
Supplementary Material: Deep Compositional Phase Diffusion for Long Motion Sequence Generation

1 Experiment Details

1.1 Training Details

For the model structure, all the transformers, including those in ACT-PAE, TPDM, and SPDM, are structured with the following configurations, i.e., $nhead=8$, $dim_feedforward=1024$, $dropout=0.1$. We adopt the CLIP text encoder [1] with version *ViT-B/32*. All the proposed modules in our framework, including ACT-PAE, TPDM, and SPDM, are implemented in PyTorch and trained with AdamW [2] optimizer with a learning rate of 1×10^{-4} . For the ACT-PAE, the motion encoder and decoder are 8-layer transformer encoders with the encoded phase latent channel size 512, and are trained for 6500 epochs in a batch size of 96. For the two TPDMs, the denoising transformers are 8-layer transformers and are trained for 2500 epochs in a batch size of 48. Finally, the SPDM is an 8-layer transformer encoder with pre-trained *CLIP-ViT-B/32* [1] frozen model, which is trained for 9000 epochs in a batch size of 256. All modules are trained with 2x4090 Nvidia GPU for one day.

1.2 Empirical Running Time of the Pipeline

We empirically measured inference time for long-term motion generation. For scenarios with 2 subsequences, the total inference time is approximately 1.705 seconds, while for 10 subsequences, it is about 2.015 seconds. The SPDM and TPDM models require approximately 0.010 and 0.015 seconds per inference, respectively. Since subsequences are stacked along the batch dimension and processed in parallel by SPDM and TPDM, the model inference time per loop remains around 0.015 seconds, independent of the number of subsequences. In contrast, the phase mixing and rearrangement of the inferred subsequences require 0.002 seconds for 2 subsequences and 0.005 seconds for 10 subsequences. After 100 diffusion loops, the total diffusion stage takes roughly 1.7 seconds for 2 subsequences and 2.0 seconds for 10 subsequences. For the final motion decoding step, decoding times are 0.005 seconds for 2 subsequences and 0.015 seconds for 10 subsequences. In contrast to SPDM and TPDM inference, the subsequence rearrangement and linear blending steps do scale with the number of subsequences, as linear blending across overlapping regions cannot be efficiently parallelized.

1.3 Descriptions of Evaluation Metrics

For the metrics on evaluating compositional motion generation performance, we use pre-trained t2m evaluators from priorMDM [3], which employs the same Bidirectional GRU architecture as the T2M [4] evaluation pipeline. The *Frechet Inception Distance (FID)* measures the motion latent distribution difference between the 2 multivariate Gaussian distributions modelled by the generated motions and real motions, while *Multi-modal Distance (MMD)* calculates the Euclidean distance between motion and text latent codes in the same data pair. Note that the T2M metrics also include *Motion-retrieval Rank Precision (R-prec.)*, which samples a batch of 32 motion latent codes and then assesses each motion code’s average top 1/2/3 matching accuracy compared to the 32 text latent codes within a batch, and *Diversity (DIV)*, which measures the variance of generated motion latent features. However, *R-prec.* duplicates *MMD* since both assess text-motion alignment using identical motion and text latent codes. We prefer *MMD* over *R-prec.* because *R-prec.* is more susceptible to the randomness of test batch sampling. Finally, *DIV* is not a clear performance indicator, as the preferred diversity level is unknown.

Note that we evaluated the compositional motion generation metrics in three groups: 1) *Semantic* ($\mathbf{X}_p, \mathbf{X}_s$), 2) *Transition* (\mathbf{X}_t), and 3) *Combined* ($\mathbf{X}_p, \mathbf{X}_t, \mathbf{X}_s$). The distinction between these groups

lies in their evaluation targets. For a motion data tuple $(\mathbf{X}_p, \mathbf{X}_t, \mathbf{X}_s)$ and corresponding text data tuple (C_p, C_s) , *Semantic (Smt.)* focuses on (\mathbf{X}_p, C_p) and (\mathbf{X}_s, C_s) , while *Transition (Trn.)* evaluates (\mathbf{X}_t, C_p) , (\mathbf{X}_t, C_s) with the assumption that the transition should retain the semantic information from the overlapping segments. *Overall* evaluates all four tuples: (\mathbf{X}_p, C_p) , (\mathbf{X}_t, C_p) , (\mathbf{X}_t, C_s) , and (\mathbf{X}_s, C_s) . After decomposing the text-motion tuples, they are fed into pre-trained t2m evaluators to obtain *FID* and *MMD*.

For the motion inbetweening metrics, we adapt the metrics from [5, 6] to the HumanML3D [4] data format. Since the HumanML3D format uses a canonicalized pose representation with 6D rotation [7] and joint velocities, we replace the L2 distances of *joint positions (L2P)* and *joint rotation in quaternions (L2Q)* with *joint velocities (L2-Vel)* and *joint rotation in 6D rotation (L2-Rot6D)*. For the *Normalized Power Spectrum Similarity (NPSS)*, which evaluates the differences in joint angle power spectrum distributions between the ground truth masked motions and the generated motions in the inbetweening region, we modify it to use 6D rotation instead of Euler angles. Note that this also allows us to measure the difference from the ground truth motions within the inbetweening region more accurately, as it helps avoid issues such as Euler angle discontinuity and quaternion ambiguity. Furthermore, we also assess the *root mean squared jerk (RMS-Jerk)* [8], which is one of the traditional jerk-based smoothness metrics for evaluating movement smoothness. A low *RMS-Jerk* is generally preferred because it indicates smoother motion with fewer rapid changes in acceleration. High *RMS-Jerk* values suggest abrupt changes, which can lead to motion artifacts such as sudden starts, stops, or changes in direction.

