

DART: A DIFFUSION-BASED AUTOREGRESSIVE MOTION MODEL FOR REAL-TIME TEXT-DRIVEN MOTION CONTROL

Anonymous authors

Paper under double-blind review

ABSTRACT

Text-conditioned human motion generation, which allows for user interaction through natural language, has become increasingly popular. Existing methods typically generate short, isolated motions based on a single input sentence. However, human motions are continuous and can extend over long periods, carrying rich semantics. Creating long, complex motions that precisely respond to streams of text descriptions, particularly in an online and real-time setting, remains a significant challenge. Furthermore, incorporating spatial constraints into text-conditioned motion generation presents additional challenges, as it requires aligning the motion semantics specified by text descriptions with geometric information, such as goal locations and 3D scene geometry. To address these limitations, we propose **DART**, a **D**iffusion-based **A**utoregressive motion primitive model for **R**eal-time **T**ext-driven motion control. Our model, DART, effectively learns a compact motion primitive space jointly conditioned on motion history and text inputs using latent diffusion models. By autoregressively generating motion primitives based on the preceding history and current text input, DART enables real-time, sequential motion generation driven by natural language descriptions. Additionally, the learned motion primitive space allows for precise spatial motion control, which we formulate either as a latent noise optimization problem or as a Markov decision process addressed through reinforcement learning. We present effective algorithms for both approaches, demonstrating our model’s versatility and superior performance in various motion synthesis tasks. Experiments show our method outperforms existing baselines in motion realism, efficiency, and controllability.

1 INTRODUCTION

Text-conditioned human motion generation has gained increasing popularity in recent years for flexible user interaction via natural languages. Existing text-conditioned motion models (Tevet et al., 2023; Guo et al., 2024; Zhang et al., 2023a; Guo et al., 2022; Jiang et al., 2024a) primarily focus on generating standalone short motions from a single descriptive sentence, thereby overlooking the inherently sequential nature of human motion. These methods fail to generate long and complex motions consisting of multiple segments with distinct conditions of action descriptions. FlowMDM (Barquero et al., 2024) is the state-of-the-art temporal motion composition method, capable of generating complex, continuous motions while providing control over the desired actions and their durations. However, FlowMDM is an offline method that requires prior knowledge of the entire action timeline and has a slow generation speed, making it unsuitable for online and real-time applications.

In addition to text-based semantic control, generating human motion within spatial constraints and achieving specific goals—such as reaching a keyframe body pose, following the joint trajectory, or interacting with objects—offers broad applications but introduces additional complex challenges. Recent works (Shafir et al., 2024; Karunratanakul et al., 2024b; Xie et al., 2024) have sought to integrate text-conditioned motion models with spatial control capabilities. However, they still face challenges in simultaneously harmonizing spatial control, motion quality, and semantic alignment with text. Moreover, these approaches are typically restricted to controlling isolated short motions in an offline setting. Conversely, interactive character control (Kovar et al., 2008; Holden et al., 2015; Ling et al., 2020; Peng et al., 2022) has been a longstanding focus in computer graphics, with

054 a primary emphasis on achieving motion realism and responsiveness to interactive control signals.
055 However, most of these works lack support for text-conditioned semantic control and are limited by
056 being trained on small, curated datasets. Incorporating text-conditioned motion generation could
057 provide a novel, efficient language interface for animators and everyday users to control the characters,
058 reducing the effort required when specifying spatial control signals is challenging or tedious.

059 To address these limitations, we propose **DART**, a diffusion-based autoregressive motion primitive
060 model for real-time text-conditioned motion composition. Moreover, the compact and expressive
061 motion space of DART provides a foundation for integrating precise spatial control through latent
062 space optimization or reinforcement learning-based algorithms. DART features three key components.

