the step $t-1$ to t.

Co-Ascent Learning Pseudo-labeling [Lee, 2013] is a simple but effective strategy that has been widely used in semi-supervised learning methods. In our method, we first train the networks on a small labeled dataset and then apply the weak model to all unlabeled data (music without dances) to predict the corresponding labels. The dataset with both true labels and pseudo labels is again used to train the network to enhance the decision boundary. During the pseudo-labeling process, we also apply the transition matrix $\mathbf{M}$ to correct the predictions of our network, and the corrected labels are further used to update the transition matrix. The update of the transfer matrix is performed based on the product of the confidences of two pseudo-labeled music phrases:

$$M_{k+1}(d_{t-1} \rightarrow d_t) = M_k(d_{t-1} \rightarrow d_t) + P(d_{t-1})P(d_t) \tag{7}$$

where M_{k+1} is the transition matrix after kth updates by using the pseudo-labels. $P(d_t)$ is the prediction confidence on the dance phrase at the time step t. Since the transition matrix and the networks can be mutually improved based on Eq. 6 and Eq. 7, we refer to this mechanism as co-ascent learning.

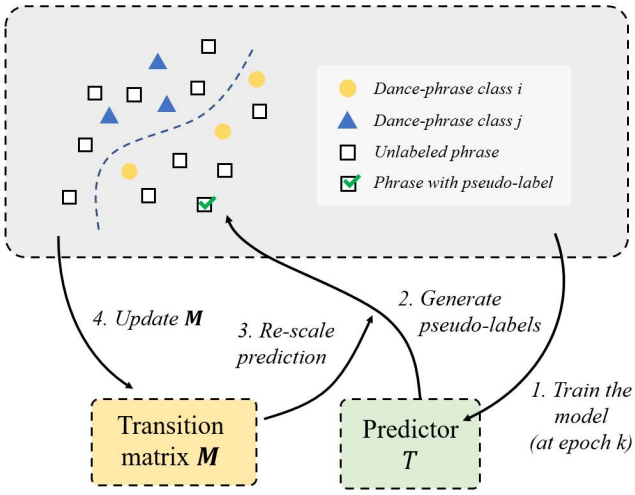

Figure 3: The pipeline of the proposed co-ascent learning. We train our predictor in a semi-supervised manner. A transition matrix is integrated to correct the pseudo-labels and is jointly updated with the predictor during training.

3.5 Implementation Details

Training Details In our method, we adopt Mel Spectrogram as the input music feature. We do not use Mel-frequency cepstral coefficients (MFCCs) since the Mel Spectrogram contains more original music information, and we aim to learn a better representation of music to replace manual features (i.e. MFCCs [Logan and others, 2000]). The input Mel Spectrogram is resized to 128×128 before fed into the encoder E, the melody and rhythm are also resized to 1×128. The dimension of music embeddings produced by the encoder is set to 512. For a detailed network configuration and the co-ascent learning pipeline, Please refer to our *supp.*

In the pre-training stage, we use Adam optimizer [Kingma and Ba, 2014] to train our model with the learning rate of 10^{-4} and stop at 200 epochs. The learning rate decay is set to 0.1 per 50 epochs. We set the loss coefficient $\beta_1 = \beta_2 = 1$ and $\beta_3 = 10$. In the supervised fine-tuning stage, we train our translator by SGD with the learning rate of 10^{-2}, momentum 0.9, weight decay 5×10^{-4} and the max-epoch number of

Group	Ablations				Index		
	Self-Supervised	Attention	Balance	Co-Ascent	Top1	Top5	Top10
I	×	×	×	×	12.3%	20.5%	23.6%
II	✓	×	×	×	14.5%	19.3%	22.3%
III	✓	✓	×	×	19.1%	23.7%	25.5%
IV	✓	✓	✓	×	19.3%	**25.0%**	**27.2%**
V	✓	✓	✓	✓	**19.8%**	24.8%	26.8%

Table 1: Results of our controlled experiment (A higher score indicates better performance).