Among all these metrics, *Overall FID* is the most crucial metric for the compositional motion generation task, as it comprehensively evaluates the overall quality of the generated motion. Following this, *Overall MM-Dist* assesses the overall alignment between the text and motion, which is also significant. For motion inbetweening, *NPSS* stands out as the most important metric, as it is more closely aligned with human assessments of motion quality [5, 9].

1.4 Settings for the Comparison Models

For priorMDM [3], we utilize its provided doubletake text-to-motion pipeline with a default handshake length of 30, while adjusting the input motion lengths to ensure that the total length of the final motion meets our specifications. To apply priorMDM to motion inbetweening tasks, we follow the double-take strategy outlined in the paper, but we perform motion replacement during the first take. To be concise, after synthesizing a valid handshake interval connected to the input motion, we proceed to the second step, which involves determining the number of transition boundary frames. Additionally, we inject semantic conditions during the second step diffusion when evaluating the CMIB task.

For PCMDM [10], we utilize the provided compositional transition sampling pipeline with the default parameter settings of 2 inpainting frames.

For TEACH [11], we utilize its provided text-to-motion sampling pipeline with a default slerp window of 8.

For RMIB [5], we retrieve the first and last frames within the inbetweening region and input them into the provided pipeline for further processing.

For RSMT [12], we begin by processing the entire motion to extract the phase latent vector. We then retrain RSMT on the BABEL-TEACH dataset using a phase latent size of 10. After retraining, we follow the autoregressive frame prediction structure as in RMIB.

For the single text-conditioned motion generation models, such as MDM [13], MLD [14], and CMB [15], we employ a pre-trained *CLIP-ViT-B/32* model as the text encoder to train a transformer encoder for motion generation tasks. During training, these models are trained to denoise semantic segments individually. In the compositional motion generation tasks, MDM and MLD generate two semantic segments independently, without being inherently aware of the potential transitions between them. To facilitate smooth transitions, additional frames are generated for the preceding and succeeding motions to create a 30-frame overlapping region for linear blending. In the motion in-betweening tasks, the CMB model uses an interpolated pose segment from adjacent motions as input during training. We also provide such an interpolated pose segment during inference.

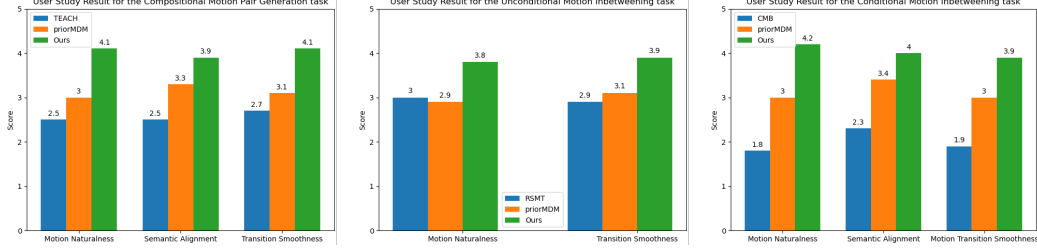


Figure 1: Results of the user evaluation on the generated motions are presented. The figures for the Compositional Motion Pair Generation, Unconditional Motion Inbetweening (UMIB), and Conditional Motion Inbetweening (CMIB) tasks are displayed from left to right, respectively. It is important to note that UMIB does not consider text input, so its evaluation regarding Semantic Alignment is omitted.

Other models, such as InfiniMotion [16] and M2D2M [17], are not included in the comparison due to the unavailability of their source code. Note that we also attempted to use PhaseBetweener [6]. However, despite retraining the DeepPhase [18] model on BABEL-TEACH according to their instructions, the provided Unity inference pipeline does not support importing trained models with different parameter settings. These limitations have prevented us from applying PhaseBetweener as a comparison for evaluating the UMIB task.

2 User Study

We also conducted user studies to assess whether the generated motion quality aligns with human perception. For this evaluation, we prepared 30 motion clips for each task: Compositional Motion Pair Generation, Unconditional Motion Inbetweening (UMIB) with 60 transition frames, and Conditional Motion Inbetweening (CMIB) with 120 transition frames. These clips included visualizations of the motions generated by our method as well as those produced by other comparison methods discussed in the main paper.

In the user study for each task, 5 motion clips were randomly selected from the 30 motion clips. Each participant was asked to rate them on a scale of 1 to 5 (with 5 indicating the highest quality) across three aspects described below:

- **Motion Naturalness.** On a scale of 1 to 5, where 1 represents very unnatural and 5 signifies very natural, how would you rate the naturalness and realism of the motions? Are these motions created by a machine or performed by a real human?
- **Semantic Alignment.** On a scale of 1 to 5, where 1 indicates unrelated and 5 denotes a perfect match, how well do the motions align with the text descriptions?
- **Motion Transition Smoothness.** On a scale of 1 to 5, where 1 represents very unnatural and 5 signifies very natural, how smooth and natural are the transitions from the “1st motion” to the “2nd motion”?

We have collected answers from 30 independent judges, and the results are shown in Fig. 1. Our method received the highest scores across all three aspects of compositional motion pair generation, with the most significant advantage noted in motion transition smoothness. Furthermore, in the case of motion inbetweening with 60 transition frames, our method outperformed the others, particularly excelling in motion transition smoothness. This clearly demonstrates the benefits of utilizing phase latent space for motion analysis. Lastly, for conditional motion inbetweening with 120 transition frames, our method achieved the highest scores across all three evaluated aspects, with the most notable advantage observed in motion naturalness. This highlights its capability to generate semantically aligned motion over a 120-frame range (equivalent to 4 seconds) while ensuring smooth transitions between adjacent motion clips.

Table 1: Ablation Studies on the proposed module and design.