063 First, DART represents human motion as a collection of motion primitives (Zhang & Tang, 2022),
064 which are autoregressive representations consisting of overlapping short motion segments tailored for
065 online generation and control. These short primitives also provide a clearer alignment with atomic
066 action semantics compared to longer sequences, enabling effective learning of a text-conditioned
067 motion space. By focusing on shorter primitives, DART avoids the complexities and extensive data
068 demands of modeling entire motion sequences, allowing for high-quality motion generation with
069 only a few diffusion steps.

070 Next, DART learns a text-conditioned motion primitive space from large-scale data using a latent
071 diffusion architecture, which contains a denoiser network for generating motions in a compact latent
072 motion space, and a decoder network for reconstructing the detailed motions. Leveraging this learned
073 latent motion space, DART employs an autoregressive rollout to synthesize motion sequences from
074 real-time text inputs, enabling the efficient generation of motions of arbitrary length. Compared with
075 the offline temporal motion composition method FlowMDM (Barquero et al., 2024), DART provides
076 real-time response and ten times faster generation speed.

077 Lastly, the versatile and powerful motion space learned in DART enables precise spatial control
078 through latent space optimization or RL-based policy learning. We present effective learning and
079 optimization algorithms that explore the learned DART motion space to obtain motion sequences that
080 precisely follow both textual and spatial constraints. We evaluate DART across various motion synthesis
081 tasks, including generating long, continuous sequences from sequential text prompts, in-between
082 motion generation, scene-conditioned motion, and goal-reaching synthesis. The experimental results
083 show that DART is a simple, unified and highly effective motion model, consistently outperforming
084 or matching the performance of baselines.

086 2 RELATED WORKS

089 **Conditional Motion Generation.** Generating realistic and diverse human motions is a long-standing
090 challenge in computer vision and graphics. Apart from generating highly realistic human motions
091 (Kovar et al., 2008; Holden et al., 2020; Clavet et al., 2016; Zinno, 2019), conditional generation is
092 another important factor that aligns motion generation with human intentions and various application
093 constraints. Text-conditioned motion generation (Tevet et al., 2023; Zhang et al., 2022; Petrovich
094 et al., 2022; Guo et al., 2022; Jiang et al., 2024a; Zhang et al., 2023a) has become increasingly
095 popular since it allows users to modulate motion generation with flexible natural languages. Audio
096 and speech-driven motion synthesis methods (Alexanderson et al., 2023; Tseng et al., 2023; Siyao
097 et al., 2022; Ao et al., 2022; 2023) have also made significant progress recently. Moreover, there
098 exist many applications that require spatial awareness and precise control in motion generation, such
099 as interactive character control (Ling et al., 2020; Peng et al., 2022; Starke et al., 2022; Luo et al.,
100 2024; Starke et al., 2024), human-scene interactions (Hassan et al., 2021; Starke et al., 2019; Zhao
101 et al., 2022; 2023; Li et al., 2024a; Xu et al., 2023; Jiang et al., 2024b; Wang et al., 2024; Liu et al.,
102 2024; Li et al., 2024b), and human-human(noid) interactions (Liang et al., 2024; Zhang et al., 2023c;
103 Christen et al., 2023; Cheng et al., 2024; Shan et al., 2024). Synthesizing high-quality motions with
104 precise spatial control remains challenging, and DART is a step toward a general and efficient motion
105 model that supports precise control tasks.

106 **Diffusion Generative Models.** Denoising Diffusion Models (Ho et al., 2020; Song et al., 2021a;b)
107 are generative models that learn to predict clean data samples by gradually annealing the noise from
a standard Gaussian sample. Diffusion models have achieved unprecedented performances in many
generation tasks including images, videos, and 3D human motions (Tevet et al., 2023; Rombach et al.,

