한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 32, No. 4, P. 1~12

양방향 Mamba를 활용한 긴 춤 동작 생성

Vanessa Tan° 김혜민 최수진 노준용*

한국과학기술원 문화기술대학원

{vanessa.tan, spot1223, 97choisj, junyongnoh}@kaist.ac.kr

ChoreoMamba: Long-Sequence Dance Generation with Bidirectional Mamba

Vanessa Tan° Haemin Kim Soojin Choi Junyong Noh*

KAIST GSCT

Abstract

Recent advances in music-to-dance generation have enhanced immersive digital experiences, but scaling these models to long sequences remains challenging. Existing approaches often rely on transformer-based architectures with quadratic complexity, which limits scalability. Others adopt stitching or interpolation strategies that can introduce temporal discontinuities. To address these limitations, we introduce ChoreoMamba (CM), a music-to-dance diffusion-based framework for efficient long-sequence generation conditioned on both music and dance style. Built on a bidirectional Mamba architecture, our model generates dance sequences up to 180 seconds while preserving motion quality, rhythmic alignment, and computational efficiency. Experimental results show that our method outperforms state-of-the-art approaches across multiple sequence lengths and generalizes well to in-the-wild inputs. A user study with both dancers and non-dancers shows that our method achieves higher mean opinion scores in beat alignment, style alignment, and naturalness compared to prior methods. Supplementary video can be seen in this link: tiny.cc/choreomamba

요 약

최근 음악 기반의 춤 생성 분야의 발전은 디지털 경험의 몰입감을 향상시켰지만, 기존의 방법들을 긴 동작을 생성하도록 확장하는 것은 여전히 어려운 과제로 남아있다. 기존의 방법들은 주로 이차 복잡도를 가지는 Transformer 기반의 구조를 사용하기 때문에 확장성이 제한된다. 또한, 일부 방법들은 긴 동작 생성을 위해 동작 연결(stitching)이나 보간(interpolation) 전략을 사용하는데, 이러한 방식은 시간 축에 대해 불연속적인 움직임을 생성해 동작 품질을 저하시킬 수 있다. 본 논문에서는 이러한 한계점을 해결하기 위해 음악과 춤 스타일을 동시에 조건으로 활용하는 확산 모델 기반 심층 네트워크인 ChoreoMamba(CM)를 제안한다. 제안된 네트워크는 양방향 Mamba 구조를 기반으로 최대 180초 길이의 춤 동작을 생성하면서도, 동작 품질, 리듬 정합성, 그리고 계산 효율성을 유지한다. 실험 결과, 제안된 방법은 다양한 시퀀스 길이에 걸쳐 기존의 방법보다 향상된 성능을 보였고, 실제 새로운 음악 입력에도 일반화가 가능함을 보였다. 또한, 무용수와 비전문가를 모두 포함한 사용자 평가 결과, 기존 방법들에 비해 비트 정렬성, 스타일 부합성, 그리고 자연스러움 측면에서 더 높은 평균 평가 점수를 획득하였다.

Keywords: music-to-dance, motion generation, diffusion, state space models
키워드: 음악-춤 생성, 동작 생성, 확산 모델, 상태 공간 모델

1 Introduction

Dance motion plays an important role in digital media such as games, films, and virtual environments, where synchronized choreography enhances immersion and storytelling. Recent advances in generative models have enabled automatic music-to-dance motion generation [1, 2, 3, 4]. However, most existing approaches remain constrained to short sequences due to computational and mem-

*corresponding author: Junyong Noh / KAIST GSCT (junyongnoh@kaist.ac.kr)

Received : 2026.04.18./ Review completed : 1st 2026.05.11./ Accepted : 2026.06.06.
DOI : 10.15701/kcgs.2026.32.4.1
ISSN : 1975-7883(Print)/2383-529X(Online)

ory limitations, making it difficult to generate a continuous dance motion. Extending these models to long durations is challenging, as dance performances require consistent motion quality, smooth transitions, and stable movement over time. Some methods attempt to address this by stitching short motion segments [5, 2, 4], but this often introduces temporal discontinuities and misalignment with the music. Other approaches rely on attention-based architectures whose computational and memory costs increase rapidly with sequence length, limiting their scalability [6, 4, 2]. As a result, generating long and coherent dance motion while maintaining stable performance remains a challenging problem.

To address this, we propose ChoreoMamba (CM), a diffusion-based music-to-dance motion generation method built on the bidirectional Mamba architecture [7]. Unlike attention-based models, bidirectional Mamba captures long-range temporal dependencies with linear complexity, making it better suited for extended motion sequences. As the denoising backbone, it lowers the computational cost of each denoising step relative to transformer-based diffusion models while using bidirectional temporal context to improve motion continuity and music-motion alignment. Our method supports sequences of varying lengths and generates temporally coherent dance motions while preserving motion quality and style control. The key contributions of our work are as follows.

- We introduce a music-to-dance generation framework based on bidirectional state space models, enabling efficient long-sequence motion generation.
- Our model generates long, high-quality dance sequences (up to 180 seconds) that remain visually coherent and musically synchronized.

2 Related Work

2.1 Music-Driven Dance Motion Generation

Deep learning–based dance motion generation methods, particularly transformer-based architectures, model relationships between music and motion to generate dance sequences [1, 2, 3, 8]. By capturing temporal dependencies between audio features and motion, these models improve motion quality and music–motion alignment. However, transformer architectures scale poorly to long sequences due to the quadratic computational complexity of the attention mechanism, limiting their efficiency and scalability. Diffusion models have recently been adopted for motion generation due to their ability to improve motion diversity, quality, and controllability [3, 9, 2, 4]. To produce long motion sequences, some approaches stitch short motion segments [10, 11], while others use multi-stage diffusion models that progressively refine motion through coarse-to-fine generation [5]. Although these approaches extend motion duration, stitching may introduce discontinuities between segments, and multi-stage methods may lead to synchronization inconsistencies when aligning motion with the full music duration. Recent single-stage diffusion models such as EDGE [2] and LDA [4] generate motion sequences within a single model, showing reasonable temporal consistency and synchronization with music. However, these models rely on transformer or conformer-based denoisers, which remain computationally expensive for long sequences. To address this limitation, we integrate a state-space model (SSM) in the denoiser network, enabling efficient long-sequence generation while maintaining motion quality and temporal synchronization.