Design choice	Compositional Motion Pair Generation					
	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
not ϵ -model	4.103	3.544	3.411	4.914	7.514	5.931
w/o emp. proj.	0.710	2.576	0.958	3.569	7.488	5.133
w/o frame-token Both	40.526	11.348	19.451	7.678	6.721	7.187
w/o frame-token SPDM	34.33	9.057	15.591	7.756	6.728	7.226
w/o frame-token TPDM	4.777	4.837	3.621	4.712	7.036	5.677
w/o linear blending	0.729	2.295	1.061	3.570	5.782	4.970
Ours	0.736	1.807	0.782	3.509	6.545	4.711

Table 2: Ablation Studies on the time window parameterization and positional embedding.

T	PE	Compositional Motion Pair Generation					
		Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
frameT	PE	0.826	1.952	0.946	3.746	6.505	5.14
normT	PE	0.681	3.538	1.409	3.461	6.608	4.782
mixT	PE	0.499	2.064	0.828	3.474	6.583	4.751
frameT	Comp-PE	3.017	3.495	2.742	5.766	6.428	6.328
normT	Comp-PE	0.553	2.559	0.988	3.465	6.584	4.803
mixT	Comp-PE	0.736	1.807	0.782	3.509	6.545	4.711

3 Ablation Studies Details

This section presents the results of the ablation studies for our proposed method. The results are organized as follows: Sec. 3.1 examines the impact of different design choices for diffusion modules. Sec. 3.2 explores the effects of time window parameterization T and positional embedding PE . Sec. 3.3 evaluates the influence of varying the number of phase channels. Sec. 3.4 investigates the role of emphasis projection weight and ϵ -modelling in GMD [19]. Finally, Sec. 3.5 analyzes the selection of phase mixing parameters across different tasks. The experimental results demonstrate that our selected parameters achieve the best *Overall FID* on the compositional motion generation task, the best *NPSS* on the motion inbetweening task, and yield the highest count of **best** metric results.

Additionally, we conducted hyperparameter tuning on key parameters within the comparison models. For priorMDM, we focused on adjusting the handshake length, while for TEACH, we tuned the sleep window. In the case of single text-to-motion models like MDM and MLD, we optimized the length of the overlap region. The experimental results in Sec. 3.6 demonstrate that our model consistently outperforms these methods across various hyperparameter settings.

3.1 Diffusion module design choices

We conducted ablation studies on various design choices for the SPDM and TPDM. The experiments included scenarios where the ϵ -model is not used (not ϵ -model), emphasis projection is omitted (w/o emp. proj.), both SPDM and TPDM do not utilize *frame-level tokens* (w/o frame-token Both), and cases where either SPDM or TPDM lacked *frame-level tokens* (w/o frame-token SPDM and w/o frame-token TPDM). Finally, we performed an experiment without applying linear blending after decoding motion segments with ACT-PAE (w/o linear blending), in which the generated transitional segment was discarded. The findings, summarized in Tab. 1, highlight that all design choices are essential for the effective training of TPDM and SPDM. Specifically, if SPDM does not incorporate *frame-level tokens*, it fails to account for crucial motion duration information during motion generation, reducing the framework’s performance to levels similar to those of MLD [14]. This oversight leads to SPDM producing inappropriate phase outputs, which negatively impact TPDM, resulting in even worse *FID* performance than when SPDM is completely removed, as indicated by the result $r_p, r_s, r_t = 1$ in Tab. 5. Additionally, the absence of the ϵ -model and emphasis projection significantly degrades performance in compositional motion pair generation tasks. Lastly, omitting the final linear blending step results in only a minor decrease in quality, as phase information exchange with TPDM continues throughout the diffusion process. However, slight discontinuities may still arise, leading to a slight increase in *Trn. FID*.

Table 3: Ablation Studies on the number of phase channels. Note that compositional motion generation result for RSMT is not available as it does not take input text condition.

Method	Q	UMIB 60				Compositional Motion Pair Generation					
		L2-Vel ↓	L2-Rot6D ↓	NPSS ↓	RMS-Jerk ↓	Smt. FID ↓	Trn. FID ↓	Overall FID ↓	Smt. MMD ↓	Trn. MMD ↓	Overall MMD ↓
RSMT	10	0.0345	0.2151	0.8552	1.2490	-	-	-	-	-	-
RSMT	128	0.0347	0.2199	0.7944	1.6435	-	-	-	-	-	-
RSMT	512	0.0348	0.2213	0.7742	1.8572	-	-	-	-	-	-
Ours	16	0.0105	0.2088	0.4039	1.1292	11.567	5.451	7.603	7.247	6.546	7.350
Ours	128	0.0120	0.2182	0.4896	0.1810	2.029	3.353	2.124	4.285	6.451	5.509
Ours	512	0.0101	0.2124	0.3651	0.0963	0.736	1.807	0.782	3.509	6.545	4.711

Table 4: Ablation Studies on the emphasis projection weight.

c	UMIB 60				Compositional Motion Pair Generation					
	L2-Vel ↓	L2-Rot6D ↓	NPSS ↓	RMS-Jerk ↓	Smt. FID ↓	Trn. FID ↓	Overall FID ↓	Smt. MMD ↓	Trn. MMD ↓	Overall MMD ↓
5	0.0127	0.2187	0.4956	0.7254	0.649	2.538	0.847	3.653	6.485	5.194
10	0.0104	0.2092	0.3968	0.1668	0.464	2.216	0.838	3.678	6.604	5.020
15	0.0101	0.2124	0.3651	0.0963	0.736	1.807	0.782	3.509	6.545	4.711
20	0.0105	0.2173	0.3659	0.1079	0.899	1.593	0.987	3.736	6.383	5.043
25	0.0105	0.2088	0.4039	0.1149	1.178	2.011	1.192	3.511	6.494	5.084

3.2 Time Window Parameterization and Positional Embedding

The ablation study results for the choices of time window parameterizations among {**frameT**, **normT**, **mixT**} and positional embeddings {**Comp-PE**, **PE**} are shown in Tab. 2. We explored all combinations of time window parameterizations and positional embeddings and found that the combination of [**mixT**, **Comp-PE**] yielded the best results for compositional motion generation tasks, while [**mixT**, **PE**] came in second. Note that the choice of positional embeddings does not directly enhance performance, as seen in the results for **frameT**. Instead, effective performance relies on the appropriate selection of time window parameterizations to achieve optimal results.

