Keyframe Control of Music-driven 3D Dance Generation

Zhipeng Yang†, Yu-Hui Wen†, Shu-Yu Chen, Xiao Liu, Yuan Gao, Yong-Jin Liu*, Lin Gao*, Hongbo Fu

Abstract—For 3D animators, choreography with artificial intelligence has attracted more attention recently. However, most existing deep learning methods mainly rely on music for dance generation and lack sufficient control over generated dance motions. To address this issue, we introduce the idea of keyframe interpolation for music-driven dance generation and present a novel transition generation technique for choreography. Specifically, this technique synthesizes visually diverse and plausible dance motions by using normalizing flows to learn the probability distribution of dance motions conditioned on a piece of music and a sparse set of key poses. Thus, the generated dance motions respect both the input musical beats and the key poses. To achieve a robust transition of varying lengths between the key poses, we introduce a time embedding at each timestep as an additional condition. Extensive experiments show that our model generates more realistic, diverse, and beat-matching dance motions than the compared state-of-the-art methods, both qualitatively and quantitatively. Our experimental results demonstrate the superiority of the keyframe-based control for improving the diversity of the generated dance motions.

Index Terms—3D animation, generative flows, multi-modal, music-driven, choreography.

1 Introduction

Dance motions are kinematically complex and diverse for long-term spatio-temporal structures [1], [2], which make it challenging to generate high-quality dance animations. Recently, deep learning models have been successfully applied to achieve automatic dance generation [3], [4], [5], [6], [7], [8]. These works mainly rely on music to drive the synthesis of dance motions. Recently, some methods have been proposed to achieve style control over the generated dance [2], [9], [10]. However, as choreography is a creative process, dance generation mainly driven by music or controlled by style is not flexible enough to control the dance poses at specific time steps, while such specific controls are often required by choreographers [11], [12]. On the other hand, keyframe-based control, which allows artists to depict their ideas via a sparse set of keyframes, has been widely adopted for creating 2D and 3D animations [13], [14]. Inspired by these works, we are interested in applying keyframe-based control to music-driven 3D dance generation and employing deep learning models to perform keyframe interpolation.

Based on the above observations, we extend the idea of keyframe interpolation for music-driven dance generation, so that 3D animators can enjoy the automated feature of dance generation while still having a sufficient control of generated animations. Traditional keyframe interpolation techniques (e.g., [15], [16], [17]) mainly use smoothness cues to fill missing frames between keyframes. We present a novel transition generation technique, which leverages a piece of input music to infer intermediate poses between a sparse set of key poses. Specifically, our transition generation technique should have the following properties: (i) producing motions with high diversity and realism, (ii) aligning well with the input music, and (iii) achieving a flexible control by the key poses, whose intervals might not be fixed.

To achieve them, we base our transition generation technique on recent advances in modeling complex distributions [18], [19]. Specifically, we build a probabilistic model for dance generation conditioned on the input music and key poses. We use normalizing flows to generate the dance motion in the current frame given its past-context dance motions and condition signals, including the features extracted from the input music and its corresponding target key pose (i.e., a key pose closest to a current frame). The normalizing flows are a set of invertible functions to map dance motions to inherent feature vectors, such that the probability distribution of dance motions is transformed to a simple Gaussian distribution. Thanks to the exact mapping between the two distributions, our model achieves highly diverse and realistic dance generation. However, as shown in our experiments (Section 4.5), by only adding condition signals of key poses, the probabilistic model cannot generate transition dance motions between keyframes robustly. This is mainly because the distance between a target pose and its corresponding generated pose is not clear and each generated dance motion is not aware of the number of left frames to its target keyframe due to the varying intervals between key poses. To solve this problem, we add a key pose loss term that describes the distance between the input key poses and their corresponding generated poses to our objective function for modeling the distribution of dance motions. Furthermore, we introduce a time embedding at each timestep as an additional condition to achieve a robust transition generation of varying lengths. Moreover, our dance motions are represented by joint rotations and global translations,

- *†Equal contribution*
- ** Corresponding author*
- *Z. Yang, S.-Y. Chen and L. Gao are with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China. E-mail: yangzhipeng19s@ict.ac.cn, chenshuyu@ict.ac.cn, gaolin@ict.ac.cn*
- *Y.-H. Wen and Y.-J. Liu are with the CS Dept, BNRist, Tsinghua University, Beijing 100190, China. Email: wenyh1616@tsinghua.edu.cn, liuyongjin@tsinghua.edu.cn*
- *X. Liu and Y. Gao are with Tomorrow Advancing Life Education Group, Beijing 100190, China. Email: liuxiao15@tal.com, gaoyuan23@tal.com*
- *H. F is with the School of Creative Media, City University of Hong Kong. Email: hongbofu@cityu.edu.hk*

Fig. 1: 3D dance motions (in an opaque style) generated by our method conditioned on a piece of music and a sparse set of key poses (in a semi-transparently style). For clarity, the generated frames are shown with an interval of 10. Our method is able to generate transition dance motions of varying lengths between the key poses.

and thus can be easily transferred to a novel 3D character for animation, as shown in Figure 1. The generated 3D character dance animations can be widely used in AR/VR, Film industry and Entertainment industry.

Our main contributions are summarized as follows. First, we propose a novel transition generation technique for synthesizing music-driven dance motions based on normalizing flows. Thanks to the probabilistic nature of normalizing flows, different yet plausible dance transitions can be generated between arbitrary pair of key frames. Second, we introduce a new key pose loss to the objective function of learning the conditional distribution of dance motions given the input music. Third, we introduce transition-control signals, i.e., a combination of time embedding and key poses, to achieve a robust transition of varying lengths between given key frames. Experimental results show that our method generates more realistic, diverse, and music-synchronized dance motions compared to the state-of-the-art methods.

2 Related Work

2.1 Human Motion Generation

2.1.1 Motion Prediction

For motion prediction, human motions are synthesized with multiple frames of movements as a past context. Recurrent Neural Networks (RNNs) have been used for motion generation given the past context. For example, Fragkiadki et al. [20] proposed an Encoder-Recurrent-Decoder (ERD) architecture by incorporating separate encoder and decoder networks before and after recurrent layers for modeling human motions. Jain et al. [21] represented human motion sequences as spatio-temporal graphs, which are modeled by structural RNNs. Some recent works have developed new architectures based on RNNs to improve the performance of motion forecasting [22], [23]. GANs have also been used to generate human motions [24] and achieved impressive improvements in producing highly convincing random human motions, though GANs are difficult to train [25].

2.1.2 Motion Control

Motion control is referred as application scenarios in which dense control signals (e.g., motion path, motion velocity, motion direction), usually user-defined, are used to drive animation generation. Motion graphs [26] are able to produce motions by traversing nodes and edges that map to character states or motions clips from captured databases. To achieve flexible control over motion graphs, some works propose extra conditions or local PCA models on pose candidates [27], [28]. Deep learning approaches have been proposed to improve the scalability and flexibility of motion control. For example, Holden et al. [29], [30] proposed a feed-forward convolutional neural network to build a constrained animation synthesis framework, which is conditioned by root trajectory and end-effectors. Mode-aware [31] neural networks can automatically choose a mixture of network weights at run-time to generate motions. Recently, Ling et al. [32] proposed a motion VAEs model to generate target position guided animation, which is able to avoid barriers. A recent work [19] proposed a probabilistic model to generate realistic human motions under the control of root trajectory, based on normalizing flows [33], which have gained much attention for highly realistic image samples [18]. Normalizing flows make it possible to compute and maximise the likelihood of the real motion data, and thus produce highly realistic motions [34]. Please refer to [35] for much more motion synthesis research works.

2.1.3 Transition Generation

Transition generation is defined as a type of control with a sparse set of keyframes, between which large gaps of motions must be filled [36]. This task is closely related to the keyframe interpolation problem [17]. Recently, deep learning methods have been applied to solve the keyframe interpolation problem for character animation. For example, Zhang et al. [37] built a RNN conditioned on input keyframes to generate intermediate-frame motions by sampling from example motions. Harvey et al. [38] proposed a transition generation method based on LSTM to accelerate the creation of transition motions. Although they have achieved impressive results, they generate only deterministic outputs.The LSTM has also been employed for human mesh sequences generation [39] and speech-driven facial animation [40]. On the contrary, Harvey et al. [36] introduced a schedule target-noise vector into a motion predictor for their transition generation, to enable variations in generated transition motions. We propose a novel transition generation technique by using normalizing flows, whose probabilistic nature is benefit for generating different yet plausible dance motions given the same piece of music.

