Controllable 3D Dance Generation Using Diffusion-Based Transformer U-Net

Puyuan Guo, Tuo Hao, Wenxin Fu, Yingming Gao, Ya Li*

Beijing University of Posts and Telecommunications, Beijing, China
{guopy, haotuo_absolute, fuwenxin2003, yingming.gao, yli01}@bupt.edu.cn

Abstract

Recently, dance generation has attracted increasing interest. In particular, the success of diffusion models in image generation has led to the emergence of dance generation systems based on the diffusion framework. However, these systems lack controllability, which limits their practical applications. In this paper, we propose a controllable dance generation method based on the diffusion model, which can generate 3D dance motions controlled by 2D keypoint sequences. Specifically, we design a transformer-based U-Net model to predict actual motions. Then, we fix the parameters of the U-Net model and train an additional control network, enabling the generated motions to be controlled by 2D keypoints. We conduct extensive experiments and compared our method with existing works on the widely used AIST++ dataset, demonstrating that our approach has certain advantages and controllability. Moreover, we also test our model on in-the-wild videos and find that it is capable of generating dance movements similar to the motions in the videos as well.

Introduction

The 3D dance generation task refers to generating 3D dance movements that match the given music, which has broad application scenarios such as film production, video games, and virtual reality. By utilizing dance generation models, we can efficiently obtain 3D motion data, thereby saving a significant amount of time.

The researches on dance generation models develop along with the emergence of generative models, evolving from long short-term memory (LSTM) models (Tang, Jia, and Mao 2018; Tang, Mao, and Jia 2018; Ye et al. 2020; Chen et al. 2021; Zhuang et al. 2022) to Transformers (Li et al. 2020, 2021, 2022a). After that, generative adversarial networks (GAN) are also applied to this task (Lee et al. 2019; Sun et al. 2021; Ferreira et al. 2021). Recently, diffusion models (Ho, Jain, and Abbeel 2020), which are very popular in the field of image generation (Dhariwal and Nichol 2021; Ramesh et al. 2022; Saharia et al. 2022; Rombach et al. 2022), and generative pre-trained transformers (GPT), used in large language models (Radford et al. 2019; Devlin et al. 2019), have provided new approaches for the dance movement generation task (Li et al. 2022b; Tseng, Castellon, and Liu 2023; Zhuang et al. 2023; Li et al. 2024). Although these models can generate high-quality dance movements, they still lack the ability to control the generated results. However, this control capability is crucial for practical applications. For the GPT models, Huang et al. (2024) designed a pose codebook, where each code corresponds to the semantic actions of different body parts. By replacing codes during the generation process, they achieved control over the generated movements. However, for diffusion models, although some works (Shafir et al. 2023a; Dai et al. 2024) have contributed to generating motions along specific trajectories, researches on controlling body movements are still lacking. Therefore, we propose a controllable dance generation framework based on diffusion models, which can guide the generation of body movements according to 2D keypoints.

Inspired by ControlNet (Zhang, Rao, and Agrawala 2023), we first train a dance movement generation U-Net using a larger-scale dataset with music as the only condition. This U-Net is composed of Transformer decoder layers. We then use this model as a base, fix its parameters, and duplicate its "downsampling" layers to create a control network, using 2D keypoints as inputs and music features as conditions. The outputs of each layer in the control network serve as a control signal for the corresponding "upsampling" layers, guiding the final generated results to resemble the 2D keypoints. This approach enables the control network to acquire the ability to affect the model's results using control signals. During the inference process, inputting 2D keypoints as control signals and musics as the conditions allows for the generation of dance movements that resemble the keypoint movements.

In this paper, our contributions are as follows:

- We propose a novel Transformer U-Net model for 3D dance movement generation, which not only generates high-quality dance motions but also adopts to the controllable network architecture we subsequently introduce.
- We propose a controllable dance generation framework based on diffusion models. By combining with the aforementioned U-Net model, we can use 2D keypoints as control signals to guide the final generated results.

*Corresponding author.

Related Works

As previously mentioned, researches on the dance movement generation task have been influenced by the development of generative models. Initially, sequence generation employed recurrent neural networks (RNN) (Rumelhart, Hinton, and Williams 1986) such as LSTM (Hochreiter and Schmidhuber 1997) and gated recurrent unit (GRU) (Cho et al. 2014). These models were also used for dance movement generation. For example, Tang, Jia, and Mao (2018) trained LSTM models on music features and dance features separately, calculating a reconstruction loss for music features and a predicted loss for dance features to update the network. However, due to their recursive nature, RNNs need to predict future results based on previous ones, which not only leads to slow generation speed but also poses the risk of gradient explosion or vanishing during the training process. Subsequently, the introduction of the attention mechanism and Transformer models (Vaswani et al. 2017) attempted to address the issues, offering better performance in sequence generation. For instance, Li et al. (2021) designed an audio transformer and a motion transformer to map music features and dance features into a unified latent space. They then used a cross-modal transformer to learn the relationship between the two features, enabling the generation of dance movements based on musics.