3.3 Number of Phase Channels

The number of phase channels in PAE will directly influence the model expressiveness. Therefore, we conduct experiments on the number of phase channels. We also experiment on RSMT [12] for comparison. From the results shown in Tab. 3, the performance of our method improves in both UMIB and compositional motion generation tasks when the number of phase channels increases. In contrast, the performance for RSMT slightly decreases. This performance change is caused by the design improvement of the ACT-PAE in our method. As discussed in the methodology section in the main paper, the ACT-PAE encoder directly predicts the 4 phase parameters **F**, **A**, **B**, **S**, which allows ACT-PAE to model the phase latent space freely to represent different motions. However, for the traditional PAE [18] used in RSMT, the use of the FFT module on calculating the phase parameters restricts the output frequency value to be a combination of the preset frequency band (see `torch.fft.rfftfreq` in PyTorch library and the traditional PAE [18] code for details), which limitation of free use on the phase latent space degenerate the PAE expressiveness, make it performs poorly even when we largely scale up the number of phase channels.

3.4 Emphasis Projection Weight and ϵ -modelling

As discussed in GMD [19], the emphasis projection is critical for machine learning models to concentrate on the global trajectory, and the ϵ -model performs better when it integrates guidance signals from multiple sources. To investigate this further, we conducted ablation studies on the emphasis projection weight, with the results presented in Tab. 4. We found that setting the emphasis projection weight to $c = 15$ yields optimal results, delivering the best performance in terms of both *Overall FID* and *NPSS* on both UMIB and Compositional Motion Pair Generation tasks. Notably, for compositional motion pair generation tasks, *Smt. FID* performed better with ($c=10$), while *Trn. FID* and *Trn. MMD* showed improved performance at ($c=20$). This suggests a trade-off between modeling semantic segments and transitioning segments, with ($c=15$) effectively balancing these aspects.

Table 5: Ablation Studies on the phase mixing parameterization within Compositional Motion Pair Generation task.

r_p, r_s	r_t	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
0	1	1.755	3.102	1.960	4.098	5.939	5.353
0.5	1	1.974	3.227	1.912	4.156	5.965	5.318
1	1	5.626	5.042	3.568	7.955	8.279	7.67
$\frac{k}{K}$	1	1.295	3.204	1.429	3.837	5.857	5.161
$1 - \frac{k}{K}$	1	2.838	3.847	2.375	4.941	6.264	5.739
$(\frac{k}{K})^3$	1	0.736	1.807	0.782	3.509	6.545	4.711
$1 - (1 - \frac{k}{K})^3$	1	5.671	6.531	3.930	6.186	7.451	6.582
$(\frac{k}{K})^2$	1	0.742	2.293	1.185	3.487	5.72	4.759
$(\frac{k}{K})^4$	1	0.634	1.867	1.157	3.471	5.623	4.765
$(\frac{k}{K})^3$	0	1.301	1.66	1.361	4.109	6.868	5.533
$(\frac{k}{K})^3$	0.5	0.764	1.871	1.087	3.646	5.913	5.111
$(\frac{k}{K})^3$	1	0.736	1.807	0.782	3.509	6.545	4.711
$(\frac{k}{K})^3$	$\frac{k}{K}$	0.792	1.746	1.019	3.684	6.086	5.151
$(\frac{k}{K})^3$	$1 - \frac{k}{K}$	0.633	1.882	0.974	3.612	5.752	5.061

Table 6: Ablation Studies on phase mixing parameterization in Unconditional Motion Inbetweening task (UMIB 60)

r_i	r_{t_1}, r_{t_2}	UMIB 60			
		L2-Vel ↓	L2-Rot6D ↓	NPSS↓	RMS-Jerk ↓
0	1	0.0125	0.2276	0.6599	0.2125
0.5	1	0.0110	0.2125	0.4732	0.1204
1	1	0.0101	0.2124	0.3651	0.0963
$\frac{k}{K}$	1	0.0110	0.2130	0.4955	0.1341
$1 - \frac{k}{K}$	1	0.0108	0.2084	0.4368	0.1115
$(1 - \frac{k}{K})^3$	1	0.0115	0.2168	0.5205	0.1404
$1 - (\frac{k}{K})^3$	1	0.0104	0.2021	0.3740	0.1014
1	0	0.0140	0.2686	0.7190	0.1867
1	0.5	0.0112	0.2240	0.4722	0.1078
1	1	0.0101	0.2124	0.3651	0.0963
1	$\frac{k}{K}$	0.0119	0.2297	0.4925	0.1208
1	$1 - \frac{k}{K}$	0.0109	0.2155	0.4584	0.1067
1	$(1 - \frac{k}{K})^3$	0.0118	0.2405	0.5696	0.1295
1	$1 - (\frac{k}{K})^3$	0.0105	0.2025	0.3759	0.0991

3.5 Phase Mixing Parameters

We conducted ablation studies on the choice of phase mixing parameters for the compositional motion pair generation and UMIB tasks. It is important to note that the optimal parameters for CMIB can be derived from the results of the compositional motion pair generation and UMIB experiments, as the framework settings for these tasks are nearly identical.

