Self-supervised Dance Video Synthesis Conditioned on Music

Xuanchi Ren
HKUST

Haoran Li
HKUST

Zijian Huang
HKUST

Qifeng Chen
HKUST

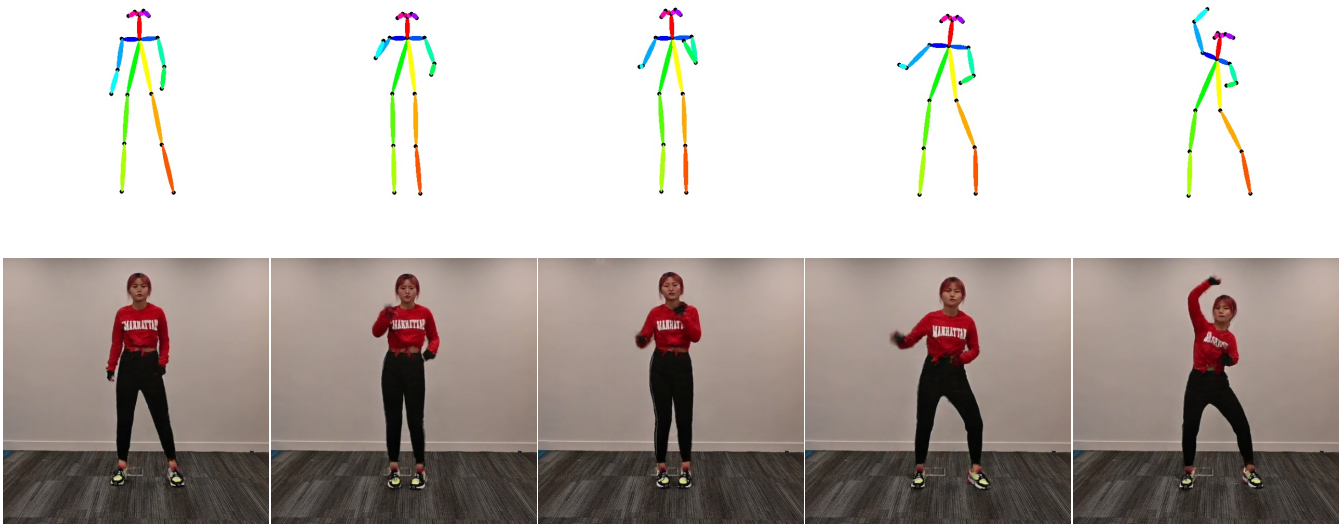


Figure 1: Our synthesized dance video conditioned on the song *"I Wish"*. We show 5 frames from a 5-second synthesized video. The top row shows the skeletons, and the bottom row shows the corresponding synthesized video frames. More results are shown in the supplementary video at https://youtu.be/UNHv7uOUExU.

ABSTRACT

We present a self-supervised approach with pose perceptual loss for automatic dance video generation. Our method can produce a realistic dance video that conforms to the beats and rhymes of given music. To achieve this, we firstly generate a human skeleton sequence from music and then apply the learned pose-to-appearance mapping to generate the final video. In the stage of generating skeleton sequences, we utilize two discriminators to capture different aspects of the sequence and propose a novel pose perceptual loss to produce natural dances. Besides, we also provide a new cross-modal evaluation to evaluate the dance quality, which is able to estimate the similarity between two modalities (music and dance). Finally, experimental qualitative and quantitative results demonstrate that our dance video synthesis approach produces realistic and diverse results. Source code and data are available at https://github.com/xrenaa/Music-Dance-Video-Synthesis.

CCS CONCEPTS

• **Applied computing** → **Media arts**; • **Human-centered computing** → *HCI theory, concepts and models*; • **Computing methodologies** → Neural networks.

KEYWORDS

Video synthesis, Generative adversarial network, Music, Dance, Cross-modal evaluation

1 INTRODUCTION

Dance videos have become unprecedentedly popular all over the world. Nearly all the top 10 most-viewed YouTube videos are music videos with dancing [14]. While generating a realistic dance sequence for a song is a challenging task even for professional choreographers, we believe an intelligent system can automatically generate personalized and creative dance videos. In this paper, we study automatic dance video generation conditioned on any music. We aim to synthesize a coherent and photo-realistic dance video that conforms to the given music. With such dance video generation technology, a user can share a personalized dance video on social media, such as TikTok. In Figure 1, we show sampled images of our synthesized dance video given the music *"I Wish"* by *Cosmic Girls*.

The dance video synthesis task is challenging for various technical reasons. First, the generated dance should be synchronized with the music and reflect the music's content, while the relationship between dance and music is hard to capture. Second, it is technically

challenging to model the dance movement, which has a long-term spatio-temporal structure. Third, a learning-based method is expected to require a large amount of training data of paired music and dance.

Researchers have proposed various ideas to tackle these challenges. A line of literature [2, 22, 38] treated dance synthesis as a retrieval problem, which limits the creativity of generated dance. To model the space of human body dance, Tang et al. [41] and Lee et al. [26] used L_1 or L_2 distance, which is demonstrated to disregard some specific motion characteristics by Martinez et al. [32]. For building a dance dataset, one option is to obtain a 3D dataset by using an expensive motion capture equipment with professional artists [41]. While this approach costs time and money, an alternative is to apply OpenPose [5, 6, 47] to get dance skeleton sequences from online videos and correct them frame by frame [25]. However, this method is still labor-intensive and thus not suitable for extensive applications.

To overcome such obstacles, we introduce a self-supervised system trained on online videos as input without additional human assistance. We propose a Global Content Discriminator with an attention mechanism to deal with cross-modal mapping and maintain global harmony between music and dance. A Local Temporal Discriminator is utilized to model the dance movement and focus on local coherence. Moreover, we introduce a novel pose perceptual loss that enables our model to train on noisy pose data generated by OpenPose.

To facilitate this dance synthesis task, we collect a dataset containing 100 online videos of representative categories: 40 in K-pop, 20 in Ballet, and 40 in Popping. To analyze the performance of our framework, we use various metrics to analyze realism, diversity, style consistency. Besides, we proposed a cross-modal evaluation to analyze the music-matching ability of our model. Both qualitative and quantitative results demonstrate that our framework can choreograph at a similar level with real artists.

The contributions of our work are summarized as follows.

- With the proposed pose perceptual loss, our model can be trained on a noisy dataset (without human labels) to synthesize realistic and diverse dance videos that conform to any given music.
- With the Local Temporal Discriminator and the Global Content Discriminator, our framework can generate a coherent dance skeleton sequence that matches the music rhythm and style.
- For our task, we build a dataset containing paired music and skeleton sequences, which will be made public for research. To evaluate our model, we propose a novel cross-modal evaluation that measures the similarity between music and a skeleton sequence.

