Rethinking Sampling for Music-Driven Long-Term Dance Generation

Tuong-Vy Truong-Thuy[1,2][0009−0008−2541−576X], Gia-Cat Bui-Le[1,2][0009−0006−9936−3814], Hai-Dang Nguyen[1,2][0000−0003−0888−8908], and Trung-Nghia Le*[1,2][0000−0002−7363−2610]

[1] University of Science, VNU-HCM, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
{tttvy20,blgcat20}@apcs.fitus.edu.vn, nhdang@selab.hcmus.edu.vn, ltnghia@fit.hcmus.edu.vn

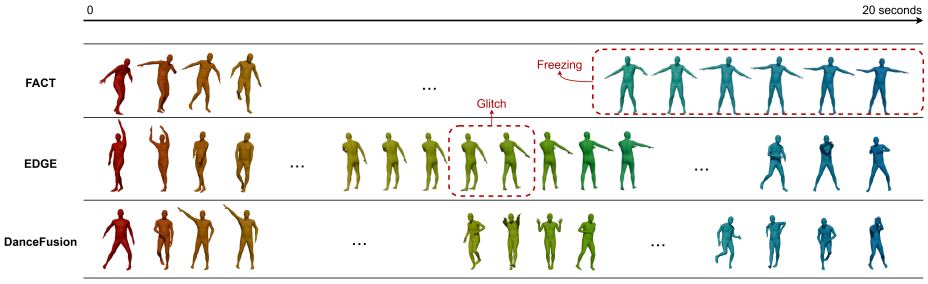


Fig. 1: Our DanceFusion, can generate natural dance sequences that retain both temporal coherence and spatial diversity. In contrast, autoregressive methods (*i.e.*, FACT [24]) experience a freezing problem and non-autoregressive methods (*i.e.*, EDGE [45]) suffer from glitches.

Abstract. Generating dance sequences that synchronize with music while maintaining naturalness and realism is a challenging task. Existing methods often suffer from "freezing" phenomena or abrupt transitions. In this work, we introduce DanceFusion, a conditional diffusion model designed to address the complexities of creating long-term dance sequences. Our method employs a past and future-conditioned diffusion model, leveraging the attention mechanism to learn the dependencies among music, past, and future motions. We also propose a novel sampling method that completes the transitional motions between two dance segments by treating previous and upcoming motions as conditions. Additionally, we address abruptness in dance sequences by incorporating inpainting strategies into a part of the sampling process, thereby improving the smoothness and naturalness of motion generation. Experimental results demonstrate that DanceFusion outperforms state-of-the-art methods in generating high-quality and diverse dance motions. User

* Corresponding author.

study results further validate the effectiveness of our approach in generating long dance sequences, with participants consistently rating DanceFusion higher across all key metrics. Code and model are available at https://github.com/trgvy23/DanceFusion.

Keywords: Music-to-Dance · Conditional Diffusion Model · Sampling Strategy

1 Introduction

Dancing has been a vital part of human culture, serving as a medium for entertainment and communication. In recent years, interest has grown in generating dances synchronized with music. This has various applications, such as virtual dancing in video games or animated films, choreography assistance, and personalized dance training. Despite these promising applications, significant challenges remain. Primarily, the generated dance sequences must align with the given music's beat and rhythm while preserving a sense of naturalness. Additionally, the dance moves need to be physically realistic and aesthetically pleasing.

Generating long dance sequences introduces additional complications due to the high temporal complexity associated with lengthy music pieces. Current methods are generally categorized into autoregressive and non-autoregressive approaches, each with its unique advantages and disadvantages. Autoregressive methods [12, 42, 48] generate future movements based on preceding ones, effectively capturing the flow of dance but often struggling with the "freezing" effect, where movements become static after a short period. In contrast, non-autoregressive methods [45] treat each motion segment independently, using simple interpolation techniques to connect segments and maintain continuity. Although more computationally efficient and capable of mitigating the cumulative error found in autoregressive methods, this approach can lead to abrupt and unnatural transitions between poses, as illustrated in Fig. 1.

This paper introduces DanceFusion, a novel approach to generating realistic 3D dance sequences synchronized with music. Our method leverages a transformer-based diffusion model conditioned on the music, as well as past and future dance movements. We chose a conditional diffusion model [9,40] due to its ability to produce high-quality samples with diverse variations, making it particularly suitable for dance synthesis. By conditioning the model on these three factors, DanceFusion aims to produce dance sequences that are well-synchronized with the music while ensuring smooth and continuous motion. Moreover, DanceFusion supports the generation of dance sequences of any length, accommodating extensive performances.

To maintain visual quality in long sequences and address issues like unnatural transitions or the "freezing" phenomenon, we propose a novel sampling method. We segment the music into independent sections and generates dance movements for each segment separately. We then employ our past and future conditioned diffusion model to create smooth transitions between adjacent segments. This

ensures that the transitions blend seamlessly with both adjacent segments, reducing abrupt movements and alleviating the "freezing" issue while preserving the strengths of autoregressive methods in capturing temporal naturalness.

We demonstrate the effectiveness of our proposed music-driven framework through various experiments. The experimental results on the AIST++ dataset [24] show that DanceFusion significantly outperforms state-of-the-art (SOTA) methods, achieving remarkable improvements in both fidelity and diversity. Furthermore, we conducted an extensive user study to validate the effectiveness of DanceFusion. Specifically, our method outperforms competitors by generating high-quality, diverse dance motions that closely resemble real-life choreography. Our code is available at https://github.com/trgvy23/DanceFusion.

