

---

# Deep Compositional Phase Diffusion for Long Motion Sequence Generation

---

**Ho Yin Au**

Hong Kong Baptist University  
cshyau@comp.hkbu.edu.hk

**Jie Chen\***

Hong Kong Baptist University  
chenjie@comp.hkbu.edu.hk

**Junkun Jiang**

Hong Kong Baptist University  
csjkjiang@comp.hkbu.edu.hk

**Jingyu Xiang**

Hong Kong Baptist University  
csjyxiang@comp.hkbu.edu.hk

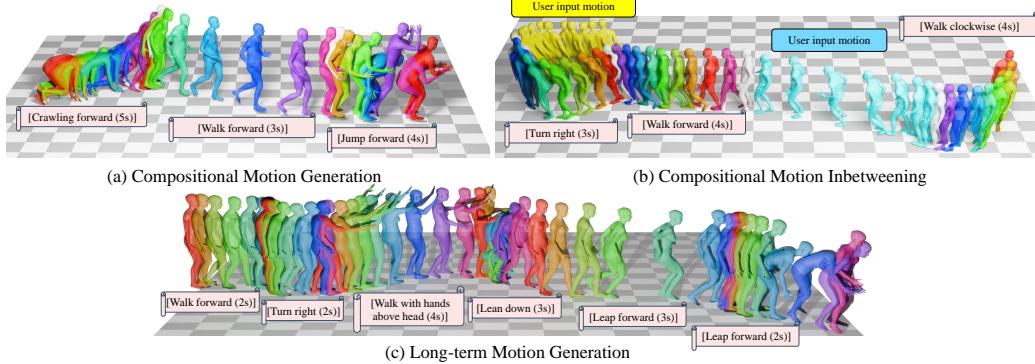


Figure 1: Our Compositional Phase Diffusion framework produces high-quality composite motion sequences with smooth transitions and semantic alignment. (a) Compositional generation involves synthesizing multiple motion segments of varying lengths simultaneously, ensuring smooth transitions between segments. (b) Motion inbetweening allows users to select segments (in blue and yellow) and create conditional or unconditional bridging motions. (c) Long-term motion generation is achieved by scaling the framework with additional modules, enabling the parallel denoising of a larger number of motion segments. The rainbow color indicates time progression.

## Abstract

Recent research on motion generation has shown significant progress in generating semantically aligned motion with singular semantics. However, when employing these models to create composite sequences containing multiple semantically generated motion clips, they often struggle to preserve the continuity of motion dynamics at the transition boundaries between clips, resulting in awkward transitions and abrupt artifacts. To address these challenges, we present Compositional Phase Diffusion, which leverages the Semantic Phase Diffusion Module (SPDM) and Transitional Phase Diffusion Module (TPDM) to progressively incorporate semantic guidance and phase details from adjacent motion clips into the diffusion process. Specifically, SPDM and TPDM operate within the latent motion frequency domain established by the pre-trained Action-Centric Motion Phase Autoencoder

---

\*Corresponding Author

(ACT-PAE). This allows them to learn semantically important and transition-aware phase information from variable-length motion clips during training. Experimental results demonstrate the competitive performance of our proposed framework in generating compositional motion sequences that align semantically with the input conditions, while preserving phase transitional continuity between preceding and succeeding motion clips. Additionally, motion inbetweening task is made possible by keeping the phase parameter of the input motion sequences fixed throughout the diffusion process, showcasing the potential for extending the proposed framework to accommodate various application scenarios. Codes are available at <https://github.com/asdryau/TransPhase>.

## 1 Introduction

Deep learning-based human motion generation holds significant potential for creating virtual humanoid animations and enhancing robotics applications. With more advanced modeling techniques [1, 2, 3] and more motion data being captured [4, 5, 6, 7], motion generation models are evolving rapidly and can be adapted to a variety of multimodal generation tasks. For example, text-to-motion generation allows animators to produce character animations by specifying semantic contexts using text prompts. Current motion generation models [8, 6, 9, 10, 11] handle variable-length motion segments with singular semantics. However, when these models are applied to long-term compositional motion generation tasks, where they must generate  $K$  motion segments sequentially for  $K$  instructions, they often struggle with smooth transitions between segments. Recently, there has been a growing focus on long-term compositional generation tasks [12, 13, 14], driven by the availability of BABEL-TEACH [4, 12], a dataset with text annotations for pairs of temporally connected motion segments, which aids motion models in learning transitions. Methods such as priorMDM [13] use the learned transition knowledge to create transitional segments that smooth out pose differences between those generated by MDM [8]. However, these approaches often overlook the intrinsic kinematics of each segment, resulting in artifacts like over-smoothing or abrupt stops in transitions.

To generate motion clips aligned with specific semantic contexts and ensure smooth transitions, we introduce the Compositional Phase Diffusion framework. This framework simultaneously creates multiple motion clips from sequential semantic instructions, using denoised information from adjacent clips to enhance transition compatibility. The Compositional Phase Diffusion framework consists of three main components: Action-Centric Periodic Autoencoder (ACT-PAE), Semantic Phase Diffusion Module (SPDM), and Transitional Phase Diffusion Module (TPDM). ACT-PAE, which builds upon DeepPhase [15], encodes each variable-length motion segment into a unified phase manifold. SPDM and TPDM then iteratively denoise phase parameters by incorporating semantic instructions and neighbouring phase signals. This approach effectively models the intrinsic dynamics of each motion segment within the latent motion frequency domain, ensuring both semantic alignment and smooth transitions. Additionally, the framework is scalable, allowing for an arbitrary number of modules to denoise multiple motion segments in parallel. This capability highlights its flexibility and scalability in generating motion sequences of varying lengths and facilitating motion inbetweening tasks.

Extensive experimental results demonstrate that the Compositional Phase Diffusion framework excels in both long-term compositional motion generation and motion inbetweening tasks, attributed to the semantic and transition-aware diffusion process. The key contributions are as follows:

- We introduce the Compositional Phase Diffusion framework, a scalable and efficient solution for various motion generation tasks. This framework can process an arbitrary number of motion segments of varying lengths simultaneously by leveraging parallel module execution, ensuring smooth and coherent transitions between clips.
- Our framework incorporates three key components: the Action-Centric Periodic Autoencoder (ACT-PAE), the Semantic Phase Diffusion Module (SPDM), and the Transitional Phase Diffusion Module (TPDM). By operating within a unified phase latent space established by ACT-PAE, SPDM and TPDM collaboratively denoise motion representations while preserving semantic phase information and aligning transitional dynamics.
- Extensive experiments validate the effectiveness of our framework, demonstrating significant improvements in long-term compositional motion generation and motion inbetweening tasks, showcasing its ability to produce high-quality, contextually relevant animations.

## 2 Related work

**Motion Phase Modeling.** Pioneering approaches [16, 17, 18] incorporate explicit phase inputs, such as foot contact during walking, to achieve smooth motion extrapolation and transition. DeepPhase [15] further extends this concept by developing a Periodic Autoencoder (PAE) that encodes motion segments into phase latent parameters, i.e., frequency ( $\mathbf{F}$ ), amplitude ( $\mathbf{A}$ ), offset ( $\mathbf{B}$ ), and phase shift ( $\mathbf{S}$ ). These parameters help generate periodic motion patterns and smooth transitions, minimizing artifacts like over-smoothing and sudden stops. Building upon PAE, PhaseBetweener [19] and RSMT [20] tackle motion inbetweening tasks by autoregressively generating motion frames and phase parameters. Meanwhile, DiffusionPhase [21] adopts the MLD [22] framework to denoise the periodic latents based on input text and conditioned pose. However, the fixed-length convolution scheme of PAE leads to instability in training objectives, as variable-length motions are encoded into a varying number of phase latent codes.