2022; Ho et al., 2022). Diffusion models can accept flexible conditions to modulate the generation, such as text prompts, images, audio, and 3D objects (Rombach et al., 2022; Tevet et al., 2023; Alexanderson et al., 2023; Tseng et al., 2023; Zhang et al., 2023b; Xu et al., 2023; Li et al., 2023). Most existing diffusion-based motion generation methods focus on offline generations (Tevet et al., 2023; Barquero et al., 2024; Chen et al., 2023b; Karunratanakul et al., 2024b; Cohan et al., 2024; Dai et al., 2025; Chen et al., 2023a) producing short, isolated motion sequences at once while neglecting the autoregressive nature of human motions. Among these methods, DNO (Karunratanakul et al., 2024b) is closely related to our optimization-based control approach, as both use diffusion noises as the latent space for editing and control. However, the key distinction lies in our latent motion primitive-based diffusion model, which, unlike DNO’s diffusion model trained with full motion sequences in raw space, achieves superior performance in harmonizing spatial control with text semantic alignment during experiments. There are also works that incorporate history conditions in diffusion-based motion prediction. Recent works (Xu et al., 2023; Jiang et al., 2024b) use history-conditioned diffusion models to autoregressively generate human-scene interaction sequences. Van Wouwe et al. (2024) uses autoregressive diffusion for real-time human motion reconstruction from IMU sensors. Han et al. (2024) uses history-conditioned diffusion to generate long motions from complex text prompts. Shi et al. (2024) and Chen et al. (2024) adapt diffusion models for real-time character motion generation and control. While Shi et al. (2024) learns character control policies using diffusion noises as the action space, akin to our reinforcement-learning-based control, their method focuses on single-frame autoregressive generation and lacks support for text conditions, which offers a compact and powerful interface for users to control character behaviors. In contrast, DART is an efficient and general motion model that scales effectively to large motion datasets with text annotations (Mahmood et al., 2019; Punnakkal et al., 2021). DART supports natural language interfacing and provides a versatile foundation for various motion generation tasks with spatial control.

3 METHOD

3.1 PRELIMINARIES

Problem Definition. We focus on the task of text-conditioned online motion generation with spatial control. Given an H frame seed motion $\mathbf{H}_{seed} = [\mathbf{h}^1, \dots, \mathbf{h}^H]$, a sequence of N text prompts $C = [c^1, \dots, c^N]$, and spatial goals g , the objective is to autoregressively generate continuous and realistic human motion sequences $\mathbf{M} = [\mathbf{H}_{seed}, \mathbf{X}^1, \dots, \mathbf{X}^N]$, where each motion segment \mathbf{X}^i matches the semantics of the corresponding text prompt c^i and satisfies the spatial goal constraints g . This task imposes challenges in high-level action semantic control, precise spatial control, and smooth temporal transition in motion generation.

Autoregressive Motion Primitive Representation. We model long-term human motions as the sequential composition of motion primitives (Zhang & Tang, 2022) with overlaps for efficient generative learning and online inference. In contrast to directly modeling long motion sequences, the primitive representation decomposes globally complex sequences into short and simple primitives, resulting in a more tractable data distribution for generative learning. This enables DART to learn high-quality primitive generation using as few as 10 diffusion steps. Moreover, the autoregressive and simple nature of the primitive representation makes it inherently suitable for fast online generations. Furthermore, motion primitives convey more interpretable semantics than individual frames, enhancing the learning of text-conditioned motion space.

Each motion primitive $\mathbf{P}^i = [\mathbf{H}^i, \mathbf{X}^i]$ is a short motion clip containing H frames of history motion $\mathbf{H}^i = [\mathbf{h}^{i,1}, \dots, \mathbf{h}^{i,H}]$ that overlap with the previous motion primitive, and F frames of future motion $\mathbf{X}^i = [\mathbf{x}^{i,1}, \dots, \mathbf{x}^{i,F}]$. The history motion of the i -th motion primitive \mathbf{H}^i consists of the last H frames of the previous motion primitive $\mathbf{X}^{i-1, F-H+1:F}$. Therefore, infinitely long motions can be represented as the rollout of such overlapping motion primitives as $\mathbf{M} = [\mathbf{H}_{seed}, \mathbf{X}^1, \dots, \mathbf{X}^N]$.