In summary, our contributions are as follows:

- We introduce a conditional diffusion model for generating realistic dance movements, conditioned on music as well as past and future motions. Our method enhances the quality of the generated sequences and allows for the creation of arbitrarily long dance sequences.
- We propose a novel sampling method based on diffusion models to generate coherent dance motions with smooth transitions between adjacent segments, resulting in more natural and fluid dance sequences.
- Quantitative and qualitative experimental evaluations demonstrate the superiority of our method compared to existing approaches.

2 Related Work

2.1 Human Motion Generation

The surge to create lifelike human motion has long captivated researchers in computer vision and computer graphics. Previously, the methods typically involved using graph-based techniques [1,18], where motion sequences were broken down into smaller components and then reassembled based on predefined rules. However, the advent of deep neural networks has revolutionized this domain, offering significantly greater precision and versatility in generating human motion.

A notable breakthrough in this area has been the application of diffusion models for text-to-motion synthesis. MDM [44] demonstrated outstanding performance by generating high-fidelity motion sequences directly from textual descriptions. Furthermore, recent studies have demonstrated the ability to produce motion under various conditions. Some conditions are based on the motion itself, such as the motion-inbetweening task [14,34], while others involve specific requirements like action-labeled [2,21,44] or scene context [13,47]. Despite these impressive advancements in improving the realism and diversity of generated motions, creating dance movements remains a substantial challenge due to the complexity of movements.

2.2 Music-Driven Dance Generation

Early efforts in music-driven dance generation primarily used retrieval-based techniques, selecting pre-defined motion segments from a database and arrang-

ing them according to the music [5, 29, 37]. In contrast, recent approaches treat dance generation as a motion synthesis problem, employing various network architectures, such as CNNs [11, 19], RNNs [12, 43], GANs [20, 41], and Transformers [22–24]. These networks typically take music and preceding dance sequences as input, predicting subsequent dance movements in an autoregressive manner. Notably, Li *et al.* [24] introduced the Full Attention Cross-modal Transformer (FACT) model, which generates dance motion from music using transformers to extract meaningful representations from the input signals and a cross-modal transformer to learn the relationships between music and dance movements. However, these autoregressive methods face challenges such as error accumulation and the motion freezing phenomenon [52].

In a unique approach, VQ-VAE [46] was utilized to produce dance sequences with temporal coherence [39, 51]. 3D motions were first quantized with a VQ-VAE codebook, then a Generative Pretrained Transformer (GPT) was employed to create coherent sequences from the learned latent codes [39]. Although VQ-VAE helps maintain a pretrained codebook and ensures high-quality motion, its codebook limits dance variety.

Recently, diffusion models have demonstrated remarkable performance in generating high-quality images, videos, and motion sequences [3, 8, 15, 17, 35, 36, 38]. EDGE [45] served as a notable example of leveraging diffusion models for dance generation, treating it as a music-conditioned motion denoising problem. Using a transformer decoder architecture for music conditioning, EDGE generated multiple overlapping dance segments during the denoising process and ensured consistency between them through diffusion inpainting [27]. These segments were then stitched together using linear interpolation to create a continuous dance sequence. Although both EDGE and our DanceFusion utilize inpainting methods during sampling, EDGE's use of a binary mask results in less smooth blending between the given and synthesized parts. In contrast, we modify the mask so that the border is linearly interpolated.

3 Proposed Method

3.1 Problem Formulation

The goal of music-to-dance generation is to produce a synchronized sequence of dance motions, denoted as $D = \{d_i\}_{i=1}^{N}$, for a given music sequence $M = \{m_i\}_{i=1}^{N}$, where $d \in \mathbb{R}^{L \times 151}$ represents a dance pose, N denotes the number of frames in the music sequence determined by a specific sampling rate. Motion representation aligns with 24-joint SMPL [25] format, utilizing 6-DOF rotation representation [50] for each joint and a single root translation like EDGE [45].

Recognizing that a complete music sequence can be divided into multiple segments ($M = \{m^1, m^2, \ldots, m^L\}$, where L is the number of segments), with each segments include k frames. We can correspondingly decompose the generated dance sequence into L segments as well ($D = \{d^1, d^2, \ldots, d^L\}$). For clarity, we describe our training method using a simplified scenario with only two dance

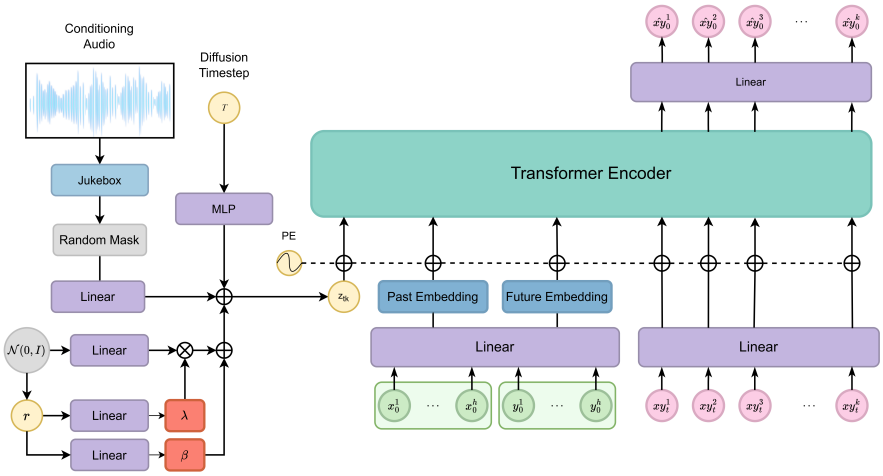


Fig. 2: Overview of DanceFusion, a music and motion-conditioned diffusion model using Transformers. The Transformer input combines music, past motions, future motion, and noised current motions. The model learns to denoise dance sequences from time $t = T$ to $t = 0$. Music embedding information is provided by a frozen Jukebox [4] model, while past and future motions are embedded with the same linear layer.