2 RELATED WORK

Generative Adversarial Networks. A generative adversarial network (GAN) [16] is a popular approach for image generation. The images generated by GAN are usually sharper and with more details compared to those with L_1 and L_2 distance. Recently, GAN is also extended to video generation tasks [29, 33, 42, 43]. The GAN model in [43] replaced the standard 2D convolutional layer with a 3D convolutional layer to capture the temporal feature, although this method can only capture characteristics in a fixed period. This limitation is overcome by TGAN [36], but with the cost of constraints imposed in the latent space. MoCoGAN [42] could generate videos that combine the advantages of RNN-based GAN models and sliding window techniques so that the motion and content are disentangled in the latent space.

Another advantage of GAN based models is that it is widely applicable to many tasks, including the cross-modal audio-to-video problem. Chen et al. [9] proposed a GAN-based encoder-decoder architecture using CNNs to convert between audio spectrograms and frames. Furthermore, Vougioukas et al. [44] adapted temporal GAN to automatically synthesize a talking character conditioned on speech signals.

Dance Motion Synthesis. A line of work focuses on the mapping between acoustic and motion features. On the base of labeling music with joint positions and angles, Shiratori et al. [22, 38] incorporated gravity and beats as additional features for predicting dance motion. Alemi et al. [2] combined the acoustic feature with the motion features of previous frames. However, these approaches are entirely dependent on the prepared database and may only create rigid motion when it comes to music with similar acoustic features.

Recently, Yaota et al. [48] accomplished dance synthesis using standard deep learning models. Tang et al. [41] proposed a model based on LSTM-autoencoder architecture to generate dance pose sequences. Ahn et al. [1] firstly determined the genre of the music by a trained classifier and chose a pose generator for the determined genre. However, their results are not on par with those by real artists. Lee et al. [25] proposed a complex synthesis-by-analysis learning framework. Their model is trained on a manually pre-processed dataset, which is not easy for the large-scale extension to different dance styles.

3 OVERVIEW

To generate a dance video from music, we split our system into two stages. In the first stage, we propose an end-to-end model that directly generates a dance skeleton sequence according to the audio input. In the second stage, we translate the dance skeleton sequences to photo-realistic videos by applying the pix2pixHD GAN [45].

The pipeline of the first stage is shown in Figure 2. Let V be the number of joints of the human skeleton where each joint is represented by a 2D coordinate. We formulate a dance skeleton sequence $X \in R^{T\times 2V}$ as a sequence of human skeletons across T consecutive frames, where each skeleton frame $X_t \in R^{2V}$ is a vector containing all joint locations. Our goal is to learn a function $G : R^{TS} \rightarrow R^{T\times 2V}$ that maps audio signals with sample rate S per frame to a joint location vector sequence.

Dance Generator. The dance generator is composed of a music encoding part and a pose generator. The input audio signals are divided into pieces of 0.1-second music. These pieces are encoded using 1D convolution and then fed into a bi-directional 2-layer GRU in chronological order, resulting in output hidden states $O = \{H_1, H_2, \cdots, H_T\}$. These hidden states are fed in the pose generator, a multi-layer perceptron, to produce a skeleton sequence of X.

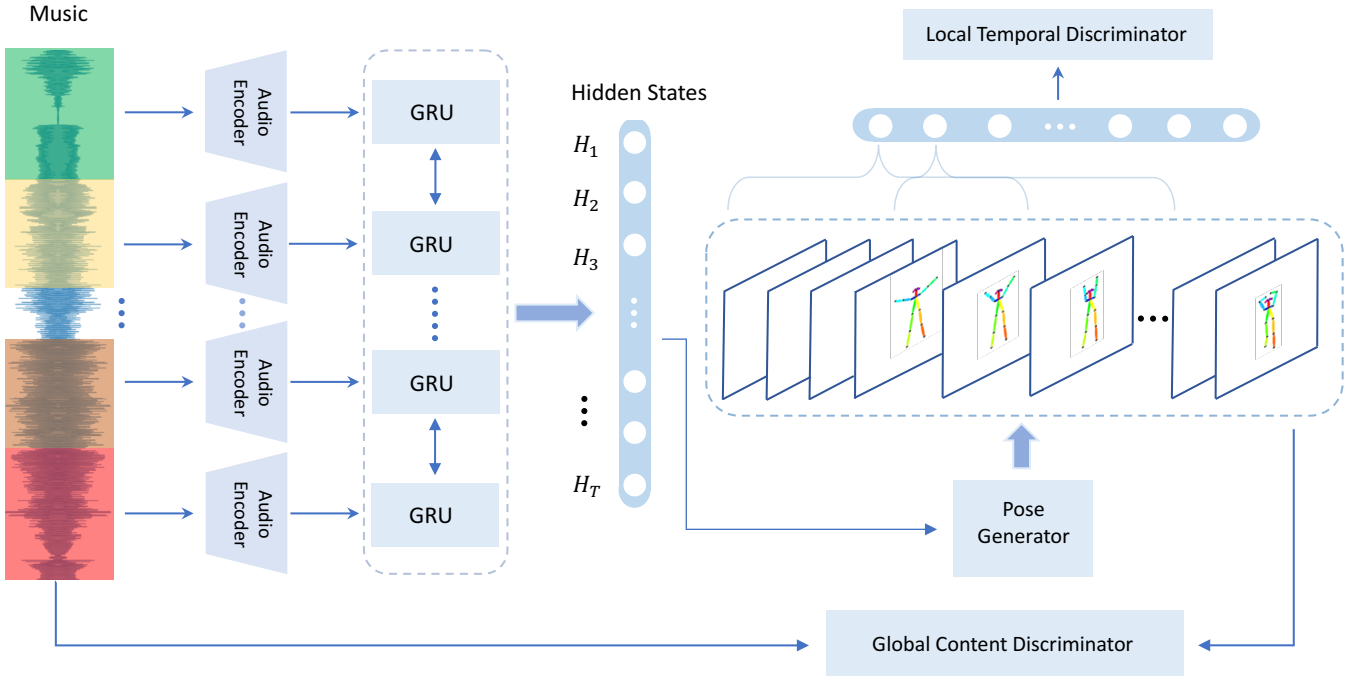


Figure 2: Our framework for music-oriented dance skeleton sequence synthesis. The input music signals are first divided into pieces of 0.1-second music. The generator in our model contains an audio encoder, a bidirectional GRU, and a pose generator. The output skeleton sequence of the generator is fed into the Global Content Discriminator to evaluate the consistency with the input music. The generated skeleton sequence is also divided into overlapping sub-sequences, which are fed into the Local Temporal Discriminator for local temporal consistency.

Local Temporal Discriminator. The output skeleton sequence X is divided into K overlapping sequences $\in R^{t\times 2V}$. Then these sub-sequences are fed into the Local Temporal Discriminator, which is a two-branch convolutional network. In the end, a small classifier outputs K scores that determine the realism of these skeleton sub-sequences.

Global Content Discriminator. The input to the Global Content Discriminator includes the music $M \in R^{TS}$ and the dance skeleton sequence X. For the pose part, the skeleton sequence X is encoded using pose discriminator as $F^P \in R^{256}$. For the music part, similar to the sub-network of the generator, music is encoded using 1D convolution and then fed into a bi-directional 2-layer GRU, resulting in an output $O^M = \{H_1^M, H_2^M, ..., H_T^M\}$ and O^M is transmitted into the self-attention component [30] to get a comprehensive music feature expression $F^M \in R^{256}$. In the end, we concatenate F^M and F^P along channels and use a small classifier, composed of a 1D convolutional layer and a fully-connected (FC) layer, to determine if the skeleton sequence matches the music.