We represent each motion frame using an overparameterized representation based on the SMPL-X (Pavlakos et al., 2019) parametric human body model. The overparameterized representation is a $D = 276$ dimensional vector including the body root translation \mathbf{t} , root orientation \mathbf{R} , local joint rotations θ , joint locations \mathbf{J} , and the temporal difference features of locations and rotations. All the frames within one motion primitive are canonicalized in a local coordinate frame centered at the first-frame body of the primitive. We use history length $H = 2$ and future length $F = 8$ in our experiments. Further details of the primitive representation are attached in the Appendix A.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215

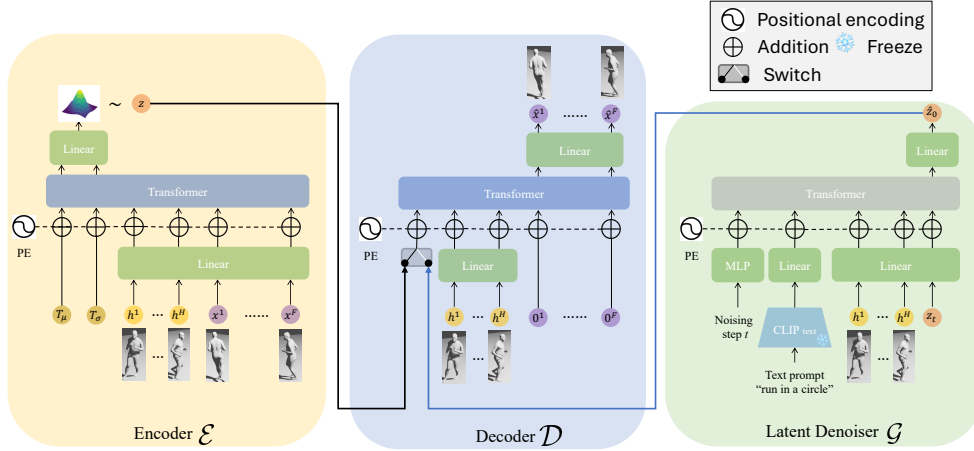


Figure 1: Architecture illustration of DART. The encoder network compresses the future frames $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^F]$ into a latent variable, conditioned on the history frames $\mathbf{H} = [\mathbf{h}^1, \dots, \mathbf{h}^H]$. The decoder network reconstructs the future frames conditioned on the history frames and latent sample z . The denoiser network predicts the clean latent sample \hat{z}_0 conditioned on the noising step, text prompt, history frames, and noised latent sample z_t . During the denoiser training, both the encoder and decoder networks remain frozen.

3.2 DART: A DIFFUSION-BASED AUTOREGRESSIVE MOTION PRIMITIVE MODEL

We propose a novel latent diffusion model (Rombach et al., 2022; Chen et al., 2023b) designed for motion primitives. This model can autoregressively generate motions with smooth transitions, conditioned on text prompts and motion history. The proposed model contains a variational autoencoder (VAE) (Kingma & Welling, 2014) that compresses the motion primitives into a compact latent space, and a latent denoising diffusion model that predicts clean latent variables from noises, conditioned on text prompts and motion history.

Latent Motion Primitive Space. We introduce a motion primitive VAE that compresses motion primitives into a compact latent space, upon which we train our latent diffusion models, rather than using the raw motion space. **This latent diffusion architecture is proved effective for text-conditioned generation of images and motion sequences in prior works LatentDiffusion and MLD** (Rombach et al., 2022; Chen et al., 2023b). The design of learning a compressed latent space of motion primitives is inspired by the observations that raw motion data from the used motion capture dataset AMASS (Mahmood et al., 2019) often contain **various levels of artifacts including glitches and jitters**, and training diffusion models on raw motion space leads to results inheriting such artifacts. This is evidenced by the significantly higher jittering in generated motions of the ablative model without VAE in Appendix. E.3. The compression achieved through the motion primitive VAE significantly mitigates the impacts of **these outlier artifacts in motion data**. The resulting latent representation is not only more compact but also more computationally efficient than the raw motion data, thereby enhancing the efficiency of our generative model and improving the control capabilities within the latent space.