segments ($D = \{d^1, d^2\}$), which can be easily extended to handle an arbitrary number of segments. To avoid ambiguity, we represent d^1 as $x = \{x_1, x_2, \ldots, x_k\}$ and d^2 as $y = \{y_1, y_2, \ldots, y_k\}$. The corresponding music frames are denoted as $m = \{m_x, m_y\}$.

3.2 Architecture of DanceFusion

Drawing inspiration from the success of models like MDM [44] and PCMDM [49], we employ a Transformer Encoder as the core of our model architecture (See Fig. 2). This choice is motivated by the Transformer's ability to handle sequences of varying lengths and its proven effectiveness in motion generation tasks [33, 44, 45]. Frozen Jukebox model [4] is used to encode music sequences into a format suitable for the Transformer.

To incorporate both the noise level and the musical context into the Transformer, we design a special token, z_t, by concatenating the noise timestep t and the music condition m, after processing them with a feed-forward network. Each frame of the noisy input x_t is then projected into the Transformer dimension and combined with positional information. This combined input, including z_t and the projected frames, is then processed by the Transformer encoder.

Finally, the encoder's output, excluding the initial token z_t, is projected back into the original motion dimensions, producing the predicted clean motion sequence, $\hat{x_0}$. Instead of directly predicting the noise, as in DDPM [9], our approach focuses on generating the clean motion itself, a strategy that has shown success

in human motion generation [44, 45]. This direct prediction approach has been found to be effective in our context as well.

3.3 Past and Future Motion Conditioned Diffusion

Our diffusion process is represented as a Markov chain process, gradually adding noise to real data. At each step of this process, Gaussian noise is injected, with the amount of noise controlled by a predefined schedule, $\{a_t \in (0,1)\}_{t=1}^{T}$. The forward noising process is mathematically defined as:

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{a_t}x_{t-1}, (1-a_t)I). \tag{1}$$

Training a model solely on music proved insufficient for generating long, coherent dance sequences, we recognized the need for a smooth transitions between segments. To address this, we introduced an additional training step focusing on generating transitions, denoted as $\hat{xy}$, between two arbitrary dance segments: the past motion x^h and the future motion y^h. This enables the concatenation of segments into a continuous sequence. The forward diffusion process remains unchanged, but the denoising process can be formulated as:

$$\begin{aligned} p_\theta(xy_{t-1}|xy_t, x^h, y^h, m_{xy}) &= \mathcal{N}(xy_{t-1}; \mu_\theta(xy_t, x^h, y^h, m_{xy}, t), \beta_t), \\ \mu_\theta(xy_t, x^h, y^h, m_{xy}, t) &= \frac{1}{\sqrt{a_t}}(xy_t - \frac{\beta_t}{\sqrt{1-a_t}}\epsilon_\theta(xy_t, x^h, y^h, m_{xy}, t)). \end{aligned} \tag{2}$$

As described in Sec. 3.1, k represents the number of frames per segment. We define $k' = \frac{k}{2}$, so x^h denotes the h frame condition of segment x ($x^{k':k'+h}$) and y^h denotes the h frame condition of segment y ($y^{k'-h:k'}$). Note that the last k' frames of past segments and the first k' frames of future segments are excluded to represent the transition between segments being generated. Consequently, the hyperparameter h frames lie within the replaced segment frames, serving as conditional frames that provide past and future context to ensure the generated transition maintains consistency with the original segment motions. We empirically chose $h = 30$ (one second at 30 fps) for optimal result and visualization. Correspondingly, m_{xy} is the concatenation of m_x^h and m_y^h, representing the music features for the respective frames.

The combined music condition m_{xy}, the past motion sequence x^h, the future motion sequence y^h, the timestep t, and the current motion xy_t are then fed into the Transformer Encoder as a single input. To maintain consistency in feature representation, we utilize the same embedding function for both past and future motions. Our model architecture is illustrated in Fig. 2.

Loss function. To enhance the physical realism of our generated dance sequences, we go beyond simply minimizing reconstruction error. Drawing inspiration from previous models [44, 45], we incorporate geometric auxiliary losses

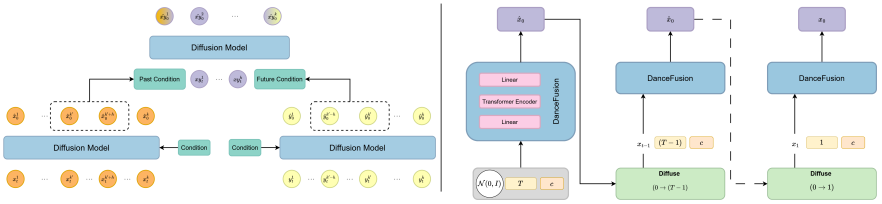


Fig. 3: (Left) Sampling method. Transitions are generated by considering frames from both preceding and following segments (*i.e.*, past and future motions). During sampling, the generation of transitions is handled as an inpainting task to maintain smoothness. **(Right) Sampling process pipeline.** The model starts with a noisy sequence $x_T \sim \mathcal{N}(0, I)$, generates an estimated final sequence $\hat{x}_0$, then noising it back to x_{T-1}, and continues this process until reaching $t = 0$.