2.2 State Space Models

SSMs, originally developed for control systems, have recently been applied to deep learning for sequence modeling [12, 13, 14, 15]. Architectures such as Structured State Space for Sequences (S4) and its successor Mamba provide an efficient alternative to transformers for long-sequence modeling by using state dynamics instead of attention mechanisms, improving scalability for long-range dependencies [12]. In this work, we use Mamba as the denoiser architecture in the diffusion process. Mamba is a hardware-aware selective SSM that dynamically adjusts its parameters based on input sequences, improving context modeling [16]. Mamba has been successfully applied to motion generation conditioned on text [17, 18, 19], speech [16, 20, 21], and motion style [22]. More recently, Park et al. [23] introduced a Mamba-based diffusion framework for music-conditioned dance generation, employing a two-stage architecture and a Gaussian-based beat representation to enhance rhythmic alignment. Similar to their work, we leverage Mamba to improve temporal modeling in dance synthesis. However, while Park et al. [23] employ a two-stage framework with separate modules for motion modeling and music-motion interaction, ChoreoMamba integrates bidirectional Mamba blocks directly into a single-stage diffusion denoiser for efficient long sequence dance generation. By incorporating both past and future context during denoising [7], our approach efficiently captures long-range temporal dependencies while maintaining motion quality and music synchronization.

3 State Space Sequence Models

State space sequence models (SSSMs) [14, 13, 12] are a class of deep learning sequence models inspired by classical continuous

state space models. Classical models map an input signal $x(t) \in \mathbb{R}$ to an output signal $y(t) \in \mathbb{R}$, similar to sequence-to-sequence models that predict the next output based on the entire input sequence. Instead, SSSMs represent the entire input as a latent state $h(t) \in \mathbb{R}^{N\times 1}$, which is updated in a constant time. Typically, SSSMs are formulated using the following ordinary differential equation:

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}h(t) \end{aligned} \tag{1}$$

where $h'(t)$ represents the updated state or first-order derivative of the latent state $h(t)$. Since $y(t)$ is a function of $h(t)$, $h'(t)$ indirectly affects $y(t)$ by shaping the trajectory of $h(t)$ [12]. Here, $\mathbf{A} \in \mathbb{R}^{N\times N}$ represents a diagonal state matrix, $\mathbf{B} \in \mathbb{R}^{N\times 1}$ and $\mathbf{C} \in \mathbb{R}^{1\times N}$ are the projection matrices of the input and output, respectively, and N denotes the hidden state size [12, 24].

For practical applications where signals are typically discrete, SSSMs are often discretized using the zero-order hold (ZOH) method, which approximates the continuous process over discrete time intervals. The discretized form is expressed as follows [14]:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A}), \quad \overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \tag{2}$$

where Δ is a time scale parameter and $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are respectively discretized versions of $\mathbf{A}$ and $\mathbf{B}$ of the continuous representation. SSSMs are designed to leverage this discretization for efficient training and long-sequence generation by using recurrent and convolutional representations [12, 17]. While the recurrent form processes sequences step by step, the convolutional representation allows parallel computation, making these models effective in balancing training speed and inference performance.

Mamba [12] is a variant of these models that introduces a contextually aware input selection mechanism by making $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and Δ dependent on the input $x(t)$ [18]. Additionally, Mamba incorporates a hardware-aware state expansion mechanism, leveraging memory hierarchies (from slower High Bandwidth Memory to faster SRAM) to improve both speed and memory efficiency [12]. Mamba's selection mechanism uses a forward unidirectional scan, while bidirectional Mamba architectures enhance this by introducing a backward feature extraction or scanning process [7, 25]. This bidirectional process enables spatial awareness similar to self-attention mechanisms, allowing for effective modeling of the global context of the input [7, 25].

4 ChoreoMamba

The objective of our method is to generate dance motion from music and a style label using a diffusion model, as illustrated in Fig. 1. We first preprocess the input and output data (Sec. 4.1) and adopt the Denoising Diffusion Probabilistic Model (DDPM) as the base diffusion process (Sec. 4.2). We then employ bidirectional Mamba as the main sequence modeling component of the denoiser (Sec. 4.3). The denoiser network consists of two components: the Residual Mamba Block (RMB) and the Mamba Refiner Block (MRB) (Sec. 4.4), both based on the bidirectional Mamba architecture [7]. The RMB models short- and long-term dependencies and improves motion diversity, while the MRB refines and smooths motion transitions. Finally, the model is fine-tuned to support long-sequence generation.

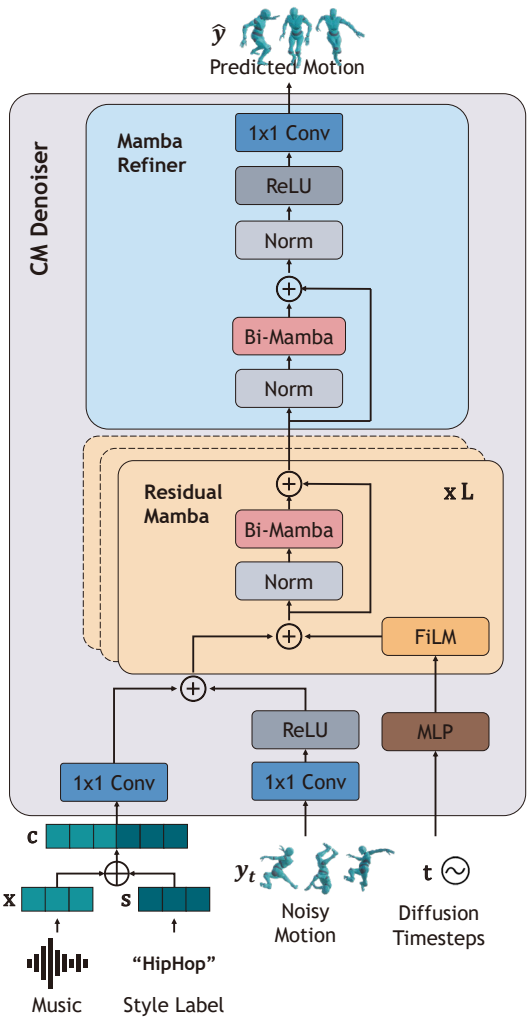


Figure 1: **ChoreoMamba Overview.** A diffusion model conditioned on music and dance style using a bidirectional Mamba backbone. At each timestep, noisy motion is denoised through L Residual Mamba Blocks (RMB) followed by a Mamba Refiner Block (MRB) to generate motion aligned with the input music and style.