Additionally, generative methods from other fields have also been applied to dance generation. Lee et al. (2019) constructed a framework for dance decomposition and composition. They designed a dance unit variational autoencoder (VAE) to decompose dance movements into reusable dance units and a music-to-movement GAN to reassemble these dance units into a complete dance conditioned on musics. Recently, diffusion models and GPTs have also been applied to dance generation, achieving notable results. Li et al. (2022b) used VQ-VAE to convert dance movements into discrete codes, and adopted GPT to generate code sequences. They then used the VQ-VAE again to decode these codes into movements. Tseng, Castellon, and Liu (2023) utilized a Transformer decoder to predict dance motions and employed diffusion models for training and inference. Due to the stability of diffusion models in training and their excellent generative capabilities, we choose to use these models as the foundation.

Although some existing diffusion-based works have successfully achieved control over human motion trajectories (Shafir et al. 2023b; Dai et al. 2024), these methods involved are relatively complex. For instance, Shafir et al. (2023b) fine-tuned the model with edited data to obtain models with single control effects. Then, by using a method similar to classifier-free guidance (Ho and Salimans 2022), they combined the results of these fine-tuned models through weighted summation, enabling different control effects. However, this approach requires separate training for different controls, and the data editing for fine-tuning is cumbersome. Therefore, we develop a controllable generation framework that can achieve different control effects on human actions using easily obtainable data without the need of multiple model training sessions.

Methodology

In this section, we initially provide an explanation of the data representations. We then introduce our proposed controllable dance generation framework. As depicted in Fig. 1, the training of this framework is divided into two stages. The first stage involves training our designed Transformer U-Net, while the second stage entails fixing the parameters of the U-Net and training an additional control network, thereby enabling the control of the generation outcomes through control signals.

Data Representations

We adopt the same settings as EDGE (Tseng, Castellon, and Liu 2023). For the music data, we use the features extracted from the 66th layer of Jukebox (Dhariwal et al. 2020; Castellon, Donahue, and Liang 2021) as the condition c. For the dance movements, we save the motion data in SMPL format (Loper et al. 2015). However, during the training process, we use six degrees of freedom (6-DOF) representation (Zhou et al. 2019) to convert the rotation angles and retain the position data, denoted as w. Additionally, in order to optimize foot movements, we also set a contact label b for four foot joints to indicate the probability of foot-ground contact. Therefore, the final dance data is the concatenation of b and w, which is denoted as $x = \{b, w\}$.

Diffusion Models

We employ guided diffusion (Dhariwal and Nichol 2021) as the framework for training and inference. However, unlike models commonly used for image generation, our model predicts the generation results rather than predicting noise. During the training phase, we obtain a sample x_0 from the dataset and uniformly sample a timestep t from $[0, T]$. According to the forward noising process defined in DDPM (Ho, Jain, and Abbeel 2020), we can get the noisy sample x_t at time t using the following formula:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{1}$$

where ϵ is a Gaussian noise, and $\bar{\alpha}_t \in (0, 1)$ are constants which follow a monotonically decreasing schedule. When $\bar{\alpha}_t$ approaches 0, we can approximate $x_T \sim \mathcal{N}(0, I)$.

Then the noisy sample x_t, the music condition c, and the timestep t are input into the prediction model x_θ to obtain the predicted result $\hat{x}_0$. $\hat{x}_0$ is compared with the ground truth x_0 to calculate the diffusion loss $\mathcal{L}_{\text{simple}}$, which is used to update the model parameters. Similar to the loss function in DDPM, the loss $\mathcal{L}_{\text{simple}}$ is expressed as follows:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t, x_0}\Big[\|x_0 - x_\theta(x_t, t, c)\|_2^2\Big], \tag{2}$$

We repeat the above process until the model converges. In addition to the reconstruction loss $\mathcal{L}_{\text{simple}}$ of the diffusion models, we also employ some auxiliary loss functions related to the physical aspects of the movements: the position loss $\mathcal{L}_{\text{pos}}$, which indicates the similarity of joint positions; the velocity loss $\mathcal{L}_{\text{vel}}$, which indicates the similarity of joint velocities; and the contact consistency loss $\mathcal{L}_{\text{foot}}$, which indicates the similarity of foot joint movements. They are defined as follows:

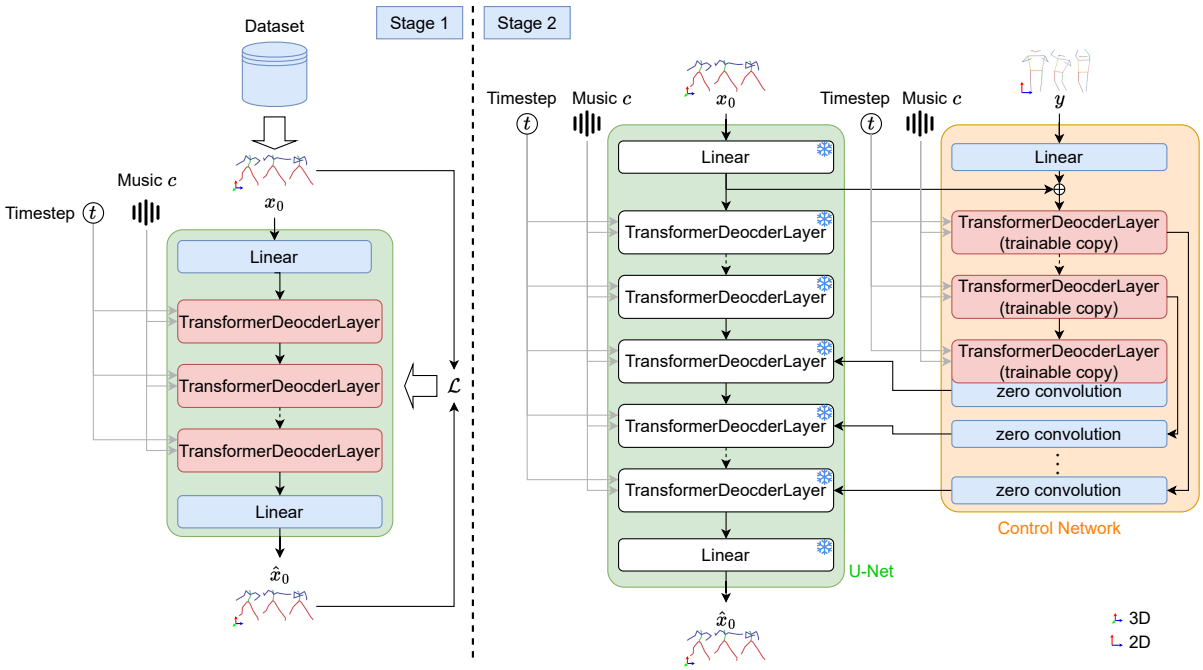


Figure 1: The training of the controllable dance generation framework. We only train the U-Net in Stage 1 and freeze its parameters in the next stage. During Stage 2, we train the control network to influence the intermediate representations of the U-Net. The snow icon denotes the parameters of the network are frozen. Similar to ControlNet (Zhang, Rao, and Agrawala 2023), the output of the zero-convolution layer is added to the corresponding U-Net layer output. To keep the diagram clear, some connections have been omitted.

$$\mathcal{L}_{\text{pos}} = \frac{1}{N}\sum_{i=1}^{N} \|FK(x^{(i)}) - FK(\hat{x}^{(i)})\|_2^2, \tag{3}$$

$$\mathcal{L}_{\text{vel}} = \frac{1}{N-1}\sum_{i=1}^{N-1} \|(x^{(i+1)} - x^{(i)}) - (\hat{x}^{(i+1)} - \hat{x}^{(i)})\|_2^2, \tag{4}$$

$$\mathcal{L}_{\text{foot}} = \frac{1}{N-1}\sum_{i=1}^{N-1} \|(FK(\hat{x}^{(i+1)}) - FK(\hat{x}^{(i)})) \cdot \hat{b}^{(i)}\|_2^2, \tag{5}$$

where $FK(\cdot)$ represents the forward kinematics operation, which can derive the corresponding joint position information from the rotation angle data. The superscript (i) denotes the motion data of the i-th frame. $\hat{b}$ represents the predicted probability of foot-ground contact. The total loss is the weighted sum of these losses:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\text{simple}} + \lambda_{\text{pos}} \cdot \mathcal{L}_{\text{pos}} + \lambda_{\text{vel}} \cdot \mathcal{L}_{\text{vel}} + \lambda_{\text{foot}} \cdot \mathcal{L}_{\text{foot}}, \tag{6}$$

where $\lambda, \lambda_{\text{pos}}, \lambda_{\text{vel}}, \lambda_{\text{foot}}$ are weights.

To achieve better generated results, we use classifier-free guidance (Ho and Salimans 2022) during training. We randomly replace the condition input with $c = \emptyset$ at a low probability. We also use guided inference which is expressed as the weighted sum of unconditionally and conditionally results:

$$\tilde{x} = w \cdot x_\theta(x_t, t, c) + (1 - w) \cdot x_\theta(x_t, t, \emptyset), \tag{7}$$

where w is the guidance weight, and we set $w = 2$ during inference.

Transformer U-Net

We design a U-Net network architecture that can not only generate high-quality dance movements but also serve as the foundation for the controllable generation framework we propose later. Given the excellent performance of Transformers in sequence modeling, we replace traditional convolution layers with Transformer decoder layers. Inspired by EDGE, we use their decoder layers to fuse the music features and dance features.

The overall structure is shown in Fig. 2. First, several decoder layers are connected in series to form the so-called "donwsampling" layers. Since we use Transformer decoder layers, the input and output dimensions remain unchanged, so there is no actual downsampling. Similarly, in the U-Net architecture, there are corresponding "upsampling" layers, but these do not alter the output dimensions either. We use one decoder layer as the intermediate layer. Because the structure of the decoder layers in U-Net are identical, the "copy and crop" operation (Ronneberger, Fischer, and Brox 2015) is replaced by concatenating the output of the corresponding "downsampling" layer to the output of the previous "upsampling" layer. A linear layer is used to adjust

its dimension so that it can be input into the current "upsampling" layer. This ensures that the input and output dimensions of each layer remain the same, avoiding data dimension mismatches, while also effectively leveraging the capability of the U-Net architecture to model contextual information.