to ensure the model aligns with three key aspects of physical realism: joint positions ($\mathcal{L}_{\text{pos}}$), velocities ($\mathcal{L}_{\text{vel}}$), and foot contact ($\mathcal{L}_{\text{contact}}$). For an input sequence x and its prediction $\hat{x}$, the loss functions are defines as follow:

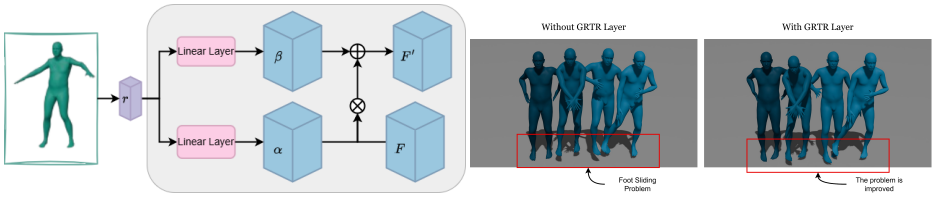
$$\begin{aligned}
\mathcal{L}_{\text{pos}} &= \frac{1}{k}\sum_{i=1}^{k} ||FK(x^{(i)}) - FK(\hat{x}^{(i)})||_2^2, \\
\mathcal{L}_{\text{vel}} &= \frac{1}{k-1}\sum_{i=1}^{k-1} ||(x^{(i+1)} - x^{(i)}) - (\hat{x}^{(i+1)} - \hat{x}^{(i)})||_2^2, \\
\mathcal{L}_{\text{contact}} &= \frac{1}{k-1}\sum_{i=1}^{k-1} ||(FK(\hat{x}^{(i+1)}) - FK(\hat{x}^{(i)})) \cdot \hat{b}^{(i)}||_2^2,
\end{aligned} \tag{3}$$

where $FK(\cdot)$ is the forward kinematic function, converting joint rotations into joint positions, superscript (i) indicates the frame index. The contact consistency loss $\mathcal{L}_{\text{contact}}$ is specifically applied to foot joints, in which $\hat{b}^{(i)} \in \{0, 1\}$ is the model's prediction of whether a foot is in contact with the ground at frame i. Finally, our overall training loss is a weighted sum of the simple reconstruction loss $\mathcal{L}_{\text{recon}}$ introduced in the work of Ho *et al.* [9] and the auxiliary losses:

$$\mathcal{L} = \lambda_{\text{recon}}\mathcal{L}_{\text{recon}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}}. \tag{4}$$

Classifier-free guidance. DanceFusion leverages a diffusion model to generate dance movements. At each time step t, we predict a clean sample $\hat{x_0} = x_\theta(x_t, c)$ and then add noise back to it to obtain x_{t-1}. This process is iteratively repeated, starting at $t = T$ and continuing until we reach the desired clean sample at x_0, as shown in Fig. 3.

We train our model x_θ using the classifier-free guidance technique [10], learning both conditioned and unconditioned distributions by randomly setting the condition c to empty ($\emptyset$), ensuring that $x_\theta(x_t, \emptyset)$ approximates the unconditioned distribution $p(x_0)$. This allows us to balance diversity and fidelity during



(a) GRTR pipeline. **(b)** Effectiveness of GRTR.

Fig. 4: Global Root Translation Refining (GRTR) layer.

sampling through performing guided inference, which is a weighted combination of unconditionally generated samples and conditionally generated samples, as expressed in Eq. (5):

$$\tilde{x}_\theta(x_t, c) = x_\theta(x_t, \emptyset) + w \cdot (x_\theta(x_t, c) - x_\theta(x_t, \emptyset)), \tag{5}$$

To ensure the generated transitions remain consistent with the preceding and following dance motions, we apply this guided strategy exclusively to the music condition and not to the past or future motion conditions.

3.4 Long-Form Sampling

Generating dance moves for music of any length is challenging. To address this, we propose to segment the music into L parts and independently generating corresponding dance moves for each segment. This method effectively mitigates the problems associated with autoregressive mechanisms, such as motion freezing and error accumulation.

However, this approach overlooks the relationships between consecutive dance segments. To overcome this limitation, we generate transitions conditioned on the h frames of the preceding and the following motion segment, as illustrated in Fig. 3. During the sampling phase, we use the sampling method from SinMDM [34] to ensure the generated transitions maintain coherence by treating transition generation as an inpainting task. Specifically, a mask includes linearly interpolated values between 0 and 1 at the borders of the inpaint and non-inpaint regions is used to indicate the transition part that requires inpainting.

3.5 Global Root Translation Refining (GRTR) Layer

Foot sliding or drifting, a common problem in dance generation, arises from a misalignment between the global translation of the root joint and the local rotations of other body joints [48]. To address this issue, inspired by previous works [31, 32, 48], we employ a Global Root Translation Refining (GRTR) layer that enable the model to learn the interdependence between these factors through an affine transformation. The GTRT layer aims to adjust latent codes of motion representations F through a modulation mechanism (See Fig. 4a): $F = \alpha F + \beta$, where $\alpha = f_\alpha(r), \beta = f_\beta(r)$ are two linear functions of the input 3D global

Table 1: Comparison with SOTAs on the AIST++ dataset. ↑ means higher is better, ↓ means lower is better, and → means closer to ground-truth is better.