4.1 Data Representation

The input audio is represented as music features $\mathbf{x}$. We initially extracted 29 features capturing timbre, pitch, dynamics, and beats, and applied Principal Component Analysis (PCA) to reduce dimensionality, resulting in $\mathbf{x} \in \mathbb{R}^{F\times 6}$, where F is the number of frames. The style vector $\mathbf{s} \in \mathbb{R}^{F\times d}$ is represented as a one-hot encoded dance style label, where d is the number of styles. The conditioning input is formed by concatenating music and style features, $\mathbf{c} = \mathbf{x} \oplus \mathbf{s} \in \mathbb{R}^{F\times(6+d)}$. The output dance motion $\mathbf{y}$ is represented using a 19-joint Motorica skeleton [4] (see Fig. 2) with parent-relative rotations in exponential map representation, along with hip vertical position, root translation, and root rotational velocity, resulting in $\mathbf{y} \in \mathbb{R}^{F\times 61}$ [26, 4]. Root translation is computed by projecting the hip joint onto the xz-plane to capture horizontal movement independent of the hip's vertical displacement.

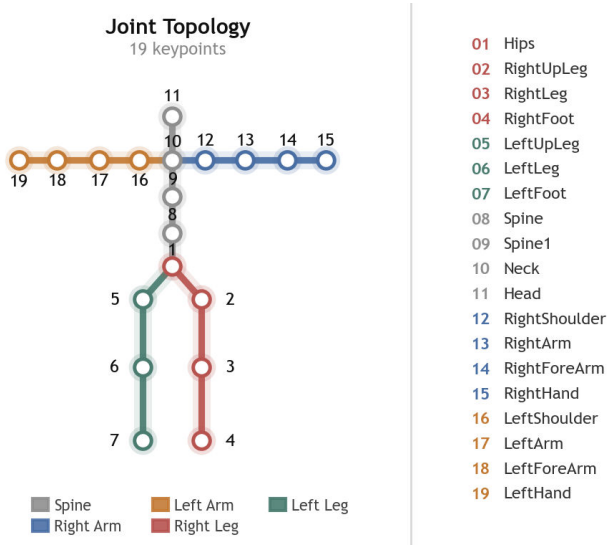


Figure 2: Visualization of the 19-joint skeleton from the Motorica dataset [4] used for motion representation.

4.2 Diffusion Process

Our diffusion model is based on DDPM [27] and LDA [4], consisting of a forward diffusion process and a reverse denoising process. In the forward process, Gaussian noise is gradually added to the dance motion $\mathbf{y}$ through a Markov process, producing a sequence $\{\mathbf{y}_t\}_{t=0}^{T}$ that approaches Gaussian noise:

$$q(\mathbf{y}_t|\mathbf{y}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{y}_{t-1}, (1-\alpha_t)\mathbf{I}) \tag{3}$$

where $\alpha_t \in (0,1)$ follows a decreasing noise schedule [27, 2]. In the reverse process, the denoiser network ϵ_θ predicts the noise at each timestep conditioned on $\mathbf{c}$, and the model iteratively removes the noise to recover the original motion, approximating $p(\mathbf{y}_0|\mathbf{c})$.

The model is trained using the standard noise prediction objective:

$$\mathcal{L}simple = ||\epsilon - \epsilon_\theta(\mathbf{y}_t, t, \mathbf{c})||_2^2 \tag{4}$$

where $\epsilon \sim \mathcal{N}(0,1)$ [27]. Aligned with Alexanderson et al. [4], we find that this objective is sufficient and does not require additional loss terms. This is due to the use of high-quality motion capture data, which provides clean and physically consistent motion sequences with minimal artifacts such as jitter or foot sliding, allowing the diffusion objective to learn stable motion dynamics without auxiliary constraints.

4.3 Bidirectional Mamba

Bidirectional Mamba serves as the core sequence modeling component, capturing both past and future temporal context while maintaining linear computational complexity with respect to sequence length [12, 15]. Fig. 3 shows the details of our bidirectional Mamba layer. Inspired by previous work [7, 25], CM employs a bidirectional Mamba architecture that processes the input signal in both forward and backward directions. The input signal is initially projected into two distinct latent states, $\mathbf{z}_1$ and $\mathbf{z}_2$, through feedforward networks. The latent state $\mathbf{z}_1$ passes through both forward and backward 1D convolutional layers, followed by their corresponding forward and backward SSMs. In parallel, latent state $\mathbf{z}_2$ is processed through a SiLU activation layer and then multiplied separately with the outputs of the forward and backward SSMs. The resulting products are combined by addition and projected back to the output signal using a feedforward network.

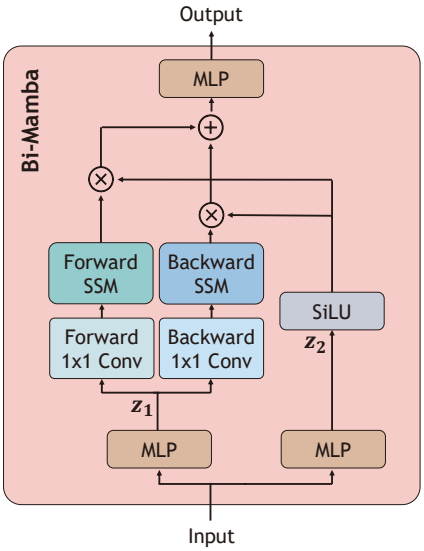


Figure 3: **Bidirectional Mamba.** Our model employs a Mamba architecture that processes sequences in both forward and backward directions [7, 25].