First, we evaluate the selection of phase mixing parameters in the context of the text-to-motion task. The text-to-motion experiment is conducted based on the framework illustrated in Fig. 2 of the main paper, where the phase mixing parameters r_p, r_s focus on the semantically conditioned segments $\mathbf{X}_p, \mathbf{X}_s$, while r_t is applied to the transitioning segment \mathbf{X}_i . The results of these experiments are presented in Tab. 5. Our findings indicate that setting r_p, r_s to $(\frac{k}{K})^3$ yields the best performance in compositional motion pair generation metrics. Note that after we evaluate the first set of phase mixing parameter choices in $\{0, 0.5, 1, \frac{k}{K}, 1 - \frac{k}{K}\}$, the result of $\frac{k}{K}$ emerged as the most effective, suggesting the possibilities of having better choice around $\frac{k}{K}$. Consequently, we extended our experiments to include two additional choices $(\frac{k}{K})^3, 1 - (1 - \frac{k}{K})^3$ for the second evaluation stage. As $(\frac{k}{K})^3$ outperformed these choices, we proceeded to test $(\frac{k}{K})^2, (\frac{k}{K})^4$. Ultimately, $(\frac{k}{K})^3$ consistently remained the best choice across the metrics of *Overall FID* and *Overall MM-Dist*.

Next, for UMIB, we follow the configuration depicted in Fig.3 in the main paper with a transition length of 60 and evaluate the phase mixing parameters r_i for the inbetweening segment \mathbf{X}_i , as well as r_{t_1}, r_{t_2} for the transitioning segments $\mathbf{X}_{t_1}, \mathbf{X}_{t_2}$, respectively. Although SPDM was not initially

Table 7: Hyperparameter tuning result for the comparison models within Compositional Motion Pair Generation task

TEACH (<i>slerp_ws</i>)	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
4 frames	0.962	2.359	1.011	3.168	7.471	4.798
8 frames	0.941	2.375	1.041	3.185	7.479	4.821
16 frames	0.952	2.356	1.118	3.231	7.477	4.859
24 frames	1.036	2.470	1.279	3.347	7.567	4.936
32 frames	1.196	2.697	1.437	3.481	7.590	5.012
45 frames	1.825	2.913	1.890	3.726	7.598	5.153
priorMDM (<i>handshake_len</i>)	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
10 frames	1.097	2.708	0.806	3.736	7.408	4.846
20 frames	1.118	2.928	0.825	3.785	7.369	4.896
30 frames	1.148	2.961	0.839	3.732	7.399	5.025
MDM (<i>overlap_len</i>)	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
15 frames	0.795	2.288	0.935	3.527	6.390	4.867
30 frames	1.084	2.526	1.146	3.793	6.429	4.923
45 frames	1.100	2.582	1.251	4.034	6.504	5.123
MLD (<i>overlap_len</i>)	Smt. FID↓	Trn. FID↓	Overall FID↓	Smt. MMD↓	Trn. MMD↓	Overall MMD↓
15 frames	12.88	15.47	13.04	8.426	7.389	7.625
30 frames	13.88	15.20	14.25	8.478	7.407	7.632
45 frames	16.34	18.93	16.40	8.457	7.394	7.641
Ours	0.736	1.807	0.782	3.509	6.545	4.711

applied to transitioning segments, we incorporate an instance with an empty string ” as input for each transitioning segment during the phase mixing parameter ablation. The results of these experiments are summarized in Tab. 6, indicating that setting all parameters to 1 achieves the best performance. This finding suggests that eliminating any semantic influence, including that from the empty string ”, and concentrating solely on phase continuity during the transition is optimal for the UMIB application. Note that after we evaluate the first set of phase mixing parameter choices in $\{0, 0.5, 1, \frac{k}{K}, 1 - \frac{k}{K}\}$, the result of $1 - \frac{k}{K}$ is the second best. To explore this, we extended our experiments to include two additional choices $(1 - \frac{k}{K})^3, 1 - (\frac{k}{K})^3$ in between $1 - \frac{k}{K}$ and 1. However, these alternatives did not surpass the performance of the parameter set to 1. Additionally, we evaluated the phase mixing setting for transitioning motion r_t in the compositional motion pair generation experiment and found that setting $r_t = 1$ also yielded the best results in this context.

3.6 Hyperparameters on comparison models in compositional motion pair generation task

While evaluating on compositional motion pair generation task, the performance of TEACH and priorMDM are slightly influenced by their key hyperparameter setting. While our primary comparisons use their proposed settings (e.g. handshake length of 30 and slerp window of 8), we also assess their performance under different settings. Additionally, we investigate the impact of varying the overlap region length when linearly blending motion outputs from single text-conditioned motion generation models, such as MDM and MLD. The results are shown in Tab. 7, indicating that our proposed method still outperforms in *Overall FID* and *Overall MM-Dist* by a margin. On the other hand, TEACH outperforms on *Smt. MM-Dist* and MDM outperforms on *Trn. MM-Dist*, even with the adjustments made to their hyperparameters.

4 Additional Visualization Results

Additional visualization results for compositional motion pair generation, UMIB, and CMIB are presented in Fig. 2, Fig. 4, and Fig. 5, respectively.

In the compositional motion pair generation results, the top example illustrates our model’s ability to transition smoothly from *step up* to *step down*, maintaining consistent upward and leg-stepping movements. Also, Transitions from *sit down* to *stand up* and *stand up* to *sit down* demonstrate strong consistency between root trajectory and pose sequence, avoiding the root orientation artifacts seen in TEACH and priorMDM. The bottom example highlights our method’s capability to generate motion

according to the text condition, unlike priorMDM and TEACH, which fail to repeatedly open and close the arm.