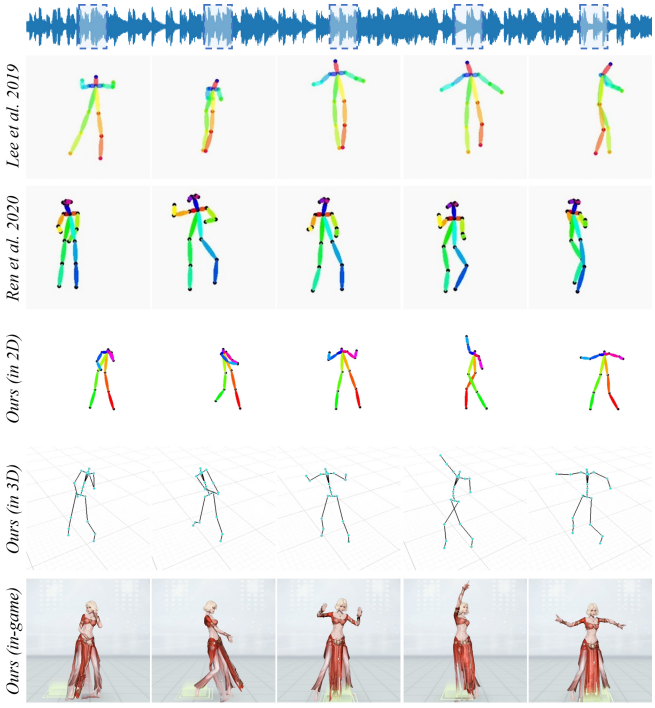

Figure 4: Comparisons between our method (last 3 rows) and previous methods (first 2 rows) on the music "Sorry". Our method can generate choreography directly for the in-game application (the last row).

500. In the co-ascent stage, we set the learning rate to 10^{-5}, update pseudo labels every 5 epochs, initialize the transition matrix $\mathbf{M}$ based on the style of dance phrases (i.e. the similar dance moves are allowed to transfer) and further clip the range of $\mathbf{M}$ within $[0.01, 1]$ to improve stability. Other configurations are kept the same as our fine-tuning stage.

Blending of Dance Phrases Considering that the dance moves in adjacent phrases are not always able to transit smoothly [Harvey *et al.*, 2020], we use a common technique called blending[4] to smooth the movements between two dance phrases.

[4]https://docs.unity3d.com/Packages/com.unity.timeline@1.6/manual/clp_blend.html

4 Experiments

4.1 Dataset and Experimental Setup

We test our method on the music-dance creation platform of a role-playing game named "Heaven mobile" and also generate both 2D and 3D animation for experiments. We built two datasets for our task.

Labeled Dance-Music Dataset In this dataset, we first recorded 1,101 different dance phrases (∼2.3 hours) by using motion capturing devices (Vicon V16 cameras). Five professional dancers took part in the motion capture for one month. We then collected about 600 songs (∼33 hours) with different genres that are suitable for choreography. We segment these songs into about 16773 music phrases and invite six experts to arrange the dance phrases for each of the music phrases (multiple music phrases may correspond to the same kind of dance phrases). For performance evaluation, we split this dataset into a training set (90 %) and a test set (10 %).

Unlabeled Music Dataset In addition to the labeled dataset, we also collected an unlabeled dataset which is 20x larger than the labeled one. The dataset consists of about 10k songs in various styles (∼686 hours). We segment each song of this dataset into music phrases and finally 293,579 music phrases are extracted and orderly packaged.

Evaluation Metrics We evaluate the performance of different music-to-dance translation models qualitatively (subjective evaluation by the expert jury). We also adopt quantitative indicators proposed by Lee *et al.* [Lee *et al.*, 2019] to evaluate the rhythm of the generated dance, such as beat coverage, beat hit rate, etc.

4.2 Music-to-Dance Translation Results

Fig. 4 shows a group of translation results by using our method and previous state-of-the-art methods on the music "Sorry" (also used in the previous work [Ren *et al.*, 2020]). The music-dance video generated by our method not only accurately captures the rhythm in the song, but also contains rich musical feelings and movement strength. Besides, since our method generates the entire dance phrase-by-phrase, players can easily replace any dance phrase by selecting the preferred one from the dance library, while the previous methods lack such interactivity in the game environment.

4.3 Controlled Experiment

Controlled experiments are conducted to verify the importance of each component in our network. We evaluate five configurations of our method, including:

Method	Group1 Rating	Group2 Rating	Group3 Rating	**Overall Rating**
Dancing to music [Lee *et al.*, 2019]	2.11 ± 0.68	2.14 ± 0.59	2.23 ± 0.79	2.16 ± 0.69
Dance Video Synthesis [Ren *et al.*, 2020]	3.41 ± 0.54	3.39 ± 0.49	3.36 ± 0.48	3.39 ± 0.50
Ours	$\mathbf{4.02 \pm 0.75}$	$\mathbf{4.09 \pm 0.63}$	$\mathbf{4.00 \pm 0.60}$	$\mathbf{4.04 \pm 0.67}$

Table 2: The experimental results of the subjective evaluation (A higher score indicates better performance).

