Smooth and Flexible Camera Movement Synthesis via Temporal Masked Generative Modeling

Chenghao Xu[1], **Guangtao Lyu**[1], **Jiexi Yan**[2*], **Muli Yang**[3], **Cheng Deng**[1*]
[1] School of Electronic Engineering, Xidian University, Xi'an, Shaanxi, China,
[2] School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, China,
[3] Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
{chx,guangtaolyu}@stu.xidian.edu.cn,
{jxyan1995,muliyang.xd,chdeng.xd}@gmail.com

Abstract

In dance performances, choreographers define the visual expression of movement, while cinematographers shape its final presentation through camera work. Consequently, the synthesis of camera movements informed by both music and dance has garnered increasing research interest. While recent advancements have led to notable progress in this area, existing methods predominantly operate in an offline manner—that is, they require access to the entire dance sequence before generating corresponding camera motions. This constraint renders them impractical for real-time applications, particularly in live stage performances, where immediate responsiveness is essential. To address this limitation, we introduce a more practical yet challenging task: online camera movement synthesis, in which camera trajectories must be generated using only the current and preceding segments of dance and music. In this paper, we propose TemMEGA (Temporal Masked Generative Modeling), a unified framework capable of handling both online and offline camera movement generation. TemMEGA consists of three key components. First, a discrete camera tokenizer encodes camera motions as discrete tokens via a discrete quantization scheme. Second, a consecutive memory encoder captures historical context by jointly modeling long- and short-term temporal dependencies across dance and music sequences. Finally, a temporal conditional masked transformer is employed to predict future camera motions by leveraging masked token prediction. Extensive experimental evaluations demonstrate the effectiveness of our TemMEGA, highlighting its superiority in both online and offline camera movement synthesis.

1 Introduction

Recent advances in image generation have significantly enhanced visual storytelling in performance arts [26; 43; 42]. In dance performances, camera work is pivotal in shaping the audience's perception and interpretation of the choreography [28; 30; 39; 3; 46]. By employing multiple camera angles and transitions, producers can better capture key dance movements, offering a more immersive storytelling experience. Additionally, creative techniques such as quick cuts, slow motion, and dolly shots enhance visual impact and introduce novelty, thereby increasing the performance's overall appeal. However, the movement of the camera is influenced by several factors, including the music and the choreography itself. Moreover, effective dance cinematography requires a variety of shot types and a focus on human-centered elements. As a result, the automatic generation of camera movements based on music and dance remains a compelling yet complex challenge.

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

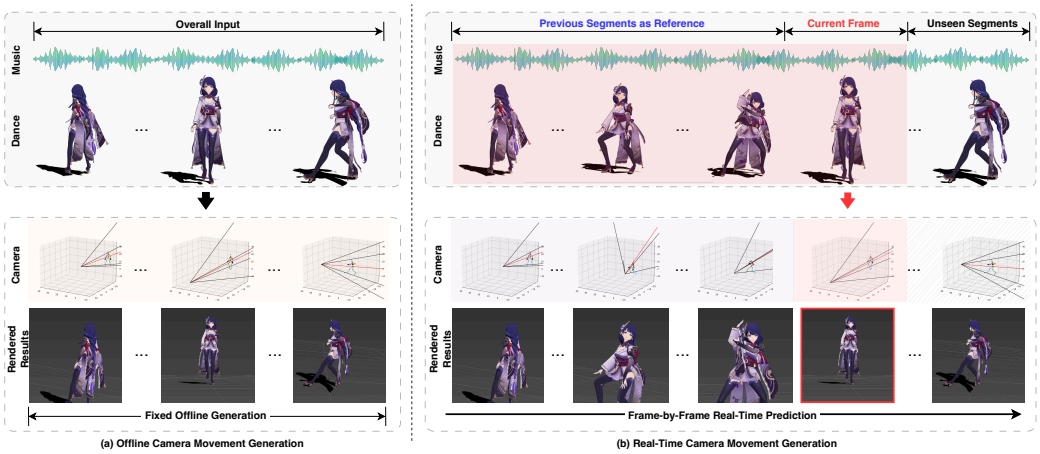

Figure 1: **Illustration of the traditional offline camera movement generation task and our proposed real-time camera movement synthesis.** (a) Offline Camera movement generation: The entire dance video with music is available to synthesize the corresponding camera movements; (b) Online camera movement generation: Camera movements are generated frame by frame. For the current frame, we employ the previous and current segments of dance and music as input to synthesize the corresponding camera movement.

Previously, significant attention had been given to camera planning and control [27; 45], primarily in gaming and film scenes. Recently, several methods have aimed to tackle the more challenging task of dance camera synthesis. Among these, DanceCamera3D [37] introduced the first 3D dance-camera-music dataset (DCM) and demonstrated the feasibility of synthesizing camera movements driven by music and dance. Additionally, Cine-AI [40] simplifies the problem by reducing it from 3D to 2D, excluding the camera's roll and pitch orientation. This simplification significantly diminishes the expressiveness of the camera movements and reduces the complexity of the task. Moreover, DanceCamAnimator [38] integrates human animation knowledge into the problem of music- and dance-driven camera synthesis, employing this knowledge to generate 3D camera movements by following animators' hierarchical camera-making procedures.

Although significant progress [7; 9; 12; 14; 15; 18; 19; 4] has been achieved in camera movement synthesis, a major limitation persists in the real-world application of these offline methods due to their reliance on having access to the entire dance video as input, as shown in Figure 1(a). In practice, online requirements must be met in the camera movement generation process, meaning that camera movements need to be swiftly generated during live stage performances, where the complete dance video is not available; instead, only prior segments of the performance are accessible. Consequently, we focus on a more practical yet challenging task, namely, online camera movement synthesis illustrated in Figure 1(b).