4.4 Denoiser for Long-Sequence Generation

As shown in Fig. 1, the denoiser takes two inputs: the conditioning signal $\mathbf{c}$ and the noisy motion $\mathbf{y}_t$. Both inputs are embedded using 1D convolutional layers, while the diffusion timestep t is embedded using a feedforward network. The network consists of two main components: RMB and MRB, both built around bidirectional Mamba layers.

The RMB begins with a Feature-wise Linear Modulation (FiLM) layer [28], which incorporates the diffusion timestep embedding into the denoiser. The timestep embedding is transformed by an MLP and used to condition the intermediate features, allowing the timestep embedding to adjust the scale and bias of each feature channel. This makes the RMB aware of the current noise level before temporal modeling with the bidirectional Mamba layer. The FiLM-conditioned features are then summed with the conditioning signal and noisy motion embeddings, passed through layer normalization, and processed by a bidirectional Mamba layer with skip connections to stabilize training and improve information propagation. Stacking L RMB layers allows the network to model both short- and long-term temporal dependencies, while FiLM modulation improves motion diversity. The output from the RMB is passed to the MRB, which performs fine-grained refinement of the residual features. The refiner consists of layer normalization and a bidirectional Mamba layer with skip connections, without FiLM modulation, as its primary role is feature refinement rather than conditioning. A final normalization layer, ReLU activation, and a 1D convolution layer map the refined features back to the pose space, producing the denoised output $\hat{\mathbf{y}}$.

To further support long-sequence generation, we adopt a two-stage training strategy in which the model is first pretrained on short sequences and then fine-tuned on longer sequences, allowing it to learn basic motion patterns before adapting to extended motions. The model can also generate sequences longer than those seen during training due to the long-range modeling capability of the Mamba architecture [12, 15, 7].

5 Experiments

5.1 Datasets

We use two datasets: Motorica [4] and FineDance [3]. Motorica contains full-length music paired with long motion sequences across eight dance styles; we excluded the Casual style due to missing music. The final Motorica dataset contains 17,557 seconds of paired music-motion data with an average sequence length of 177 seconds, and all sequences were resampled to 30 FPS in BVH format. FineDance contains both short and full-length dance sequences across 16 genres in SMPL format, which we converted to BVH and filtered to remove noisy samples, resulting in 12,382 seconds of paired music-motion data at 24 FPS.

5.2 Implementation Details

Our denoiser architecture consists of a 16-layer residual bidirectional Mamba network with a hidden dimension of 256 and an SSM dimension of 16. We used a linear noise schedule with 150 diffusion steps and trained the model using the AdamW optimizer [29] with a learning rate of 0.001 and a batch size of 8 on an NVIDIA RTX A6000 GPU. Training was performed in two stages: the model was first trained for 10 epochs on short sequences (20 seconds for FineDance and 60 seconds for Motorica), and then fine-tuned for 5 epochs on long sequences (60 seconds for FineDance and 150 seconds for Motorica), resulting in a total of 15 epochs.

5.3 Baselines

We compare our model with baseline methods capable of generating sequences longer than 10 seconds using a single diffusion model. EDGE [2] generates long sequences by stitching short segments using a transformer architecture, while LDA [4] is a conformer-based model that enables long-sequence generation through a distance-based attention bias. Due to memory constraints, EDGE generates sequences up to 90 seconds and LDA up to 120 seconds. For fair comparison, all models were trained on both datasets using the same motion representation and evaluated across varying sequence lengths.

5.4 Evaluation Metrics

To evaluate model performance, we use Diversity (DIV), Beat Alignment Score (BAS), Jitter error, inference time, and number of parameters. **Diversity** measures variation in the generated motion compared to ground truth and is reported as DIV_k in kinematic feature space and DIV_g in geometric space [30, 1]. The **Beat Alignment Score** measures synchronization between motion and music by computing the average distance between motion beats and music beats [1, 31]. We evaluate motion smoothness using **Jitter Error** instead of Fréchet Inception Distance (FID). While FID measures the distribution difference between generated and ground-truth motions [1], it can be unreliable for long sequences due to limited data samples and feature distribution bias [2]. Jitter error measures motion smoothness by computing the average jerk across all joints [32]. We also report **Inference Time** and **Number of Parameters** to evaluate model efficiency, where inference time

Table 1: Comparison of our method (CM) with diffusion-based models LDA [4] and EDGE [2] across various sequence lengths on the FineDance and Motorica datasets. Metrics are denoted as: ↑ (higher is better) and ↓ (lower is better).

Dataset	Sequence Length	Method	DIV_g ↑	DIV_k ↑	BAS ↑	Jitter Error ↓	Inference Time ↓	# Parameters ↓
FineDance	40s	GT	7.561	16.422	0.161	408.46	-	-
		LDA	3.751	10.861	**0.138**	3247.51	189.31s	71.1M
		EDGE	4.128	10.735	0.135	4148.96	74.54s	60.7M
		CM	**4.376**	**11.680**	0.134	**3176.39**	**65.67s**	**10.8M**
	60s	LDA	3.493	10.790	**0.141**	3154.80	350.03s	71.1M
		EDGE	3.518	8.786	0.138	3257.29	108.45s	60.7M
		CM	**4.324**	**11.455**	0.136	**3067.13**	**75.21s**	**10.8M**
Motorica	90s	GT	7.498	13.277	0.172	123.83	-	-
		LDA	3.847	2.543	0.171	248.34	756.26s	71.1M
		EDGE	3.326	2.388	0.166	309.01	188.89s	60.7M
		CM	**4.152**	**2.793**	**0.179**	**171.29**	**170.72s**	**10.8M**
	120s	LDA	3.833	2.719	0.168	249.99	1233.03s	71.1M
		EDGE	-	-	-	-	-	-
		CM	**4.169**	**3.066**	**0.178**	**170.19**	**192.39s**	**10.8M**

measures generation speed and the number of parameters reflects memory and computational requirements.

6 Results and Discussion

This section presents the experimental results of our model. We first report quantitative results and compare our model with baseline methods using the evaluation metrics described in Sec. 5.4 (Sec. 6.1), followed by qualitative examples of generated dance motions (Sec. 6.2). We then analyze the contribution of key architectural components through ablation studies (Sec. 6.3). Finally, we assess the perceptual quality of the generated motions through a user study (Sec. 6.4).