2.2 Music to Dance Synthesis

Dance is a type of artistic body motions performed with music. Music-driven dance generation is generally studied in application scenarios where music control signals are used to drive dance generation. This task has been studied widely in 3D scenarios by using deep learning methods. Many previous works use RNNs [3], [41], [42] to generate dances conditioned on music. RNN-based solutions suffer from the error accumulation problem in long-time motion generation. To address this issue, transformers have been imported into music-to-dance tasks [43] to predict future dance movements conditioned on music beats and a past seed dance sequence. However, most of these RNNs and transformer-based methods yield only a single sequence of dance for a given input.

To generate different dance motions for the same input and thus increase the diversity of generated results, some recent works adopt GANs for this task. For example, Lee et al. [7] proposed a GAN-based model to decompose and compose dance motions from music beats. Ferreira et al. [6] used GCNs as a motion generator in the framework of GAN for music-to-dance generation. Ren et al. [8] applied a multi-layer perceptron as a motion generator and introduced a new pose perceptual loss to produce natural

dance motions. Sun et al. [4] proposed a GAN-based cross-modal association model to capture correlation between music and dance. A probabilistic model based on normalizing flows [44] has been used to generate diverse dance motions conditioned on a piece of music and previous poses, which are encoded by a multi-modal transformer.

The above mentioned works show that deep neural networks can greatly benefit the task of music-driven dance synthesis. However, their controllability can be improved, since their generation process is driven only by the input music, which provides a relatively weak control signal.

2.3 Dance Control

Some recent methods have achieved motion or style control over generated dance motions. For example, Zhuang et al. [1] proposed to generate dance motions with a dance melody line describing desired motion speeds. Zhuang et al. [9] proposed a novel autoregressive generative model to generate dance motions by sampling from a conditional distribution that takes a musical type and musical features as condition signals. Their method uses control signals directly extracted from the music [9] to generate style-consistent dance from music. However, it still lacks sufficient controllability over dance motions. Chen et al. [2] proposed a choreography-oriented graph-based dance synthesis method called ChoreoMaster, which learns a unified embedding space for music and dance segments. To learn choreomusical rhythm embeddings for music and dance, a rhythm signature classification network is trained with the ground truth rhythm signatures, which are manually labeled by professional artists. The method proposed by Aristidou et al. [10] takes a global structure of a dance genre into consideration to control the contextual and cultural styles of dance motions. In more details, the global structure is controlled by the distribution of motion motifs, which are used to represent motion word clusters. Each motion word is a temporal set of poses depending on the musical beat.

All the above-mentioned works use dense control signals for dance generation. In contrast, we propose to synthesize dance motions conditioned on the input music and constrained by a sparse set of key poses, thus allowing for a more flexible control over generated dance motions.

3 METHODS

In this work, we propose a novel transition generation technique for choreography, which generates dance motions between keyframes with an input piece of music, based on normalizing flows. Next, we will introduce the details of our method.

3.1 Normalizing Flow

First, we briefly review normalizing flow, which has been less explored for dance generation. Given an observed variable x and a latent variable z, the parametric model of their joint distribution is denoted as $p(x, z)$ (called a *generative model* [45]). To learn the parametric model of a given dataset $X = \{x_1, ..., x_N\}$, it is typical to perform maximum marginal likelihood by maximizing $\log p(X) = \sum_{i=1}^{N} \log p(x_i)$. Generally, the marginal likelihood is intractable to compute or differentiate directly for flexible generative models that are parameterized by neural networks. By introducing an approximate posterior distribution for the latent variables $q(z|x)$ (called an *inference model* [45]), we are able to obtain a lower bound on the log-likelihood of each observation [46]:

$$\log p(x) \geq \log p(x) - D_{KL}(q(z|x)||p(z|x)) = \mathcal{L}(x; \theta), \tag{1}$$

where $D_{KL}(q(z|x)||p(z|x))$ is the Kullback-Leibler (KL) divergence, which is non-negative. By minimizing the KL divergence, $\mathcal{L}(x; \theta)$ is able to match the true objective $\log p(x)$. Considering both requirements for optimizing the objective and fast inference, the key is to choose a flexible and computationally-feasible approximate posterior distribution $q(z|x)$ to match the true posterior distribution $p(z|x)$ [45].

Normalizing Flow, a sequence of K invertible parameterized transformations f_k, is introduced in the context of stochastic gradient variational inference to build a flexible posterior distribution z_K from a relatively simple distribution z_0 [33]:

$$z_0 = q(z_0|x), z_K = f_K(...(f_1(z_0, x))). \tag{2}$$

Thus, the probability density function for z_K can be computed efficiently with the Jacobian determinant $\det|\frac{\partial z_k}{\partial z_{k-1}}|$ of each transformation f_k:

$$\log q(z_K|x) = \log q(z_0|x) - \sum_{k=1}^{K} \log \det|\frac{\partial z_k}{\partial z_{k-1}}|. \tag{3}$$

3.2 Transition Generation for Choreography

Now we introduce the details of our transition generation method for music-driven dance motions, which are constrained by a sparse set of key poses.

3.2.1 Data Representation

We represent a human motion by a skeleton structure with $j = 24$ joints. Specifically, we use an exponential map [47] vector $r_t \in \mathbb{R}^{j*3}$ along with a root joint position $p_t \in \mathbb{R}^3$ to represent a pose at frame t. We also extract the forward, sideways, and angular velocity $v_t \in \mathbb{R}^3$ of root to describe the movement path of dance sequences on the ground. In addition, we extract the foot contact signals for each position. Especially, we gather four-dimensional 0-1 vectors $c_t \in \mathbb{R}^4$ to denote the contact relations between floor and feet (left ankle, left toes, right ankle, right toes). Then, we get pose features x_t by merging p_t, r_t, v_t and c_t.

For audio data, we adopt Mel Frequency Cepstrum Coefficients (MFCCs), which are widely used for speech analysis and music recognition, and extract a 40-channel MFCC feature m_t from the corresponding music signal at frame t by the audio processing toolbox Librosa [48].

3.2.2 Transition Generation

We train a probabilistic model to generate a dance motion x_t at frame t conditioned on τ past-context dance motions $x_{t-\tau:t-1}$ and condition signals $s_t = (m_{t-\tau:t}, e_\gamma, E_t)$, where $m_{t-\tau:t}$ represents music signals from frames $t - \tau$ to t, e_γ denotes the target key pose at the frame γ, and E_t represents the time embedding at frame t. Our framework is shown in Figure 2. By introducing the idea of normalizing flow, we define a sequence of invertible transformations f_k for the approximate posterior distribution as:

$$z_0 = q(z_0|x_{t-\tau:t}, s_t), z_K = f_K(...(f_1(z_0, x_{t-\tau:t}, s_t))). \tag{4}$$