**Text-to-Motion Generation.** Several methods have been utilized diffusion models [2, 3] with a single text prompt, including MDM [8], MLD [22], MotionDiffuse [11], and DiffusionPhase [21]. Notably, MDM [8] applies the Diffusion Model to raw pose sequences conditioned on text encoded by CLIP [23]. Building upon MDM, PhysDiff [24] and GMD [25] have been developed to enhance physical plausibility and trajectory control in the generated motion. However, due to the limited segment lengths of datasets like HumanML3D [6] and BABEL [4], with maximum frames of 196 and 250, respectively, these models struggle to generate longer motion sequences.

**Learning-based Motion Inbetweening** is achieved through two main approaches. 1) Autoregressive frame generation: Motion frames are sequentially generated to connect segment boundaries [26, 27]. Methods like DiffusionPhase [19] and RSMT [20] further incorporate motion phase modeling for smoother, phase-aware transitions. 2) Segment interval infilling: Transitional segments of specified length are created to bridge segment boundaries [28, 29]. Methods like CMB [30] and MDM [8] extend this by integrating semantic conditions into the inbetweening motion generation process.

**Long Motion Sequence Generation** can be approached in two ways: sequential generation and parallel generation. Sequential generation methods such as TEACH [12], PCMDM [14], M2D2M [31], and InfiniMotion [32] generate motion segments one after another in an autoregressive manner. Analogous to traditional motion graph based approaches [33, 34, 35, 36], these methods require that the generated segments not only align with the current input semantics, but also transition smoothly from previously generated segments. For parallel generation, priorMDM [13] generates semantic motion segments independently and then synthesizes blending transitional segments using a diffusion model. Note that the frameworks above typically model transitions in the raw motion space, which may lead to slight discontinuities at the segment boundaries. To address this, motion inbetweening techniques are usually employed to smooth the transition boundaries. For example, TEACH [12] uses spherical linear interpolation (SLERP) to create motion frames connecting boundary poses.

## 3 Compositional Phase Diffusion

We propose three key components for the framework: the Action-Centric Periodic Autoencoder (ACT-PAE), the Transitional Phase Diffusion Module (TPDM), and the Semantic Phase Diffusion Module (SPDM). ACT-PAE creates a motion latent manifold that captures important semantic and transition-aware phase information for each motion segment  $\mathbf{X} \in \mathbb{R}^{N \times E}$  and represent them as a set of latent variables  $\mathbf{P} = [\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}]$ . Leveraging such ACT-PAE latent space, TPDMs refine phase latents of the current segment using the **phase dynamics information from adjacent motions**, while SPDM incorporates **semantic information** into the diffusion process. Details of these components will be covered in Sec. 3.1.

With these innovative elements, we adapt the Compositional Phase Diffusion framework to various motion generation tasks. For the compositional motion generation and motion inbetweening tasks, SPDMs and TPDMs gradually integrate semantic information and phase dynamics information from adjacent segments throughout the denoising process of sequentially connected segments. For the long-term motion generation task, the phase dynamics of a motion segment will progressively propagate bidirectionally along the timeline during the denoising process. This promotes mutual phase dynamics adjustment between segments, increases their transition-awareness, and thereby enhances overall motion consistency. By blending a series of transition-aware motion segments, we create a cohesive motion sequence composed of a series of semantically meaningful segments and seamless transitions in between. Further details are provided in Sec. 3.2.

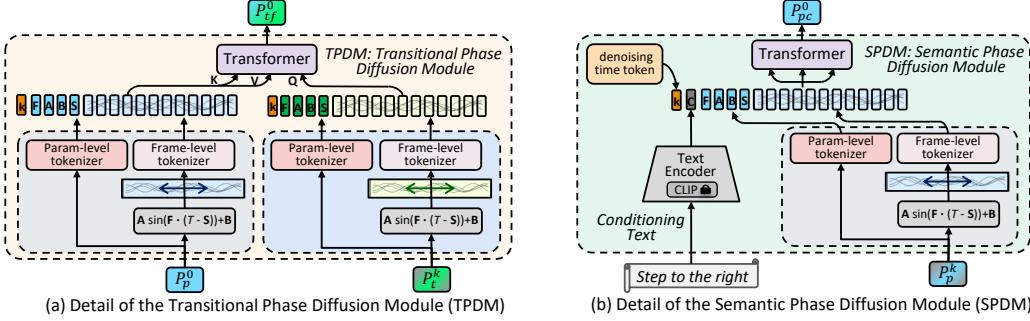


Figure 2: Module detail of TPDM and SPDM. (a) The TPDM uses the clean phase latents from the adjacent segment  $P_p^0$  to denoise the current motion phase  $P_t^k$ , making the current denoising motion align with the motion dynamics of the adjacent motion segment. (b) The SPDM utilizes the text embedding  $C$  from CLIP to denoise the motion phase parameters of the current denoising motion.

### 3.1 Key Components

#### 3.1.1 ACT-PAE: Action-Centric Periodic Autoencoder

Our ACT-PAE builds upon the transformer-based motion autoencoder architecture from ACTOR [37]. ACT-PAE encoder first processes input motion  $\mathbf{X} \in \mathbb{R}^{N \times E}$  of  $N$  frames into four phase parameters  $\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S} \in \mathbb{R}^Q$ . Unlike PAE [15], which processes motion using a convolution scheme and derives phase parameters from the FFT results, the ACT-PAE encoder directly processes variable-length motion using a transformer and predicts their phase parameters. To enforce latent space periodicity, these parameters are parameterized into a periodic signal  $\mathbf{Q} \in \mathbb{R}^{N \times Q}$  using the following equation:

$$\mathbf{Q} = \mathbf{A} \sin(\mathbf{F} \cdot (T - S)) + \mathbf{B}. \quad (1)$$

Here  $T$  represents the time difference of each frame in the motion *relative to the center* of the motion segment. The  $\sin$  function parameterizes  $\mathbf{F} \cdot (T - S)$  into a periodic sine wave, transforming frequency and phase shift information into a sinusoidal basis. This representation allows ACT-PAE to capture the underlying phase dynamics of the motion effectively. Finally, the ACT-PAE decoder takes  $\mathbf{Q}$  to predict the motion  $\hat{\mathbf{X}}$ . The entire ACT-PAE is trained with L2 loss.

The main advantage of ACT-PAE lies in its ability to capture unified phase dynamics within semantically meaningful motion (e.g.,  $\mathbf{X}_p$  and  $\mathbf{X}_s$ ). Its architecture handles variable-length motion input, which eliminates the need for fixed-window motion slicing and thus preserves complete semantic and transition-aware phase information in phase latents. By doing so, ACT-PAE standardizes the training objective for the subsequent motion diffusion modules more effectively than the fixed-window process used in PAE, which results in an undetermined number of phase latents. Importantly, we have changed the sinusoidal positional embedding module  $PE$  and the time window  $T$  to accommodate variable-length motion encoding. For example,  $T$  can be parameterized for normalized action progression ( $-1$  to  $1$  across  $N$  frames), or actual time duration ( $-\frac{N}{2}$  to  $\frac{N}{2}$  across  $N$  frames). Details of  $PE$  and  $T$  adjustments will be provided in the Appendix.