6.1 Quantitative Results

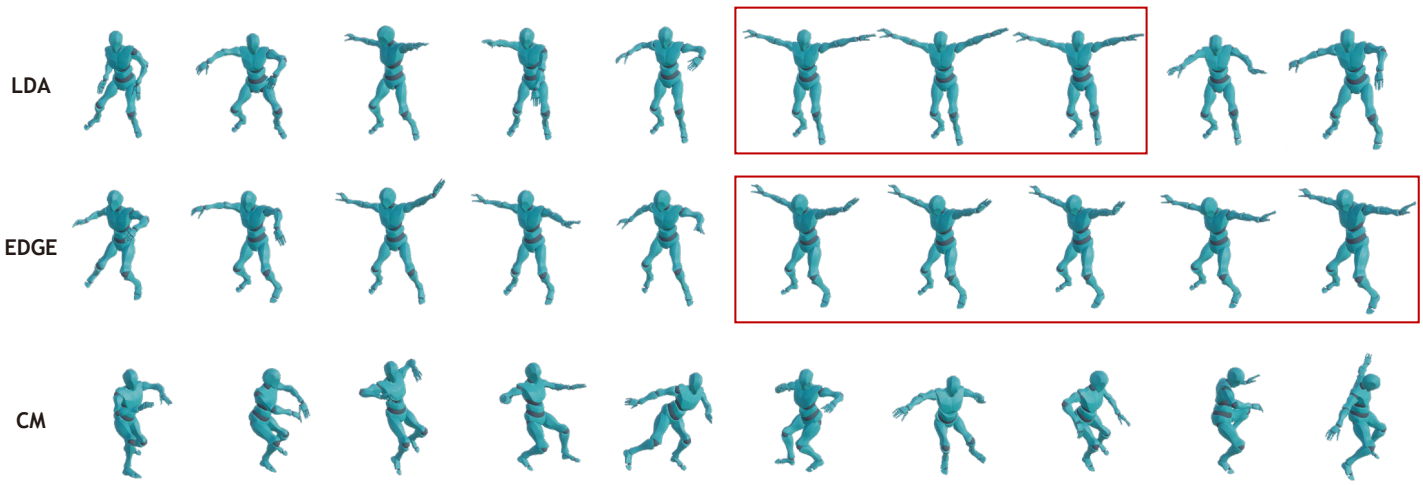
The quantitative evaluation of our method (CM) compared to other methods across different datasets and sequence lengths is shown in Table 1. The results show that CM consistently outperformed baseline models across both the FineDance and Motorica datasets, demonstrating its adaptability to varying sequence lengths while maintaining efficiency. Specifically, CM achieved high DIV_g and DIV_k scores, indicating that it generates more diverse motions compared to the baselines.

A key observation is the dataset-dependent relationship between BAS and Jitter. On FineDance, CM achieved slightly lower BAS than LDA but produced the lowest Jitter error, while baseline models showed consistently high Jitter values. To interpret this trend, we measured ground-truth Jitter and found that FineDance exhibits substantially higher motion jitter than Motorica (see GT Jitter row in Table 1), which introduces spurious beat candidates and artificially inflates BAS. In contrast, on Motorica, where ground-truth Jitter is low, CM achieved both the highest BAS and the lowest Jitter across all evaluated sequence lengths. This highlights an important nuance: a high BAS does not necessarily indicate accurate rhythmic alignment if the motion is excessively jittery, and therefore evaluating both metrics together provides a more reliable assessment of motion quality.

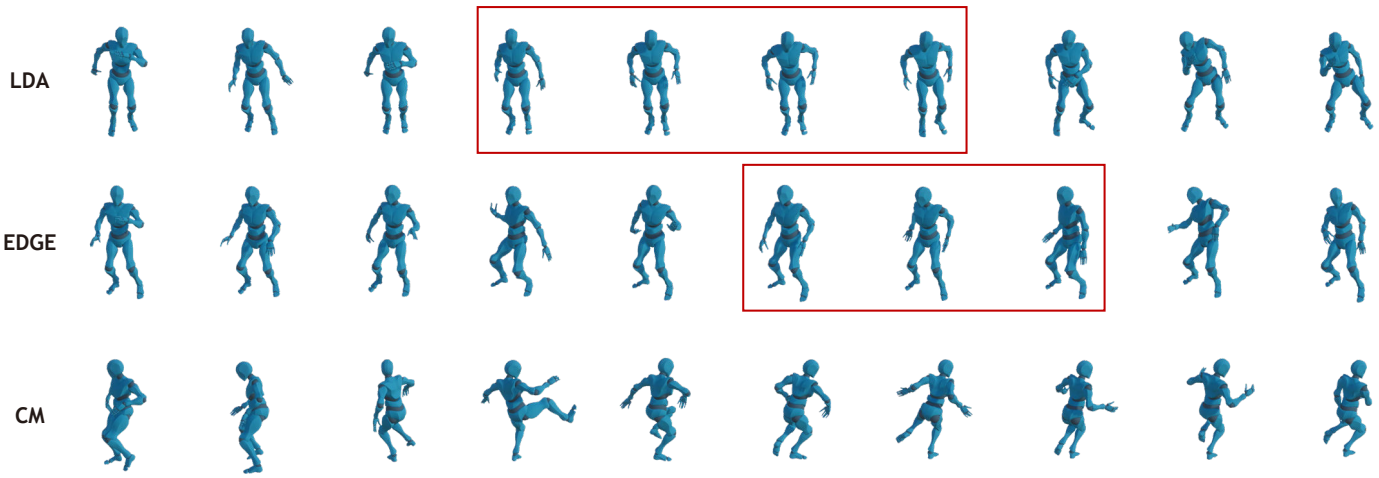
For practical deployment, inference time and model complexity are essential metrics as they directly affect a model's usability in real-world applications. With only 10.8M parameters as shown in Table 1, CM required a significantly lower inference time than LDA and EDGE, particularly for long sequences. This reduced complexity and computational load underscore the efficiency of the Mamba architecture in generating high-quality motion, as they scale linearly with input length, making CM an effective choice for resource-constrained settings.