Pose Perceptual Loss. Recently, graph convolutional networks (GCN) [28, 39, 49] have been extended to model skeletons since the human skeleton has a graph-based representation. Thus, the features extracted by GCN contains high-level spatial structural information about the human skeleton structure. Matching intermediate features in a pre-trained GCN network gives a better constraint on both detail and layout of a pose than the traditional metrics such as L_1 or L_2 distance. Figure 3 shows the pipeline of our pose perceptual loss. With the pose perceptual loss, our output skeleton sequence does not need post-processing for temporal smoothing.

4 POSE PERCEPTUAL LOSS

Perceptual loss or feature matching loss [3, 10, 15, 21, 34, 45, 46] has been used to measure the similarity between two images in image processing and synthesis. For the tasks that generate human skeleton sequences, prior work [4, 26, 41] only uses L_1 or L_2 distance for measuring pose similarity. However, L_1 or L_2 loss is not invariant in translation and scale. Moreover, the poses generated by OpenPose [5] are noisy, as shown in Figure 5. Correcting inaccurate human poses on a large number of videos is labor-intensive: a two-minute video with 10 FPS will have 1200 poses to verify and correct.

To tackle these difficulties, we propose a novel pose perceptual loss. We propose to directly match features in a pose recognition network that takes human skeleton sequences as input. We use ST-GCN [49] that is a Graph Convolutional Network (GCN) for a pose recognition to extract deep features. ST-GCN utilizes a spatial-temporal graph to form the hierarchical representation of skeleton sequences to learn both spatial and temporal patterns from data. As shown in Figure 4, our generator can stably generate poses with the pose perceptual loss.

Given a pre-trained GCN network Φ, we define a collection of layers Φ as $\{\Phi_l\}$. For a training pair (P, M) where P is the ground-truth skeleton sequence (from Openpose) and M is the corresponding

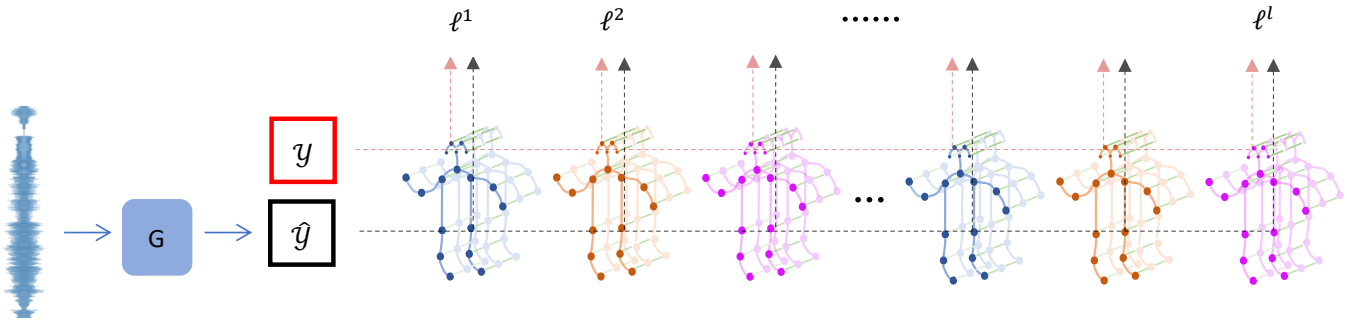


Figure 3: The overview of the pose perceptual loss based on ST-GCN. G is our generator in the first stage. y is the ground-truth skeleton sequence, and $\hat{y}$ is the generated skeleton sequence.

piece of music, the pose perceptual loss is

$$\mathcal{L}_P = \sum_l \lambda_l \|\Phi_l(P) - \Phi_l(G(M))\|_1. \quad (1)$$

Here G is the generator in the first stage of our framework. The hyperparameters $\{\lambda_l\}$ balance the contribution of each layer l to the loss.

5 IMPLEMENTATION

5.1 Pose Discriminator

To evaluate if a dance skeleton sequence is realistic, we believe the most indispensable factors include the intra-frame representation for joint co-occurrences and the inter-frame representation for skeleton temporal evolution. To extract features of a pose sequence, we explore multi-stream CNN-based methods and adopt the Hierarchical Co-occurrence Network framework [27] to enable discriminators to differentiate real and fake pose sequences.

Two-stream CNN. The input of the pose discriminator is a skeleton sequence X. The temporal difference is interpolated to be of the same shape of X. Then the skeleton sequence and the temporal difference are fed into the network directly as two input streams. Their feature maps are concatenated along channels, and then we use convolutional and fully-connected layers to extract features.

5.2 Local Temporal Discriminator

One of the objectives of the pose generator is to achieve temporal coherence of the generated skeleton sequence. For example, when a man moves his left foot, his right foot should keep still for multiples frames. Similar to PatchGAN [20, 46, 51], we propose a Local Temporal Discriminator to achieve coherence between consecutive frames. Besides, the Local Temporal Discriminator contains a trimmed pose discriminator and a small classifier composed of two fully-connected layers.

5.3 Global Content Discriminator

Dance is closely related to music, and the harmony between music and dance is a crucial criterion to evaluate a dance sequence. Inspired by [44], we proposed the Global Content Discriminator to deal with the relationship between music and dance.

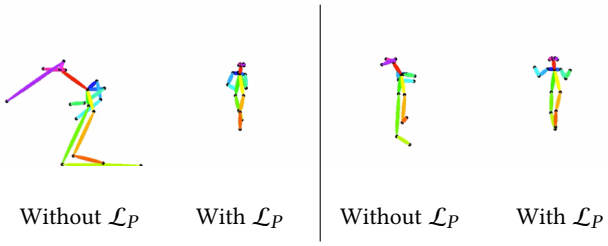


Figure 4: In each pair of images, the first image is a skeleton generated by the model without pose perceptual loss, and the second image is a skeleton generated by the model with pose perceptual loss according to the same piece of music.

As we mentioned previously, music is encoded as a sequence $O^M = \{H_1^M, H_2^M, ..., H_T^M\}$. Though GRU can capture long term dependencies, it is still challenging for GRU to encode the entire music information. In our experiment, only using H_T^M to represent music feature F^M will lead to a crash of the beginning part of the skeleton sequence. Therefore, we use the self-attention mechanism [30] to assign a weight for each hidden state and gain a comprehensive embedding. In the next part, we briefly describe the self-attention mechanism used in our framework.

Self-attention mechanism. Given $O^M \in R^{T\times k}$, we can compute its weight at each time step by

$$r = W_2 \tanh(W_1 O^{M^\top}), \quad (2)$$

$$a_i = -\log\left(\frac{\exp(r_i)}{\sum_j \exp(r_j)}\right), \quad (3)$$

where r_i is i-th element of the r, $W_1 \in R^{k\times l}$, and $W_2 \in R^{l\times 1}$. a_i is the assigned weight for i-th time step in the sequence of hidden states. Thus, the music feature F^M can be computed by multiplying the scores $A = [a_1, a_2, ..., a_n]$ and O^M, written as $F^M = AO^M$.