#### 3.1.2 SPDM: Semantic Phase Diffusion Module

SPDM is designed to denoise phase parameters so that the corresponding decoded motion segment is aligned to the semantic condition. In text-to-motion settings, SPDM employs the pre-trained *CLIP-ViT-B/32* [23] to encode the input text conditions into embedding vector  $C_p$ , as shown in Fig. 2(b). This embedding guides the denoising process of the phase parameters  $P_p^k$ , which are encoded by ACT-PAE, for the semantically conditioned motion  $\mathbf{X}_p$  via  $P_{pc}^0 = \mathcal{F}_S(k, C_p, P_p^k)$ . Here,  $k$  indicates the denoising time step. Note that the input phase parameters  $P$  are parameterized as both *param-level tokens* [ $\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}$ ] and *frame-level tokens* which constitute the periodic signal  $\mathbf{Q}$  created using Equation 1. These frame-level tokens explicitly outline the spatio-temporal motion context in the phase parameters, assisting SPDM in monitoring the current semantic context during the phase parameter denoising process. Finally, a self-attention transformer [1] is employed to derive semantically-denoised parameters  $P_{pc}^0$ .

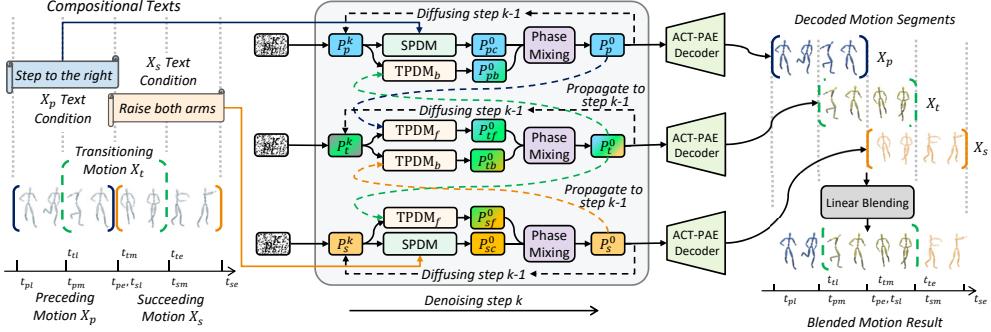


Figure 3: Illustration of the phase diffusion pipeline for the compositional motion generation task. SPDMs and TPDMs guide the denoising of motion segments through **semantic information** and the **phase dynamics information from adjacent motions**, respectively. The denoised results are combined via phase mixing and either diffused back to step  $k - 1$  or fed into the ACT-PAE decoder at the final step to produce motion segments, which are then linearly blended to create a long sequence.

### 3.1.3 TPDM: Transitional Phase Diffusion Module

TPDM is designed to denoise phase parameters such that the resulting decoded motions are transitionally aligned with adjacent motions. Depending on the specific application scenario, these adjacent motions may come from either the forward or backward direction, which will be explained in Sec. 3.2.

Fig. 2(a) illustrates the TPDM architecture, which leverages the clean phase parameters of the forward, preceding motion  $\mathbf{P}_p^0$  to denoise the phase parameters of the transitioning motion  $\mathbf{P}_t^k$ ,  $\mathbf{P}_{tf}^0 = \mathcal{F}_{T_f}(k, \mathbf{P}_t^k, \mathbf{P}_p^0)$ .  $f$  indicates that the denoising process is conditioned by the phase dynamics of the forward, preceding motion. Similar to the SPDM design, both *param-level tokens* and *frame-level tokens* are computed for  $\mathbf{P}_p^0$ . The motion context provided by the frame-level tokens assists the TPDM in ensuring alignment of the motion dynamics during the denoising process. Finally, a cross-attention transformer [1] processes all the  $\mathbf{P}_t^k$  and  $\mathbf{P}_p^0$  tokens to predict phase noise  $\mathbf{P}_{tf}^0$ .

As can be seen in Fig. 3, there are at least two TPDM modules involved in compositional motion generation: TPDM<sub>f</sub> utilizes preceding motion phase  $\mathbf{P}_p^0$  to denoise  $\mathbf{P}_t^k$ ,  $\mathbf{P}_{tf}^0 = \mathcal{F}_{T_f}(k, \mathbf{P}_t^k, \mathbf{P}_p^0)$  and TPDM<sub>b</sub> utilizes succeeding motion phase  $\mathbf{P}_s^0$  instead:  $\mathbf{P}_{tb}^0 = \mathcal{F}_{T_b}(k, \mathbf{P}_t^k, \mathbf{P}_s^0)$ . These two modules work together to ensure that dynamic coherence is maintained in both *forward* and *backward* directions throughout the composited long sequence.

SPDM and TPDMs are implemented as  $\epsilon$ -models, and their training procedures follow those of traditional diffusion frameworks [25, 3]. Details for SPDM and TPDM are provided in the Appendix.

## 3.2 Applications

### 3.2.1 Compositional Motion Pair Generation

The compositional motion pair generation task focuses on creating two sequentially connected motion segments,  $\mathbf{X}_p$  and  $\mathbf{X}_s$ . To ensure a smooth transition while maintaining semantic alignment, we develop a compositional motion diffusion pipeline that progressively incorporates the **semantic information** and the **phase dynamics information from adjacent segments** in the diffusion process. This phase dynamics information exchange enhances phase alignment between  $\mathbf{X}_p$  and  $\mathbf{X}_s$ , and facilitates the creation of an intermediate transition segment  $\mathbf{X}_t$ <sup>2</sup>, which is linearly blended into the output to further smooth the segment boundary.

The pipeline detail is shown in Fig. 3 and described in Algorithm 1 in the Appendix. During the denoising step  $k$ , SPDM semantically denoises the phase latents  $\mathbf{P}_p^k$  and  $\mathbf{P}_s^k$  for  $\mathbf{X}_p$  and  $\mathbf{X}_s$  based on their respective semantic conditions. TPDM<sub>f</sub> and TPDM<sub>b</sub> then estimate  $\mathbf{P}_{pb}^0$ ,  $\mathbf{P}_{tf}^0$ ,  $\mathbf{P}_{tb}^0$  and  $\mathbf{P}_{sf}^0$  by combining and mixing information from temporally adjacent phase latents (i.e.,  $\mathbf{P}_p^0$ ,  $\mathbf{P}_t^0$ , and  $\mathbf{P}_s^0$ ) from the earlier denoising step  $k + 1$ . For instance,  $\mathbf{P}_t^k$  is denoised with  $\mathbf{P}_p^0$  and  $\mathbf{P}_s^0$  from step  $k + 1$ ,

<sup>2</sup>We define the concept of the transition motion  $\mathbf{X}_t$  to be the segment covering exactly the second half of  $\mathbf{X}_p$  and the first half of  $\mathbf{X}_s$ .