6.2 Qualitative Results

The qualitative results of our method, shown in the supplementary video, demonstrate its capability to generate long-sequence dance motions that align with both input music and specified style labels. To further assess the robustness of the model, we tested it on diverse in-the-wild music that were not part of the dataset and compared its performance against baselines. In Fig. 4, the results of



(a) Generated dance motions for the music, "Turn Down for What by DJ Snake and Lil Jon".



(b) Generated dance motions for the music, "Up by Karina".

Figure 4: **Visual comparison of dance sequences generated by different methods.** The input is in-the-wild music with a locking dance style. Each column shows the same frame across the three methods for direct comparison. Baseline methods exhibit motion freezing (highlighted in red boxes), while our method produces seamless and dynamic movements in both the upper and lower body.

baseline models show motion freezes due to being trained on short sequences, leading to abrupt pauses and disjointed movements during tempo changes. This limitation highlights our model's advantage in handling long-sequence and diverse inputs.

6.3 Ablation Studies

In this ablation study, we evaluated the impact of key components and configurations of our model architecture on diversity (DIV_g and DIV_k), BAS, and Jitter error metrics. All experiments were performed on models trained on the Motorica dataset, providing insight into the effects of each architectural modification.

FiLM Layer Ablation. Table 2 presents the results of the FiLM layer ablation evaluated on 10-second sequences. The FiLM layer modulates timestep features during training, potentially improving the diversity and stability of the generated motions. Unlike simple bias addition, FiLM enables adaptive feature conditioning by scaling relevant music features, leading to improved diversity and reduced jitter. The results show that incorporating FiLM increases both DIV_g and DIV_k and lowers Jitter error, leading to smoother and more stable motions. However, FiLM slightly reduced BAS, likely due to its modulation across diffusion steps. By applying feature-wise affine transformations at each step, FiLM adjusts representations before passing them to Mamba, introducing subtle shifts in motion timing relative to the music beats. Although the FiLM layer slightly decreased the BAS, this reduction was small

and did not significantly impact overall performance, considering the stability and diversity benefit it introduced. Table 3 further presents the results on 60-second sequences. The model with FiLM achieved the best overall configuration, showing improvements in DIV_k, BAS, and Jitter error, which is generally consistent with the short-sequence evaluation. For a clear comparison of visual quality improvements with the use of FiLM, please refer to the supplementary video.

Table 2: Effect of the FiLM layer on model performance, evaluated on 10-second sequences.

Method	$\text{DIV}_g \uparrow$	$\text{DIV}_k \uparrow$	BAS ↑	Jitter ↓
Ground Truth	7.498	13.277	0.172	123.83
w/o FiLM	5.868	3.233	**0.168**	211.03
w/ FiLM	**6.143**	**3.348**	0.164	**204.68**

Table 3: Effect of the FiLM layer on model performance, evaluated on 60-second sequences.

Method	$\text{DIV}_g \uparrow$	$\text{DIV}_k \uparrow$	BAS ↑	Jitter ↓
Ground Truth	7.498	13.277	0.172	123.83
w/o FiLM	**3.645**	2.385	0.168	446.17
w/ FiLM	2.006	**4.062**	**0.169**	**420.01**

Mamba Refiner Block Ablation. Table 4 presents the results of the Mamba Refiner Block (MRB) ablation evaluated on 10-second sequences. MRB is designed to enhance motion quality during the final stage of generation by refining the motion features learned by the preceding residual Mamba blocks. The model with MRB achieved a higher BAS and reduced Jitter error, highlighting its role in providing rhythmically aligned and stable motion. A slight trade-off was observed in diversity, as the MRB model marginally reduced DIV_g. However, the gain in motion quality appears to outweigh this minor diversity loss as shown in the supplementary video. We also tested an (L-1)-sized RMB with MRB and found that it resulted in lower jitter and improved DIV_g, suggesting that removing the final residual block does not degrade motion stability. These findings highlight MRB's role in refining motion features learned earlier in the model. Table 5 further reports the results on 60-second sequences. The model with MRB achieved the best DIV_k, and BAS, showing that the benefit of MRB is also observed in long sequence generation. Although the Jitter error is slightly higher than the model without MRB, the overall results suggest that MRB improves motion diversity and beat alignment while maintaining comparable stability.

Table 4: Effect of the Mamba Refiner Block (MRB) on model performance, evaluated on 10-second sequences.

Method	$\text{DIV}_g \uparrow$	$\text{DIV}_k \uparrow$	BAS ↑	Jitter ↓
Ground Truth	7.498	13.277	0.172	123.83
w/o MRB	6.143	3.348	0.164	327.51
w/ MRB	6.131	**3.495**	**0.169**	253.24
(L-1) w/ MRB	**6.351**	3.367	0.161	**174.54**

Table 5: Effect of the Mamba Refiner Block (MRB) on model performance, evaluated on 60-second sequences.

Method	$\text{DIV}_g \uparrow$	$\text{DIV}_k \uparrow$	BAS ↑	Jitter ↓
Ground Truth	7.498	13.277	0.172	123.83
w/o MRB	3.320	3.295	0.164	**324.19**
w/ MRB	**6.280**	**4.746**	**0.169**	337.55
(L-1) w/ MRB	3.125	3.620	**0.169**	328.14

While structurally similar to the RMB, the MRB serves a distinct purpose within the network. Throughout the sequence, each residual block contributes to the overall motion by stabilizing gradient flow and capturing broad contextual features. However, as each RMB layer sequentially adds its output back to the input across the L layers, errors may accumulate. The MRB, positioned near the output layer, performs a final bidirectional scan across the accumulated outputs to refine and selectively emphasize the most critical information from these layered states. This final stage helps address any inconsistencies left by the residual blocks, providing a degree of refinement that the residual blocks alone are not optimized to achieve.

Number of Residual Mamba Blocks Table 6 shows the effect of varying the number of RMBs in the architecture. Optimal performance is achieved with 16 layers, where the highest diversity and BAS scores were observed, indicating that this configuration strikes the best balance across all metrics. In particular, Jitter error is minimized with 12 layers, but this comes at the cost of slightly reduced diversity. Conversely, increasing the number of layers to 20 resulted in a decline in both diversity and BAS scores, suggesting that exceeding 16 layers may introduce redundancy or excessive smoothing, which can hinder the model's expressiveness and performance.