5.4 Other Loss Function

GAN loss $\mathcal{L}_{adv}$. The Local Temporal Discriminator (D_{local}) is trained on overlapping skeleton sequences that are sampled using $S(\cdot)$ from a whole skeleton sequence. The Global Content Discriminator (D_{global}) distinguishes the harmony between the skeleton sequence and the input music m. Besides, we have $x = G(m)$ and

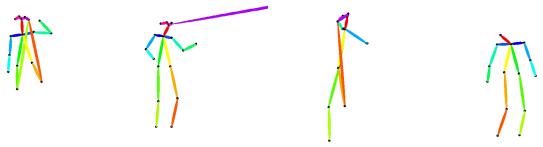


Figure 5: Noisy pose data caused by occlusion and overlapping. Correcting such noisy frames brings tremendous difficulties to enlarge the dance dataset, especially in terms of time and labor.

the ground truth skeleton sequence p. We also apply a gradient penalty [17] term in D_{global}. Therefore, the adversarial loss is defined as

$$\begin{aligned}\mathcal{L}_{adv} =& \mathbb{E}_p[\log D_{local}(S(p))]+ \\ & \mathbb{E}_{x,m}[\log[1 - D_{local}(S(x))]]+ \\ & \mathbb{E}_{p,m}[\log D_{global}(p,m)]+ \\ & \mathbb{E}_{x,m}[\log[1 - D_{global}(x,m)]]+ \\ & w_{GP}\mathbb{E}_{\hat{x},m}[(\| \nabla_{\hat{x},m} D(\hat{x},m)\|_2 - 1)^2],\end{aligned} \tag{4}$$

where w_{GP} is the weight for the gradient penalty term.

L_1 **distance** $\mathcal{L}_{L_1}$**.** Given a ground truth dance skeleton sequence Y with the same shape of $X \in R^{T\times 2V}$, the reconstruction loss at the joint level is

$$\mathcal{L}_{L_1} = \sum_{j\in[0,2V]} \|Y_j - X_j\|_1. \tag{5}$$

Feature matching loss $\mathcal{L}_{FM}$**.** We adopt the feature matching loss from [45] to stabilize the training of Global Content Discriminator D:

$$\mathcal{L}_{FM} = \mathbb{E}_{p,m} \sum_{i=1}^{M} \|D^i(p,m) - D^i(G(m),m)\|_1, \tag{6}$$

where M is the number of layers in D and D^i denotes the i^{th} layer of D. In addition, we omit the normalization term of the original $\mathcal{L}_{FM}$ to fit our architecture.

Full Objective. Our full objective is

$$\arg\min_G \max_D \mathcal{L}_{adv} + w_P\mathcal{L}_P + w_{FM}\mathcal{L}_{FM} + w_{L_1}\mathcal{L}_{L_1}, \tag{7}$$

where w_P, w_{FM}, and w_{L_1} represent the weights for each loss term.

5.5 Pose to Video

Recently, researchers have been studying motion transfer, especially for transferring dance motion between two videos [8, 31, 46, 50]. Among these methods, we adopt pix2pixHD GAN proposed by Wang et al. [45] rather than a state-of-the-art method because of its simplicity and effectiveness. Given a skeleton sequence and a video of a target person, the framework could transfer the movement of the skeleton sequence to the target person. Although pix2pixHD GAN [45] is an image-based method, our synthesized dance videos achieve temporal coherence for the effectiveness of our local temporal discriminator. As shown in Figure 9, the quality of our video is better than Lee et al. [25], which use vid2vid GAN [46] to transfer skeleton sequences to videos. Additional result is shown in Figure 10.

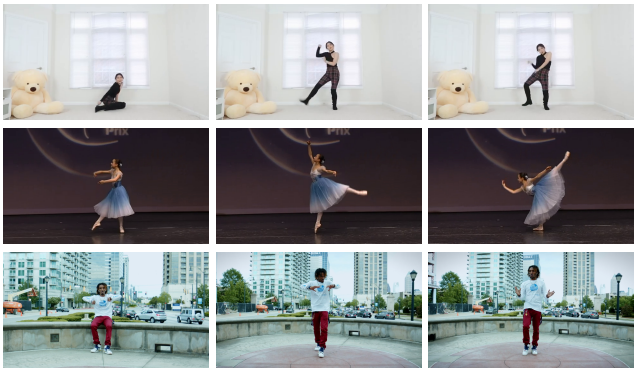


Figure 6: Video frames sampled from online dance videos of different categories. From top to down: K-pop, Ballet, and Popping. In total, we collect 100 online videos in our dataset.

6 EXPERIMENTS

6.1 Experimental Setup

We will evaluate the following baselines and our model.

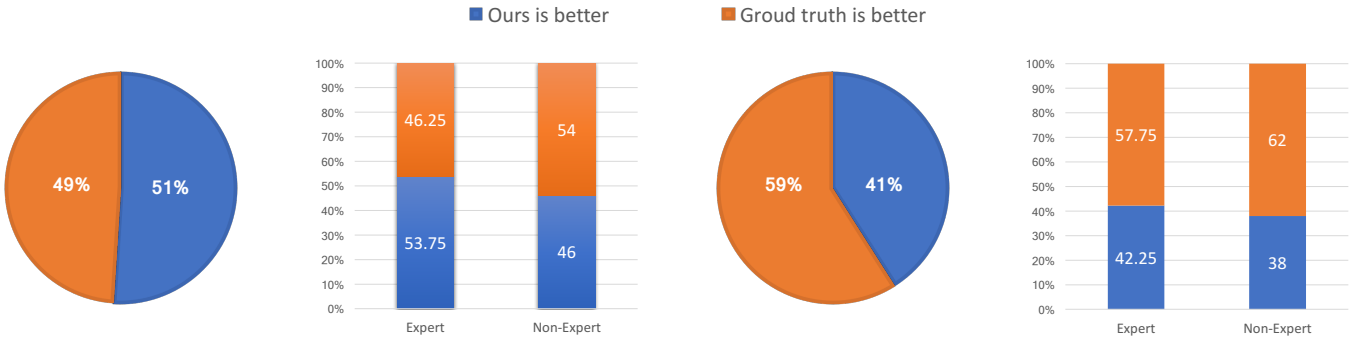
- $\boldsymbol{L_1}$. In this L_1 baseline, we only use L_1 loss to train the generator.
- **Global D**. Based on the $\boldsymbol{L_1}$ baseline, we add a Global Content Discriminator.
- **Local D**. Based on the **Global D** baseline, we add a Local Temporal Discriminator.
- **Our model**. Based on the **Local D** baseline, we add pose perceptual loss. These conditions are used in Table 2.