In the UMIB task, priorMDM exhibits hyperactivity by producing rapid movements to align with the succeeding motion, while RSMT struggles to create transitional motion that connects smoothly to the succeeding motion. For the CMIB task, priorMDM generates inbetweening motion that misaligns with the input text condition, while our method produces inbetweening motion that satisfies both the input text condition and ensures a smooth transition to adjacent segments.

5 Discussions

5.1 Extended Applications

The average time for generating a compositional motion pair is approximately 2 seconds. Leveraging our simultaneous denoising design within the denoising pipeline, even very long motion sequences composed of multiple clips can be denoised in about 2 seconds by stacking these clips in the batch dimension, provided the batch size is manageable within GPU memory constraints. This inference time is within the acceptable range for interactive motion synthesis and editing tools, enabling animation artists to create extended compositional motion sequences with seamless transitions, thereby enhancing and accelerating the workflow for producing high-quality animations.

Furthermore, the phase latent space constructed by ACT-PAE introduces a novel approach to time series modeling. By reparameterizing the phase parameters \mathbf{F} , \mathbf{A} , \mathbf{B} , \mathbf{S} into frame-level tokens \mathbf{Q} , ACT-PAE can integrate motion length and progress information into the latent code denoising process, offering a significant advantage over the VAE encoded latent space in MLD. This reparameterization of \mathbf{Q} also regularizes the generated motion to be smoother, resulting in significantly lower *RMS-Jerk* compared to existing motion generation models. The ACT-PAE latent space offers a promising direction for time-series modeling, particularly when smoothness is a critical factor.

5.2 Limitations

Our ACT-PAE has slightly reduced model expressiveness compared to VAE due to the phase parameter reparameterization in Equation 1, which introduces smoothness and periodic regularization while compromising expressiveness, favoring low-frequency motion content and avoiding high-frequency details. This regularization helps reduce overfitting and results in inherently smooth motions, as shown by motion visualization and low *RMS-Jerk* in UMIB evaluations. However, this increased smoothness may reduce the model’s expressiveness, making it more difficult to capture subtle motion patterns, such as minor movements related to body balance or object contact, which contribute minimally to the reconstruction loss during ACT-PAE training. To address these limitations, a residual network could be explored to enhance ACT-PAE’s output by adding fine-grained details, complementing the focus on low-frequency motion content in our Compositional Phase Diffusion pipeline.

Beyond expressiveness issues, our method struggles with motion inbetweening in short segments. Similar to other PAE-based models, our approach requires multiple frames to assess the motion phase conditions in the input segments. Fortunately, our method can handle variable lengths, allowing phase signal extraction from input motion segments with as few as three motion frames. Future work is necessary to enhance accuracy and stability when processing short motion segments.

Moreover, as mentioned in the **Remark on Dataset** in Sec. 4 of the main paper, our framework relies on data samples formatted as subsequence pairs to effectively learn transitions between sequences. This challenge restricts the applications of all current long-term motion generation models. To broaden their potential applications, it is essential to create datasets of subsequence motion pairs labeled in different modalities, such as breaking down long dance sequences into pairs using bar information from musical scores.

Finally, our model requires the input prompt to be formatted as a sequence of K instructions, each with a specified length. To adapt our model for handling free-form long text input, natural language processing tools such as ChatGPT can be employed to analyze and decompose lengthy text prompts into semantic conditions for each subsequence.

References

- [1] 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.
- [2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [3] 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.
- [4] 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.
- [5] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. Teach: Temporal action composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pages 414–423. IEEE, 2022.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] Jihoon Kim, Taehyun Byun, Seungyoun Shin, Jungdam Won, and Sungjoon Choi. Conditional motion in-betweening. *Pattern Recognition*, 132:108894, 2022.
- [16] 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.

- [17] 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.
- [18] 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.
- [19] 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.