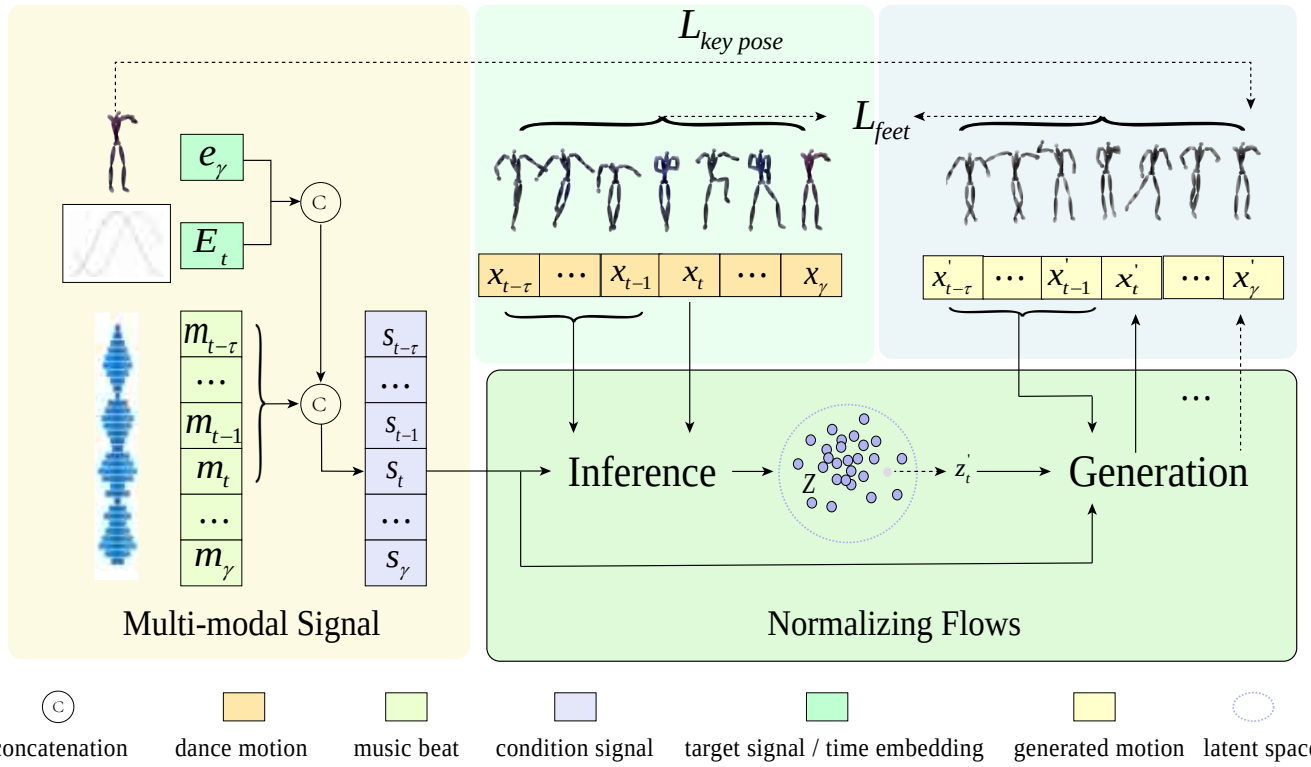


Fig. 2: Framework of our transition generation for choreography based on normalizing flows. The transformation from $x_t (\forall t = \tau ... \gamma)$ to the latent space Z is conditioned on past-context dance motions $x_{t-\tau:t-1}$ and condition signals $s_t = (m_{t-\tau:t}, e_\gamma, E_t)$. $m_{t-\tau:t}$ represents music signals from frames $t-\tau$ to t, and e_γ denotes the target key pose at the γ-th frame, while E_t is the embedding of time signal t for a correct transition generation. To generate x'_t, we sample z'_t in the latent space and transform it using reversed flows under the control of the condition signals s_t and past-context dance motions $x'_{t-\tau:t-1}$. $L_{\text{key pose}}$ describes the distance between the generated pose x'_γ at the target frame γ and its corresponding key pose. After generating a full motion sequence $x'_{\tau:\gamma}$, L_{feet} describes the distance between feet contacts of the generated motion sequence $x'_{\tau:\gamma}$ and the ground truth sequence $x_{\tau:\gamma}$.

Then, we define an invertible autoregressive transformation with a simple Jacobian determinant as follows [45]:

$$z_k = \mu_k + \delta_k \odot z_{k-1}, \quad \forall k = 1...K \tag{5}$$

where $\mu_k = (1 - \delta_k) \odot a_k$ and $\delta_k = \text{sigmoid}(b_k)$ are transformation parameters from an autoregressive model $[a_k, b_k] \leftarrow \text{LSTM}(z_{k-1}, x_{t-\tau:t}, s_t; \theta)$. Then, the determinant of $\frac{\partial z_k}{\partial z_{k-1}}$ is $\Pi_{d=1}^{D} \delta_{k,d}$. With $z_0 = \mu_0 + \delta_0 \odot \epsilon (\epsilon \sim \mathcal{N}(0, I))$, $\log q(z_K|x)$ is computed efficiently by:

$$\begin{aligned} \mathcal{L}_{flows} &= \log q(z_K|x_{t-\tau:t}, s_t) \\ &= -\sum_{d=1}^{D} (\frac{1}{2}\epsilon_d^2 + \frac{1}{2}\log(2\pi) + \sum_{k=0}^{K} \log \delta_{k,d}). \end{aligned} \tag{6}$$

The flexibility of the distribution of $q(z_K|x_{t-\tau:t}, s_t)$ after the iteration of z_K, which increases with the expressivity of the autoregressive model and the flow length K, determines the ability to fit to the true posterior $p(z|x_{t-\tau:t}, s_t)$.

Inspired by *Glow*, which is proposed for parameterizing complex distributions [18], [19], we develop the architecture of our probabilistic model based on a sequence of flow steps, as illustrated in Figure 3. Each flow step contains three invertible layers: an activation and normalizing (Actnorm) layer, a linear transformation layer, and an affine coupling layer. Mathematically, we denote h_t as the hidden features of x_t after the first two invertible layers, and then split h_t to two equal parts $[h_t^{lo}, h_t^{hi}]$. In the affine coupling layer, we perform an affine transformation to one half of the hidden features h_t^{hi}. Specifically, the transformation parameters are determined by the other half hidden features h_t^{lo}, the past-context information $x_{t-\tau:t-1}$ and condition signals s_t. Finally, the output z_k of the k-th flow step frame can be calculated as:

$$z_k = [z_k^{lo}, z_k^{hi}] = [h_k^{lo}, \mu_k + \delta_k \odot h_k^{hi}], \quad \forall k = 1...K \tag{7}$$

where $\mu_k = (1 - \delta_k) \odot a_k$ and $\delta_k = \text{sigmoid}(b_k)$ are transformation parameters from the autoregressive model $[a_k, b_k] \leftarrow \text{LSTM}(z_{k-1}, x_{t-\tau:t}, s_t; \theta)$. We notice that the equation of z_k in Eq. 7 is a particular version of Eq. 5, so the computation of Eq. 6 still works [45].

As described above, we add the condition signal of the target key pose to generate dance motions for transition generation. However, the transition generation is not robust. To solve the problem, we firstly add a key pose loss term to Eq. 6 to constrain the generated motions at the keyframes to match the corresponding key poses. Then, we propose to use a time embedding to achieve a flexible control over the generated dance motions of varying lengths. The details are described in the following.

3.2.3 Losses

We propose a key pose loss to generate a sequence of dance motions to match the target key pose at the frame γ. In more

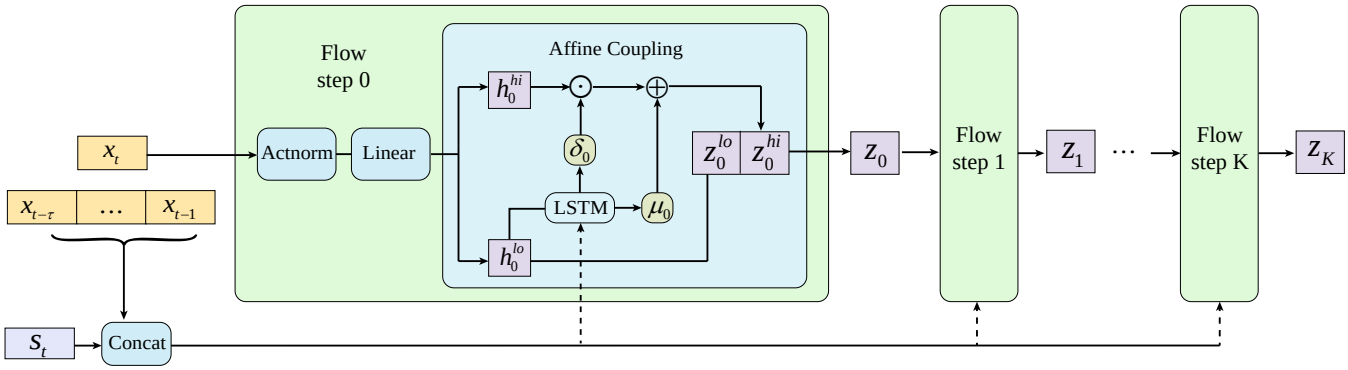


Fig. 3: Architecture of our normalizing flow model that consists of a sequence of flow steps. Each flow step f contains three invertible layers: an activation and normalizing layer, a linear transformation layer, and an affine coupling layer. Then, we perform an affine transformation to one half of the hidden features h_k^{hi} with the parameters from the other half h_k^{lo}, the past-context information $x_{t-\tau:t-1}$ and condition signals s_t. The output of the k-th flow step z_k is a concatenation of z_k^{lo} and z_k^{hi}.

details, we use z'_t randomly sampled from the latent space, the past-context motions $x_{t-\tau:t-1}$, and the condition signals ($m_{t-\tau:t}$, e_γ, and E_t) as input to generate the dance motion $x'_{\tau:\gamma}$ during training: $x'_t = {f_1}^{-1}(...f_K^{-1}(z'_t, x_{t-\tau:t-1}, m_{t-\tau:t}, e_\gamma, E_t))$. Then, we calculate the key pose loss for the generated sequence with an L2 norm $\mathcal{L}_{key\ pose} = \|\text{FK}(x'_\gamma) - \text{FK}(e_\gamma)\|_2$, where Forward Kinematics (FK) is performed to calculate the global joint positions of the human motion. We also use a contact-based loss $\mathcal{L}_{feet} = \frac{1}{\gamma-\tau}\sum_{t=\tau}^{\gamma}\|{c'}_t - c_t\|_1$ [36], to constrain the distance between the generated feet contacts ${c'}_t$ of ${x'}_t$ and ground-truth contacts c_t of x_t.