6.2 Dataset

To build our dataset, we apply OpenPose [5, 6, 47] to collected online videos of three representative categories of dance, as shown in Figure 6 to obtain the skeleton sequences. In total, We collected 100 videos about 3 minutes with a single dancer, and there are 40 k-pop videos, 20 ballet videos, and 40 popping videos. All the extracted skeleton sequences are cut into clips of 5s. There are 1782 k-pop clips, 448 ballet clips, and 1518 popping clips in our dataset. To avoid overlapping(from the same song) between the training set and test set, we select the last 10% of each type of dance for testing, and the remaining part is used for training.

6.3 Evaluation

6.3.1 User Study. To evaluate the quality of the generated skeleton sequences, we conduct a user study on Amazon Mechanical Turk, following the protocol proposed by Lee et al. [25] to compare the synthesis skeleton sequence and the ground-truth skeleton sequence. To make this study fair, we verify the ground truth skeletons and interpolate the missing frames. The users are first asked to answer a background question: "Do you learn to dance or have knowledge about dance?" and they are labeled as "Expert" or "None-Expert" based on their answer. Then, given a pair of dances with the same music clip, each participant needs to answer two questions: "Which dance is more realistic regardless of music?" and "Which dance matches the music better?". The results are summarized in



(a) Motion Realism Preference

(b) Style Consistency Preference

Figure 7: Results of user study on comparisons between our synthesized skeleton sequence and the ground truth. For each comparison, the participant should select the dances that "are more realistic regardless of music" and "better match the style of music". Each number denotes the percentage of preference. Our result is more preferred by Experts (familiar with dance).

	FID	Diversity	Cross-modal
Real	–	26.12	0.043
L_1	37.92	17.71	0.312
Global D	18.04	20.33	0.094
Local D	15.86	19.57	0.068
Our model	**3.80**	**25.63**	**0.046**

Table 1: Comparison between our model and baselines. For FID and the cross-modal evaluation, lower is better. For Diversity, higher is better. The details of the baselines are shown in Section 6.1.

Figure 7. Compared to the real dances, 51.2% of users prefer our approach in terms of motion realism and 40.83% in style consistency, which shows that our model can choreograph at a similar level with real artists. We randomly sample 20 pairs of five-second skeleton sequences in the user study, and 30 participants are involved.

6.3.2 Cross-modal Evaluation. It is challenging to evaluate if a dance sequence is suitable for a piece of music. To our best knowledge, there is no existing method to evaluate the mapping between music and dance. Therefore, we propose a cross-modal metric, as shown in Figure 8, to estimate the similarity between music and dance.

Given a training set $X = \{(P, M)\}$ where P is a dance skeleton sequence and M is the corresponding music. Then with a pre-trained music feature extractor E_m [11], we aggregate all the music embedding $F = \{E_m(M), M \in X\}$ in an embedding dictionary.

With our generator G, we can get the synthesized skeleton sequence $P_u = G(M_u)$ for the given music M_u. Also, we find a skeleton sequence that represents the music M_u to compare with P_u. We first obtain the music feature F_u by $F_u = E_m(M_u)$, and then let F_v be the nearest neighbor of F_u in the embedding dictionary. In the end, we use its corresponding skeleton sequence P_v to represent the music M_u. The second step is to measure the similarity between two skeleton sequences with the novel metric learning objective based on a triplet architecture and Maximum Mean Discrepancy, proposed by Coskun et al. [13]. More implementation details about this metric will be shown in the supplement.

6.3.3 Quantitative Evaluation. In addition to our cross-modal evaluation, which proves that our result matches the music, we also adopt visual quality measurement following Fréchet Inception Distance (FID) [18] and diversity measurement from [25]. For FID, we generate 70 dances based on randomly sampled music and use our pre-trained ST-GCN to extract feature as there is no standard feature extractor for skeleton sequences. For diversity, we also generated 70 dances based on randomly sampled music and compute the FID between the 70 random combinations of them (To avoid randomness, we make 200 random processes and take the average score).

As shown in Table 2, all our proposed components steadily improve the score of FID, Diversity, and Cross-modal metrics, and our model is on par with the real artist. In particular, the pose perceptual loss contributes most significantly.

6.3.4 Qualitative Evaluation. The qualitative comparison between Lee et al. [25] and ours is illustrated in Figure 9. The dance videos for different targets are illustrated in Figure 10. Our skeleton synthesis results for different music styles are illustrated in Figure 11. Furthermore, visual ablation study and more demonstration will be presented in our supplementary video.

7 CONCLUSION

We have presented a two-stage framework to generate dance videos, given almost any music. With our proposed pose perceptual loss, our model can be trained on dance videos with noisy pose skeleton sequence (without human labels). Our approach can create arbitrarily long, good-quality videos. We hope that this pipeline of synthesizing skeleton sequence and dance video combining with pose perceptual loss can support more future work, including more creative video synthesis for artists.

Beyond sharing personalized dance music videos, another fun application of our system is to create a choreography for popular vocal groups whose members are all virtual animated characters. A

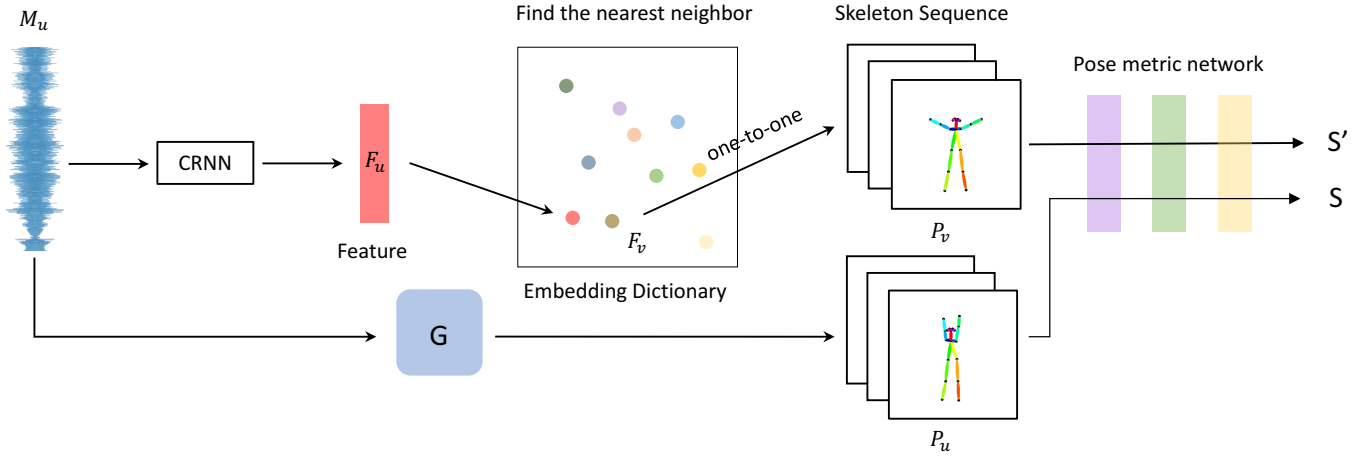


Figure 8: Cross-modal evaluation. We first project all the music pieces in the training set of the K-pop dataset into an embedding dictionary by a music classification network CRNN [11]. We train the pose metric network based on the K-means clustering result of the embedding dictionary. For the K-means clustering, we choose K = 5, according to the Silhouette Coefficient. The similarity between M_u and P_u is measured by $\|S - S'\|^2$.