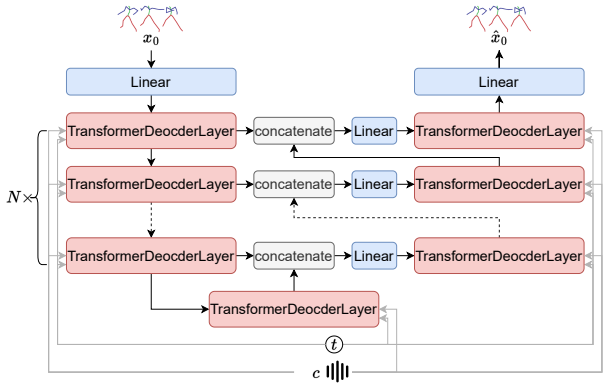


Figure 2: The architecture of Transformer U-Net.

Controllable Generation Framework

Inspired by ControlNet, we first train the aforementioned U-Net using a larger music-dance pair dataset and then freeze its parameters. As shown in Fig. 1, during the training phase, we use a dataset that contains 2D keypoints, dance movements, and musics. We replicate the "downsampling" layers in the U-Net along with their parameters to form the control network. The control network takes the 2D keypoints y as input data, while using the same music condition c and timestep t. For the U-Net part, the dance motion data x_0 is used as input, also with the same music condition c and timestep t, but its parameters are frozen.

The output generated by the replicated "downsampling" layers is passed through separate zero-convolution layers and then added to the output of the corresponding "downsampling" layers in the U-Net. The resulting intermediate representation is then processed through the concatenation and dimension transformation, serving as the input to the corresponding "upsampling" layers. After these operations, the U-Net outputs the predicted dance movements, which is used to calculate the total loss $\mathcal{L}$ in Eq. (6) and to update the parameters of the control network. This process is repeated until the entire model converges. The inference phase is similar to the process described above, with the addition of 2D keypoint data as the control signal, and the entire model includes both the U-Net and the control network.

Experiments

Experimental Setup

Datasets. We use PhantomDance dataset (Li et al. 2022a) to train the U-Net. The motion data in this dataset is manually created using industrial animation software and stored in SMPL format. The currently available dance movements contain 9.42 hours at a frame rate of 30 fps, covering more than 13 genres. For the controllable generation framework, we use AIST++ dataset (Li et al. 2021) to train the models. The primary reason for using this dataset is that it provides 2D keypoint data corresponding to 3D motions. The available data totals 5.19 hours with a frame rate of 60 fps. This dataset is obtained by performing 3D reconstruction on a dance motion video dataset AIST Dance Video DB (Tsuchida et al. 2019). The 2D keypoint data include 9 camera angles, but we only use the data from the C09 camera angle. Additionally, the 2D keypoints are stored in COCO format (Lin et al. 2014).

Implementation Details. We first train the Transformer U-Net using the PhantomDance dataset. Similar to EDGE, we segment the motion and corresponding music data into 5-second clips, with each segment overlapping every 0.5 seconds. These clips are all used to train the U-Net, and the converged model parameters are saved. Then, we train the control network using the train set of AIST++ dataset. The data undergo the same segmentation process, including the segmentation of 2D keypoint data. During training, we first load the pre-trained model parameters of U-Net, then construct the corresponding control network, and finally freeze the U-Net model parameters. Both models are trained using the Adan optimizer (Xie et al. 2024), with a learning rate of 4×10^{-4} and a weight decay of 0.02. To reduce memory usage, we employ a mixed-precision training strategy. The training is conducted on two NVIDIA GeForce RTX 4090 GPUs, and it takes approximately one day to complete. During the inference phase, we provide musics and keypoints for the controllable framework and use DDPM sampling method to generate better results.

Additionally, due to the use of datasets with different data ranges, we design a unified normalizer to handle both datasets. We normalize each dimension of the 3D motion data separately, which requires determining the minimum and maximum values for each dimension. However, based on the physical meaning of each dimension, we can set specific values for normalization. For the motion data, which consists of 151 dimensions, the first 4 dimensions represent the contact probabilities of four foot joints with the ground; therefore, the minimum value is 0 and the maximum value is 1. The 5th to 7th dimensions indicate body position, requiring the statistics of both datasets to obtain their range. The remaining dimensions denotes the rotation angles, with a minimum value of -1 and a maximum value of 1. By setting up a unified normalizer in this way, we can process datasets with different ranges and avoid issues such as vanishing gradients during training.