Our motion primitive VAE employs a transformer-based architecture based on **MLD** (Chen et al., 2023b), comprising an encoder \mathcal{E} and decoder \mathcal{D} , as shown in Fig. 1. The encoder takes as input the history motion frames \mathbf{H} and future motion frames \mathbf{X} as well as the learnable distribution tokens T_μ and T_σ , which are responsible for predicting the latent distribution. The latent sample z is obtained from the predicted distribution via reparameterization (Kingma & Welling, 2014). The decoder then predicts the future frames $\hat{\mathbf{X}}$ from zero tokens conditioned on the latent sample z and the history frames \mathbf{H} . The motion primitive VAE is trained with the future frame reconstruction loss L_{rec} , auxiliary losses L_{aux} that penalize unnatural motion reconstruction, and a small Kullback-Leibler (KL) regularization term L_{KL} . We refer to Appendix C for further details about the motion primitive VAE.

Algorithm 1 Autoregressive rollout generation using latent motion primitive model

```

216 Input: primitive decoder  $\mathcal{D}$ , latent variable denoiser  $\mathcal{G}$ , history motion seed  $\mathbf{H}_{seed}$ , text prompts
217  $C = [c^1, \dots, c^N]$ , total diffusion steps  $T$ , classifier-free guidance scale  $w$ , diffusion sampler  $\mathcal{S}$ .
218 Optional Input: Latent noises  $Z_T = [z_T^1, \dots, z_T^N]$ 
219 Output: motion sequence  $\mathbf{M}$ 
220  $\mathbf{H} \leftarrow \mathbf{H}_{seed}$ 
221  $\mathbf{M} \leftarrow \mathbf{H}_{seed}$ 
222 for  $i \leftarrow 1$  to  $N$  do ▷ number of rollouts
223   sample noise  $z_T^i$  from  $\mathcal{N}(0, 1)$  if not inputted
224    $\hat{z}_0^i \leftarrow \mathcal{S}(\mathcal{G}, z_T^i, T, \mathbf{H}, c^i, w)$  ▷ diffusion sample loop with classifier-free guidance
225    $\hat{\mathbf{X}} \leftarrow \mathcal{D}(\mathbf{H}, \hat{z}_0^i)$ 
226    $\mathbf{M} \leftarrow \text{CONCAT}(\mathbf{M}, \hat{\mathbf{X}})$  ▷ concatenate future frames to generated sequence
227    $\mathbf{H} \leftarrow \text{CANONICALIZE}(\hat{\mathbf{X}}^{F-H+1:F})$  ▷ update the rollout history with last H generated frames
228 end for
229 return  $\mathbf{M}$ 