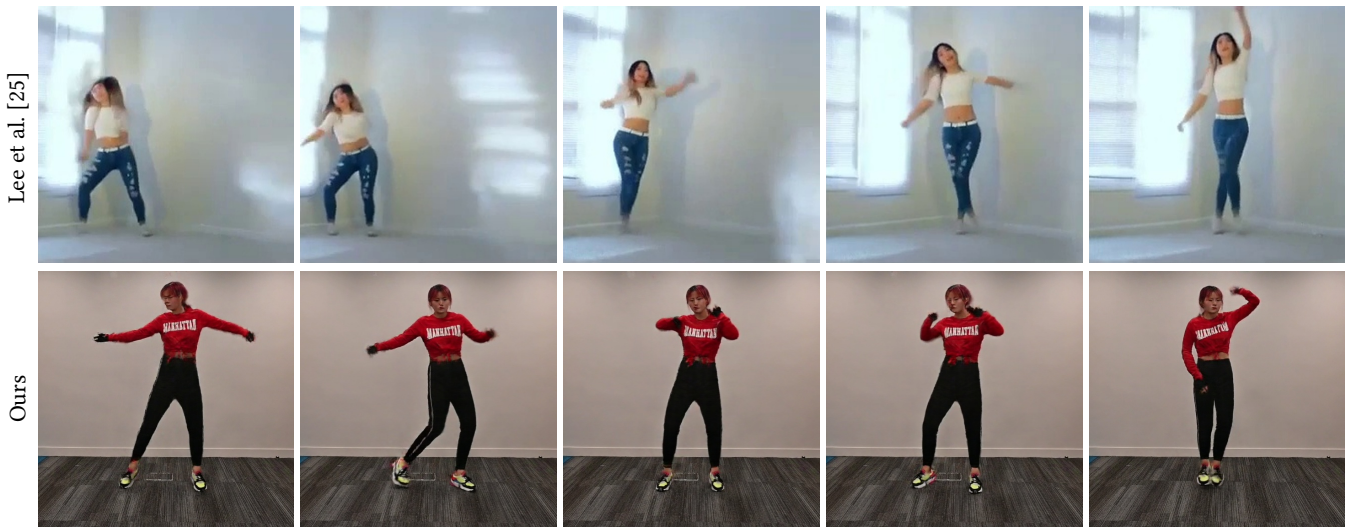


Figure 9: Compared with Lee et al. [25], our generated dance skeleton sequence is more temporal-coherent, and thus the synthesized dance video is more plausible. Though they use the state-of-the-art vid2vid GAN [46], the background in the video is still not stable. The result of Lee et al. [25] is extracted from their demo video.

,

mobile application Sway [19] can create personalized dance videos with a limited set of dance sequences and music, which can be enriched with our system to generate unseen dance and music. Moreover, our framework can be extended to work on a humanoid robot so that the robot dances with music. Since our model is learning-based, another possible application is to learn the dance style of a specific superstar. With our system pre-trained on the dance by Michael Jackson, one can generate novel dance videos in his style for any music.

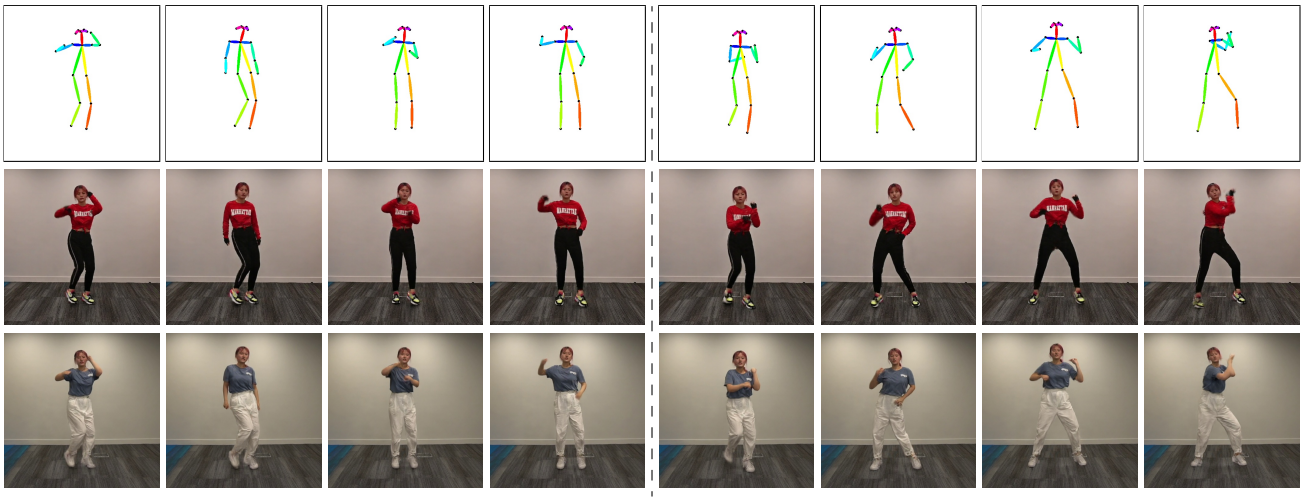


Figure 10: Synthesized dance video conditioned on the music *"LIKEY"* by *TWICE*. For each 5-second dance video, we show four frames. The top row shows the skeleton sequence, and the bottom rows show the synthesized video frames conditioned on different target videos.

Figure 11: Qualitative results conditioned on different styles of music clips. Every two rows show the skeleton sequences based on a different style of music, from top to down: Popping, Ballet, and K-pop. In each row, we sample one pose every 0.3s.

REFERENCES

[1] Hyemin Ahn, Jaehun Kim, Kihyun Kim, and Songhwai Oh. 2020. Generative Autoregressive Networks for 3D Dancing Move Synthesis from Music. *IEEE Robotics and Automation Letters* (2020).

[2] Omid Alemi, Jules Françoise, and Philippe Pasquier. 2017. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. *networks* (2017).

[3] Joan Bruna, Pablo Sprechmann, and Yann LeCun. 2016. Super-Resolution with Deep Convolutional Sufficient Statistics. In *ICLR*.

[4] Haoye Cai, Chunyan Bai, Yu-Wing Tai, and Chi-Keung Tang. 2018. Deep video generation, prediction and completion of human action sequences. In *ECCV*.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.

[7] Daniel Castro, Steven Hickson, Patsorn Sangkloy, Bhavishya Mittal, Sean Dai, James Hays, and Irfan Essa. 2018. Let's Dance: Learning From Online Dance Videos. In *arXiv:1801.07388*.

[8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody dance now. In *ICCV*.

[9] Lele Chen, Zhiheng Li, Ross K. Maddox, Zhiyao Duan, and Chenliang Xu. 2018. Lip Movements Generation at a Glance. In *ECCV*.

[10] Qifeng Chen and Vladlen Koltun. 2017. Photographic Image Synthesis with Cascaded Refinement Networks. In *ICCV*.