Evaluation Metrics

Fidelity. We follow the evaluation method of Bailando (Li et al. 2022b) and use the Fréchet Inception Distance (FID) to assess the quality of the generated motions. The FID metric calculates the similarity between the distributions of the generated results and ground truth. This metric is divided into two categories: FID_g, which calculates geometric features, and FID_k, which calculates kinematic features. FID_g measures the geometric relationship between specific

Method	$FID_g \downarrow$	$FID_k \downarrow$	$BAS \uparrow$	$d_{avg} \downarrow$
Ground Truth	10.60	17.10	0.24	-
FACT* (Li et al. 2021)	19.88	53.98	0.22	-
Bailando* (Li et al. 2022b)	**10.08**	28.16	0.23	-
EDGE* (Tseng, Castellon, and Liu 2023)	25.08	32.61	0.23	1.37
LODGE* (Li et al. 2024)	37.78	37.91	0.24	-
Transformer U-Net (Ours)	22.76	22.87	**0.25**	1.40
Control Network (Ours)	18.87	**15.70**	0.22	**0.44**

Table 1: Comparison with existing methods on the AIST++ dataset. ↓ means lower is better, and ↑ means higher is better. **Bold** denotes the best performance among these methods. * indicates that we fix some bugs in the FID metric calculation method and recalculate all metrics.

body parts in the motion sequences, while FID_k assesses the kinematic aspects of the motion, such as velocity and acceleration.

Music Consistency. We use the Beat Alignment Score (BAS) proposed by FACT (Li et al. 2021) to measure the rhythmic alignment between the generated motions and the music conditions. This metric first extracts the beats of both the motion and the corresponding music separately and then calculates the average distance between the kinematic beat and the music beat. The kinematic beat is defined as the moments corresponding to the local minima of the body's average velocity. The calculation formula for this metric is shown as follows:

$$BAS = \frac{1}{m}\sum_{i=1}^{m}\exp\Big(-\frac{\min_{\forall t_j^y \in B^y}\|t_i^x - t_j^y\|_2^2}{2\sigma^2}\Big), \quad (8)$$

where $B^x = \{t_i^x\}$ denotes the kinematic beats, $B^y = \{t_j^y\}$ denotes the music beats and σ is the normalization factor.

Motion Consistency. We evaluate the performance of the controllable generation framework by measuring the average distance between corresponding joints in the generated motions and ground truth. The closer the corresponding joints are, the smaller the gap between the generated and real results, indicating a stronger control effect of the 2D keypoints. The metric is expressed as follows:

$$d_{avg} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_{\text{gen}} - x_{\text{gt}})^2}, \quad (9)$$

where n denotes the number of joints in SMPL format, x denotes the positions of every joints in the motions, and the subscripts "gen" and "gt" represent the generated and real movements respectively.

Results and Analysis

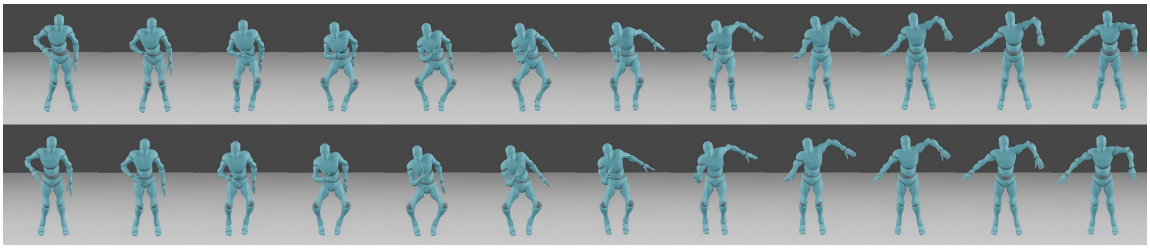
Performance of Transformer U-Net

We trained our proposed U-Net using the AIST++ dataset and compared it with other works. The results are presented in Table 1. Although the U-Net does not achieve state-of-the-art performance in terms of dance motion quality, it surpasses some diffusion-based methods. This indicates that the skip connections in the U-Net structure enhance the generation effect. By appending features extracted from shallow layers to these of deep layers, they compensate for the lack of some details in the features extracted from deep layers. Additionally, the U-Net exhibits the best rhythm consistency among all methods in the table. However, our model surpasses the ground truth in this metric. In fact, there are some fundamental issues with the BAS metric. BAS calculates the negative exponent of the average distance between the neighboring kinematic beats and music beats. This is not strictly correct that the calculated BAS is poor in some rhythmically aligned dances where one kinematic beat corresponds to every other music beat. This may indicate that when the generated motions are very similar to the real results, this metric loses its effectiveness.