Finally, the total objective function of our method is defined as:

$$\mathcal{L}_{total} = \omega_1\mathcal{L}_{flows} + \omega_2\mathcal{L}_{key\ pose} + \omega_3\mathcal{L}_{feet}, \quad (8)$$

where ω_1, ω_2, ω_3 are the weights for the loss terms ($\omega_1 = 1.0$, $\omega_2 = 0.5$, and $\omega_3 = 0.1$ in our implementation.)

3.2.4 Time Embedding

We propose a time embedding for transition generation to enable a flexible control over generated dance motions of varying lengths. It is not sufficient to generate a dance motion by simply using an additional condition signal of its target key pose, because the transition generation must be aware of the frame position until arrival at the target keyframe. In this way, we use positional encodings, which shift smoothly and uniquely to represent the location of each frame [36], [49]:

$$E_{t,2l} = sin(\frac{nt}{basis^{2l/\mathcal{D}}}),\ E_{t,2l+1} = cos(\frac{nt}{basis^{2l/\mathcal{D}}}), \quad (9)$$

where nt is the time step between the start frame and the target frame. nt can evolve forward and backward in time. We have conducted experiments to show that it is better to use nt evolving backward, which represents the number of frames left to reach the target frame. $\mathcal{D}$ denotes the dimension of the input motion, and $l \in [0, ..., \mathcal{D}/2]$. The $basis$ is set to $10,000$ to represent the the rate of change in frequencies along the input dimension as [36], [49].

4 Experiments and Evaluation

In this section, we conduct qualitative and quantitative experiments to evaluate our transition generation for choreography. Since our method is based on a probabilistic distribution modeled by normalizing flows, different yet plausible dance motions can be sampled repeatedly from the distribution. We compare our method with several related works that are able to generate different dance motions given the same input music.

4.1 Dataset

We use a public music-to-dance dataset AIST++ [43], which contains 1,408 motion sequences of 10 dance genres. For training, we get 14,645 clips of dance motions and each clip contains 40 frames. We build the test set as suggested by the authors of AIST++ [43], as follows. Firstly, we select one music piece from each of the 10 genres. Then, we randomly select two dancers, each of which performs two different dance motions for the music piece. Finally, we get in total 40 unique choreographies paired with music beats in the test set.

4.2 Implementation Details

Our method of transition generation for choreography is implemented in PyTorch [50] and could also be implemented on other deep learning platforms such as Jittor [51]. In our framework, the number of flow steps K is set to 16. In each flow step, the autoregressive model consists of two LSTM layers, and the hidden channel is set as 512. The batch size is set to 100 during training. The learning rate is set to 0.0001 with exponential decay rates (β_1, β_2) = (0.9,0.999). The dropout rate of past-context motions is set to 0.7 [19]. To generate a dance motion at frame t, our model takes as input τ = 10 previous frames of dance motions and condition signals, including music features between the frames ($t - \tau$ to t), the target key pose at frame γ, and a time embedding at frame t. γ is set as 30 in the training stage and it can be set to arbitrary values by users in the test stage. All the models are trained on an Nvidia RTX 2080Ti GPU.

4.3 Comparisons

In the following, we compare our model to several baselines and the state-of-the-art (SOTA) methods that are able to generate diverse dance motions given the same input music.

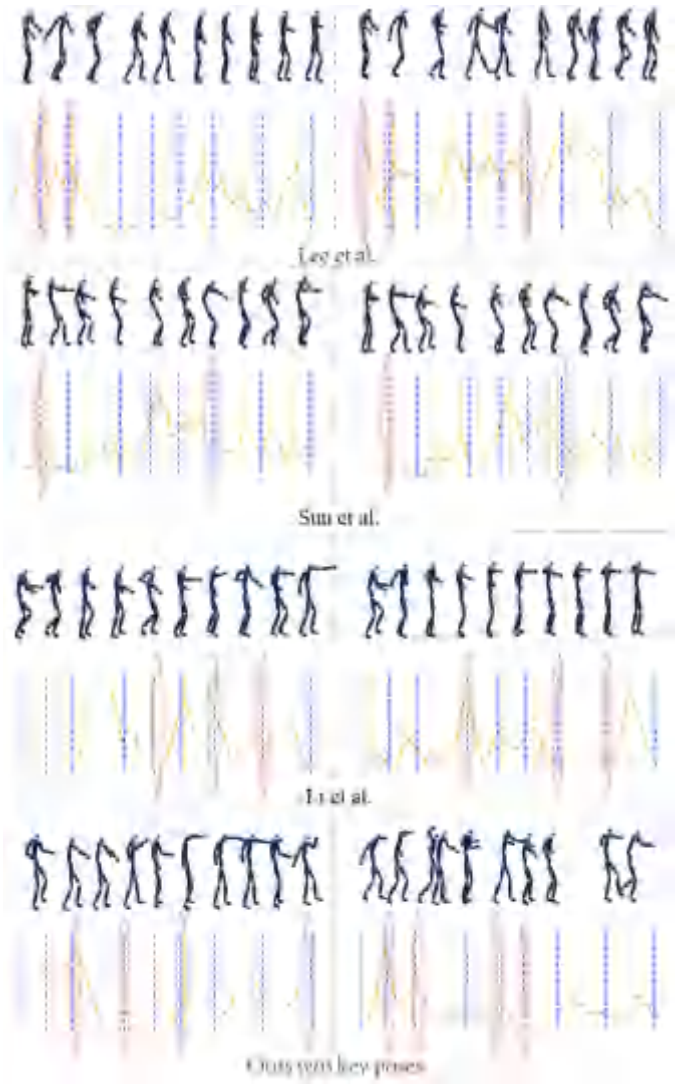


Fig. 4: Qualitative comparisons of our method and the SOTA works. The odd rows show two dance sequences driven by the same input music with the corresponding beat-matching figures below them. In each beat-matching figure, kinematic velocity lines (yellow curve) are plotted with kinematic beats (yellow dashes) and musical beats (blue dashes). The red elliptics show the matched beats. Our baseline method (Ours w/o key poses) generates dance motions with high realism, variability and beat-matching performance. For clarity, the generated frames are shown with an interval of 30. Please refer to the supplemental video for the full animation results with the corresponding input music.

4.3.1 Adversarial Learning Scheme

For music-driven dance generation, we compare our model with the SOTA methods [4], [7], [43] based on an adversarial learning scheme to generate different dance motions with the same input. These methods firstly construct a motion generator to generate dance motions from an input piece of music, and then apply discriminators to distinguish the generated motions from the training dance samples. The method of Lee et al. [7] was originally developed for generating 2D dance motions while the methods of Sun et al. [4] and Li et al. [43] were trained for 3D dance motions generation. For a fair comparison, we train and evaluate their models using the same 3D dance dataset, which is used in the approach by Li et al. [43].

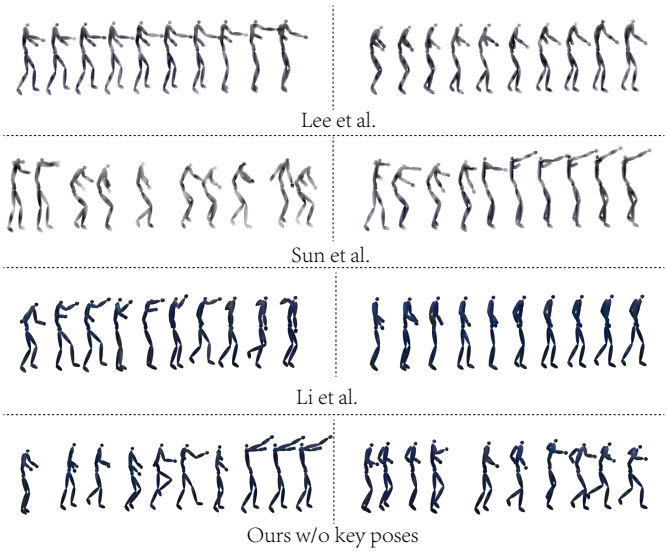


Fig. 5: Qualitative comparisons of our method and the SOTA works. It shows two dance sequences driven by different input music segments. Our baseline method (Ours w/o key poses) generates dance motions with high realism performance. For clarity, the generated frames are shown with an interval of 30. Please refer to the supplemental video for the full animation results with the corresponding input musics.