Sequence Length for Pretraining Table 7 presents the impact of pretraining sequence lengths by comparing models pretrained on 60-second sequences with those pretrained on 10-second sequences. Both models were subsequently fine-tuned and evaluated on 150-second sequences for this ablation study. Models pre-

Table 6: Effect of varying the number of Residual Mamba Blocks on model performance, evaluated on 10-second sequences.

Method	$DIV_g \uparrow$	$DIV_k \uparrow$	BAS ↑	Jitter ↓
8 Layers	6.131	3.495	0.164	253.24
12 Layers	6.343	3.514	0.164	**221.26**
16 Layers	**6.355**	**3.973**	**0.166**	245.68
20 Layers	6.097	3.192	0.16	241.07

trained on 60-second sequences consistently outperformed those pretrained on 10-second sequences across all metrics. These findings highlight the advantages of using long pretraining sequences to enhance diversity, alignment, and naturalness in generation.

Table 7: Effect of pretraining sequence length on model performance after fine-tuning and evaluation on 150-second sequences. FT denotes the fine-tuned model initialized from the corresponding pretraining length.

Method	$DIV_g \uparrow$	$DIV_k \uparrow$	BAS ↑	Jitter ↓
FT (10s)	3.876	2.423	0.176	175.35
FT (60s)	**4.169**	**3.066**	**0.179**	**170.19**

6.4 User Study

We conducted a user study to assess Beat-Music Alignment (BMA), Style-Motion Alignment (SMA), naturalness, and overall dance quality, using a Mean Opinion Score for each metric. The study included 25 participants, composed of 48% dancers and 52% non-dancers. Each participant evaluated 12 dance motions, with four motions per method. The participants rated each dance motion's quality on a 5-point Likert scale. The results, summarized in Table 8, show that the participants preferred CM across all metrics. This suggests that CM is effective at synchronizing dance motions with the music while also capturing stylistic nuances and maintaining a natural motion quality.

Table 8: Results of the user study on different metrics.

Method	BMA ↑	SMA ↑	Naturalness ↑	Overall ↑
LDA	3.85	3.7	3.68	3.743
EDGE	3.47	3.48	3.3	3.42
CM	**3.96**	**3.72**	**3.99**	**3.89**

7 Limitations

Our current model is limited to generating sequences up to 180 seconds due to the maximum sample lengths available in the dataset. Although the model can train and generate longer sequences without memory constraints, the lack of sufficient long-duration training data restricts its ability to learn temporal patterns over extended durations. Consequently, performance on longer sequences is constrained by the available training data. Future work could address this limitation by expanding datasets to include more sequences longer than 180 seconds or by developing data augmentation techniques that extend sequence duration while preserving alignment between music and motion.

8 Conclusion and Future Work

We introduced a bidirectional Mamba diffusion model for efficient long-sequence dance generation conditioned on music and style. Our model was evaluated across multiple sequence lengths using diversity, alignment, naturalness, and efficiency metrics, and outperformed previous state-of-the-art methods while generalizing well to in-the-wild inputs and multiple dance styles. Future work includes refining loss functions to better balance motion stability and rhythmic consistency, and extending the model to full-length music and choreographic analysis.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00333478).

References

[1] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "Ai choreographer: Music conditioned 3d dance generation with aist++," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 381–13 392.

[2] J. Tseng, R. Castellon, and C. K. Liu, "Edge: Editable dance generation from music," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 448–458.

[3] R. Li, J. Zhao, Y. Zhang, M. Su, Z. Ren, H. Zhang, Y. Tang, and X. Li, "Finedance: A fine-grained choreography dataset for 3d full body dance generation," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 10 200–10 209.

[4] S. Alexanderson, R. Nagy, J. Beskow, and G. E. Henter, "Listen, denoise, action! audio-driven motion synthesis with

diffusion models," *ACM Trans. Graph.*, vol. 42, no. 4, jul 2023. [Online]. Available: https://doi.org/10.1145/3592458

[5] R. Li, Y. Zhang, Y. Zhang, H. Zhang, J. Guo, Y. Zhang, Y. Liu, and X. Li, "Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives," pp. 1524–1534, 2024.

[6] N. Le, T. Do, K. Do, H. Nguyen, E. Tjiputra, Q. D. Tran, and A. Nguyen, "Controllable group choreography using contrastive diffusion," *ACM Trans. Graph.*, vol. 42, no. 6, dec 2023. [Online]. Available: https://doi.org/10.1145/3618356

[7] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *ArXiv*, vol. abs/2401.09417, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267028142

[8] 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, dec 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480570

[9] X. Gao, L. Hu, P. Zhang, B. Zhang, and L. Bo, "Dancemeld: Unraveling dance phrases with hierarchical latent codes for music-to-dance synthesis," *ArXiv*, vol. abs/2401.10242, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:267060925

[10] M. Petrovich, O. Litany, U. Iqbal, M. J. Black, G. Varol, X. Bin Peng, and D. Rempe, "Multi-track timeline control for text-driven 3d human motion generation," pp. 1911–1921, 2024.

[11] M. Li, C. Zhai, S. Yao, Z. Xie, K. Chen, and Y.-G. Jiang, "Infinite motion: Extended motion generation via long text instructions," *ArXiv*, vol. abs/2407.08443, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271097653

[12] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2024. [Online]. Available: https://arxiv.org/abs/2312.00752

[13] A. Gu, K. Goel, and C. R'e, "Efficiently modeling long sequences with structured state spaces," *ArXiv*, vol. abs/2111.00396, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:240354066

[14] A. Gupta, A. Gu, and J. Berant, "Diagonal state spaces are as effective as structured state spaces," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2024.

[15] R. Waleffe, W. Byeon, D. Riach, B. Norick, V. A. Korthikanti, T. Dao, A. Gu, A. Hatamizadeh, S. Singh, D. Narayanan, G. Kulshreshtha, V. Singh, J. Casper, J. Kautz, M. Shoeybi, and B. Catanzaro, "An empirical study of mamba-based language models," *ArXiv*, vol. abs/2406.07887, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270391285

[16] C. Fu, Y. Wang, J. Zhang, Z. Jiang, X. Mao, J. Wu, W. Cao, C. Wang, Y. Ge, and Y. Liu, "Mambagesture: Enhancing co-speech gesture generation with mamba and disentangled multi-modality fusion," pp. 10 794–10 803, 2024.