In this paper, we introduce TemMEGA, a novel Temporal Masked Generative Modeling framework for both online and offline camera movement synthesis. Our approach is built upon three key components. First, the discrete camera tokenizer (DCT) is trained using the vector quantized variational autoencoder (VQ-VAE). The DCT transforms and quantizes raw camera movement data into a sequence of discrete motion tokens in latent space, based on a camera codebook. To more effectively capture the temporal context of dance and music segments, we introduce the consecutive memory encoder (CME), which provides a more accurate history summary by jointly modeling long- and short-term temporal memories. Specifically, long- and short-term segments of dance motions and music are encoded into fixed tokens. Finally, we mask the tokens to be predicted and employ a conditional masked transformer (CMT) to predict the masked tokens in real-time, conditioned on both the unmasked tokens and the long- and short-term memory. Extensive comparative and ablation studies on public datasets validate the effectiveness of our framework.

In summary, our main contributions include:

- We introduce the practical task of online camera movement synthesis, with the potential to significantly expand applications of camera movement generation, particularly in live stage performances.
- We propose a novel temporal masked generative modeling framework, TemMEGA for smooth and flexible generate camera movement synthesis in both online and offline manner. Our TemMEGA consists of three main components, *i.e*. discrete camera tokenizer, consecutive memory encoder, and conditional masked transformer.
- Comprehensive experiments on public datasets demonstrate that our method achieves state-of-the-art performance, confirming its effectiveness.

2 Related Work

2.1 Camera Control and Planning

Automatic cinematography has gained significant attention due to the expertise and labor required for manually producing film-like videos, despite the importance of artistic video content in media, entertainment, and gaming industries. Jiang *et al*. [18] propose extracting camera behaviors from film clips for re-application in virtual environments. Similarly, Rao *et al*. [28] generate dynamic storyboards from story and camera scripts, while Wu *et al*. [39] develop a GAN-based controller to produce actor-driven camera movements considering spatial, emotional, and aesthetic factors. Rucks *et al*. [30] introduce CamerAI to replicate chase camera techniques in third-person games, and Evin *et al*. [7] present Cine-AI to simulate movie directors' cinematographic techniques for enhancing game cutscenes. In the domain of aerial cinematography, studies [14; 16; 13; 10] focus on automating drone camera movements based on artistic principles. However, controlling cameras for dance sequences is more complex due to the need to synchronize with music and dance motions.

To address this, Wang et al. [37] introduced the 3D dance-camera-music dataset DCM and developed DanceCamera3D, a transformer-based diffusion model for dance camera synthesis. Nonetheless, it overlooks the mix of continuous shots and abrupt transitions in dance cinematography. DanceCamAnimator [38] improves on this by integrating animator knowledge into a three-stage process—keyframe detection, keyframe synthesis, and tween function prediction—offering precise control over variable-length sequences.

2.2 Dance Synthesis

Music-conditioned 3D dance generation merges dance and machine learning, producing dance sequences that align with music's melody and rhythm. Existing approaches are split into two types: retrieval-based and direct generation methods. Retrieval-based approaches [24; 8] segment dances into fixed-length pieces to match the music structure, but are limited by BPM and fixed segment lengths, making synchronization challenging. Direct generation methods [1; 32; 33; 41] address these limitations by generating dance movements from scratch.

Recent advances in deep learning have led to the rise of diffusion-based and discrete generation techniques. Diffusion models, known for their noise-refinement process, generate coherent dance sequences aligned with musical cues. For instance, EDGE [33] employs conditional diffusion models to create dance movements using Jukebox [6] for audio feature extraction. Discrete generation follows a two-stage process. First, VQ-VAE [35] transforms dance movements into compact, discrete features. Next, natural language processing techniques, such as autoregressive and mask modeling, generate and reconstruct dance sequences, ensuring temporal coherence and fluidity while synchronizing with the music.

3 Method

3.1 Problem Formulation

The existing music-dance-to-camera synthesis methods [37; 38] take a dance video with T frames of music features $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_T\}$ and dance motions $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_T\}$ as input conditions, to generate camera movement sequence $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_T\}$, which is a offline paradigm.

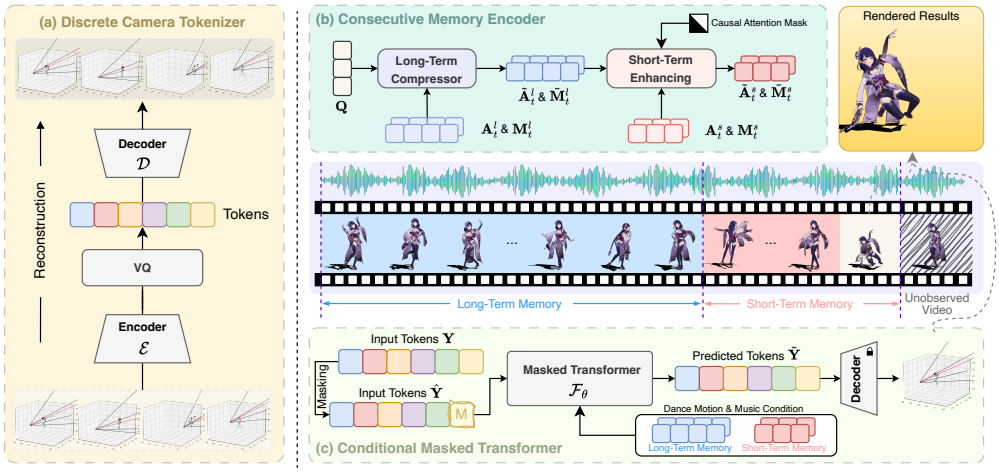

Figure 2: **The framework of our proposed TemMEGA.** Our method can be mainly divided in three components, *i.e*. discrete camera tokenizer (DCT), consecutive memory encoder (CME) and conditional masked transformer (CMT).