[11] Keunwoo Choi, György Fazekas, Mark B. Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *ICASSP*.

[12] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. 2018. Human motion analysis with deep metric learning. In *ECCV*.

[13] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. 2018. Human Motion Analysis with Deep Metric Learning. In *ECCV*.

[14] DIGITALTRENDS. 2020. The most-viewed YouTube videos of all time. https://www.digitaltrends.com/web/most-viewed-youtube-videos/

[15] Alexey Dosovitskiy and Thomas Brox. 2016. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *NeurIPS*.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NeurIPS*.

[17] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *NeurIPS*.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*. 6626–6637.

[19] Humen, Inc. 2020. Sway: Magic Dance. https://getsway.app/

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*.

[22] Jae Woo Kim, Hesham Fouad, and James K. Hahn. 2006. Making Them Dance. In *AAAI Fall Symposium: Aurally Informed Performance*.

[23] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. 2011. HMDB: a large video database for human motion recognition. In *ICCV*.

[25] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. 2019. Dancing to Music. In *NeurIPS*.

[26] Juheon Lee, Seohyun Kim, and Kyogu Lee. 2018. Listen to Dance: Music-driven choreography generation using Autoregressive Encoder-Decoder Network. *CoRR* (2018).

[27] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. 2018. Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. In *IJCAI*.

[28] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. 2019. Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition. In *CVPR*.

[29] Yitong Li, Martin Renqiang Min, Dinghan Shen, David E. Carlson, and Lawrence Carin. 2017. Video Generation From Text.

[30] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *ICLR*.

[31] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. 2019. Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis. In *ICCV*.

[32] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On Human Motion Prediction Using Recurrent Neural Networks. In *CVPR*.

[33] Michaël Mathieu, Camille Couprie, and Yann LeCun. 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*.

[34] Anh Mai Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NeurIPS*.

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[36] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. 2017. Temporal Generative Adversarial Nets with Singular Value Clipping. In *ICCV*.

[37] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *CVPR*.

[38] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. 2006. Dancing-to-Music Character Animation. *Comput. Graph. Forum* (2006).

[39] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. 2019. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. In *CVPR*.

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* (2012).

[41] Taoran Tang, Jia Jia, and Hanyang Mao. 2018. Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis. In *ACM Multimedia*.

[42] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MoCoGAN: Decomposing motion and content for video generation. In *CVPR*.

[43] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating Videos with Scene Dynamics. In *NeurIPS*.

[44] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2018. End-to-End Speech-Driven Facial Animation with Temporal GANs. In *BMVC*.

[45] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs. In *CVPR*.

[46] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *NeurIPS*.

[47] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *CVPR*.

[48] Nelson Yalta. 2017. Sequential Deep Learning for Dancing Motion Generation.

[49] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

[50] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L. Berg. 2019. Dance Dance Generation: Motion Transfer for Internet Videos. *CoRR* (2019).

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

A AUDIO ENCODER

To extract the feature of the music, we adopt a deep audio encoder from [44], consisting of 1D convolutional layers and a fully-connected (FC) layer.

stage	specification	output sizes
input data	-	1×1600
1D conv_1	kernal 80, stride 16	1×1600
1D conv_2	kernal 4, stride 2	16×100
1D conv_3	kernal 4, stride 2	32×50
1D conv_4	kernal 10, stride 5	64×25
1D conv_5	kernal 5, stride 1	128×5
fc	a FC layer	256

Table 2: Details of our audio encoder. The audio encoder extracts 256 dimensional features from audio pieces containing 1600 samples. The output sizes of the 1D convolutional layer is denoted as $number\ of\ feature\ maps \times feature\ size$**.**

B POSE GENERATOR

The pose generator [4] is responsible for generating the skeleton sequence in the first step of our system. Our pose generator takes hidden states as input and outputs the skeleton sequence. As shown in Figure 14, the pose generator is a multi-layer perceptron.

C POSE DISCRIMINATOR

Given a skeleton sequence $X \in R^{T \times 2V}$, we firstly reshape it as $T \times V \times 2$. For the point-level feature learning stage, the kernel sizes along the joint dimension are kept 1, so they are forced to learn point-level representation from 2D coordinates for each joint independently. After that, we transpose the feature maps with parameter $(0, 2, 1)$ so that the joint dimension is moved to channels of the tensor. Then in the next stage, all subsequent convolution layers extract global co-occurrence features from all joints of a person [27]. In the end, we flatten the feature map and gain a feature embedding by using a fully-connected layer.

D GLOBAL CONTENT DISCRIMINATOR

The Global Content Discriminator is shown in Figure 13.

E POSE METRIC NETWORK

For our novel cross-modal metric, we adopt the pose metric network, proposed by [12]. We firstly introduce the formulation of MMD measures used in this pose metric network. Given two different distributions p and q,

$$\begin{aligned} \text{MMD}[k, X, Y] = &\frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} k(x_i, x_j) - \\ &\frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j) + \\ &\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} k(y_i, y_j) \end{aligned} \tag{8}$$

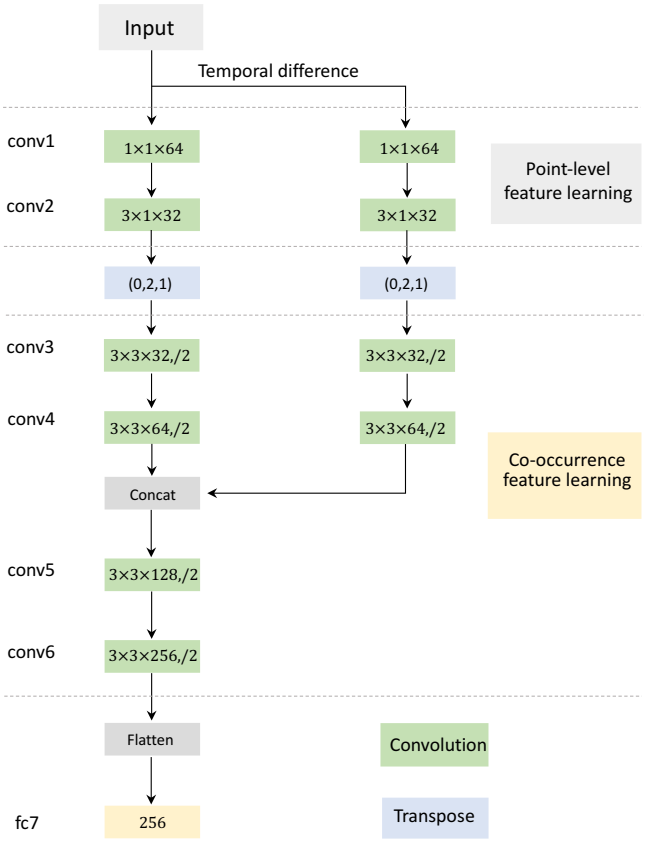


Figure 12: Overview of pose discriminator, which is a two-branch CNN. For the blocks of the convolution layer, the last dimension represents the number of output channels. A trailing "/2" means an appended MaxPooling layer with stride 2 after convolution.

where $X := x_1, x_2, ..., x_m$ is the sample set from p and $Y := y_1, y_2, ..., y_n$ is the sample set from q. Besides,

$$k(x, y) = \sum_{q=1}^{K} k_{\sigma_q}(x, y) \tag{9}$$

where k_{σ_q} is a Gaussian kernal with bandwidth parameter σ_q, and K is number of kernels.