4.3.2 Ours w/o Key Poses

The related works [4], [7], [43] generate dance motions guided by music only. As a baseline, we train our model in terms of the input music only, to evaluate the superiority of using the key poses and compare the baseline with the related works.

4.3.3 Ours w/o Time Embedding

As illustrated in Section Method, we add a time embedding to make the generated movements to arrive at the target poses properly, thus achieving a robust transition of varying lengths between key poses. We train our model without the time embedding to evaluate its importance.

4.4 Qualitative Evaluation

We firstly evaluate the realism and beat concordance of the compared methods. Figure 4 shows the dance motions generated by our approach and the SOTA methods. For a fair comparison, we use our baseline method (Ours w/o key poses) that generates dance motions conditioned only on music to compare with the SOTA methods. It can be seen that our results are more realistic than the existing methods. Specifically, the results generated by Lee et al. [7] and Sun et al. [4] involve little movement, which is unreal in dance composing situations. The motions generated by Li et al. [43] also dance to freezing gestures after a few seconds. Furthermore, the even rows in Figure 4 show that the number of the matched beats (red ellipces) of our method is higher than that of the other methods. The results illustrate that our method outperforms the other methods in beat alignment. What's more, the odd rows in Figure 4 show that our method generates more variable dance motions given the same input music than the SOTA methods. In

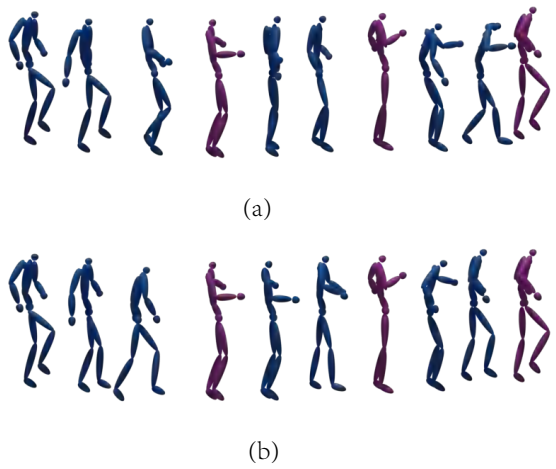


Fig. 6: Variability evaluation. (a) and (b) show different dance motions (in blue) generated by our full model conditioned on the same input music segment and key poses (in pink). The full video can be found in the supplemental video.

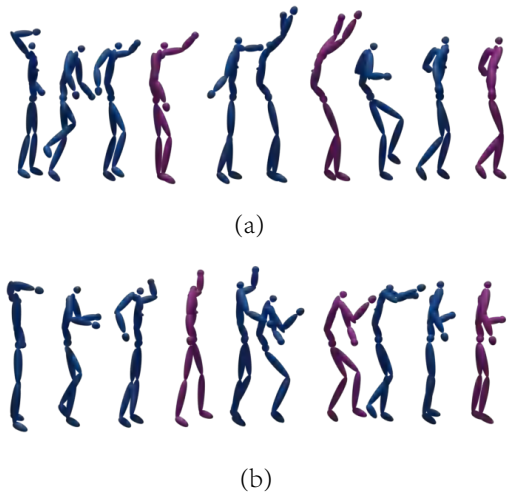


Fig. 7: Diversity evaluation. (a) and (b) show different dance motions (in blue) generated by our full model conditioned on the same input music segment but different key poses (in pink).

fact, the method of Sun et al. [4] generates the same two dance sequences when given the same music beats.

Furthermore, Figure 5 shows the diversity comparison results between our baseline method (Ours w/o key poses) and SOTA works [4], [7], [43]. Given different input music pieces, the methods of Lee et al. [7] and Sun et al. [4] generate similar dance movements in two sequences or low variable poses in one complete sequence, while our method generates diverse results, which leads higher realism performance. Specifically, the dance motions generated by Lee et al. [7] tend to freeze after several frames. This is because the method of Lee et al. [7] encodes dance motions with the gated recurrent unit (GRU), which has the error accumulation problem [43]. The method of Sun et al. [4] generates motion clips and combines them as a full sequence, so it leads to the similarity

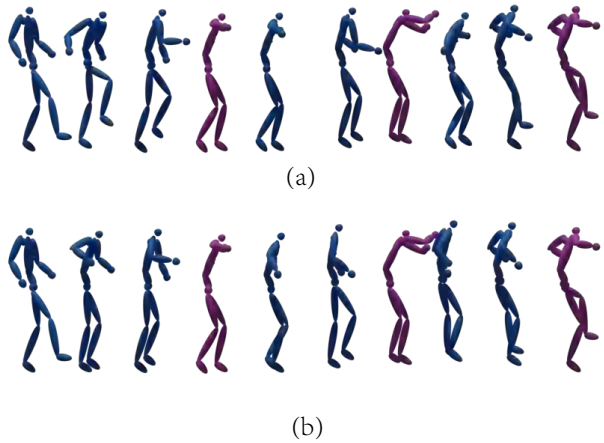


Fig. 8: Diversity evaluation. (a) and (b) show different dance motions (in blue) generated by our full model conditioned on different input music segments but the same key poses (in pink).

of motions when given a music piece which has similar recurrent beats. As a conclusion, our model outperforms the SOTA works in generating realistic, temporal consistent and diverse dance motions.

Figure 6 shows different results generated by our full model given the same input music beats and key poses. It reveals the variability of our full method. Such different yet plausible dance motions can provide more inspirations for 3D animators.

As shown in Figure 7, our method is able to generate realistic and diverse dance sequences given the same input music segments but different key poses, thus facilitating 3D animators to edit dance motions and produce diverse dance motions for different scenarios. Furthermore, we show more experimental results to reveal the diversity of dance motions generated by our full model. Different outputs with different input music beats but the same key poses can be found in Figure 8. Figure 9 gives different outputs with different input music beats and key poses. Given the different inputs, our model generates different yet plausible dance motions.

In Figure 10, we evaluate how the generated motions at the keyframes (i.e., the generated key poses) respect the input key poses (GT), and compare our method with two baselines, i.e., our method without key pose loss (Ours w/o key pose loss) and our method without time embedding (Ours w/o time embedding). It can be seen that the generated key poses of our full method has the highest similarity to the input key poses, while the other settings cannot generate proper poses. The results illustrate the superiority of our full model in the robust transition generation by introducing the key pose loss and the time embedding.

Furthermore, we evaluate the variability of the results of Harvey et al. [36] by generating different results giving the same inputs. The qualitative results are shown in Figure 11. It can be concluded that our transition results have a better performance on motion richness, which benefits choreography applications.

We have also experimented our method on a different dataset, which has been used in the work of Tang et al. [42]. The skeleton architecture of this dataset is different from that of AIST++ [43], so we retrain our model on the dataset. As shown in Figure 12 and the supplementary video, our method is able to generate realistic dance motions that correspond to the input music.

TABLE 1: Quantitative Evaluation Results.