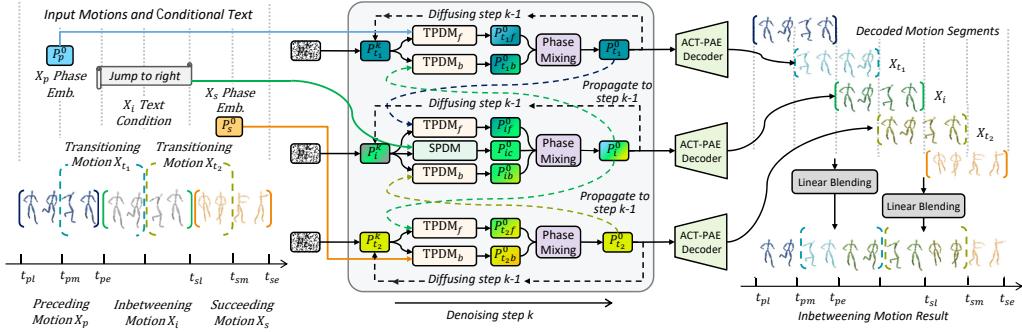


Figure 4: The phase diffusion pipeline for the motion inbetweening tasks. The inbetweening motion  $\mathbf{X}_i$  along with two transitional motions  $\mathbf{X}_{t1}$  and  $\mathbf{X}_{t2}$  are simultaneously generated. These motions are subsequently linear blended to form the final inbetweening output.

while the resulting  $\mathbf{P}_t^0$  from step  $k$  helps denoising  $\mathbf{P}_p^{k-1}$  and  $\mathbf{P}_s^{k-1}$  in step  $k-1$ , demonstrating the exchange of phase dynamics throughout diffusion process.

The predictions from both SPDM and TPDM are then combined using the Phase Mixing Equation:

$$\mathbf{P}_t^0 = r \frac{\mathbf{P}_{\cdot f}^0 + \mathbf{P}_{\cdot b}^0}{2} + (1 - r) \mathbf{P}_{\cdot c}^0 \quad (2)$$

The variable  $r$  is defined as  $r = (\frac{k}{K})^3$  for semantic conditioned motion segments (i.e.,  $\mathbf{X}_p$  and  $\mathbf{X}_s$ ), and  $r = 1$  otherwise (i.e.,  $\mathbf{X}_t$ ). Note that the value of  $r$  for semantically denoised segments is determined by the ratio of total steps  $K$  and current denoising step  $k$ , ensuring transition compatibility early and enriching semantic details progressively. Finally, DDIMScheduler [3]  $\mathcal{F}_D$  is utilized to estimate the clean phase latent and the next step phase latent based on the mixed phase latent.

### 3.2.2 Motion Inbetweening

The motion inbetweening task aims to generate an inbetweening motion  $\mathbf{X}_i$ , which is of a specified length to bridge the gap between two separated motions  $[\mathbf{X}_p, \mathbf{X}_s]$ . The pipeline for the task is illustrated in Fig. 4. The process begins by encoding these segments into latent codes  $\mathbf{P}_p^0$  and  $\mathbf{P}_s^0$  with the ACT-PAE encoder, then uses TPDMs to guide the generation of inbetweening motion  $\mathbf{X}_i$  and two transitioning motions  $\mathbf{X}_{t1}$  and  $\mathbf{X}_{t2}$ . An optional SPDM can be incorporated for  $\mathbf{X}_i$ , modifying the task from unconditional (UMIB) to conditional (CMIB). Additionally, the pipeline demonstrates the flexibility and scalability of the Compositional Phase Diffusion framework by enabling the compositional generation of more motion segments of varying lengths through parallel processing with an increasing number of modules.

### 3.2.3 Long-term Motion Generation

Long-term motion sequence generation extends beyond short-term compositional motion pair generation by producing much longer continuous motion, composed of hundreds or thousands of motion segments. While short-term tasks focus on semantics and transitions within a few segments, long-term generation involves monitoring kinetic dynamics, which can impact motion over extended sequences and potentially disrupt motion realism and physical plausibility. To adapt our compositional motion framework for long-term generation, we can unroll it to process each segment with the [TPDM<sub>f</sub>, SPDM, TPDM<sub>b</sub>] triplet and denoise them based on semantics and adjacent phase conditions. By rearranging and batching the input for each module, the denoising process of all segments can be done in parallel, making the overall denoising time independent of the number of segments.

The bidirectional TPDM mechanism in our framework ensures that phase information propagates progressively throughout the sequences, rather than being confined to specific local segments. This mitigates the risk of substantial phase dynamics misalignments between adjacent semantic segments and simplifies the adjustments required by transition segments. Unlike existing methods that struggle with handling substantial differences in motion phase dynamics between segments, which often result in the loss of smooth transitions or semantic alignment, our model continuously refines both motion phase dynamics and text alignment to preserve long-term motion integrity.

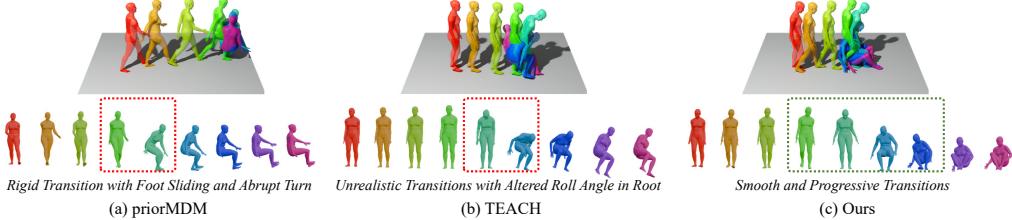


Figure 5: Compositional motion pair result visualization for [walk(2.4s), sit down(3.6s)]. The motion frames are colored from red to purple in a rainbow gradient to represent the progression of time. Note that priorMDM exhibits an unrealistic, sudden turn during the sit-down action, which is reflected in its low FID score. TEACH’s result includes footing skating in the walk motion. In contrast, our framework generates a fluid walking motion that transitions smoothly into a sit-down action.

Table 1: Quantitative results for **Compositional Motion Pair Generation** on the BABEL-TEACH test set. **Bold** and underline indicates the **best** and the second-best result.

Comp. Motion Pair	Motion Realism — FID $\downarrow$			Text Alignment — MMD $\downarrow$		
	Smt.	Trn.	Overall	Smt.	Trn.	Overall
MDM-30 [8]	1.084	2.526	1.146	3.793	<b>6.429</b>	4.923
MLD-30 [22]	13.88	15.20	14.25	8.478	7.407	7.632
TEACH [12]	<u>0.941</u>	2.375	1.041	<b>3.185</b>	7.479	<u>4.821</u>
PCMDM [14]	1.056	<u>1.898</u>	<u>0.837</u>	4.548	7.323	5.423
priorMDM [13]	1.148	2.580	0.839	3.732	7.399	5.025
Ours	<b>0.736</b>	<b>1.807</b>	<b>0.782</b>	<u>3.509</u>	6.545	<b>4.711</b>

## 4 Experiments

### 4.1 Implementation and Evaluation Details

#### 4.1.1 Training and Evaluation Dataset

We use the BABEL-TEACH dataset [4, 12] for training and evaluation, as it provides annotated subsequence pairs essential for long-term motion generation [12, 13, 14], facilitating the learning of transitions between subsequences. These annotated pairs are derived from decomposing fine-grained text subsequence annotations from BABEL [4]. For example, a sequence such as [walk, sit down, stand up, move arms] is split into pairs like [walk, sit down], [sit down, stand up], and [stand up, move arms].

Following the data processing pipeline outlined in recent long-term motion generation [12, 13, 14], we set the minimum and maximum lengths for each subsequence as 45 and 250 frames, respectively. The pipeline then groups the textually annotated subsequences into pairs. Note that any overlapping offsets identified by the pipeline above are redistributed among the annotated motion subsequences. As a result, the training dataset contains 4370 subsequence pairs, while the testing dataset includes 1582 subsequence pairs. Moreover, we follow PCMDM [14] and priorMDM [13] to transform the motion data into HumanML3D [6] format. Initially, the root trajectory is represented by only 4 out of 263 parameters, omitting essential root orientation details. Therefore, we supplement a 6D rotation [38] for the root, increasing the total parameters to  $E = 269$ . Note that all models being compared are trained on the same dataset and representation to ensure a fair comparison.