Method	Motion Quality		Motion Diversity		BAS ↑	PFC ↓	Winning Rate
	$\mathbf{FID}_k \downarrow$	$\mathbf{FID}_g \downarrow$	$\mathbf{Dist}_k \rightarrow$	$\mathbf{Dist}_g \rightarrow$			
Ground-truth	-	-	10.31	7.65	0.278	1.332	69.06%
FACT [24]	35.35	12.40	5.94	6.18	0.221	2.2543	83.59%
Bailando [39]	28.15	**9.63**	7.83	**6.33**	0.220	1.754	73.59%
EDGE [45]	36.42	23.67	4.47	4.32	0.226	1.6545	70.78%
DanceFusion ($w = 1.0$)	**27.60**	21.13	6.77	4.10	**0.234**	1.4115	-
DanceFusion ($w = 2.5$)	31.53	23.00	**7.97**	4.97	0.233	**1.3759**	-

translation r. As illustrated in Fig. 4b, the GRTR layer significantly reduces foot sliding, maintaining a consistent foot trajectory across all frames.

4 Experiments

4.1 Experimental Settings

Our method was implemented and evaluated on a machine with a single A100 GPU of 80 GB memory. DanceFusion was trained for 20,000 steps using the AdamW optimizer [16,26] with a learning rate of 2×10^{-4} and a weight decay of 10^{-4}. The minibatch size was set to 32 for evaluation and 128 for training. We employed a Transformer encoder with 4 attention heads and 512-dimensional hidden representations as our backbone network. During both training and evaluation, we divided the dataset into 6-second segments and downsampled the dance motion data to 30 fps. Similar to Tseng *et al.* [45], we set the values $\lambda_{\text{recon}} = 0.636, \lambda_{\text{pos}} = 1.0, \lambda_{\text{vel}} = 2.964$, and $\lambda_{\text{contact}} = 10.942$ in Eq. (4).

We used AIST++ dataset [24], which comprises 1,408 high-quality dance motions paired with music, for our evaluation. We followed the train/test splits as defined by the original dataset.

4.2 Comparison with State-of-the-art Methods

We compare our DanceFusion against music-to-danceSOTAs, including FACT [24], Bailando [39], and EDGE [45]. FACT [24] leverages a full attention cross-modal transform model to generate long sequences of realistic 3D dance movements. Building upon this, Bailando [39] shows remarkable improvements in qualitative performance. Lastly, EDGE [45], a transformer-based diffusion model, represents the current SOTA model in dance generation.

Motion Quality. In line with previous studies [24, 39, 45], we evaluate motion quality by measuring the distance between the motion features of both

the generated and the ground-truth motions, using the Frechet Inception Distance (FID) [7]. The features For this computation are extracted from *fairmotion* [6], which include kinematic features [30], denoted as FID_k, and geometric features [28], denoted as FID_g. Our DanceFusion outperforms EDGE in both FID_k and FID_g. Notably, we achieve the best FID_k score of 27.6, outperforming all other methods and indicating superior motion quality. Although FACT achieves better FID_g scores, the generated motions often appear nonsensical, with repetitive poses during the test music piece. This discrepancy raises concerns about the metrics' ability to accurately reflect the quality of generated motion, a concern also noted by Tseng *et al.* [45].

Motion Diversity. To evaluate DanceFusion's ability to generate diverse dance motions in response to various input music tracks, we calculate the mean Euclidean distance in the feature space, as outlined in Bailando [39]. The motion diversity metrics in the kinematic and geometric feature spaces are denoted as Dist_k and Dist_g, respectively. We achieve the best performance on Dist_k with the score of 7.97, an improvement of 0.14 over Bailando, the second-best method. Although FACT and Bailando perform better in terms of Dist_g, we argue that neither performs well with in-the-wild music, both suffering from freezing motion issues.

Motion-Music Correlation. To assess how well the generated dance sequences are synchronized with the accompanying music, we use the Beat Alignment Score (BAS) [24]. This score calculates the average time temporal distance each beat in the music and the nearest matching beat in the dance sequence. As shown in Tab. 1, DanceFusion beats all other methods on this metric. These findings highlight our model's proficiency in improving the correlation between music and motion.

Physical Plausibility. To evaluate the physical plausibility of our generated dance sequences, we adopt the Physical Foot Contact score (PFC) metric, as proposed in EDGE [45]. This metric assesses the realism of foot-ground interactions without assuming static contact throughout the sequence. The results are presented in Tab. 1. Notably, our approach outperforms the SOTA methods and achieves a score close to the ground truth motion capture data.