Method	Beat Coverage	Beat Hit Rate	Music Hit Rate	Beat Overlap
Dancing to music* [Lee *et al.*, 2019]	39.4%	65.1%	25.6%	0.226
Dance Revolution*† [Huang *et al.*, 2020]	21.8%	68.4%	14.9%	0.140
Dancing to music [Lee *et al.*, 2019]	78.7%	56.5%	44.5%	0.315
Dance Video Synthesis [Ren *et al.*, 2020]	**94.8%**	55.5%	52.6%	0.365
Ours**	87.4%	**63.2%**	**55.2%**	**0.417**

* *Beat Coverage* and *Beat Hit Rate* in these rows are reported by the original papers [Lee *et al.*, 2019; Huang *et al.*, 2020].
† The pre-trained model of this work has not yet been released, thus we report the original results only for reference.
** Our dance beat detection algorithm is designed to track both strong and weak dance beats for better evaluation.

Table 3: The experimental results of the quantitative evaluation. (A higher score indicates better performance)

Group I: A ResNet-50 encoder is only adopted and is initialized by ImageNet pre-trained weights.

Group II: A ResNet-50 encoder is adopted and is initialized with the proposed self-supervised training method.

Group III: We fix the encoder trained by self-supervised losses and fine-tuning the attention-based predictor on the labeled dataset.

Group IV: We further balance the labeled dataset on top of Group III to ensure that each type of dance movement has an equivalent sampling frequency during the training.

Group V: We apply co-ascent learning on top of Group IV.

The results are listed in Table 1. We can see that our full implementation (Group V) achieves significant improvement than baselines, the self-supervised learning (Group III) brings a noticeable improvement on our results (+6.8% on top1 than Group I), but using self-supervised pre-trained weights may lead to an overfitting problem (+2.2% on top1 than Group I). Besides, the co-ascent learning also shows improvements on top1 (+0.5%) - although the scores are somewhat incremental, we find that co-ascent learning provides prediction results with a much more consistent style.

4.4 Subjective Evaluation

Since the predictor faces to a 1000-classification problem and the choreography can be very flexible, dance phrases can often be exchangeable. In other words, a higher retrieval accuracy in this task may not necessarily indicate better performance (even may indicate overfitting on the proxy task).

To better evaluate the quality of the generated dance phrases, subjective evaluations are further conducted. In this experiment, we first collect three groups of music: 1) music used in the previous method [Ren *et al.*, 2020], 2) music from our unlabeled test set, 3) unseen style music outside of our dataset (e.g. popular songs on youtube). Note that all these musics are not shown in our training dataset. Then we compare the dance videos generated by our full-implemented method with two previous state of the art methods [Ren *et al.*, 2020; Lee *et al.*, 2019].

For each group of the result, we invite nine certified dance teachers (with more than 10 years of dancing experience) and twelve professional dancers (with $5 \sim 10$ years of dancing experience) to evaluate the results of the three methods, where 5 points represent the senior dancer level while 1 point is at the beginner level. The result videos are randomly segmented into a set of 30s clips. As shown in the first three rows of Fig.4, we only show 2D animations of all three methods to the expert jury for a fairness evaluation. The statistics of the rating for different video groups are listed in Table 2. The experts agree our method generates high-quality dance movements in terms of both fluency and strength of the movements.

4.5 Quantitative Evaluation

We also quantitatively evaluate our method by using indicators recommended by Lee *et al*. [Lee *et al.*, 2019], where the beat coverage and beat hit rate are evaluated. To calculate the above indicators, for each music-dance animation, we count its dance beat number (N_d), music beat number (N_m) and beat hit number (N_h) respectively. Then, the *Beat Coverage* and the *Beat Hit Rate* are defined as $\frac{N_d}{N_m}$ and $\frac{N_h}{N_d}$. We further derive extra two indicators for a better evaluation, which are *Music Hit Rate* ($\frac{N_h}{N_m}$, similar to Recall) and *Beat Overlap* ($\frac{N_h}{N_m+N_d-N_h}$, similar to IOU).

Similar to Lee *et al*. 's work [Lee *et al.*, 2019], we implement a self-adaptive dance beat detection algorithm across different comparison methods on 2D animations mentioned in Sec 4.4 (Please refer to our *supp.* for more details). The evaluation results are shown in Table 3. Although our method is retrieval-based, we can still accurately assign dance phrases for music phrases with the highest rhythmic consistency. This is mainly owing to the good representation of rhythm learned by our model in our self-supervised learning stage.

5 Limitation

Our method has two limitations. One limitation is that our method is difficult to generalize on very smooth music since the rhythm of this kind of music is difficult to be captured by our encoder. Another limitation is that games usually adopt the linear blending method to transit between dance movements, which may cause model clipping on the large movement changes. We will focus on these problems in the future.