Method	Beat Hit Rate (↑)	Beats Coverage (↑)	FID (↓)	KPD (↓)	Variability (↑)	Diversity(↑)
Real Dance	48.2%	39.4%	3.71	-	-	-
Lee et al. [7]	42.4%	31.3%	28.75	-	12.53 ± 0.87	19.76 ± 0.97
Sun et al. [4]	42.9%	33.5%	26.47	-	-	16.82 ± 1.12
Li et al. [43]	43.8%	37.5%	15.62	-	31.45 ± 0.98	39.45 ± 1.21
Ours w/o key poses	**44.7%**	**39.6%**	13.42	-	$\mathbf{38.39 \pm 1.30}$	42.75 ± 1.14
Ours w/o key pose loss	44.4%	39.5%	13.92	10.90	31.70 ± 0.91	45.08 ± 1.23
Ours w/o time embedding	44.4%	39.4%	12.08	8.96	29.74 ± 1.18	41.24 ± 1.01
Ours	44.5%	39.4%	**10.42**	**4.53**	29.31 ± 1.75	$\mathbf{48.65 \pm 1.92}$

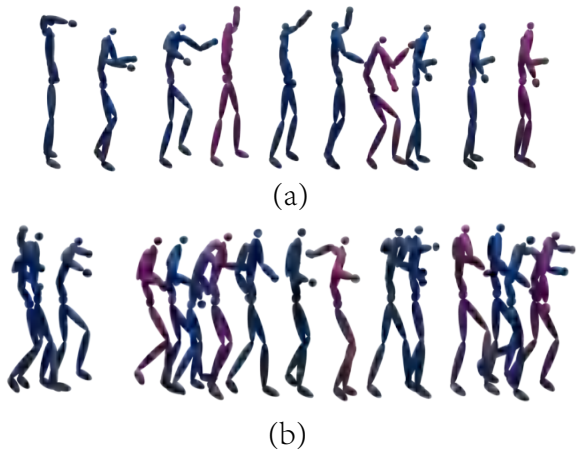


Fig. 9: Illustration of the diversity in our generated dance motions. Different yet plausible motions are generated by our method, given different input music segments and key poses (in pink). Result (a) shows short-time generation, while result (b) shows long-time generation. The full video and more variable results can be found in the supplemental video.

Fig. 11: Variability comparison with an existing transition method.

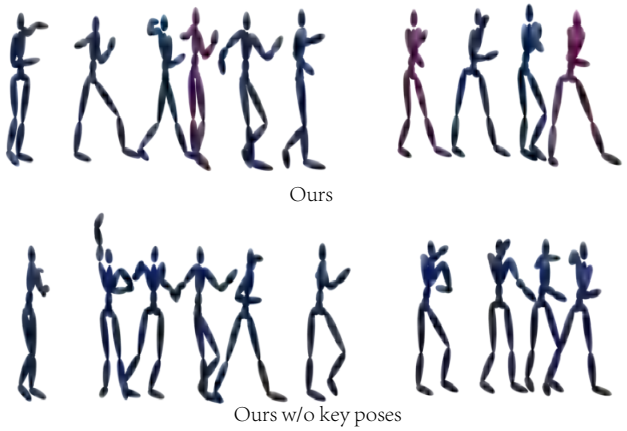


Fig. 12: Our results on a different dataset of Tang et al. [42].

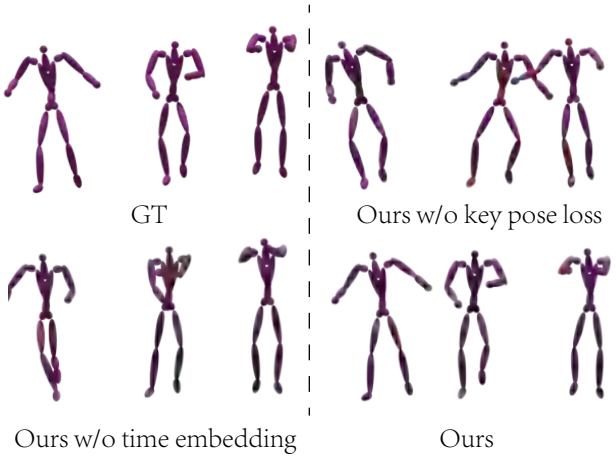


Fig. 10: Comparison of our full method to two baselines to generate motions constrained by key poses. We show three ground-truth poses and the corresponding results generated by the three compared methods. It reveals that our method achieves a more accurate target frame generation.

4.5 Quantitative Evaluation

4.5.1 *Beat Hit Rate and Coverage*

We evaluate the concordance between our generated dance motions and the input music by calculating the beat hit rate. Firstly, we calculate a kinematic velocity curve that measures the distance of the corresponding joint positions between neighboring frames, and extract the kinematic beats by finding the local maxima from the kinematic velocity curve. The music beats are extracted by the public Librosa toolbox [48]. Then, we gather the total number of kinematic beats B_k, the total number of musical beats B_m, and the number of well-aligned kinematic beats (the gap between kinematic and music beats is not more than 2 frames) with the music beats B_a. Finally, the beat hit rate is defined as B_a/B_k, which describes the ratio of the matched kinematic beats to the total kinematic beats. The beats coverage B_k/B_m measures the ratio of kinematic beats to musical beats.

As shown in Table 1, the beat hit rate generated by our method (44.5%) is closer to that of the real dance motions than those generated by the SOTA works [4], [7]. It indicates that our method has a better performance in generating dance motions that match the beats of the input music. Our baseline model (Ours w/o

TABLE 2: Transition Generation Results.

Method	L2P(30)	L2P(60)	L2P(90)	AVG L2P
Interpolation	1.98	4.07	7.84	4.63
Harvey et al. [36]	**1.34**	2.75	5.63	3.24
Ours	1.53	**2.68**	**4.12**	**2.77**

key poses) has a slightly higher beat hit rate (44.7%) than our full model using key poses as constraints. This is mainly because the input key poses are set without considering the music beats, thus resulting in the degradation of the beat matching performance. For professional and experienced choreographers, the key poses may be set more properly. In addition, we would develop a new tool for analyzing the music beats to set key poses automatically for ordinary users.

For the beats coverage, our approaches have similar scores with real dance, revealing that our results align the musical beats better than those by the SOTA methods. Note that since the beats coverage depends on the total number of kinematic beats, the shaking motions may cause more kinematic beats, thus making the beats coverage higher than smooth motions.

4.5.2 FID

Fréchet inception distance (FID) [52] evaluates the distance between the distributions of real and generated motions to reveal the visual quality. Following [7], we trained an encoder for dance motions as a feature extractor.

The FID scores are shown in Table 1. Our full model outperforms the SOTA works [4], [7], [43] and our baseline models. Our full model generates more accurate key poses, which leads a closer distance to real dance motions when compared to our baseline models. The FID score of "Ours w/o key poses" model has a better performance over the methods of Lee et al. [7], Sun et al. [4], and Li et al. [43], indicating that our normalizing flow based model is able to generate dance motions with high visual quality.

4.5.3 Key Pose Distance (KPD)

To measure the accuracy of the generated poses at the keyframes, we perform a quantitative evaluation on the distance between the input and generated poses at the keyframes. Given all generated dance frames, we gather the ground-truth poses P_t in the test sets and the corresponding frames P_g in the generated motions. Then, we compute the average squared distance of joint positions between P_t and P_g to measure the key pose distance. Since the SOTA methods do not take key poses as input, we compare our model with two baseline methods, our method without introducing the key pose loss (Ours w/o key pose loss) and our method without using the time embedding (Ours w/o time embedding). As shown in Table 1, our full model outperforms the baselines in terms of the key pose distance, indicating that it is necessary to impose the key pose loss and time embedding into our model for a robust transition generation. Moreover, we conduct experiments with a forward time embedding, from which the key pose distance (5.48) is higher than that of our full model. The result shows that the time embedding evolving backward performs better for a robust transition generation.

4.5.4 Variability and Diversity

We use variability and diversity as two metrics for further evaluation, as suggested by [7]. In more details, the variability measures the ability of a method to generate different dance motions conditioned on the same input. The diversity describes the performance in producing diverse motions given different inputs. Mathematically, the variability is calculated by the average squared distance of motion features extracted by the same extractor used in the FID measures between all combinations of 15 dances randomly generated from the same input. The diversity is calculated by the average squared distance of motion features between all combinations of generated dance motions from different input sequences in the test set.

As shown in Table 1, the diversity scores of our normalizing flow base models (i.e., Ours w/o key poses, Ours w/o key pose loss, Ours w/o time embedding, and our full approach) are similar and all higher than those of the SOTA works. The results reveal the superiority of normalizing flows in modeling the exact distribution of real dance motions. This is similarly reflected by the variability scores. Note that our baseline method (Ours w/o key poses) has the highest variability score but a lower diversity score when compared to the other two keyframe-based baseline methods (Ours w/o time embedding and our full approach). This is because the key poses providing extra signals lead to more similar movements when giving the same signals (for variability) and more different movements in the case of different signals (for diversity).