User Study. To thoroughly assess the visual quality of our method, we conducted an extensive user study comparing dance sequences generated by DanceFusion with those produced by other methods using the AIST++ dataset. The study included 64 participants who individually viewed 40 pairs of video clips, each lasting 7 to 10 seconds. Each pair featured one sequence generated by DanceFusion (with guidance $w = 1.0$) and one by a competing approach. Participants were asked to select the video with superior overall quality in the dance

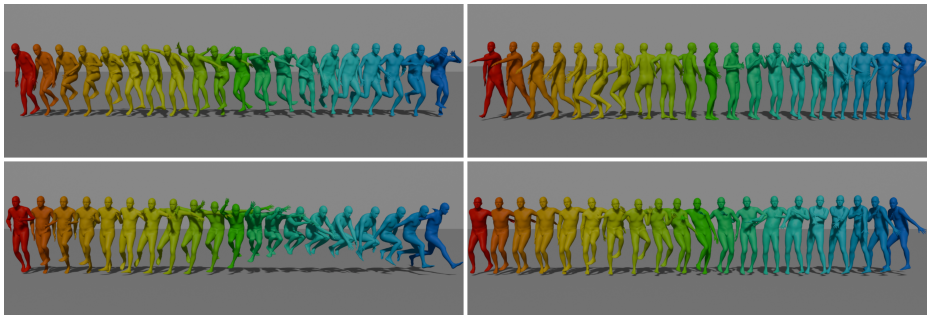


Fig. 5: Our DanceFusion is capable of generating diverse and physically plausible dance.

sequence, considering factors such as physical plausibility, visual appeal, and diversity of dancing motions. Notably, music was not provided in this study, as it was not necessary for the goal of evaluating the visual quality of the generated dance sequences. Additionally, initial participants provided feedback that they preferred to watch the videos with muted music, as the short duration of each melody made it difficult for them to focus on the dance sequences.

The results shown in Tab. 1 demonstrate that our method substantially outperforms EDGE, achieving an impressive 70.78% winning rate. Notably, DanceFusion surpasses the ground-truth dance performances 69.06% of the time, even though the baseline dances are motion-captured from professional dancers. This indicates that DanceFusion can generate dance movements that are indistinguishable from those performed by real-life dancers. Figure 5 showcases several examples where DanceFusion's capabilities are fully displayed. These results highlight our method's ability to produce highly realistic and physically plausible 3D human movements.

4.3 Ablation Study

Guidance Weight at Inference Time. The guidance weight coefficient w in Eq. (5) is a crucial parameter in classifier-free guidance. A larger w yields higher fidelity to the condition but may reduce accuracy relative to the true distribution of the original data. Tab. 2 displays results for different w values. As w increases, diversity also increases, while a lower w produces motions more accurate to the ground-truth.

Effect of Conditioning Frames. We conducted an in-depth analysis of various hyperparameter choices for h, starting with one second and exploring different options. As shown in Tab. 2, $h = 30$ is the optimal value, striking a balance between fidelity and diversity. We also found that too high value causes insufficient temporal space for meaningful transitions, while too low value fails to provide sufficient context and constraints for the model.

Table 2: Ablation study of DanceFusion.

Method	$\mathbf{FID}_k \downarrow$	$\mathbf{FID}_g \downarrow$	$\mathbf{Dist}_k \rightarrow$	$\mathbf{Dist}_g \rightarrow$	BAS ↑	PFC ↓
Ground-truth	-	-	10.31	7.65	0.278	1.332
$w = 1.0$	27.60	**21.13**	6.77	4.10	**0.234**	1.4115
$w = 1.5$	**26.13**	21.28	7.29	4.51	0.225	1.3865
$w = 2.0$	28.79	21.87	7.67	4.79	0.230	1.5792
$w = 2.5$	31.53	23.00	**7.97**	**4.97**	0.233	**1.3759**
$h = 20$	**26.96**	22.72	6.74	4.26	0.226	1.5951
$h = 30$	27.60	21.13	**6.77**	4.10	**0.234**	**1.4115**
$h = 40$	45.66	**18.81**	6.01	**4.43**	0.230	1.5356
w/o inpainting	28.56	24.83	6.77	**4.11**	0.230	1.9795
with inpainting	**27.60**	**21.13**	**6.77**	4.10	**0.234**	**1.4115**
w/o GRTR	44.45	21.20	5.29	**4.30**	0.232	1.4671
with GRTR	**27.60**	**21.13**	**6.77**	4.10	**0.234**	**1.4115**

Inpainting Strategy During Sampling. Tab. 2 indicates that the performance declines when sampling without the inpainting method. Specifically, both FID_k and FID_g increase compared to when the inpainting strategy is applied, suggesting a decline in motion quality. While other metrics show negligible differences, sequences without inpainting exhibit subtle glitches.

Effect of GRTR Layer. Tab. 2 shows the improvement of using the GRTR layer in term of PFC score, indicating the efficiency of the GRTR layer in solving issues of foot sliding.

5 In-The-Wild Music-To-Dance Evaluation

While DanceFusion has shown excellent performance on the AIST++ dataset, it also shows impressive results for in-the-wild music. To address crucial aspects of generalization and to highlight our method's capability, we conducted a thorough user study using in-the-wild music.

Metrics. We defined four key metrics to comprehensively evaluate the ability to generate long-term dance sequences, including Smoothness, Pose Diversity, Beat Alignment and Overall Impression. These metrics cover a thorough assessment of the model's effectiveness and capabilities. Specifically, participants were asked to rate the generated dance sequences on a scale of 1 to 5 based on the following criteria: *Smoothness*: How smooth and fluid did you find the dance sequence? *Diversity*: How diverse were the poses in the dance sequence? *Beat Alignment*: How well did the dance sequence align with the musical beat? *Overall*: What is your overall impression of the dance sequence?

Table 3: Comparative performance of DanceFusion and EDGE in term of MOS.