**Remark on Dataset.** To the best of our knowledge, BABEL-TEACH is currently the only dataset that provides subsequence pair annotations. Long-term motion generation models, such as TEACH [12], PCMDM [14], and our own model, require data samples in the form of continuous subsequence pairs to effectively learn transitions between sequences. Motion datasets annotated in other formats or modalities cannot be efficiently utilized either for training or for fair evaluation of our task.

#### 4.1.2 Evaluation Metrics

We assess the results of **compositional motion pair generation**, **long-term motion generation**, and **conditional motion inbetweening** based on two key aspects: *Fréchet Inception Distance* (FID) for Motion Realism and *Multimodal Distance* (MMD) for Text Alignment, following the T2M [6]

Table 2: Quantitative results for **Long-term Motion Generation** on the BABEL-TEACH test set with a single extended text sequence of 3,164 texts (302,298 frames, 168 minutes). **Bold** and underline indicates the **best** and the **second-best** result.

Long-term	Motion Realism — FID $\downarrow$			Text Alignment — MMD $\downarrow$		
	Smt.	Trn.	Overall	Smt.	Trn.	Overall
MDM-30 [8]	1.094	2.051	1.365	3.877	6.411	4.958
MLD-30 [22]	14.36	13.44	17.02	8.423	7.407	7.690
TEACH [12]	0.785	1.645	1.780	<b>3.175</b>	<u>5.483</u>	4.984
PCMDM [14]	1.068	0.934	<u>0.876</u>	4.188	5.721	5.156
priorMDM [13]	2.288	1.067	1.536	4.299	5.536	5.060
Ours	<b>0.773</b>	<b>0.909</b>	<b>0.847</b>	<u>3.642</u>	<b>5.389</b>	<b>4.849</b>

evaluation protocol utilized in PCMDM [14] and priorMDM [13] for long-term motion assessment. We exclude *R-precision* (R-prec.) because it overlaps with *Multimodal Distance* (MMD) and omit *Diversity* (Div.) due to its unclear role in evaluating motion performance. For clarity, we segment the generated motions into semantic and transitional parts and evaluate the aforementioned metrics across three groups: 1) Semantic (Smt.), which focuses solely on the semantic segments, 2) Transition (Trn.), which assesses only the transitional segments, and 3) Overall, which evaluates both semantic and transitional segments together. For the contextual alignment of transitional segments, we use the text from both the preceding and succeeding motions, assuming that the transition should retain the semantic information from overlapping segments.

For **unconditional motion inbetweening**, we assess the Transition Realism by using L2 losses and NPSS [39], as described in [27, 19], by comparing them to the ground truth inbetweening motion. Specifically, we focus on L2 losses for joint velocity (*L2-Vel*) and 6D rotation [38] (*L2-Rot6D*) to provide a direct and explicit evaluation of human motion in the HumanML3D [6] format. Additionally, we assess Transition Smoothness using root mean squared jerk [40] (RMS-Jerk) over joint rotations. Detailed descriptions of these evaluation metrics will be included in the supplementary material.

## 4.2 Compositional Motion Generation Performance Evaluation

### 4.2.1 Compositional Motion Pair Generation

The compositional motion pair experiment follows the setup illustrated on the left in Fig. 3, with the objective of generating motions  $\mathbf{X}_p$ ,  $\mathbf{X}_t$ , and  $\mathbf{X}_s$  based on the corresponding text condition pairs  $(C_p, C_s)$ . We compare the performance of our method with long motion generation models, including TEACH [12], PCMDM [14], and priorMDM [13], among with single text-conditioned models MDM-30 [8] and MLD-30 [22] as baselines. In single text-conditioned models, additional frames are generated for preceding and succeeding motions to create a 30-frame overlapping region, which is then blended linearly to form smooth transitions. The evaluation employs the *FID* and *MMD* metrics across three groups: 1) Semantic ( $\mathbf{X}_p$ ,  $\mathbf{X}_s$ ), 2) Transition ( $\mathbf{X}_t$ ), and 3) Overall ( $\mathbf{X}_p$ ,  $\mathbf{X}_t$ ,  $\mathbf{X}_s$ ). The experiment results are summarized in Tab. 1, showing that our model produces realistic motions and achieves the best overall *FID* and *MMD* scores. Strong overall performance demonstrates the potential to generate high-quality compositional motions with natural transitions. Although TEACH shows strong contextual alignment in semantic segments, its lower *Overall FID* score indicates a compromise in motion quality, leading to motion artifacts such as changes in the root roll angle, as shown in Fig. 5.

**Remark.** The performance comparison with MDM-30 highlights the effectiveness of SPDM in single-sequence text-to-motion generation. Since only a very short blending window is applied between segments, evaluating *Smt. FID* and *Smt. MMD* for MDM-30 is essentially equivalent to evaluating two independently generated motion sequences, effectively reflecting the single-sequence text-to-motion generation scenario. Therefore, the superior performance of our method on *Smt. FID* and *Smt. MMD* compared to MDM-30 underscores the competitive performance of SPDM in the standalone text-to-motion generation task.

### 4.2.2 Long-term Motion Generation

To assess the long-term motion generation performance, we combine all text conditions from the testing dataset into a single extended text sequence of 3,164 texts, and apply comparison models to

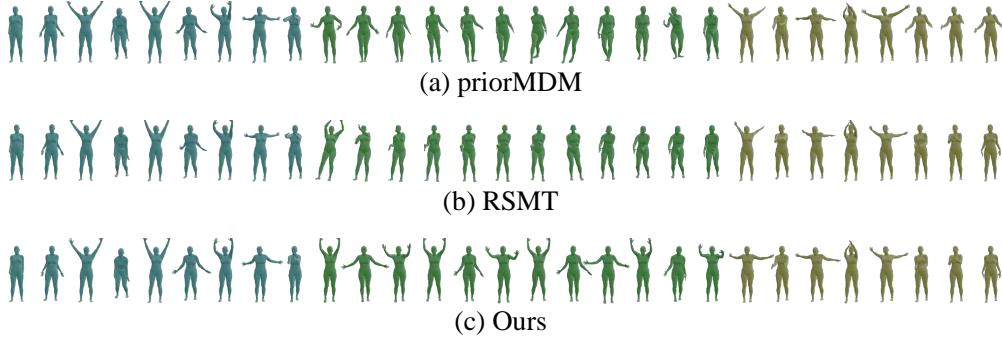


Figure 6: Visualization of the **UMIB** with 120 transition frames: preceding motion in blue, transitioning motion in green, and succeeding motion in yellow.

Table 3: Quantitative results for **Unconditional Motion Inbetweening (UMIB)** on BABEL-TEACH [12] test set. We report the performance under settings of transition lengths at 60, 120, and 180 frames. **Bold** and underline indicates the **best** and the second-best result.