Considering the more practical setting, *i.e*. online music-dance-to-camera synthesis as illustrated in Figure 1(b), we generate the current camera movement $\mathbf{c}_t$ by taking the current and previous t frames of music features $\mathcal{A}_t = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_t\}$ and dance motions $\mathcal{M}_t = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_t\}$.

Specifically, we follow FACT [21] to extract music features, denoted as $m_t \in \mathbb{R}^{35}$, using Librosa [23]. For dance motions and camera movements, we adopt the approach of DanceCamera3D [37], using the global positions of 60 human joints, represented as $\mathbf{m}_i \in \mathbb{R}^{60\times 3}$, and MMD format camera representation in polar coordinates, denoted as $\mathbf{c}_i \in \mathbb{R}^{3+3+1+1}$. This includes the global position of the reference point, the camera's rotation and distance relative to the reference point, and the camera's field of view (FOV).

3.2 Temporal Masked Generative Modeling

Our objective is to develop a unfied solution for both offline and online music-dance-to-camera synthesis that efficiently generates camera movements in real time by utilizing previous and current dance and music segments. To accomplish this, we propose a novel temporal masked generative modeling (TemMEGA) framework to replace previous diffusion-based methods, which are limited to offline operation, i.e., relying on the entire dance video as input to generate the corresponding camera movement sequence.

As illustrated in Figure 2, our framework consists of three key components. First, the Discrete Camera Tokenizer (DCT) is designed to transform camera movements into a sequence of discrete camera tokens while preserving rich correlated information about the camera movements. Second, the Consecutive Memory Encoder (CME) enhances the conditional information (previous and current music and dance motion) and provides a more accurate history summary of the temporal condition by compressing long- and short-term memory in a segment-based manner. Finally, the Temporal Conditional Masked Transformer (CMT) is trained to predict masked current camera tokens based on the pre-computed long- and short-term memories of both music and dance motion.

Discrete Camera Tokenizer. To effectively facilitate the synthesis of camera movements, we pre-train a discrete camera tokenizer (DCT). This is achieved using the Vector Quantized Variational Autoencoder (VQ-VAE) architecture [35; 41], which enables the generation of discrete representations of camera shot data through the quantization of encoder outputs into discrete tokens, mapped to entries or codes from a learned codebook via vector quantization. Our DCT framework comprises a camera encoder $\mathcal{E}$ and a camera decoder $\mathcal{D}$. The objective of vector quantization is defined as follows:

$$\mathcal{L}_{VQ} = ||\text{sg}[\mathcal{E}(\mathbf{c}_i)] - \hat{\boldsymbol{\nu}}_i||_2^2 + \beta||\mathcal{E}(\mathbf{c}_i) - \text{sg}[\hat{\boldsymbol{\nu}}_i]||_2^2. \quad (1)$$

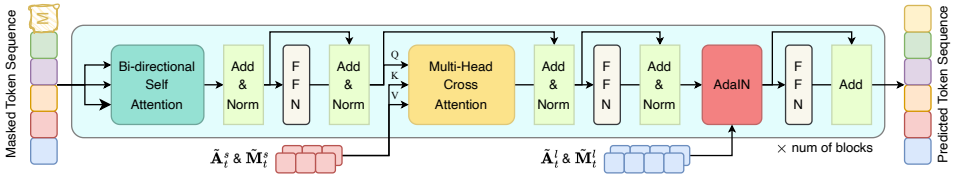

Figure 3: **Architecture of the masked transformer.** The model is a multi-layer transformer with a bidirectional attention structure. It takes as input the masked camera tokens, along with the long- and short-term memories of music and dance motion, to predict the camera tokens. The long- and short-term memories of music and dance motion are integrated into the network at various stages through self-attention layers and AdaIN layers, respectively.

Here, for the i-th latent feature $\boldsymbol{\nu}_i$, the estimated embedding $\hat{\boldsymbol{\nu}}_i$ can be found by searching the nearest embedding in the codebook $\mathcal{X}$ through the quantization process $Q(\cdot)$:

$$\hat{\boldsymbol{\nu}}_i = Q(\boldsymbol{c}_i) := \underset{\boldsymbol{x}_k \in \mathcal{X}}{\arg\min} \|\boldsymbol{\nu}_i - \boldsymbol{x}_k\|_2. \tag{2}$$

Based on the estimation latent representation $\hat{\boldsymbol{V}} = [\hat{\boldsymbol{\nu}}_1, \hat{\boldsymbol{\nu}}_2, \cdots, \hat{\boldsymbol{\nu}}_T]$, the reconstructed camera movements can be produced by the decoder $\boldsymbol{D}(\cdot)$, *i.e.*, $\tilde{\boldsymbol{X}} = \boldsymbol{D}(\hat{\boldsymbol{V}})$.

Additionally, we incorporate moving averages during codebook updates and reset inactive codebooks, techniques commonly used to improve codebook utilization in VQ-VAE. These strategies enable the robust and efficient transformation of camera movements into a sequence of discrete camera tokens.

Consecutive Memory Encoder. To more effectively handle the previous conditional information, we draw inspiration from some existing works [44; 36] and introduce a Consecutive Memory Encoder (CME) to separate the long- and short-term memories of the preceding music and dance motion segments. This allows for modeling short-term context while extracting meaningful correlations from the long-term history. By doing so, we compress the long-term history without losing important fine-grained details.

As illustrated in Figure 2(b), we explicitly divide the previous music and dance motion segments into long- and short-term memories. Specifically, for the prediction of the t-th frame, the short-term memory retains only a limited number of recent frames of music and dance motion, denoted as $\mathbf{A}_t^s = \{\mathbf{a}_i\}_{i=t-L_s+1}^{t}$ and $\mathbf{M}_t^s = \{\mathbf{m}_i\}_{i=t-L_s+1}^{t}$, respectively. Here, L_s represents the length of the short-term memory. The other memory, referred to as long-term memory, stores features from frames further removed from the current time. It is defined as $\mathbf{A}_t^l = \{\mathbf{a}_i\}_{i=t-L_s-L_l+1}^{t-L_s}$ and $\mathbf{M}_t^l = \{\mathbf{m}_i\}_{i=t-L_s-L_l+1}^{t-L_s}$, where L_l denotes the length of the long-term memory, which is significantly longer than the short-term memory.