[17] Z. Zhang, A. Liu, I. Reid, R. Hartley, B. Zhuang, and H. Tang, "Motion mamba: Efficient and long sequence motion generation," pp. 265–282, 2025.

[18] X. Wang, Z. Kang, and Y. Mu, "Text-controlled motion mamba: Text-instructed temporal grounding of human motion," *ArXiv*, vol. abs/2404.11375, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269187718

[19] Z. Zhang, A. Liu, Q. Chen, F. Chen, I. Reid, R. Hartley, B. Zhuang, and H. Tang, "Infinimotion: Mamba boosts memory in transformer for arbitrary long motion generation," *ArXiv*, vol. abs/2407.10061, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271212240

[20] F. Jafari, S. Berretti, and A. Basu, "Jambatalk: Speech-driven 3d talking head generation based on hybrid transformer-mamba model," *ArXiv*, vol. abs/2408.01627, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271710434

[21] F. Zhang, N. Ji, F. Gao, B. Zhao, J. Wu, Y. Jiang, H. Du, Z. Ye, J. Zhu, W. Zhong, L. Yan, and X. Ma, "Dim-gesture: Co-speech gesture generation with adaptive layer normalization mamba-2 framework," *ArXiv*, vol. abs/2408.00370, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271600755

[22] Z. Qian, Z. Xiao, Z. Wu, D. Yang, M. Li, S. Wang, S. Wang, D. Kou, and L. Zhang, "Smcd: High realism motion style transfer via mamba-based diffusion," *ArXiv*,

vol. abs/2405.02844, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269605365

[23] S. Park, I. Choi, D. Soon, Y. Jeon, and K. Joo, "Not like transformers: Drop the beat representation for dance generation with mamba-based diffusion model," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2026, pp. 1767–1776.

[24] Y. Fu, C. Chen, and Y. Yu, "Lamamba-diff: Linear-time high-fidelity diffusion models based on local attention and mamba," *ArXiv*, vol. abs/2408.02615, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:271710169

[25] M. H. Erol, A. Senocak, J. Feng, and J. S. Chung, "Audio mamba: Bidirectional state space model for audio representation learning," *IEEE Signal Processing Letters*, vol. 31, pp. 2975–2979, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270258415

[26] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *J. Graphics, GPU, & Game Tools*, vol. 3, pp. 29–48, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:9489978

[27] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[28] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville, "Film: visual reasoning with a general conditioning layer," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[29] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *ArXiv*, vol. abs/1711.05101, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3312944

[30] L. Siyao, W. Yu, T. Gu, C. Lin, Q. Wang, C. Qian, C. C. Loy, and Z. Liu, "Bailando: 3d dance generation by actor-critic gpt with choreographic memory," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 040–11 049, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247627867

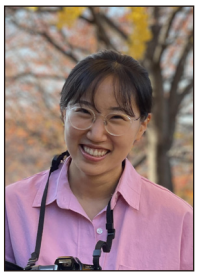
[31] V. Tan, J. Nam, J. Nam, and J. Noh, "Motion to dance music generation using latent diffusion model," in *SIGGRAPH Asia 2023 Technical Communications*, ser. SA '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3610543.3626164

[32] X. Yi, Y. Zhou, and F. Xu, "Transpose: real-time 3d human translation and pose estimation with six inertial sensors," *ACM Trans. Graph.*, vol. 40, no. 4, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459786

〈저자소개〉

Vanessa Tan

- 2010 – 2015: Bachelor’s Degree in Electronics and Communications Engineering, University of the Philippines Diliman
- 2015 – 2016: Associate Software Engineer, Accenture Inc.
- 2016 – 2018: Research Associate, Ubiquitous Computing Lab
- 2016 – 2019: Master’s Degree in Electrical Engineering, University of the Philippines Diliman
- 2019 – 2021: University Researcher, Philippine Space Agency
- 2025: Research Intern, Sony CSL
- 2022 – Present: Ph.D. in Culture Technology, KAIST
- Areas of Interest: Character Animation, Music Processing, Generative Models
- https://orcid.org/0009-0001-8174-6909



김혜민

- 2015 – 2020: Bachelor’s Degree in Computer Science and Engineering, Ewha Womans University
- 2019 – 2020: Research Intern, Visual Media Lab, KAIST
- 2020 – 2022: Master’s Degree in Culture Technology, KAIST
- 2022 – Present: Ph.D. in Culture Technology, KAIST
- Areas of Interest: Character Animation, Motion Stitching, Motion Inbetweening
- https://orcid.org/0009-0008-1016-4793

최수진

- 2016 – 2020: Bachelor’s Degree in Computer Education, Sungkyunkwan University
- 2020 – 2022: Master’s Degree in Culture Technology, KAIST
- 2022 – Present: Ph.D. in Culture Technology, KAIST
- Areas of Interest: Character Animation, Motion Retargeting, Automatic Rigging and Skinning
- https://orcid.org/0000-0002-0832-6545

노준용

- 1990 – 1994: Bachelor’s degree in Electrical Engineering, University of Southern California
- 1994 – 1996: Master’s in Computer Engineering, University of Southern California
- 1996 – 2002: Ph.D. in Computer Science, University of Southern California
- 2003 – 2006: Rhythm and Hues Studio, Graphics Scientist
- 2006 – Present: Professor, KAIST Graduate School of Cultural Technology
- 2011 – Present: KAIST Chair Professor
- 2012 – Present: Adjunct Professor, School of Computing, KAIST
- 2016 – 2020: Director of KAIST Cultural Technology Research Institute
- 2016 – 2020: Head of Department, Graduate School of Culture and Technology, KAIST
- 2026 – Present: Adjunct Professor, College of AI, KAIST
- Areas of interest: Computer Graphics, Computer Vision, Facial Modeling, Facial Animation, Character Animation, Image & Video Manipulation/Generation
- https://orcid.org/0000-0003-1925-3326