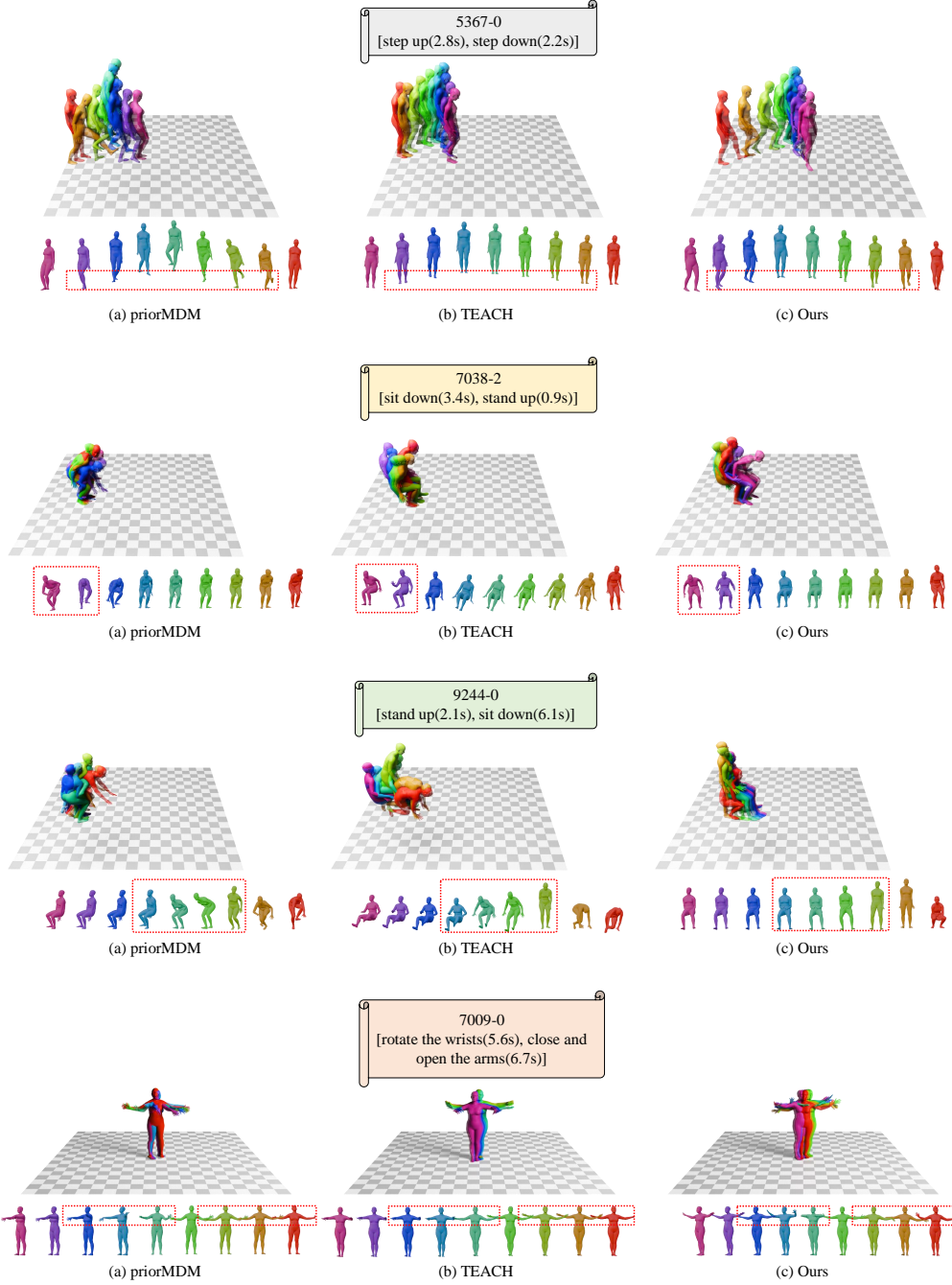


Figure 2: Visualization of Compositional Motion Generation result. Motion frames are coloured from red to purple in rainbow order to show time progression.

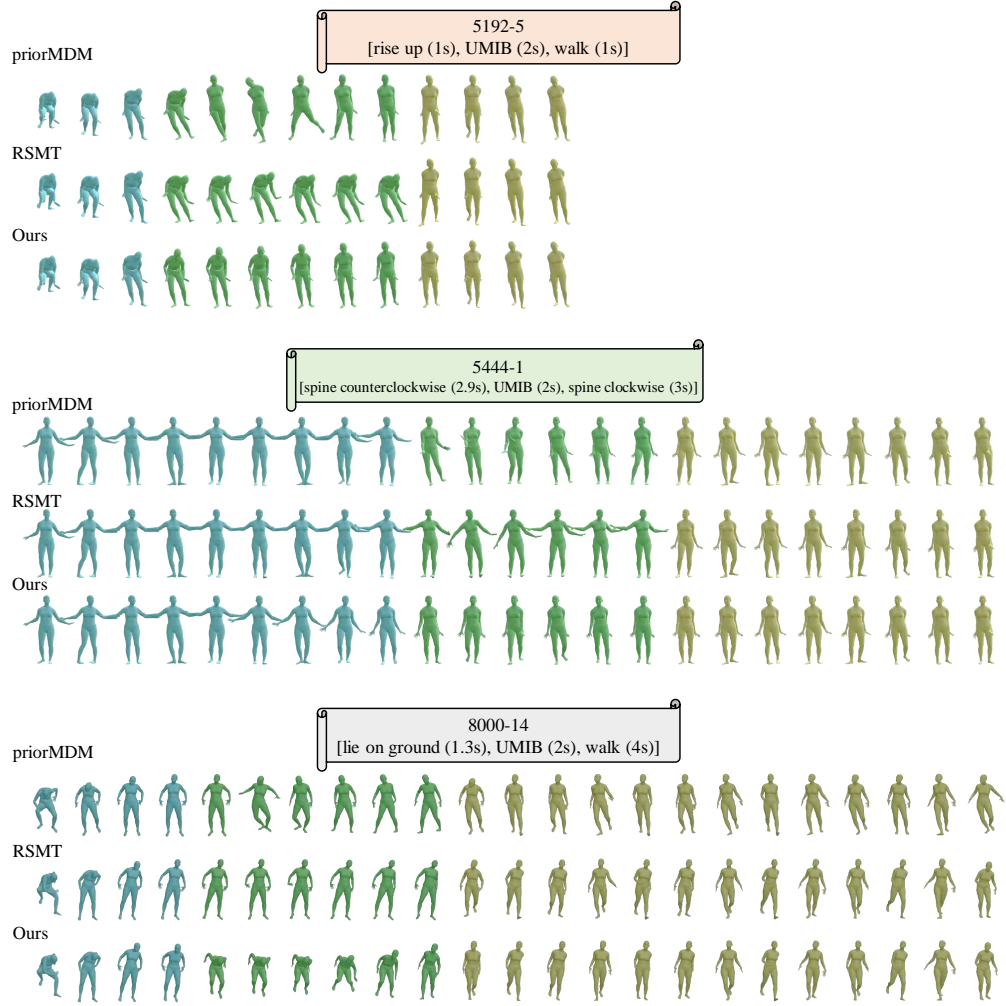


Figure 3: Visualization of UMIB results with 60 transition frames. The preceding motion is depicted in blue, the generated transitioning motion in green, and the succeeding motion in yellow.