To further improve the quality of the compressed long-term memory and enhance the short-term memory, we compress and abstract the long-term memory into a fixed-length latent representation, which is then integrated into the short-term memory. Specifically, we first divide $\mathbf{A}_t^l$ and $\mathbf{M}_t^l$ into non-overlapping memory segments. Next, we apply a weight-shared transformer decoder block with K learnable tokens as the long-term memory queries to query each segment. Through this process, the memory segments are transformed into K segment-level abstract features. Each feature is then average-pooled into a single vector, and these vectors are concatenated to form the compressed long-term segmented memory. Finally, we input the concatenated vectors into two transformer encoder blocks to obtain the final compressed long-term memories, $\tilde{\mathbf{A}}_t^l$ and $\tilde{\mathbf{M}}_t^l$.

To further enhance the short-term memory $\mathbf{A}_t^s$ and $\mathbf{M}_t^s$, we utilize it as a query to retrieve relevant context from the compressed long-term memory. A transformer causal decoder block is employed to aggregate the compressed long-term memory into the short-term memory $\tilde{\mathbf{A}}_t^s$ and $\tilde{\mathbf{M}}_t^s$. The resulting compressed long-term memory, along with the enhanced short-term memory, is then fed into the subsequent Temporal Conditional Masked Transformer (CMT) as conditional input.

Temporal Conditional Masked Transformer. As shown in Figure 2(c), we design a bidirectional masked transformer $\mathcal{F}_\theta$, parameterized by θ, to model the camera tokens. Inspired by MAGE [22], the camera tokens $\mathbf{Y}$ are first obtained by passing the encoder output through a vector quantizer

Table 1: Quantitative results on the DCM dataset in the online setting. The best results are indicated as **Bold**, and the second ones are indicated as Underline. - denotes that the self-comparison is meaningless. * denotes that we retrain and retest the method in the online setting.

Method	Quality		Diversity		Dancer Fidelity		User Study
	$\text{FID}_k \downarrow$	$\text{FID}_s \downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_s \uparrow$	DMR $\downarrow$	LCD $\downarrow$	TemMEGA WinRate $\uparrow$
GT	-	-	3.275	1.731	0.00142	-	32.15%±3.07%
DanceCamera3D* [37]	4.634	0.761	1.488	1.109	0.0066	0.197	83.43%±2.36%
TemMEGA w/o $\tilde{\mathbf{A}}_t^l$&$\tilde{\mathbf{M}}_t^l$	4.367	0.618	1.525	1.64	0.0045	0.180	61.36%±1.96%
TemMEGA	**4.025**	**0.599**	**1.589**	**1.187**	**0.0035**	**0.177**	-

Table 2: Quantitative results on the DCM dataset in the offline setting. The best results are indicated as **Bold**, and the second ones are indicated as Underline. - denotes that the self-comparison is meaningless.

Method	Quality		Diversity		Dancer Fidelity		User Study
	$\text{FID}_k \downarrow$	$\text{FID}_s \downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_s \uparrow$	DMR $\downarrow$	LCD $\downarrow$	TemMEGA WinRate $\uparrow$
GT	-	-	3.275	1.731	0.00142	-	40.35%±2.62%
DanceRevolution [17]	10.267	2.368	1.491	1.118	0.0062	0.154	88.14%±2.05%
FACT [21]	5.205	0.960	1.505	1.007	0.0899	0.310	85.64%±1.61%
DanceCamera3D [37]	3.749	0.280	1.631	1.326	0.0025	0.147	73.64%±2.67%
DanceCamAnimator [38]	3.453	0.268	**3.140**	1.293	0.0022	0.152	65.64%±4.54%
TemMEGA	**3.237**	**0.255**	1.961	**1.347**	**0.0020**	**0.141**	-

during training. We then randomly mask out a varying fraction of the sequence elements, replacing the tokens with a special [MASK] token. The masked camera token sequence $\hat{\mathbf{Y}}$, along with the long- and short-term memories of music and dance motion $\tilde{\mathbf{A}}_t^l, \tilde{\mathbf{M}}_t^l, \tilde{\mathbf{A}}_t^s, \tilde{\mathbf{M}}_t^s$, serve as the inputs for our bidirectional masked transformer $\mathcal{F}_\theta$. Mathematically, the masked transformer $\mathcal{F}_\theta$ is optimized by minimizing the negative log-likelihood of the target predictions:

$$\mathcal{L}_{\text{CMT}} = \sum -\log \mathcal{F}_\theta(\tilde{\mathbf{Y}} | \hat{\mathbf{Y}}, \tilde{\mathbf{A}}_t^l, \tilde{\mathbf{M}}_t^l, \tilde{\mathbf{A}}_t^s, \tilde{\mathbf{M}}_t^s). \tag{3}$$

To effectively integrate the conditional input, we carefully design the masked transformer, as depicted in Figure 3. Following GestureDiffuCLIP [2], in our masked transformer, we extract long-term memory features by compressing and enhancing them, obtaining a fixed-length token representation. These long-term (historical) features capture macro-level information such as the overall style of the music. To incorporate this non-sequential macro-level information into the generation process, we leverage the style transfer capability of the adaptive instance normalization (AdaIN). In contrast, short-term memory features maintain a direct temporal correspondence with the sequence to be generated. Therefore, to ensure the generated sequence aligns temporally with the input short-term information, it is crucial to establish strong temporal interactions between the short-term memory and the generated sequence. To achieve this, we employ a cross-attention mechanism. Moreover, the bidirectional self-attention mechanism enables the prediction of masked tokens by leveraging context from both directions.