UMIB	Transition Realism									Transition Smoothness			
	L2-Vel $\downarrow$			L2-Rot6D $\downarrow$			NPSS $\downarrow$			RMS-Jerk $\downarrow$			
	Length	60	120	180	60	120	180	60	120	180	60	120	180
RMIB [27]		0.0353	0.0307	0.0297	0.3510	0.3797	0.3789	2.0908	5.7334	9.1055	1.9216	1.3794	1.1009
RSMT [20]		0.0345	0.0273	0.0246	0.2151	0.2438	0.2654	0.8552	2.8986	4.836	1.6435	1.4670	1.3249
CMB [30]		0.0200	0.0172	0.0175	0.2194	0.2209	0.2289	0.4085	1.0278	1.6723	2.0658	1.9885	1.9298
MDM [8]		0.0302	0.0275	0.0319	0.2961	0.3309	0.3099	1.0369	3.2291	4.1440	1.9283	2.1205	2.0688
priorMDM [13]		<u>0.0151</u>	<u>0.0141</u>	<u>0.0144</u>	0.2398	0.2455	0.2479	0.5640	1.2107	1.7469	<u>0.4058</u>	<u>0.4495</u>	<u>0.2803</u>
Ours		<b>0.0101</b>	<b>0.0102</b>	<b>0.0125</b>	<b>0.2124</b>	<b>0.2205</b>	<b>0.2220</b>	<b>0.3651</b>	<b>0.9296</b>	<b>1.6308</b>	<b>0.0963</b>	<b>0.1054</b>	<b>0.1213</b>

generate long motion across 302,298 frames (168 minutes). As shown in Tab. 2, experiment results indicate that our method achieves competitive performance in both motion realism (*Overall FID*) and text alignment (*Overall MMD*) even in long motion generation scenarios. This demonstrates the superiority of our phase modeling approach for transition-aware motion generation, compared to other methods that model transitions directly in the raw motion space. Among the other compared methods, TEACH continues to struggle with poor transition generation, resulting in a high *Trn. FID*. PriorMDM and PCMDM face challenges in generating realistic semantic segments that align with the text input. Note that MDM and MLD show similar performance in this task as in compositional motion pair generation, mainly because the transitions between subsequences are created through blending rather than being generated by machine learning models.

### 4.3 Motion Inbetweening Performance Evaluation

The unconditional motion inbetweening (UMIB) experiment follows a setup similar to Fig. 4, where a specific number of frames around the transition boundary of testing motion pairs are masked to evaluate various methods for reconstructing the masked motion content. We assess the *L2-Vel*, *L2-Rot6D*, and *NPSS* metrics by comparing the generated segments with the actual masked content, while the *RMS-Jerk* metrics evaluate motion smoothness. The effectiveness of motion inbetweening is analyzed and compared with autoregressive frame prediction methods, RMIB [27] and RSMT [20], as well as interval infilling methods, CMB [30], MDM [8], and priorMDM [13], across three different inbetweening length settings, all within the training motion length range of [45, 250]. The results are shown in Tab. 3, demonstrating the superior performance of our proposed framework across all inbetweening length settings. Additionally, Fig. 6 illustrates the smoothness and realism of our generated results, as reflected in the metric values. In contrast, the results generated by priorMDM tends to exhibit hyperactivity by producing random motion content unrelated to the adjacent motion, which negatively impacts the overall inbetweening performance. Lastly, RSMT results reveal a failure to connect the succeeding motions. This highlights the limitations of autoregressive processing with fixed-window phase latents, which also justifies both priorMDM and our method for managing variable-length motion as a cohesive entity.

In addition to the UMIB experiment, we also conducted the conditional motion inbetweening (CMIB) experiment to assess the effectiveness of conditioning the inbetweening region with text context. The results of this experiment are detailed in the Appendix.

#### 4.4 Ablation Studies and User Studies

We assess the effects of our proposed modules and recommended hyperparameters on compositional motion generation and motion inbetweening tasks. Firstly, the integration of *frame-level tokens* within SPDM and TPDM significantly enhances their performance in denoising *param-level tokens*. Secondly, we assess the phase mixing parameter setting, revealing that  $r = (\frac{k}{K})^3$  is optimal for semantic conditional scenarios, while  $r = 1$  is best for unconditional scenarios. Moreover, in the user study, our approach attains the highest scores for motion realism and smoothness. Further details of the ablation studies will be provided in the supplementary material.

### 5 Conclusion and Future Work

We present the Transitional Phase Diffusion Module (TPDM) and the Semantic Phase Diffusion Module (SPDM), which operate within the periodic latent space generated by the Action-Centric Periodic Autoencoder. These modules inject semantic guidance and neighbouring phase information into the motion denoising process, enabling the generation of semantically meaningful motion clips with smooth transitions. The proposed Compositional Phase Diffusion pipeline, which incorporates both the TPDM and SPDM modules, can be adapted for compositional motion generation and motion inbetweening tasks. Its flexibility to handle multiple motion segments simultaneously enhances its capability to tackle complex motion sequencing tasks. Extensive experiments and evaluations have showcased the framework’s effectiveness in compositional motion generation and motion inbetweening tasks. Further exploration of these frameworks holds promise for the development of advanced motion-generation techniques in the future. As our framework applies compositional diffusion in motion generation using a basic phase mixing technique, potential performance improvement may be achievable by incorporating advanced methods like score-based or potential-based diffusion. Additionally, incorporating learnable parameters or an adaptive mechanism for phase mixing could further enhance results. However, implementing such features is challenging and requires more detailed data modelling and complex architectures. Future research will focus on adjusting the architecture and data representation to incorporate these advanced diffusion techniques.

### References

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [3] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [4] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021.
- [5] Junkun Jiang, Jie Chen, and Yike Guo. A dual-masked auto-encoder for robust motion capture with spatial-temporal skeletal token completion. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5123–5131, 2022.
- [6] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.
- [7] Junkun Jiang, Jie Chen, Ho Yin Au, Mingyuan Chen, Wei Xue, and Yike Guo. Every angle is worth a second glance: Mining kinematic skeletal structures from multi-view joint cloud. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[8] Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

[9] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pages 580–597. Springer, 2022.

[10] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8255–8263, 2023.

[11] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[12] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gü̈l Varol. Teach: Temporal action composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pages 414–423. IEEE, 2022.

[13] Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *The Twelfth International Conference on Learning Representations*, 2024.

[14] Zhao Yang, Bing Su, and Ji-Rong Wen. Synthesizing long-term human motions with diffusion models via coherent sampling. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3954–3964, 2023.

[15] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

[16] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[17] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 39(4):54–1, 2020.

[18] Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–18, 2022.

[19] Paul Starke, Sebastian Starke, Taku Komura, and Frank Steinicke. Motion in-betweening with phase manifolds. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–17, 2023.

[20] Xiangjun Tang, Linjun Wu, He Wang, Bo Hu, Xu Gong, Yuchen Liao, Songnan Li, Qilong Kou, and Xiaogang Jin. Rsmt: Real-time stylized motion transition for characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, 2023.

[21] Weilin Wan, Yiming Huang, Shutong Wu, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Diffusionphase: Motion diffusion in frequency domain. *arXiv preprint arXiv:2312.04036*, 2023.

[22] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[24] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16010–16021, 2023.

[25] Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2151–2162, 2023.

[26] Félix G Harvey and Christopher Pal. Recurrent transition networks for character locomotion. In *SIGGRAPH Asia 2018 Technical Briefs*, pages 1–4. 2018.

[27] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.

[28] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927. IEEE, 2020.

[29] Alejandro Hernandez, Jurgen Gall, and Francesc Moreno-Noguer. Human motion prediction via spatio-temporal inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7134–7143, 2019.

[30] Jihoon Kim, Taehyun Byun, Seungyoun Shin, Jungdam Won, and Sungjoon Choi. Conditional motion in-betweening. *Pattern Recognition*, 132:108894, 2022.

[31] Seunggeun Chi, Hyung-gun Chi, Hengbo Ma, Nakul Agarwal, Faizan Siddiqui, Karthik Ramani, and Kwonjoon Lee. M2d2m: Multi-motion generation from text with discrete diffusion models. In *European Conference on Computer Vision*, pages 18–36. Springer, 2024.