6 Conclusion

In this paper, we propose a novel method for automatic music-to-dance translation. We re-formulate the music-to-dance translation as a semi-supervised dance movement retrieval problem based on the choreography theory. We also build a new music-dance dataset which consists of over 16k music phrases labeled with dance movements and also 300k unlabeled ones. We design a self-supervised pre-training and a co-ascent learning pipeline to make full use of the unlabeled music dataset. Our experimental results in our dataset suggest that our methods can generate high-quality music-dances. The ablation studies also suggest the effectiveness of the core design in our method.

Acknowledgments

We would like to thank Mr. Wei Zhang, Mr. Yuntao Xu, Mrs. Ziyou Liu, Mr. Kaihua Yu from the development group of the game "Heaven mobile" for their excellent music-dance platform and kindly support. We would like to thank Mr. Tianyao Bai, Mr. Yangfan Xu, Mr. Han Yin, Mrs. Yuxia Wu, Mr. Ziguang She from the Ray-Force Sound Team of NetEase for their professional guidance in music. We would like to thank Mr. Rui Hu and Mr. Zilei Huang from NetEase Fuxi AI Lab for their great help. We would also like to thank Mr. Yenan Lin for his art-related help.

This work is supported in part by National Key Research and Development Program of China under Grant 2020AAA0107400, National Natural Science Foundation of China under Grant U20A20222, Zhejiang Provincial Natural Science Foundation of China under Grant LR19F020004, and key scientific technological innovation research project by Ministry of Education.

References

[Alemi *et al.*, 2017] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26, 2017.

[Bengio *et al.*, 2006] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. *Book Chapter in Semi-Supervised Learning*, 2006.

[Brown *et al.*, 1992] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[Harvey *et al.*, 2020] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hsieh *et al.*, 2019] Tsung-Han Hsieh, Li Su, and Yi-Hsuan Yang. A streamlined encoder/decoder architecture for melody extraction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 156–160. IEEE, 2019.

[Hu *et al.*, 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[Huang *et al.*, 2020] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. *arXiv preprint arXiv:2006.06119*, 2020.

[Humphrey, 1959] Doris Humphrey. *The art of making dances*. Dance Horizons, 1959.

[Jehan, 2005] Tristan Jehan. *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, 2005.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Icml*, volume 99, pages 200–209, 1999.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Knösche *et al.*, 2005] Thomas R Knösche, Christiane Neuhaus, Jens Haueisen, Kai Alter, Burkhard Maess, Otto W Witte, and Angela D Friederici. Perception of phrase structure in music. *Human Brain Mapping*, 24(4):259–273, 2005.

[Lee *et al.*, 2013] Minho Lee, Kyogu Lee, and Jaeheung Park. Music similarity-based approach to generating dance motion sequence. *Multimedia tools and applications*, 62(3):895–912, 2013.

[Lee *et al.*, 2019] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems*, pages 3586–3596, 2019.

[Lee, 2013] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 2, 2013.

[Li *et al.*, 2019] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, pages 10276–10286, 2019.

[Li *et al.*, 2021] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation. *arXiv preprint arXiv:2101.08779*, 2021.

[Logan and others, 2000] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *Ismir*, volume 270, pages 1–11, 2000.

[McFee *et al.*, 2015] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[Ofli *et al.*, 2011] Ferda Ofli, Engin Erzin, Yücel Yemez, and A Murat Tekalp. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE Transactions on Multimedia*, 14(3):747–759, 2011.

[Oliver *et al.*, 2018] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[Pereyra *et al.*, 2017] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

[Poria *et al.*, 2013] Soujanya Poria, Alexander Gelbukh, Amir Hussain, Sivaji Bandyopadhyay, and Newton Howard. Music genre classification: A semi-supervised approach. In *Mexican Conference on Pattern Recognition*, pages 254–263. Springer, 2013.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Ren *et al.*, 2020] Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Self-supervised dance video synthesis conditioned on music, 2020.

[Shiratori *et al.*, 2006] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, pages 449–458. Wiley Online Library, 2006.

[Song *et al.*, 2007] Yangqiu Song, Changshui Zhang, and Shiming Xiang. Semi-supervised music genre classification. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 2, pages II–729. IEEE, 2007.

[Tang *et al.*, 2018] Taoran Tang, Hanyang Mao, and Jia Jia. Anidance: Real-time dance motion synthesize to the song. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1237–1239, 2018.

[Tarvainen and Valpola, 2017] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.