Inference. During the inference phase in the online setting, we predict only the result at the current time step t. Specifically, we append a [MASK] token following the corresponding camera tokens of the short-term memory and utilize both the long- and short-term memories as conditional inputs to predict the masked token. Finally, the predicted tokens are decoded and projected back to camera sequences through the VQ-VAE decoder. During both training and testing, we apply the [MASK] token operation solely to the position corresponding to time t and directly use the model's output at this position as the predicted result for t. Consequently, our model does not involve concepts such as masking ratio or the number of inference steps.

4 Experiments

In this section, we evaluate our proposed TemMEGA and analyze its essential characteristics.

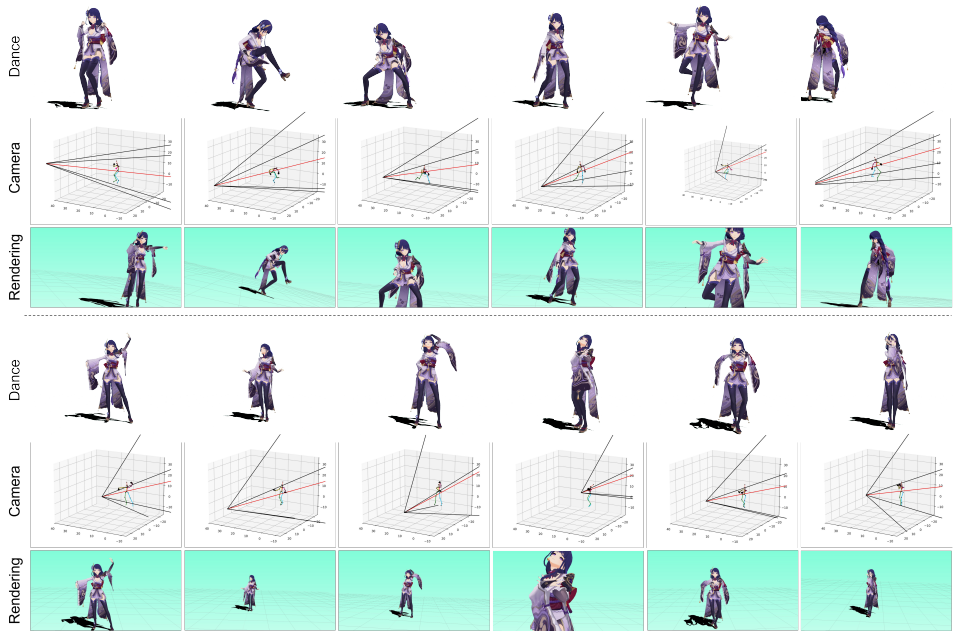

Figure 4: Visualization of the generated results utilizing our TemMEGA.

4.1 Datasets and Implementation Details

Experimental Implementation. We train our models on 4 NVIDIA A6000 48 GB with a batch size of 512. Discrete Camera Tokenizer (DCT) architecture incorporates residual blocks within its encoder and decoder components, featuring a spatial downscaling factor of 4, which consists of 4 quantization layers, each covering a codebook comprising 2048 vectors of 32-dimensional entities. The quantization dropout ratio is set to 0.2. For Consecutive Memory Encoder (CME), we use two transformer encoder blocks to compress long-term memories with 2 layers, and we enhance the short-term memories with two transformer decoder blocks with 4 layers. we set L_l ,L_s and K to 256, 32 and 8. The number of masked transformer blocks, heads, and dimensions is set to 6, 8, and 512 in the Temporal Conditional Masked Transformer (CMT). We train the models by Adam optimizer [20] with the same hyperparameters (learning rate, β_1, and β_2 are set as 0.002, 0, and 0.99, respectively) as previous works.

Datasets. In this work, we use DCM [37], a dataset consisting of 108 pieces of animator-designed paired dance-camera-music data including camera keyframe information. To ensure the fairness of the experiment, we follow the previous works and re-use the train and test splits provided by the original dataset. For the training of our framework, in the training set, we stitch the data pieces that are adjacent to the original data so that we acquire more training data with history.

4.2 Evaluation Metrics

Kinetic Feature Evaluation. Following prior works [21; 31; 37], we evaluate generated camera movement using Fréchet Inception Distance (FID) [11] for quality and average Euclidean distance (Dist) in the feature space for diversity. For kinetic evaluation, we use a kinetic feature extractor [25] following existing works [17; 21; 31]. Since this feature extractor calculates average velocity and acceleration, we compute kinetic features on split 2.5-second data to ensure the density of feature distribution which is similar to settings in AIST++ [21]. Thus, we have got FID_k for kinetic quality and Dist_k for kinetic diversity.

Shot Feature Evaluation. Shot features play a crucial role in dance camera synthesis. However, existing approaches [5; 29; 34] are confined to 2D classification with a limited number of predefined shot types. Therefore, we use a novel shot feature extractor designed for 3D scenes, incorporating cinematographic knowledge. We follow [37; 38] and calculate shot features as:

$$\text{Features}_{\text{shot}} = (S_3/S_1, S_3/S_2)\,. \tag{4}$$

Table 3: Ablation studies on the codebook: (Left) number of code; (Right) code dimension.