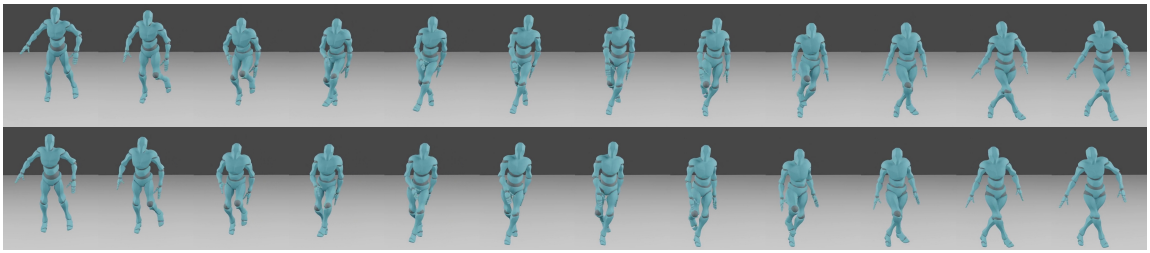
Controllable Generation using 2D Keypoints

Test on AIST++ Dataset. The quantitative results of our proposed control network are shown in Table 1. They demonstrate the excellent performance of this framework in generating high-quality movements. The main reason may be that the controlled generation of motions has a high degree of fidelity to the real data. The control network, especially the zero convolution layer, can transform control signals into biases on the output of the backbone. This makes the generated outcomes closely resemble actual movements, leading to lower FID scores. However, this framework is slightly lacking in music consistency. As we analyzed in the previous section, although BAS has contributed to the advancement of dance generation task in the past, it may lose its practical utility for generated results that are very similar to the ground truth. Due to issues with the storage format of the output data, we only compare the performance of EDGE and our proposed models in terms of motion consistency. For EDGE and Transformer U-Net, we use only musics as input and compare the generated results with the corresponding real motions to calculate d_{avg}. For the controllable generation framework, we use both musics and 2D keypoints as input and calculate the d_{avg} between generated outcomes and ground truth. As shown in the Table 1, the motions generated by Control Network are very similar to the real data, verifying its ability to control body movements.

We select the same music from the AIST++ dataset as a condition and use different 2D keypoints as control sig-



(a) The choreography ID is ch08.



(b) The choreography ID is ch09.

Figure 3: Results generated by using the same music but different control signals from the AIST++ dataset. The music we used is mLH2. The different choreography IDs denote different 2D keypoints. The first row shows the ground truth. The second row demonstrates the generated motions.

nals to verify the control capability of this framework. As shown in Table 2, we calculate the similarity between generated motions and their corresponding ground truth, and also compared the similarity between different ground truth. Fig. 3 displays the results generated by using the same music but different 2D keypoints. Despite using the same music, the generated motions still resemble the corresponding real data. This indicates that our proposed framework can effectively utilize control signals to guide the generation of body movements.

Generated Motions	$d_{avg} \downarrow$
Similar to Ground Truth	0.24
Different from Ground Truth	1.20

Table 2: Ablation study of the controllable generation framework using different control signals.

Performance on In-the-Wild Data. We tested the proposed controllable generation framework on some raw data. This data includes dance videos sourced from platforms like TikTok and Bilibili. Using pose estimation methods (Contributors 2020; Xu et al. 2022), we extracted the movements of individuals from these videos to obtain 2D keypoint data. Additionally, we utilized tools to extract music information from the videos as conditions. By inputting these two types of data into the framework, we generated 3D dance movements and visualized them. We found that our proposed framework is effective on in-the-wild data, generating dance movements similar to those of the individuals in the videos. Some of the visualization results are shown in Fig. 4.

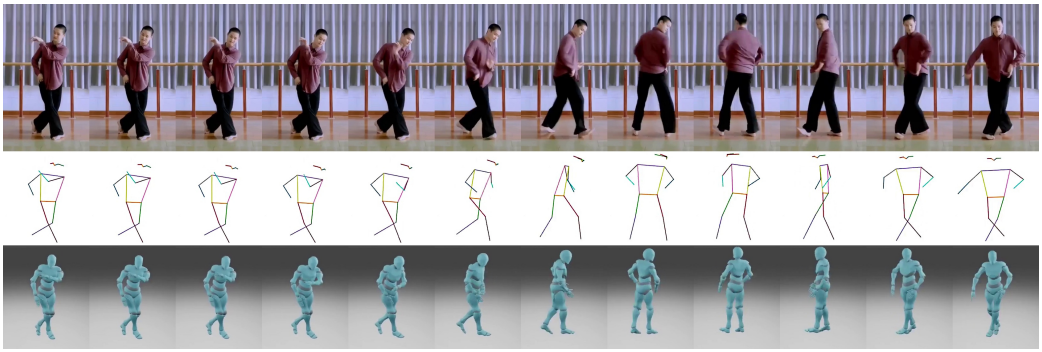
However, the framework also has some issues that lead to unstable generated results. The framework heavily relies on 2D keypoints, and when these keypoints are of poor quality, the corresponding generated movements may exhibit unrealistic behavior, as shown in Fig. 5(a). Because these methods are still data-driven and do not incorporate designs to simulate physical properties, the generated movements lack realism, particularly noticeable in foot movements, as demonstrated in Fig. 5(b).