[32] Zeyu Zhang, Akide Liu, Qi Chen, Feng Chen, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Infinimotion: Mamba boosts memory in transformer for arbitrary long motion generation. *arXiv preprint arXiv:2407.10061*, 2024.

[33] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, 2002.

[34] Jianyuan Min and Jinxiang Chai. Motion graphs++: a compact generative model for semantic motion analysis and synthesis. *ACM Transactions on Graphics*, 31(6):1–12, 2012.

[35] Ho Yin Au, Jie Chen, Junkun Jiang, and Yike Guo. Choreograph: Music-conditioned automatic dance choreography over a style and tempo consistent dynamic graph. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3917–3925, 2022.

[36] Ho Yin Au, Jie Chen, Junkun Jiang, and Yike Guo. Rechoreonet: Repertoire-based dance re-choreography with music-conditioned temporal and style clues. *Machine Intelligence Research*, pages 1–11, 2024.

[37] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[38] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[39] Anand Gopalakrishnan, Ankur Mali, Dan Kifer, Lee Giles, and Alexander G Ororbia. A neural temporal model for human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12116–12125, 2019.

[40] Raymond P Young and Ronald G Marteniuk. Acquisition of a multi-articular kicking task: Jerk analysis demonstrates movements do not become smoother with learning. *Human Movement Science*, 16(5):677–701, 1997.

## 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: The abstract and introduction outline the key challenges addressed, our contributions, and a summary of the experimental 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: We include the limitation of our work in Section 5.

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: [NA]

Justification:

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 described the experimental details and have properly cited the open-source assets and materials necessary for reproducing the experiments. The code will be made publicly available upon acceptance.

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: [\[Yes\]](#)

Justification: We used the public dataset, and we plan to publicly release our code once the status of the paper is confirmed.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: See Section 4.1 and Appendix A.

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: Due to the limited computational resources, we conducted each experiment only once.

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: See Appendix A and supplementary material.

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: We fully adhere to the NeurIPS Code of Ethics.

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: [\[Yes\]](#)

Justification: See Appendix C.

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: The 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: The paper properly credits the creators or original owners of the assets used, such as code, data, and models. We also have correctly cited the relevant literature.

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](http://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: We will disclose the code after the confirmation of the status of the paper, but currently no new assets is available.

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.

## A Technical Details of Compositional Phase Diffusion

### A.1 Compositional Motion Generation Algorithm

Algorithm implementation details of the phase diffusion pipeline for the compositional motion generation task in Sec. 3.2.1. The detail is also illustrated in Fig. 3.

---

**Algorithm 1** Compositional Phase Diffusion on generating sequence composed by 2 semantic conditioned segments

---

```

Require: forward TPDM  $\mathcal{F}_{T_f}(\cdot)$ , backward TPDM  $\mathcal{F}_{T_b}(\cdot)$ , SPDM  $\mathcal{F}_s(\cdot)$ , DiffusionScheduler  $\mathcal{F}_D(\cdot)$ 
 $\mathbf{P}_p^K, \mathbf{P}_s^K, \mathbf{P}_t^K \sim \mathbb{N}(\mathbf{0}, \mathbf{I})$   $\triangleright$  sample latent for  $\mathbf{X}_p, \mathbf{X}_s, \mathbf{X}_t$ 
 $\mathbf{P}_p^0, \mathbf{P}_s^0, \mathbf{P}_t^0 \leftarrow \mathbf{P}_p^K, \mathbf{P}_s^K, \mathbf{P}_t^K$   $\triangleright$  clean latent placeholder
for  $k$  from  $K$  to 1 do
    # predict phase latent noise using SPDM and TPDMs
     $\mathbf{P}_{pc}^0 \leftarrow \mathcal{F}_s(k, C_p, \mathbf{P}_p^k), \mathbf{P}_{pb}^0 \leftarrow \mathcal{F}_{T_b}(k, \mathbf{P}_p^k, \mathbf{P}_t^0)$ 
     $\mathbf{P}_{tf}^0 \leftarrow \mathcal{F}_{T_f}(k, \mathbf{P}_t^k, \mathbf{P}_p^0), \mathbf{P}_{tb}^0 \leftarrow \mathcal{F}_{T_b}(k, \mathbf{P}_t^k, \mathbf{P}_s^0)$ 
     $\mathbf{P}_{sf}^0 \leftarrow \mathcal{F}_{T_f}(k, \mathbf{P}_s^k, \mathbf{P}_t^0), \mathbf{P}_{sc}^0 \leftarrow \mathcal{F}_s(k, C_s, \mathbf{P}_s^k)$ 
    # perform phase mixing as in Eq. 2
     $\mathbf{P}_p^0 \leftarrow (\frac{k}{K})^3 \mathbf{P}_{pb}^0 + (1 - (\frac{k}{K})^3) \mathbf{P}_{pc}^0$   $\triangleright r_p = (\frac{k}{K})^3$ 
     $\mathbf{P}_t^0 \leftarrow \frac{\mathbf{P}_{tf}^0 + \mathbf{P}_{tb}^0}{2}$   $\triangleright r_t = 1$ 
     $\mathbf{P}_s^0 \leftarrow (\frac{k}{K})^3 \mathbf{P}_{sf}^0 + (1 - (\frac{k}{K})^3) \mathbf{P}_{sc}^0$   $\triangleright r_s = (\frac{k}{K})^3$ 
    # estimate phase latent at both step  $k - 1$  and step 0
     $\mathbf{P}_p^{k-1}, \mathbf{P}_p^0 \leftarrow \mathcal{F}_D(\mathbf{P}_p^0, \mathbf{P}_p^k, k - 1)$ 
     $\mathbf{P}_t^{k-1}, \mathbf{P}_t^0 \leftarrow \mathcal{F}_D(\mathbf{P}_t^0, \mathbf{P}_t^k, k - 1)$ 
     $\mathbf{P}_s^{k-1}, \mathbf{P}_s^0 \leftarrow \mathcal{F}_D(\mathbf{P}_s^0, \mathbf{P}_s^k, k - 1)$ 
end for
return  $\mathbf{P}_p^0, \mathbf{P}_t^0, \mathbf{P}_s^0$ 

```

---

### A.2 Adjustment to $T$ and $PE$

As discussed in Sec. 3.1.1, we have refined the sinusoidal positional embedding  $PE$  and the time window  $T$  to support motion autoencoding with variable lengths.

Positional embedding  $PE$  is crucial for accurately representing time progression in motion encoding and generation. Traditional sinusoidal positional embeddings  $PE$  only reflect the time progression signal from the leading frame of motion *leading* frame of motion  $\mathbf{X}$ . In contrast, our composite positional embedding **Comp-PE** creates duplicates of the positional embedding shifted to the *middle* and *ending* frames. These duplicates are stacked channel-wise, enhancing the model’s awareness of sequential action progression from three key locations and improving semantic understanding.

On the other hand, the time window  $T \in \mathbb{R}^{N \times Q}$  is essential for transforming the fixed-size parameters  $\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}$  into the variable length periodic signal  $\mathbf{Q}$  following Equation 1 in the main paper. In traditional PAEs [15, 21],  $T$  is defined as a fixed-length time window ranging from  $-1$  to  $1$  across 121 frames. When extending  $T$  to adapt to variable lengths, two variants arise: (**normT**) parameterizes the time window from  $-1$  to  $1$  over the frame count  $N$  to correspond with normalized action progression, and (**frameT**) parameterizes the time window from  $-\frac{N}{2}$  to  $\frac{N}{2}$  to align with the actual time duration. In this work, we employ a mixed linear parameterization (**mixT**), where  $\frac{Q}{2}$  channels within the time window  $T$  are parameterized using **frameT**, while the other half is parameterized using **normT**.