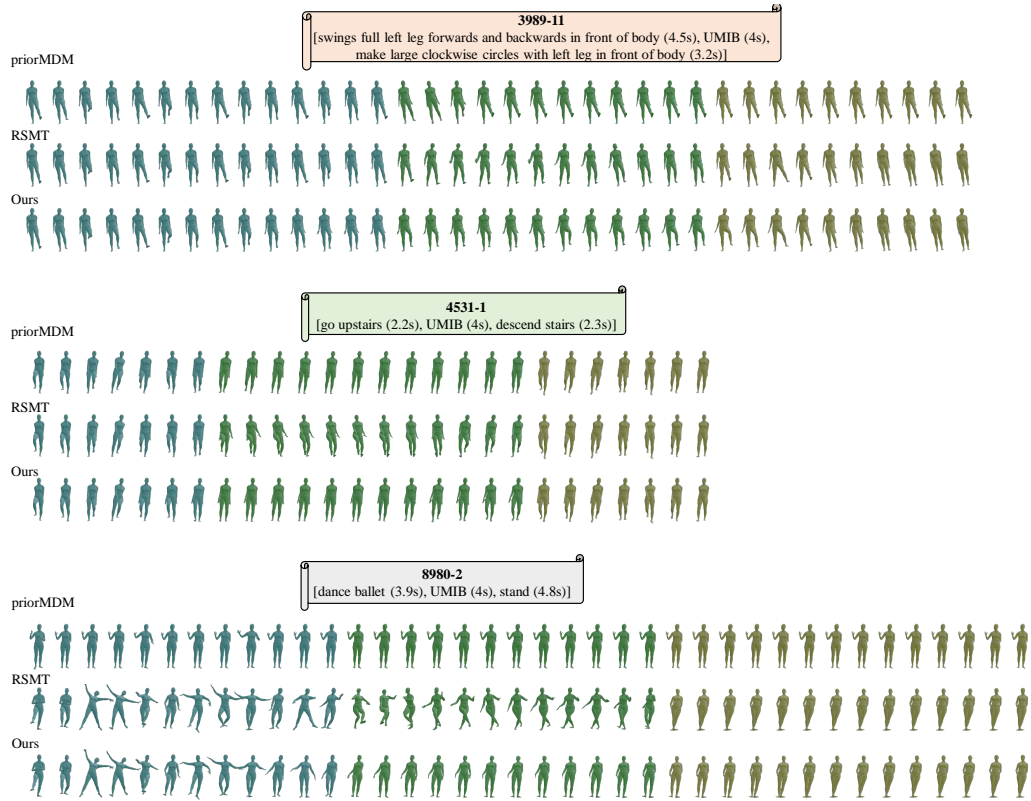


Figure 4: Visualization of UMIB results with 120 transition frames. The preceding motion is depicted in blue, the generated transitioning motion in green, and the succeeding motion in yellow.

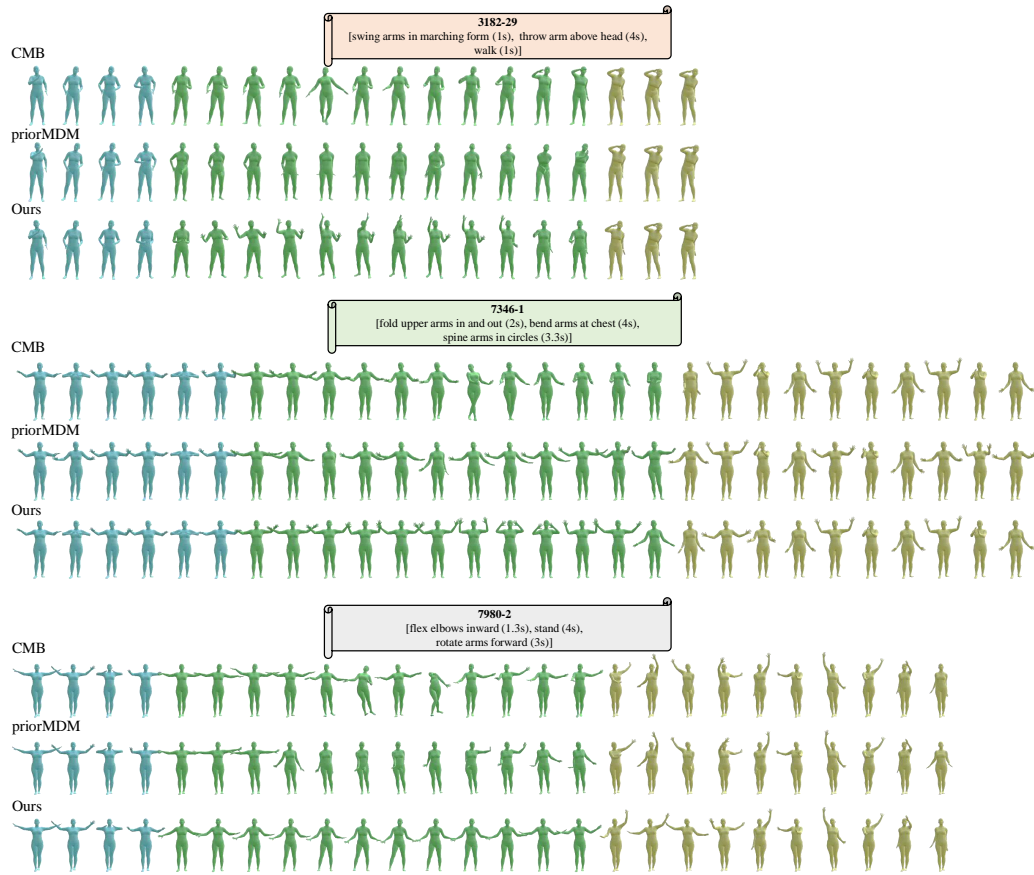


Figure 5: Visualization of CMIB result with 120 transition frames. The preceding motion is depicted in blue, the generated transitioning motion in green, and the succeeding motion in yellow.