Conclusions

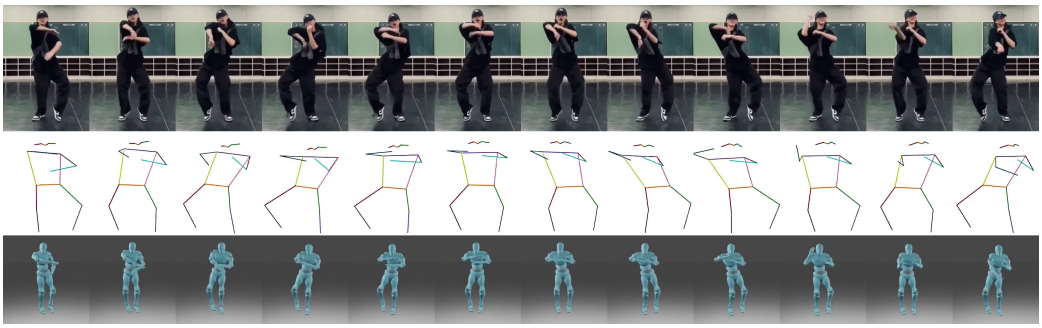
In this paper, we propose a controllable 3D dance generation framework based on a specifically designed Transformer U-Net model. The U-Net is trained on a large-scale dance dataset and enables the generation of high-quality dance movements. In our proposed framework, we use 2D keypoint sequences as control signals, allowing the generated motions to resemble the given keypoints. Experimental results show that this framework not only exhibits excellent performance but also effectively utilizes control signals to guide the generated results. We also test it on in-the-wild data and find that it still possesses control over the generated results. However, it also highlight some issues that we need to address in the future, such as a dependence on the control signals and inability to effectively generate displacement information.

Acknowledgments

The work was supported by National Key Research and Development Program of China (No. 2024YFB2808802).

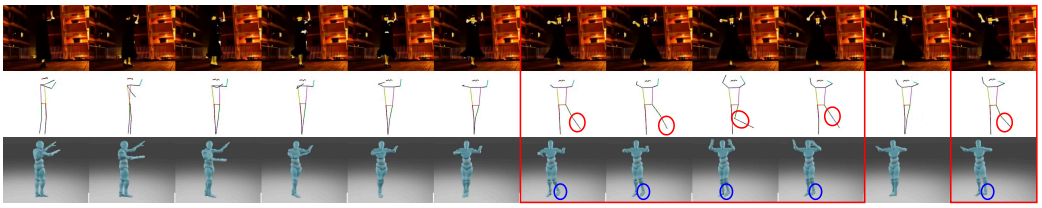


(a) Generated by using the video "Hua Jian You" from TikTok.

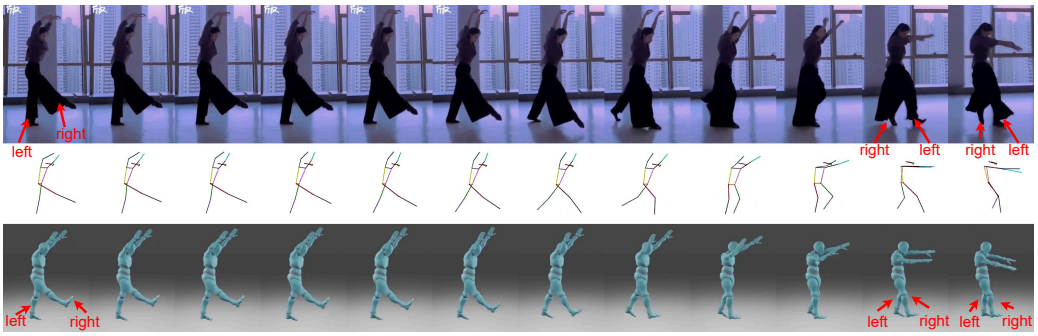


(b) Generated by using the video "Ke Mu San" from TikTok.

Figure 4: Some examples generated by using in-the-wild data. The first row shows some frames of the video. The second row demonstrates 2D keypoints recognized by using pose estimation methods. The third row illustrates the generated dance movements. The images in the same column correspond to the same time frame.



(a) The red rectangles highlight the frames with issues. The red ellipses mark the incorrect recognition results, while the blue ellipses indicate the corresponding generated motions.



(b) The generated foot joint movements do not align with the video, and the walking motion throughout the clip appears unnatural.

Figure 5: Some bad cases. The first row shows some frames of the video. The second row demonstrates 2D keypoints recognized by using pose estimation methods. The third row illustrates the generated dance movements. The images in the same column correspond to the same time frame.

References

Castellon, R.; Donahue, C.; and Liang, P. 2021. Codified audio language modeling learns useful representations for music information retrieval. arXiv:2107.05677.

Chen, K.; Tan, Z.; Lei, J.; Zhang, S.-H.; Guo, Y.-C.; Zhang, W.; and Hu, S.-M. 2021. ChoreoMaster: choreography-oriented music-driven dance synthesis. *ACM Trans. Graph.*, 40(4).

Cho, K.; van Merrienboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv:1409.1259.

Contributors, M. 2020. OpenMMLab Pose Estimation Toolbox and Benchmark. https://github.com/open-mmlab/mmpose.