Furthermore, given two dance sequence $X = \{x_1, x_2, ..., x_n\}$ and $Y = \{y_1, y_2, ..., y_n\}$ (where x_t and y_t represent the pose at time t), the similarity metric can be expressed directly as the squared Euclidean distance in the embedding space, which can be written mathematically as

$$d(f(X), f(Y)) = \|f(X) - f(Y)\|^2 \tag{10}$$

where $f(\cdot)$ is the learned embedding function that maps a motion sequence to a point in a Euclidean space and f is learned by means of a deep learning model trained with a MMD-NCA loss:

$$\mathcal{L}_{metric} = \frac{\exp(-\text{MMD}[k, f(X), f(X^+)])}{\sum_{j=1}^{M} \exp(-\text{MMD}[k, f(X), f(X_{c_j}^-)])} \tag{11}$$

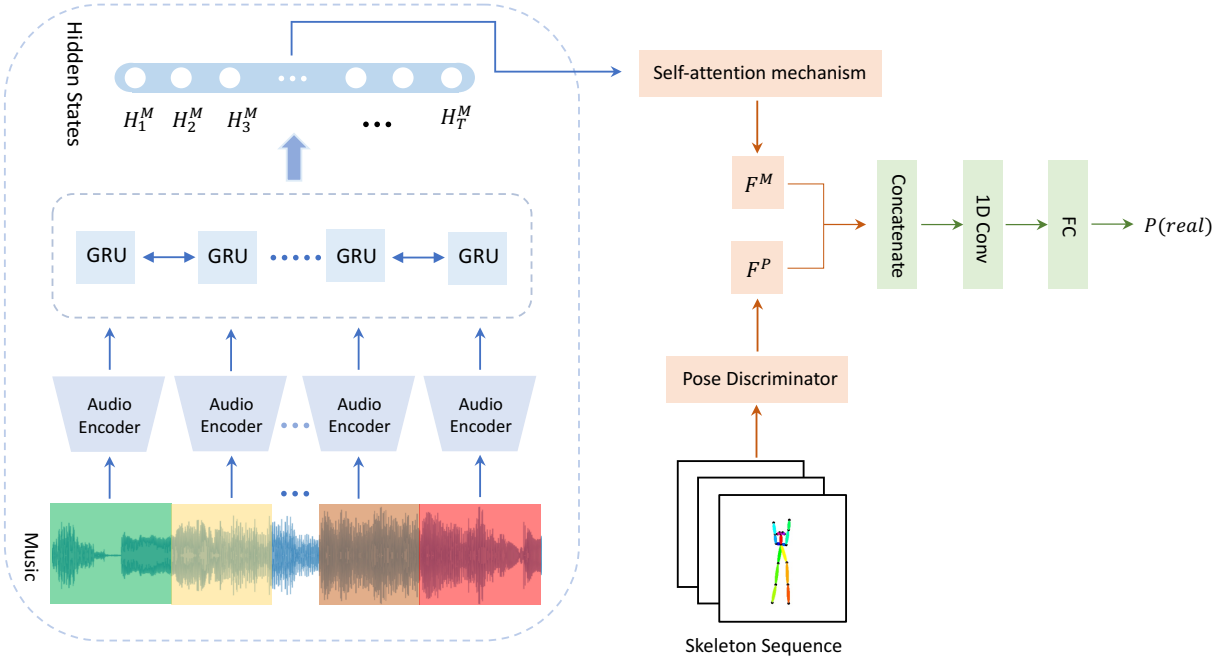


Figure 13: The architecture of Global Context Discriminator, which contains the music feature extraction part and skeleton sequence extraction part, and a small classifier consisting of a 1D convolutional layer and a fully-connected layer.

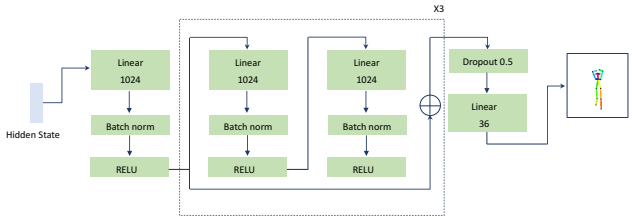


Figure 14: Overview of our pose generator. $\oplus$ stands for element-wise addition. The output of the last fully-connected layer is $\in R^{2V}$, where $V = 18$.

where X and X^+ represent skeleton sequence from the same category, while X_{c_j} represents samples from category $c_j \in C$ (C is a set of M different categories).

F POSE NORMALIZATION

For our dataset, the skeleton sequences are scaled to a similar height: we first define one video's skeleton sequence as the standard sequence and used a weighted ratio to scale sequences of other videos to match the standard skeleton sequences. All the skeleton sequences are normalized to $[-1, 1]$. And for the step of pose to video, we adjust the generated skeleton sequences to fit the subject of the target video.

G OTHER DATASET

Let's Dance Dataset. Castro et al. [7] released a dataset containing 16 classes of dance. The dataset provides information about human skeleton sequences for pose recognition. Though there are existing enormous motion datasets [24, 37, 40] with skeleton sequences, we choose *Let's Dance Dataset* to pre-train our ST-GCN for pose perceptual loss as dance is different with normal human motion.

FMA. For our cross-modal evaluation, the extraction of music features is needed. To achieve this goal, we adopt CRNN [11] and choose the Free Music Archive (FMA) dataset to train CRNN. In FMA, genre information and the music content are provided for genre classification.

H IMPLEMENTATION DETAILS

All the models are trained on an Nvidia GeForce GTX 1080 Ti GPU. For the first stage in our framework, the model is implemented in PyTorch [35] and takes approximately one day to train for 400 epochs. For the hyperparameters, we set $V = 18$, $T = 50$, $t = 5$, $K = 16$, $S = 16000$. For the self attention mechanism, we set $k = 256$, $l = 40$. For the loss function, the hyperparameters $\{\lambda_l\}$ are set to be $[20, 5, 1, 1, 1, 1, 1, 1]$ and $w_{GP} = 1$, $w_P = 1$, $w_{FM} = 1$, $w_{L_1} = 200$. Though the weight of L_1 distance loss is relatively large, the absolute value of the L_1 loss is quite small. We used Adam [23] for all the networks with a learning rate of 0.003 for the generator and 0.003 for the Local Temporal Discriminator and 0.005 for the Global Content Discriminator.

For the second stage that transfers pose to video, the model takes approximately three days to train, and the hyperparameters of it adopt the same as [8]. For the pre-train process of ST-GCN and CRNN, we also used Adam [23] for them with a learning rate of 0.002. ST-GCN achieves 46% precision on *Let's Dance Dataset*. CRNN is pretrained on the FMA, and the top-2 accuracy is 67.82%.