Method	Smoothness	Pose Diversity	Beat Alignment	Overall
EDGE [45]	3.65	3.48	3.47	3.55
DanceFusion	**3.94**	**3.84**	**3.76**	**3.99**

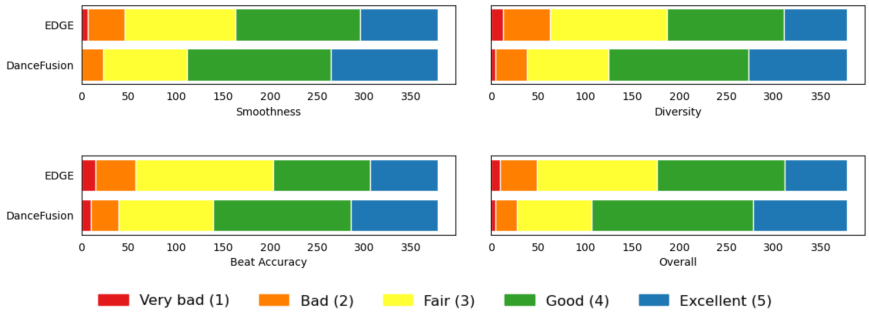


Fig. 6: Participants' ratings for each method across four key metrics in the user study. The horizontal bars represent the aggregated scores for each method.

Setup. We evaluated our method against EDGE [45], the leading approach in achieving the highest qualitative performance. Bailando [39] was excluded because it is observed to frequently produce instances where the generated dances froze. FACT [24] was omitted as well due to not support customized music inputs. We randomly selected 6 music pieces spanning diverse genres, including Pop, Rap, Chinese Classical, and K-pop. We then generated dance movements using both the EDGE model and our own. Afterward, we visualized the dances with a 3D SMPL [25] model in Blender, giving participants a clearer view of the performances, as shown in Fig. 7.

5.1 Apparatus and Procedure

We invited 64 participants, covering various levels of knowledge in artificial intelligence and choreography, to join our study. Their diverse professional backgrounds provided a range of perspectives for the evaluation process, ensuring a comprehensive and objective assessment. Participants were asked to rate the performance of each of the two methods on a scale from 1, indicating "Very poor,", to 5, indicating "Excellent," based on four metrics from their perspectives. To ensure quality responses, we filtered out those with uniform ratings (*e.g.*, all "Very bad" or all "Excellent").

5.2 Quantitative Results

Results from Tab. 3 show that DanceFusion consistently outperforms EDGE across all metrics, indicating that participants found DanceFusion superior in

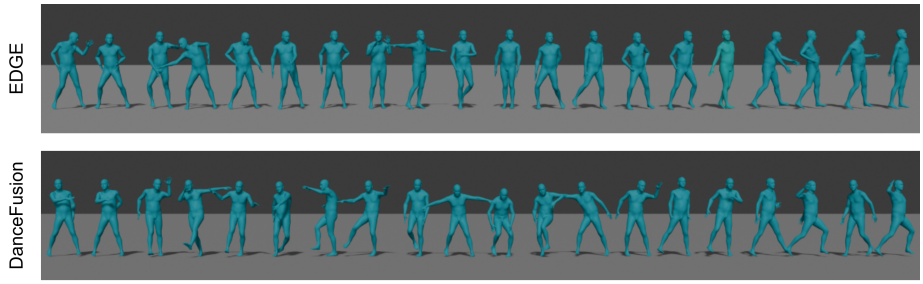


Fig. 7: Qualitative results for generating dance sequences from in-the-wild music.

generating fluid, diverse, well-aligned with the beat, and impressive long-term dance sequences. As depicted in Fig. 6, DanceFusion rarely received "Very bad" ratings and was frequently rated as "Good." In contrast, EDGE had more "Fair" ratings, highlighting its lower performance. Additionally, Fig. 7 demonstrates that DanceFusion produces dance movements with greater diversity than EDGE.

6 Failure cases

Although DanceFusion shows significant improvements over existing methods, it still has certain limitations. Notably, it faces challenges in generating smooth turn-around movements, often resulting in jerky or abrupt transitions. Additionally, it struggles to maintain consistency during more complex movements. We leave these limitations as areas for future research. Advancing research in these directions could considerably enhance the model's performance and robustness, opening doors for its use in more demanding real-world scenarios.

7 Conclusion

In this paper, we introduced DanceFusion, a novel method for generating long-term 3D dances using a diffusion model conditioned on past and future. We also proposed a specialized sampling technique to maintain consistency, resulting in natural and fluid long dance sequences. DanceFusion has been rigorously evaluated through user studies and standard measures. Experimental results demonstrates that DanceFusion can produce long and diverse dance sequences with high temporal coherence.

Acknowledgement. This research is funded by University of Science, VNU-HCM, under grant number CNTT 2024-16.