Dai, W.; Chen, L.-H.; Wang, J.; Liu, J.; Dai, B.; and Tang, Y. 2024. MotionLCM: Real-time Controllable Motion Generation via Latent Consistency Model. arXiv:2404.19759.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Dhariwal, P.; Jun, H.; Payne, C.; Kim, J. W.; Radford, A.; and Sutskever, I. 2020. Jukebox: A Generative Model for Music. arXiv:2005.00341.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 8780–8794. Curran Associates, Inc.

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

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 6840–6851. Curran Associates, Inc.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Huang, Y.; Wan, W.; Yang, Y.; Callison-Burch, C.; Yatskar, M.; and Liu, L. 2024. CoMo: Controllable Motion Generation through Language Guided Pose Code Editing. arXiv:2403.13900.

Lee, H.-Y.; Yang, X.; Liu, M.-Y.; Wang, T.-C.; Lu, Y.-D.; Yang, M.-H.; and Kautz, J. 2019. Dancing to Music. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Li, B.; Zhao, Y.; Zhelun, S.; and Sheng, L. 2022a. DanceFormer: Music Conditioned 3D Dance Generation with Parametric Motion Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1272–1279.

Li, J.; Yin, Y.; Chu, H.; Zhou, Y.; Wang, T.; Fidler, S.; and Li, H. 2020. Learning to Generate Diverse Dance Motions with Transformer. arXiv:2008.08171.

Li, R.; Yang, S.; Ross, D. A.; and Kanazawa, A. 2021. AI Choreographer: Music Conditioned 3D Dance Generation With AIST++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 13401–13412.

Li, R.; Zhang, Y.; Zhang, Y.; Zhang, H.; Guo, J.; Zhang, Y.; Liu, Y.; and Li, X. 2024. Lodge: A Coarse to Fine Diffusion Network for Long Dance Generation Guided by the Characteristic Dance Primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1524–1534.

Li, S.; Yu, W.; Gu, T.; Lin, C.; Wang, Q.; Qian, C.; Loy, C. C.; and Liu, Z. 2022b. Bailando: 3D Dance Generation by Actor-Critic GPT With Choreographic Memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11050–11059.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 740–755. Cham: Springer International Publishing. ISBN 978-3-319-10602-1.

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6).

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10674–10685.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Navab, N.; Hornegger, J.; Wells, W. M.; and Frangi, A. F., eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234–241. Cham: Springer International Publishing. ISBN 978-3-319-24574-4.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088): 533–536.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; Ho, J.; Fleet, D. J.; and Norouzi, M. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 36479–36494. Curran Associates, Inc.

Shafir, Y.; Tevet, G.; Kapon, R.; and Bermano, A. H. 2023a. Human Motion Diffusion as a Generative Prior. arXiv:2303.01418.

Shafir, Y.; Tevet, G.; Kapon, R.; and Bermano, A. H. 2023b. Human Motion Diffusion as a Generative Prior. arXiv:2303.01418.

Sun, G.; Wong, Y.; Cheng, Z.; Kankanhalli, M. S.; Geng, W.; and Li, X. 2021. DeepDance: Music-to-Dance Motion Choreography With Adversarial Learning. *IEEE Transactions on Multimedia*, 23: 497–509.

Tang, T.; Jia, J.; and Mao, H. 2018. Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, 1598–1606. New York, NY, USA: Association for Computing Machinery. ISBN 9781450356657.

Tang, T.; Mao, H.; and Jia, J. 2018. AniDance: Real-Time Dance Motion Synthesize to the Song. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, 1237–1239. New York, NY, USA: Association for Computing Machinery. ISBN 9781450356657.

Tseng, J.; Castellon, R.; and Liu, K. 2023. EDGE: Editable Dance Generation From Music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 448–458.

Tsuchida, S.; Fukayama, S.; Hamasaki, M.; and Goto, M. 2019. AIST Dance Video Database: Multi-genre, Multi-dancer, and Multi-camera Database for Dance Information Processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, 501–510. Delft, Netherlands.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xie, X.; Zhou, P.; Li, H.; Lin, Z.; and Yan, S. 2024. Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–13.

Xu, Y.; Zhang, J.; ZHANG, Q.; and Tao, D. 2022. ViTPose: Simple Vision Transformer Baselines for Human Pose Estimation. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 38571–38584. Curran Associates, Inc.

Ye, Z.; Wu, H.; Jia, J.; Bu, Y.; Chen, W.; Meng, F.; and Wang, Y. 2020. ChoreoNet: Towards Music to Dance Synthesis with Choreographic Action Unit. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, 744–752. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379885.

Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 3836–3847.

Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; and Li, H. 2019. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhuang, H.; Lei, S.; Xiao, L.; Li, W.; Chen, L.; Yang, S.; Wu, Z.; Kang, S.; and Meng, H. 2023. GTN-Bailando: Genre Consistent long-Term 3D Dance Generation Based on Pre-Trained Genre Token Network. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.

Zhuang, W.; Wang, C.; Chai, J.; Wang, Y.; Shao, M.; and Xia, S. 2022. Music2Dance: DanceNet for Music-Driven Dance Generation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(2).