4.5.5 L2 Distances of Global Positions (L2P)

To measure the ability of transition generation, we use L2P evaluation as suggested by [36]:

$$L2P = \frac{1}{|D|}\frac{1}{T}\sum_{S\in D}\sum_{t=0}^{T-1}\|\hat{p}_t^S - p_t^S\|_2, \tag{10}$$

where S denotes a sequence in the test set D, and t is the frame index of a transition sequence with a total length T. $\hat{p}_t^S$ denotes the global position of the generated pose at frame t of sequence S, and p denotes ground-truth poses.

For a fair comparison, we trained the model of Harvey et al. [36] with the same database AIST++. During training, the transition interval is set as 30 frames, and it takes past $\tau = 10$ frames as seed poses to generate a new transition movement, and this parameter is the same as our model. We use short-to-long interval settings in the test stage to show the scalability of different models. As shown in Table 2, the approach of Harvey et al. [36] has a better performance when the interval is set as 30 frames, which was used in the training stage. This is mainly because our probabilistic model samples randomly in the latent space when generating new movements, leading to a few different movements compared to the ground truth. Ours outperforms the interpolation method, in which we linearly interpolate the root position and spherically interpolate the rotation between seed poses and target poses. Our method also beats the model of Harvey et al. [36] under long interval settings. Specifically, the results generated by the approach of Harvey et al. [36] seem to arrive at a neighboring pose of a target pose within a short period and change slowly afterwards. It reveals the better scalability and flexibility of our method.

4.5.6 Perceptive Study

We further conducted a perceptive study to evaluate the realism, beat-alignment, richness, and style consistency of dance genre performance [10] of our method, in comparison with the approaches of Lee et al. [7], Sun et al. [4], and Li et al. [43]. The richness shows the variety of poses and the style consistency illustrates the matching of dance and music styles. For a fair comparison, we

TABLE 3: Perceptive Study Results.

Method	Realism	Beat-alignment	Richness	Style consistency
Lee et al. [7]	3.12	2.98	3.25	3.32
Sun et al. [4]	2.86	2.95	2.62	3.13
Li et al. [43]	2.25	2.15	2.15	1.80
Ours w/o key poses	**1.77**	**1.91**	**1.98**	**1.75**

TABLE 4: Beat Hit Rate Results.

Setting	Beat Hit Rate
Equal interval (100)	44.5%
Beat interval	45.5%

used "ours w/o key poses" method in this study. There were in total 30 participants, and most of them were science or engineering students in a local university, and ten of them had experience in dance and music. We asked the participants to sort the four randomly presented dance motions in each group of the results generated by the compared four methods (20 groups in total). Specifically, they were asked to give the scores from 1 (the best) to 4 (the worst) according to the realism and richness of dance motions, beat-alignment, and dance genre consistency performance, separately. As shown in Table 3, our method outperforms the other methods in realism and richness of movements, beat-alignment, and style consistency.

5 Discussions

5.1 Beat Hit Rate

As shown in Table 1, our baseline method without using key poses (Ours w/o key poses) has a better performance in beat matching than our full method. The main reason is that the key poses in the evaluation are set with an equal interval, and thus might not be in accordance with the musical beats. Below we show more experimental results to reveal the influence of the key poses in beat matching.

Table 4 shows the beat hit rate of two different settings for key poses. In the equal interval setting, the key poses are set per 100 frames without regarding of beat matching. In the beat interval setting, we extract the music beats by using the public toolbox Librosa [48], and then randomly select three key poses at the beat frames to generate dance sequences. The beat interval setting has a higher beat hit rate (45.5%) than the equal interval setting (44.5%) and the baseline method (Ours w/o key poses), whose beat hit rate is 44.7% (Table 1 in our paper). The experimental results illustrate that a proper setting of key poses is beneficial for generating music-matching dance motions.

5.2 Key-frame Matching and Temporal Coherency

During the transition generation stage, the coherency between key-frame matching and temporal constraint could limit the interpolation performance. Specifically, when given a short interval (e.g., 30 frames) and target poses with a large-scale variation (e.g., from standing to lying), our model failed to generate satisfying results, where the movement was like a squat gesture. When increasing the timescale of two target poses, the transition will arrive at a correct pose. However, if the variation between target poses is not obvious, the generated sequences might seem freezing. The results illustrate that a proper setting of interval and target poses is necessary for better performance.

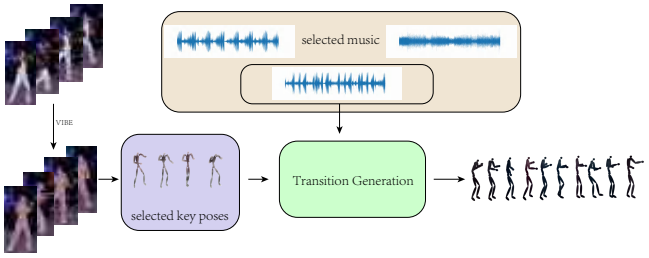


Fig. 13: Choreography pipeline for users.

5.3 Applications

We further evaluate our method for application to facilitate normal users. In more details, we conduct experiments to generate dance motions constrained by the key poses, which are easily selected from wild photos. As shown in Figure 13, users firstly gather several human dancing photos, and then reconstruct their SMPL [53] poses using the publicly available project VIBE [54]. Then, our model is used to generate a full dance sequence by using the key poses from these reconstructed results and a piece of selected music. As shown in Figure 13 and the supplementary video, the generated dance motions are realistic and in accordance with the music beats.

6 Conclusions

We have presented a normalizing flow based model for music-driven dance generation constrained by a sparse set of key poses. Our model is probabilistic, and describes the distribution of real dance poses. Thus, it enables the generation of rich natural variations of dance motions. We build the model to generate dance motions autoregressively, and thus it is able to generate arbitrarily-long dance motions. Moreover, we introduce a time embedding at each timestep for a robust transition generation between keyframes.

Although our method successfully generates a diverse set of dance sequences aligned with music beats and target poses, it has several limitations. First, our method cannot generate dance styles not covered in the training dataset. This might be improved by expanding the datasets with additional styles, like Classical, Jazz, etc. Second, the generated poses in key frames might still have small differences compared to the given key poses. The gaps particularly exist in rotation angles of end effectors (feet and hands). This issue might addressed by a further refinement in the feet and hand details.

For future research, we are interested in developing an interactive dance composing system, where users can easily input desired music pieces and target poses to drive automatic dance generation. In more details, the target poses can be chosen from a database or reconstructed from user-specified human dancing photos by a reconstruction function integrated in the system. Additionally, synthesizing dance motions conditioned on human emotion is another interesting research direction.

REFERENCES

[1] W. Zhuang, Y. Wang, J. P. Robinson, C. Wang, M. Shao, Y. Fu, and S. Xia, "Towards 3d dance motion synthesis and control." *CoRR*, 2020.

[2] C. Kang, Z. Tan, J. Lei, S.-H. Zhang, Y.-C. Guo, W. Zhang, and S.-M. Hu, "Choreomaster : Choreography-oriented music-driven dance synthesis," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, 2021.

[3] N. Yalta, S. Watanabe, K. Nakadai, and T. Ogata, "Weakly-supervised deep recurrent neural networks for basic dance step generation," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[4] G. Sun, Y. Wong, Z. Cheng, M. S. Kankanhalli, W. Geng, and X. Li, "Deepdance: music-to-dance motion choreography with adversarial learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 497–509, 2020.

[5] Z. Ye, H. Wu, J. Jia, Y. Bu, W. Chen, F. Meng, and Y. Wang, "Choreonet: Towards music to dance synthesis with choreographic action unit," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 744–752.

[6] J. P. Ferreira, T. M. Coutinho, T. L. Gomes, J. F. Neto, R. Azevedo, R. Martins, and E. R. Nascimento, "Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio," *Computers & Graphics*, vol. 94, pp. 11–21, 2021.

[7] H.-Y. Lee, X. Yang, M.-Y. Liu, T.-C. Wang, Y.-D. Lu, M.-H. Yang, and J. Kautz, "Dancing to music." in *NeurIPS*, 2019, pp. 3581–3591.

[8] X. Ren, H. Li, Z. Huang, and Q. Chen, "Self-supervised dance video synthesis conditioned on music." in *ACM Multimedia*, 2020, pp. 46–54.

[9] W. Zhuang, C. Wang, J. Chai, Y. Wang, M. Shao, and S. Xia, "Music2dance: Dancenet for music-driven dance generation," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 18, no. 2, 2022.