(a) Number of code

noc	FID_k	Dist_k	DMR	LCD
256	4.302	1.498	0.0042	0.182
512	4.168	1.521	0.0040	0.179
1024	4.088	1.563	0.0038	0.177
2048	4.025	1.589	0.0035	0.177
4096	4.035	1.590	0.0037	0.179
8192	4.125	1.568	0.0042	0.183

(b) Code dimension

cd	FID_k	Dist_k	DMR	LCD
16	4.091	1.561	0.0038	0.178
32	4.025	1.589	0.0035	0.177
64	4.019	1.595	0.0034	0.176
128	4.015	1.605	0.0034	0.178
256	4.010	1.598	0.0035	0.176

where S_1 and S_3 represent the camera plane projection areas of the dancer's full body and body parts, respectively, within the camera view, and S_2 is the total area of the camera screen. The term S_3/S_1 indicates the percentage of the body within the camera view, while S_3/S_2 reflects the proportion of the camera screen occupied by the dancer. We then compute the Fréchet Inception Distance (FID) and distance (Dist) for $\text{Features}_{\text{shot}}$ and its velocity to obtain FID_s and Dist_s, which measure shot quality and diversity. To account for the differences between shot and kinetic features, we calculate shot metrics on a frame-by-frame basis to maintain the accuracy of shot classification.

Dancer Fidelity Evaluation. Dancer fidelity means camera movement should try to capture significant body parts against the dancer's poses and avoid the long time absence of the dancer in the camera view. We follow [37; 38] to evaluate dancer fidelity with the following two metrics: **1)** Dancer Missing Rate (DMR): DMR represents the ratio of frames in which the dancer is not in the view of the camera, and **2)** Limbs Capture Difference (LCD): LCD denotes the difference of body parts inside and outside camera view between synthesized results and ground truth. Lower values of DMR and LCD indicate better dancer fidelity, as they correspond to fewer instances where the dancer is missing from the view and greater similarity between the synthesized results and the carefully adjusted ground truth.

User Study. For qualitative evaluation, we conduct a user study to compare our method with alternative approaches and the ground truth. In this study, we first randomly select 10 dance-camera input sequences from the test set, each lasting between 17 and 35 seconds. For each sequence, we sample the outputs from our method as well as from baseline methods. This process produces 40 pairs of dance videos, with each pair consisting of the output from our method and one from a baseline method. We then invite 21 participants to view these 30 video pairs in a randomized order and respond to the question, "Which camera movement better highlights the dance and music?" for each pair. The participants include dancers, animators, filmmakers, and individuals with minimal experience in camera work and dance.

4.3 Evaluation Results

Quantitative Results. We compare our TemMEGA with the state-of-the-art camera generation methods on the DCM dataset and report the experimental results in Table 1. Since DanceCamAnimator [38] is a three-stage approach that requires a complete dance video to generate camera movements, it cannot be applied to real-time generation in the real-time setting. The results indicate that the performance of our proposed method considerably outperforms DanceCamera3D [37]. Notably, our method remains effective even without utilizing long-term memory information. This suggests that both the generation architecture and the proposed Consecutive Memory Encoder play a substantial role in enhancing the quality of the generated results.

To further demonstrate the effectiveness and scalability of our method, we made simple modifications to TemMEGA to enable training and testing in an offline setting. Specifically, we remove the components involving long-term memory in TemMEGA and only use short-term memory as the condition for camera generation. Details of these model modifications can be found in the supplementary materials. We can see that TemMEGA consistently performs favorably against all the other existing methods on all evaluations. The demonstrated superiority of our method across various camera quality metrics indicates that it not only generates motions that are more lifelike compared to those produced by baseline methods, but it also excels in choreographing these movements into coherent camera

Table 4: Ablation studies on temporal receptive field: (Left) local window size L_l; (Right) stride L_s.

(a) Impact of L_l

L_l	FID_k	Dist_k	DMR	LCD
64	4.112	1.554	0.0039	0.179
128	4.068	1.575	0.0037	0.179
256	4.025	1.589	0.0035	0.177
512	4.017	1.593	0.0034	0.176
1024	4.002	1.599	0.0033	0.176

(b) Impact of L_s

L_s	FID_k	Dist_k	DMR	LCD
8	4.427	1.505	0.0054	0.192
16	4.167	1.519	0.0047	0.186
32	4.025	1.589	0.0035	0.177
64	4.012	1.597	0.0033	0.175
128	3.995	1.602	0.0032	0.174

sequences through the implementation of the proposed CMT, which helps us learn high-fidelity camera.

Qualitative Results. To better comprehensively demonstrate the effectiveness of our TemMEGA, we visualize the generated camera shots and corresponding rendered dance with diverse camera movements in Figure 4. Our method demonstrates the ability to achieve smooth dance performances with diverse shot transitions, underscoring the advantages of the TemMEGA framework, which effectively synthesizes satisfactory dance camera movements without requiring the entire dance video. More visual results can be found in the supplementary materials.

4.4 Ablation Studies

The impact for the number of code in the codebook. We conducted ablation experiments using codebooks of various lengths. Table 3 shows that a codebook length of 2048 yields the best results. When the number of codes in the codebook is reduced, the diversity of the generated outputs diminishes. On the other hand, excessively increasing the number of codes leads to a rapid decline in overall quality. This is because the size of the codebook determines the number of categories in the subsequent CMT classification. When the number of categories becomes too large, it adversely impacts CMT performance.

The impact of the code dimension of the codebook. We conducted ablation experiments on codebooks with various dimensions to assess their impact on performance. As shown in Table 3, the results indicate that the code dimension of 32 yields improved outcomes compared to other dimensions. Our experimental analysis suggests that changes in the code dimension have only a minor effect on the quality of generation, indicating relative stability across different dimensions. However, due to the demands of the real-time setting, where high generation speed is essential, we selected a dimension of 32 for the final experiments to balance performance quality with reduced computational cost.

The choice of L_l and L_s We conducted ablation experiments to assess the effects of different L_l and L_s values. As shown in Table 4, as L_l increases, the generation quality also improves, though this improvement slows when L_s exceeds 256. Table 4 demonstrates that when L_s is smaller, the generation quality increases more rapidly as L_s grows. Given that the real-time setting requires faster generation speeds, we ultimately selected L_l and L_s values of 256 and 32, respectively, as the experimental parameters.