References

1. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. ACM Transactions on Graphics (TOG) **21**(3), 483–490 (2002) 3

2. Chen, X., Jiang, B., Liu, W., Huang, Z., Fu, B., Chen, T., Yu, G.: Executing your commands via motion diffusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18000–18010 (2023) 3
3. Dabral, R., Mughal, M.H., Golyanik, V., Theobalt, C.: Mofusion: A framework for denoising-diffusion-based motion synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9760–9770 (2023) 4
4. Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A., Sutskever, I.: Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341 (2020) 5
5. Fan, R., Xu, S., Geng, W.: Example-based automatic music-driven conventional dance motion synthesis. IEEE transactions on visualization and computer graphics **18**(3), 501–515 (2011) 4
6. Gopinath, D., Won, J.: Fairmotion-tools to load, process and visualize motion capture data (2020) 10
7. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017) 10
8. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022) 4
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020) 2, 5, 7
10. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 7
11. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016) 4
12. Huang, R., Hu, H., Wu, W., Sawada, K., Zhang, M., Jiang, D.: Dance revolution: Long-term dance generation with music via curriculum learning. arXiv preprint arXiv:2006.06119 (2020) 2, 4
13. Huang, S., Wang, Z., Li, P., Jia, B., Liu, T., Zhu, Y., Liang, W., Zhu, S.C.: Diffusion-based generation, optimization, and planning in 3d scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16750–16761 (2023) 3
14. Kaufmann, M., Aksan, E., Song, J., Pece, F., Ziegler, R., Hilliges, O.: Convolutional autoencoders for human motion infilling. In: 2020 International Conference on 3D Vision (3DV). pp. 918–927. IEEE (2020) 3
15. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis & editing. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 8255–8263 (2023) 4
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9
17. Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761 (2020) 4
18. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 723–732 (2023) 3
19. Kritsis, K., Gkiokas, A., Pikrakis, A., Katsouros, V.: Danceconv: Dance motion generation with convolutional networks. IEEE Access **10**, 44982–45000 (2022) 4
20. Lee, H.Y., Yang, X., Liu, M.Y., Wang, T.C., Lu, Y.D., Yang, M.H., Kautz, J.: Dancing to music. Advances in neural information processing systems **32** (2019) 4

21. Lee, T., Moon, G., Lee, K.M.: Multiact: Long-term 3d human motion generation from multiple action labels. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1231–1239 (2023) 3
22. Li, B., Zhao, Y., Zhelun, S., Sheng, L.: Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1272–1279 (2022) 4
23. Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., Li, H.: Learning to generate diverse dance motions with transformer. arXiv preprint arXiv:2008.08171 (2020) 4
24. Li, R., Yang, S., Ross, D.A., Kanazawa, A.: Ai choreographer: Music conditioned 3d dance generation with aist++. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13401–13412 (2021) 1, 3, 4, 9, 10, 13
25. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2, pp. 851–866 (2023) 4, 13
26. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 9
27. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022) 4
28. Muller, M., Kurth, F., Clausen, M.: Chroma-based statistical audio features for audio matching. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005. pp. 275–278. IEEE (2005) 10
29. Ofli, F., Erzin, E., Yemez, Y., Tekalp, A.M.: Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. IEEE Transactions on Multimedia **14**(3), 747–759 (2011) 4
30. Onuma, K., Faloutsos, C., Hodgins, J.K.: Fmdistance: A fast and effective distance function for motion capture data. Eurographics (Short Papers) **7** (2008) 10
31. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of conference on computer vision and pattern recognition. pp. 2337–2346 (2019) 8
32. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018) 8
33. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3d human motion synthesis with transformer vae. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10985–10995 (2021) 5
34. Raab, S., Leibovitch, I., Tevet, G., Arar, M., Bermano, A.H., Cohen-Or, D.: Single motion diffusion. arXiv preprint arXiv:2302.05905 (2023) 3, 8
35. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 4
36. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22500–22510 (2023) 4
37. Shiratori, T., Nakazawa, A., Ikeuchi, K.: Dancing-to-music character animation. In: Computer Graphics Forum. vol. 25, pp. 449–458. Wiley Online Library (2006) 4

38. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022) 4
39. Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., Loy, C.C., Liu, Z.: Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11050–11059 (2022) 4, 9, 10, 13
40. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015) 2
41. Sun, G., Wong, Y., Cheng, Z., Kankanhalli, M.S., Geng, W., Li, X.: Deepdance: music-to-dance motion choreography with adversarial learning. IEEE Transactions on Multimedia **23**, 497–509 (2020) 4
42. Sun, J., Wang, C., Hu, H., Lai, H., Jin, Z., Hu, J.F.: You never stop dancing: Non-freezing dance generation via bank-constrained manifold projection. Advances in Neural Information Processing Systems **35**, 9995–10007 (2022) 2
43. Tang, T., Jia, J., Mao, H.: Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1598–1606 (2018) 4
44. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022) 3, 5, 6
45. Tseng, J., Castellon, R., Liu, K.: Edge: Editable dance generation from music. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 448–458 (2023) 1, 2, 4, 5, 6, 9, 10, 13
46. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017) 4
47. Wang, J., Yan, S., Dai, B., Lin, D.: Scene-aware generative network for human motion synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12206–12215 (2021) 3
48. Yang, S., Yang, Z., Wang, Z.: Longdancediff: Long-term dance generation with conditional diffusion model. arXiv preprint arXiv:2308.11945 (2023) 2, 8
49. Yang, Z., Su, B., Wen, J.R.: Synthesizing long-term human motions with diffusion models via coherent sampling. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 3954–3964 (2023) 5
50. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5745–5753 (2019) 4
51. Zhuang, H., Lei, S., Xiao, L., Li, W., Chen, L., Yang, S., Wu, Z., Kang, S., Meng, H.: 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). pp. 1–5. IEEE (2023) 4
52. Zhuang, W., Wang, C., Chai, J., Wang, Y., Shao, M., Xia, S.: Music2dance: Dancenet for music-driven dance generation. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **18**(2), 1–21 (2022) 4