[10] A. Aristidou, A. Yiannakidis, K. Aberman, D. Cohen-Or, A. Shamir, and Y. Chrysanthou, "Rhythm is a dancer: Music-driven motion synthesis with global structure," *IEEE Transactions on Visualization and Computer Graphics*, 2022.

[11] R. Neagle, K. Ng, and R. Ruddle, "Developing a virtual ballet dancer to visualise choreography," in *Proceedings of the Symposium on Language, Speech and Gesture for Expressive Characters*, 2004.

[12] E. R. Moghaddam, J. Sadeghi, and F. F. Samavati, "Sketch-based dance choreography," in *International Conference on Cyberworlds*, 2014, pp. 253–260.

[13] M. Izani, A. R. Eshaq, N. Zainuddin, and A. Razak, "A study on practical approach of using motion capture and keyframe animation techniques," in *Proceedings. Eighth International Conference on Information Visualisation*, 2004, pp. 849–852.

[14] T.-Y. Mou, "Keyframe or motion capture? reflections on education of character animation," *EURASIA Journal of Mathematics, Science and Technology Education*, vol. 14, no. 12, 2018.

[15] B. Choi, R. B. i Ribera, J. P. Lewis, Y. Seol, S. Hong, H. Eom, S. Jung, and J. Noh, "Sketchimo: sketch-based motion editing for articulated characters," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[16] L. Ciccone, M. Guay, M. Nitti, and R. W. Sumner, "Authoring motion cycles," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2017, pp. 1–9.

[17] L. Ciccone, C. Öztireli, and R. W. Sumner, "Tangent-space optimization for interactive animation control," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–10, 2019.

[18] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions." in *NeurIPS*, 2018, pp. 10 236–10 245.

[19] G. E. Henter, S. Alexanderson, and J. Beskow, "Moglow: Probabilistic and controllable motion synthesis using normalising flows," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–14, 2020.

[20] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics." in *ICCV*. IEEE Computer Society, 2015, pp. 4346–4354.

[21] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs." in *CVPR*. IEEE Computer Society, 2016, pp. 5308–5317.

[22] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3d human motion modelling." in *ICCV*. IEEE, 2019, pp. 7143–7152.

[23] X. Du, R. Vasudevan, and M. Johnson-Roberson, "Bio-lstm: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction." *IEEE Robotics Autom. Lett.*, vol. 4, no. 2, pp. 1501–1508, 2019.

[24] A. H. Ruiz, J. Gall, and F. Moreno, "Human motion prediction via spatio-temporal inpainting." in *ICCV*. IEEE, 2019, pp. 7133–7142.

[25] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study." in *NeurIPS*, 2018, pp. 698–707.

[26] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *ACM SIGGRAPH 2008 classes*, 2008, pp. 1–10.

[27] A. Treuille, Y. Lee, and Z. Popović, "Near-optimal character animation with continuous control," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.

[28] A. Safonova, J. K. Hodgins, and N. S. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Trans. Graph.*, vol. 23, no. 3, p. 514–521, 2004.

[29] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing." *ACM Trans. Graph.*, vol. 35, no. 4, pp. 138:1–138:11, 2016.

[30] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders." in *SIGGRAPH Asia Technical Briefs*. ACM, 2015, pp. 18:1–18:4.

[31] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Trans. Graph.*, vol. 37, no. 4, 2018.

[32] H. Y. Ling, F. Zinno, G. Cheng, and M. Van De Panne, "Character controllers using motion vaes," *ACM Trans. Graph.*, 2020.

[33] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows." in *ICML*, vol. 37, 2015, pp. 1530–1538.

[34] Y.-H. Wen, Z. Yang, H. Fu, L. Gao, Y. Sun, and Y.-J. Liu, "Autoregressive stylized motion synthesis with generative flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 13 612–13 621.

[35] S. Xia, L. Gao, Y. Lai, M. Yuan, and J. Chai, "A survey on human performance capture and animation," *J. Comput. Sci. Technol.*, vol. 32, no. 3, pp. 536–554, 2017. [Online]. Available: https://doi.org/10.1007/s11390-017-1742-y

[36] F. G. Harvey, M. Yurick, D. Nowrouzezahrai, and C. Pal, "Robust motion in-betweening," *ACM Trans. Graph.*, vol. 39, no. 4, 2020.

[37] X. Zhang and M. van de Panne, "Data-driven autocompletion for keyframe animation," in *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, 2018, pp. 1–11.

[38] F. G. Harvey and C. Pal, "Recurrent transition networks for character locomotion," in *SIGGRAPH Asia 2018 Technical Briefs*, 2018, pp. 1–4.

[39] Y.-L. Qiao, Y.-K. Lai, H. Fu, and L. Gao, "Synthesizing mesh deformation sequences with bidirectional lstm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 4, pp. 1906–1916, 2022.

[40] Y. CHAI, Y. WENG, L. WANG, and K. ZHOU, "Speech-driven facial animation with spectral gathering and temporal attention," *Frontiers of Computer Science*, vol. 16, no. 3, p. 163703, 2022.

[41] O. Alemi, J. Françoise, and P. Pasquier, "Groovenet: Real-time music-driven dance movement generation using artificial neural networks," *networks*, vol. 8, no. 17, p. 26, 2017.

[42] T. Tang, J. Jia, and H. Mao, "Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1598–1606.

[43] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "Learn to dance with aist++: Music conditioned 3d dance generation," *arXiv*, 2021.

[44] G. Valle-Pérez, G. E. Henter, J. Beskow, A. Holzapfel, P.-Y. Oudeyer, and S. Alexanderson, "Transflower: Probabilistic autoregressive dance generation with multimodal attention," *ACM Trans. Graph.*, vol. 40, no. 6, 2021.

[45] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[46] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999.

[47] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *Journal of Graphics Tools*, vol. 3, no. 3, pp. 29–48, 1998.

[48] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015, pp. 18–25.

[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6000–6010.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp.

8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[51] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 222103, pp. 1–21, 2020.

[52] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[53] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics*, vol. 34, no. 6, pp. 248:1–248:16, 2015.

[54] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zhipeng Yang is a master student at Institute of Computing Technology, University of Chinese Academy of Sciences. He received the bachelor's degree from University of Chinese Academy of Sciences (UCAS). His research interests include computer vision, artificial intelligence, and human motion analysis.

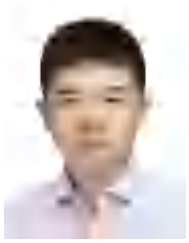
Yu-Hui Wen received the bachelor's degree from Harbin Institute of Technology (HIT), and the Ph.D. degree in computer science and technology from University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2020. She is currently a postdoc at Tsinghua University. Her research interests include machine learning, artificial intelligence, and virtual reality.

Shu-Yu Chen received the PHD degree in computer science and technology from University of Chinese Academy of Sciences. She is currently working as a research associate in Institute of Computing Technology, Chinese Academy of Sciences. Her research interests include computer graphics.

Xiao Liu is a researcher with Tomorrow Advancing Life Education Group (TAL). He received the PhD degree in computer science from Zhejiang University in 2015 and then worked for Baidu from 2015 to 2019. His research interests is the applied AI such as intelligent multimedia processing, computer vision and learning system. His research results have expounded in 40+ publications at journals and conferences such as IEEE T-IP, T-NNLS, CVPR, ICCV, ECCV, AAAI and MM. As a key team member, he achieved the best performance in various competitions, such as the ActivityNet challenges, NTIRE super resolution challenge, EmotioNet facial expression recognition challenge, etc.

Yuan Gao recived his M.S. in Computer Science from Beihang University in 2015. He is currently a Research Engineer at Tomorrow Advancing Life(TAL) education group. His main research interests are object detection and segmentation.

Yong-Jin Liu is a Professor with the Department of Computer Science and Technology, Tsinghua University, China. He received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. His research interests include computational geometry, computer graphics and computer vision. He is a senior member of the IEEE.



Lin Gao received his PhD degree in computer science from Tsinghua University. He is currently an Associate Professor at the Institute of Computing Technology, Chinese Academy of Sciences. He has been awarded the Newton Advanced Fellowship from the Royal Society and the AG young researcher award. His research interests include computer graphics and geometric processing.

Hongbo FU received a BS degree in information sciences from Peking University, China, in 2002 and a PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. He is a Full Professor at the School of Creative Media, City University of Hong Kong. His primary research interests fall in the fields of computer graphics and human computer interaction. He has served as an Associate Editor of The Visual Computer, Computers & Graphics, and Computer Graphics Forum.