The choice of K We performed ablation experiments to evaluate the impact of different levels of long-term memory compression, denoted by K. As shown in Figure 5, the extent of compression has minimal influence on the outcomes, whereas the inclusion of long-term memory significantly affects performance. Considering the high-speed requirements of the real-time setting, we optimized for a balance between computational efficiency and output quality, selecting $K = 8$ as an effective compromise.

The impact of components in CME and CMT. In Table 5, we analyze the influence of different components in CME and CMT. Cases 1–3 correspond to the ablation of CME. Removing the long-term enhancement for short-term encoding (Case 1) increases FID_k and DMR, indicating that the interaction between memories benefits visual quality and temporal consistency. When the causal mask in short-term enhancement is removed (Case 2), performance slightly drops, showing that causal modeling helps maintain motion continuity. Excluding the long-term memory as a conditioning signal (Case 3) results in the largest degradation among CME variants, proving the importance of

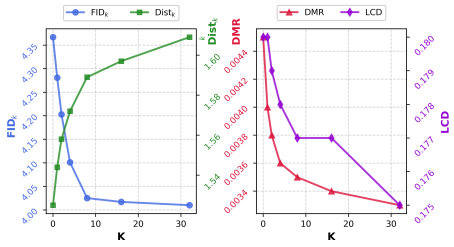

Figure 5: Ablation study for the impact of the number K.

Table 5: Ablation results of the TemMEGA model on CME and CMT components.

Case	$\text{FID}_k \downarrow$	$\text{Dist}_k \uparrow$	DMR $\downarrow$	LCD $\downarrow$
Ours	4.025	1.589	0.0035	0.177
Case 1	4.238	1.552	0.0041	0.178
Case 2	4.126	1.577	0.0038	0.178
Case 3	4.367	1.525	0.0045	0.180
Case 4	4.151	1.580	0.0037	0.178
Case 5	4.851	1.425	0.0081	0.208
Case 6	4.061	1.593	0.0036	0.177

long-term context for stable camera synthesis. Cases 4–6 investigate CMT. Injecting both memories through cross-attention (Case 4) achieves comparable but slightly worse performance than our design, suggesting that our fusion strategy is more effective. Using AdaIN for feature injection (Case 5) leads to a notable decline in all metrics, revealing that adaptive normalization is less suitable for temporal-memory fusion. Sequentially injecting long-term and then short-term memory (Case 6) performs close to the full model, demonstrating the robustness of the proposed memory arrangement.

5 Conclusion

In this paper, we introduce the TemMEGA framework, a novel approach for real-time camera movement synthesis tailored specifically for live dance performances. Unlike previous methods that rely on full-length dance videos, TemMEGA utilizes only current and past segments of dance and music, making it feasible for real-time application. Our approach leverages discrete camera tokenization, a consecutive memory encoder for capturing long- and short-term temporal dependencies, and a conditional masked transformer to generate camera movements dynamically. Experimental results on public datasets demonstrate that TemMEGA achieves state-of-the-art performance, validating its robustness and effectiveness in addressing the complexities of real-time camera movement synthesis for live dance contexts.

6 Acknowledgments

Our work is supported in part by the National Key R&D Program of China (No. 2023YFC3305600), the Joint Fund of Ministry of Education of China (8091B022149, 8091B02072404), National Natural Science Foundation of China (62132016, 62571412), and the Expert Workstation of Yunnan Province under Grant (202305AF150202).

References

[1] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26, 2017.

[2] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM TOG*, 42:1 – 18, 2023.

[3] Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*, 2025.

[4] Jinhe Bi, Yujun Wang, Haokun Chen, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. Llava steering: Visual instruction tuning with 500x fewer parameters through modality linear representation-steering. In *ACL*, 2025.

[5] Luca Canini, Sergio Benini, and Riccardo Leonardi. Classifying cinematographic shot types. *Multimedia tools and applications*, 62:51–73, 2013.

[6] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[7] Inan Evin, Perttu Hämäläinen, and Christian Guckelsberger. Cine-ai: Generating video game cutscenes in the style of human directors. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI PLAY):1–23, 2022.

[8] Rukun Fan, Songhua Xu, and Weidong Geng. Example-based automatic music-driven conventional dance motion synthesis. *IEEE TVCG*, 18(3):501–515, 2011.

[9] Mirko Gschwindt, Efe Camci, Rogerio Bonatti, Wenshan Wang, Erdal Kayacan, and Sebastian Scherer. Can a robot become a movie director? learning artistic principles for aerial cinematography. In *IEEE/RSJ IROS*, pages 1107–1114. IEEE, 2019.

[10] Mirko Gschwindt, Efe Camci, Rogerio Bonatti, Wenshan Wang, Erdal Kayacan, and Sebastian Scherer. Can a robot become a movie director? learning artistic principles for aerial cinematography. In *IROS*, pages 1107–1114, 2019.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[12] Chong Huang, Yuanjie Dang, Peng Chen, Xin Yang, and Kwang-Ting Cheng. One-shot imitation drone filming of human motion videos. *IEEE TPAMI*, 44(9):5335–5348, 2021.

[13] Chong Huang, Yuanjie Dang, Peng Chen, Xin Yang, and Kwang-Ting Cheng. One-shot imitation drone filming of human motion videos. *TPAMI*, 44(9):5335–5348, 2022.

[14] Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. Learning to film from professional human motion videos. In *CVPR*, pages 4244–4253, 2019.

[15] Chong Huang, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Tim Cheng. Learning to capture a film-look video with a camera drone. In *ICRA*, pages 1871–1877. IEEE, 2019.

[16] Chong Huang, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Tim Cheng. Learning to capture a film-look video with a camera drone. In *ICRA*, pages 1871–1877, 2019.