Note that the time window parameterizations are implemented piecewise to address the length imbalance in the transitioning motion  $\mathbf{X}_t$ <sup>3</sup>. Using the settings in Fig. 3 of the main paper as an example, **normT** parameterize the  $\mathbf{X}_t$  left frame range  $[t_{tl}, t_{tm}]$  as  $[-1, 0]$ , and the  $\mathbf{X}_t$  right frame range  $[t_{tm}, t_{te}]$  as  $[0, 1]$ . Similarly, **frameT** parameterize the  $\mathbf{X}_t$  right frame range  $[t_{pm}, t_{te}]$  as  $[0, t_{te} - t_{tm}]$  to align with the actual time duration.

<sup>3</sup> $\mathbf{X}_t$  represents the segment covering the second half of  $\mathbf{X}_p$  and the first half of  $\mathbf{X}_s$ . For example, if  $\mathbf{X}_p$  is 2 seconds and  $\mathbf{X}_s$  is 8 seconds,  $\mathbf{X}_t$  will span 5 seconds, covering the last 1 second of  $\mathbf{X}_p$  and the first 4 seconds of  $\mathbf{X}_s$ . Note that the middle frame of  $\mathbf{X}_t$  is defined at the transition boundary; when  $\mathbf{X}_p$  and  $\mathbf{X}_s$  have unequal lengths, this middle frame is offset from the center of  $\mathbf{X}_t$ .

### A.3 Details of SPDM and TPDM

As discussed in Sec. 3.1.2 and Sec. 3.1.3 in the main paper, our diffusion models are designed as  $\epsilon$ -model [25], which is trained using the  $\ell_1$  loss (formulated as  $\|\cdot\|_1$ ) to the diffusion noise  $[\epsilon_p, \epsilon_t, \epsilon_s]$  which was scheduled to diffuse the clean phase latent of the motion segments to  $[P_p^k, P_t^k, P_s^k]$  respectively. Note that the estimation of diffused latent at the next diffusion step (prev\_sample) and the clean phase latent (pred\_original\_sample) is also supported by the commonly used DDIM diffusion scheduler [3]:  $P_p^{k-1}, P_p^0 \leftarrow \mathcal{F}_D(P_p^0, P_p^k, k-1)$ .

SPDM is trained using semantically annotated motion segments in the BABEL-TEACH [12] training dataset, which are the preceding motion  $X_p$  and succeeding motion  $X_s$  in each motion subsequence pair, each associated with text annotations  $C_p$  and  $C_s$ . The training loss of SPDM on each subsequence pair is illustrated as follows:

$$\mathcal{L}_S = \|\mathcal{F}_S(k, C_p, P_p^k) - \epsilon_p\|_1 + \|\mathcal{F}_S(k, C_s, P_s^k) - \epsilon_s\|_1.$$

On the other hand, TPDM is trained based on the neighbouring information in each motion subsequence pair. Specifically, we can obtain 2 transitional segment pair  $(X_p, X_t)$ ,  $(X_t, X_s)$  for each motion data tuple  $(X_p, X_t, X_s)$ . Then,  $TPDM_f$  and  $TPDM_b$  are trained on each transitional segment pair to denoise motion phase parameters using neighbouring phase information from either the forward or backward direction. The training loss of TPDMs on each subsequence pair are illustrated as follows:

$$\begin{aligned} \mathcal{L}_{T_f} &= \|\mathcal{F}_{T_f}(k, P_t^k, P_p^0) - \epsilon_t\|_1 + \|\mathcal{F}_{T_f}(k, P_s^k, P_t^0) - \epsilon_s\|_1, \\ \mathcal{L}_{T_b} &= \|\mathcal{F}_{T_b}(k, P_p^k, P_t^0) - \epsilon_p\|_1 + \|\mathcal{F}_{T_b}(k, P_t^k, P_s^0) - \epsilon_t\|_1. \end{aligned}$$

### A.4 Implementation Details

We apply the emphasis projection with  $c = 15$ , as demonstrated in GMD [25], to incorporate root trajectory information into the motion representation. Also, our models are designed based on phase latent size  $Q = 512$ , which serves as both the latent dimension for all diffusion modules and the number of periodic signals in ACT-PAE. For the diffusion step setting in SPDM and TPDM, DDIM [3] is utilized for 1000 training steps and 100 inference steps.

## B Conditional Motion Inbetweening Evaluation

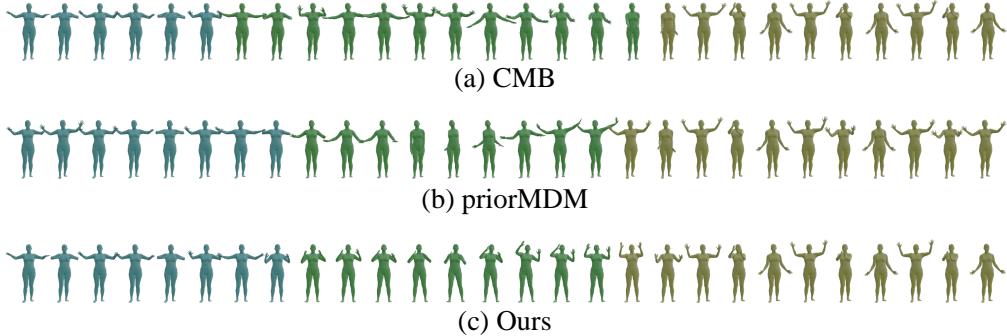


Figure 7: Visualization of the **CMIB** with 120 transition boundary frames conditioned with *bend arms up*: preceding motion in blue, transitioning motion in green, and succeeding motion in yellow.

As shown in Fig. 4, our framework can integrate SPDM into the denoising process of inbetweening segments to enable conditional motion inbetweening. We use the same testing setup as in unconditional motion inbetweening, focusing the evaluation on the inbetweening region. We evaluate our framework against CMB [30], MDM [8], and priorMDM [13]. The results, presented in Tab. 4, demonstrate that our method excels at producing natural inbetweening motion while adapting to input text semantics. As illustrated in Fig. 7, our framework creates smooth inbetweening motion that corresponds well with the input text condition *bend arms up*. In contrast, both priorMDM and CMB show hyperactivity, resulting in abrupt inbetweening motion that does not align with the text.

Table 4: Quantitative results for **Conditional Motion Inbetweening (CMIB)** on the BABEL-TEACH [12] test set. We report the performance of various methods under the settings of transition lengths at 60, 120, and 180 frames. **Bold** and underline indicates the **best** and the second-best result.

Length	Motion Realism			Text Alignment		
	Smt. FID ↓			MMD ↓		
	60	120	180	60	120	180
CMB	<u>0.693</u>	<u>1.382</u>	2.765	7.420	<u>7.544</u>	7.561
MDM	0.694	1.482	<u>2.626</u>	<u>7.411</u>	7.609	7.658
priorMDM [13]	1.613	1.392	<u>5.699</u>	<u>7.761</u>	8.075	<u>7.544</u>
Ours	<b>0.389</b>	<b>0.679</b>	<b>2.152</b>	<b>7.213</b>	<b>6.871</b>	<b>7.206</b>

## C Impact Statements

The exploration and application of phase latent spaces in this work contribute to the advancement of deep learning by offering new methodologies for signal processing and multimedia generation. It has no negative impact on society as the focus is on technological improvement rather than datasets that could be sensitive or have privacy implications.