[17] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. In *ICLR*, 2020.

[18] Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. Example-driven virtual cinematography by learning camera behaviors. *ACM TOG*, 39(4):45, 2020.

[19] Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. In *Computer Graphics Forum*, volume 43, page e15055, 2024.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[21] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, pages 13401–13412, 2021.

[22] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. *arXiv preprint arXiv:2211.09117*, 2022.

[23] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *SciPy*, pages 18–24, 2015.

[24] Ferda Ofli, Engin Erzin, Yücel Yemez, and A Murat Tekalp. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE TMM*, 14(3):747–759, 2011.

[25] K ONUMA. Fmdistance: A fast and effective distance function for motion capture data. *Proc. EUROGRAPHICS 2008 Short Papers*, 2008.

[26] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023.

[27] Pablo Pueyo, Juan Dendarieta, Eduardo Montijano, Ana C Murillo, and Mac Schwager. Cinempc: A fully autonomous drone cinematography system incorporating zoom, focus, pose, and scene composition. *IEEE T-RO*, 2024.

[28] Anyi Rao, Xuekun Jiang, Yuwei Guo, Linning Xu, Lei Yang, Libiao Jin, Dahua Lin, and Bo Dai. Dynamic storyboard generation in an engine-based virtual environment for video production. In *ACM SIGGRAPH*, 2023.

[29] Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *ECCV*, pages 17–34, 2020.

[30] James Rucks and Nikolaos Katzakis. Camerai: Chase camera in a dense environment using a proximal policy optimization-trained neural network. In *IEEE CoG*, pages 1–8. IEEE, 2021.

[31] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *CVPR*, pages 11050–11059, 2022.

[32] Taoran Tang, Hanyang Mao, and Jia Jia. Anidance: Real-time dance motion synthesize to the song. In *ACM MM*, pages 1237–1239, 2018.

[33] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *CVPR*, pages 448–458, 2023.

[34] Ioannis Tsingalis, Nicholas Vretos, Nikos Nikolaidis, and Ioannis Pitas. Svm-based shot type classification of movie content. In *Mediterranean Electrotechnical Conference*, volume 6, 2012.

[35] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, volume 30, 2017.

[36] Jiahao Wang, Guo Chen, Yifei Huang, Limin Wang, and Tong Lu. Memory-and-anticipation transformer for online action understanding. In *ICCV*, pages 13824–13835, 2023.

[37] Zixuan Wang, Jia Jia, Shikun Sun, Haozhe Wu, Rong Han, Zhenyu Li, Di Tang, Jiaqing Zhou, and Jiebo Luo. Dancecamera3d: 3d camera movement synthesis with music and dance. In *CVPR*, pages 7892–7901, 2024.

[38] Zixuan Wang, Jiayi Li, Xiaoyu Qin, Shikun Sun, Songtao Zhou, Jia Jia, and Jiebo Luo. Dancecamanimator: Keyframe-based controllable 3d dance camera synthesis. In *ACM MM*, 2024.

[39] Xinyi Wu, Haohong Wang, and Aggelos K Katsaggelos. The secret of immersion: actor driven camera movement generation for auto-cinematography. *arXiv preprint arXiv:2303.17041*, 2023.

[40] Chun Xie, Isao Hemmi, Hidehiko Shishido, and Itaru Kitahara. Camera motion generation method based on performer's position for performance filming. In *Global Conference on Consumer Electronics (GCCE)*, pages 957–960, 2023.

[41] Chenghao Xu, Guangtao Lyu, Jiexi Yan, Muli Yang, and Cheng Deng. Llm knows body language, too: Translating speech voices into human gestures. In *ACL*, pages 5004–5013, 2024.

[42] Chenghao Xu, Jiexi Yan, and Cheng Deng. Keep and extent: Unified knowledge embedding for few-shot image generation. *IEEE TIP*, 2025.

[43] Chenghao Xu, Jiexi Yan, Muli Yang, and Cheng Deng. Rethinking noise sampling in class-imbalanced diffusion models. *IEEE TIP*, 2024.

[44] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. In *NeurIPS*, volume 34, pages 1086–1099, 2021.

[45] Sizhe Yang, Yanjie Ze, and Huazhe Xu. Movie: Visual model-based policy adaptation for view generalization. In *NeurIPS*, volume 36, 2024.

[46] Zixiao Yu, Enhao Guo, Haohong Wang, and Jian Ren. Bridging script and animation utilizing a new automatic cinematography model. In *MIPR*, pages 268–273. IEEE, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes. The main claims in the abstract and introduction accurately reflect the paper's contributions and scope. They clearly outline the research objectives, methodology, and the significance of the findings.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations have been discussed in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All the theorems, formulas, and proofs in the paper have been properly numbered and cross-referenced, fulfilling the guidelines provided.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have detailed our model and experimental setup thoroughly in the Methodology and Experiments sections, providing all necessary information to reproduce the main experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer:[No]

   Justification: At this stage, we do not plan to release the code due to ongoing related research. However, we provide detailed implementation and training settings in the paper to ensure reproducibility.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: To ensure full reproducibility, we provide a comprehensive description of the experimental setup in the "Method" and "Experimental Implementation" sections, along with additional details in the Supplementary Material. This includes clear specifications of data partitions, hyperparameters, model selection criteria, and the optimization algorithms employed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experimental results do not include confidence intervals or statistical significance tests.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Appendix and Experimental Implementation provide detailed information on the computational resources required to reproduce our experiments, including hardware specifications, memory usage, and estimated execution time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research adheres to all ethical guidelines required by NeurIPS.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

Justification: Our research does not directly produce societal impacts as it focuses on technical advancements in a specific field without direct societal applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets used in our paper are publicly available datasets, and we have cited the respective literature for each dataset. Any researcher can download these datasets from the provided sources.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets. We will make the code public after the paper is accepted.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.