```

Latent Motion Primitive Diffusion. Building on the motion representation provided by our trained motion primitive encoder \mathcal{E} , we transform the motion primitives from the raw feature space to a more refined latent space that is more suitable for generative learning with diffusion models. This latent motion primitive space is then used to train a diffusion denoiser for generating motion primitives conditioned on text prompts. The denoiser architecture is shown in Fig. 1. Given the noised latent variable z_t , the diffusion step t , the history motion \mathbf{H} , and a text prompt c , we train a transformer-based latent denoiser model \mathcal{G} to predict the cleaned motion primitive latent variable $\hat{z}_0 = \mathcal{G}(z_t, t, \mathbf{H}, c)$. The diffusion step is embedded using a small MLP, while the text prompt is encoded using the CLIP (Radford et al., 2021) text encoder. The text prompt is randomly masked out by a probability of 0.1 during training to enable classifier-free guidance (Ho & Salimans, 2021) during generation. During training, the motion primitive latent variable $z_0 = \mathcal{E}(\mathbf{H}, \mathbf{X})$ can be efficiently obtained with a single inference of the fixed encoder \mathcal{E} . The cleaned latent variable \hat{z}_0 can be converted back to the future frames using the fixed decoder \mathcal{D} : $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{H}, \hat{z}_0)$. The latent denoiser \mathcal{G} is trained with the simple objective (Ho et al., 2020). Additionally, we use the reconstruction loss L_{rec} and auxiliary losses L_{aux} on the decoded future frames $\hat{\mathbf{X}}$, the same as training the motion primitive VAE. Notably, we use DDPM (Ho et al., 2020) with only 10 diffusion steps during both the training and inference of the denoiser model. This is sufficient for generating realistic samples due to the simplicity of the short motion primitives and enables highly efficient online generation. Moreover, we use the scheduled training (Ling et al., 2020; Bengio et al., 2015; Rempe et al., 2021) to progressively introduce the test-time distribution of the history motion \mathbf{H} , which improves the stability of long sequence online generation and the text prompt controllability for unseen poses. We refer to Appendix D for the details.

With the trained motion primitive decoder \mathcal{D} , latent denoiser \mathcal{G} and a diffusion sampler \mathcal{S} such as DDPM and DDIM (Ho et al., 2020; Song et al., 2021a), we can autoregressively generate motion sequences given the history motion seed \mathbf{H}_{seed} and the online sequence of text prompts C , as shown in Alg. 1. During sampling, we use classifier-free guidance (Ho & Salimans, 2021) on the text condition with a guidance scale w :

$$\mathcal{G}_w(z_t, t, \mathbf{H}, c) = \mathcal{G}(z_t, t, \mathbf{H}, \emptyset) + w \cdot (\mathcal{G}(z_t, t, \mathbf{H}, c) - \mathcal{G}(z_t, t, \mathbf{H}, \emptyset)) \quad (1)$$

Using the rollout algorithm, DART generates over 300 frames per second using a single RTX 4090 GPU, enabling real-time applications and online reinforcement-learning control as in Sec. 3.3.

3.3 CONTROLLABLE MOTION SYNTHESIS VIA DART

Text-conditioned motion generation offers a user-friendly interface for controlling motions through natural language. However, relying solely on text limits precise spatial control, such as walking to a specific location or sitting in a designated spot. Therefore, it is necessary to incorporate motion control mechanisms to achieve precise spatial goals, including reaching a keyframe body pose, following joint trajectories, and interacting with scene objects.

Algorithm 2 Latent noises optimization

```

270 Input: Latent noises  $Z_T = [z_T^1, \dots, z_T^N]$ , Optimizer  $\mathcal{O}$ , learning rate  $\eta$ , criterion  $\mathcal{L}$ , goal  $g$  (for
271 brevity, we do not reiterate the inputs of the rollout function defined in Alg. 1)
272 Output: motion sequence  $\mathbf{M}$ 
273 for  $i \leftarrow 1$  to optimization steps do
274    $\mathbf{M} \leftarrow \text{ROLLOUT}(Z_T, \dots)$ 
275    $\nabla \leftarrow \nabla_{Z_T} \mathcal{L}(\mathbf{M})$ 
276    $Z_T \leftarrow \mathcal{O}(Z_T, \nabla / \|\nabla\|, \eta)$  ▷ update using normalized gradient
277 end for
278 return  $\mathbf{M} \leftarrow \text{ROLLOUT}(Z_T)$ 

```

We formulate the motion control task as generating the motion sequence \mathbf{M} that minimizes its distance to a given spatial goal g under a criterion function $\mathcal{F}(\cdot, \cdot)$ and the regularization from the scene and physical constraints $\text{cons}(\cdot)$:

$$\mathbf{M}^* = \operatorname{argmin}_{\mathbf{M}} \mathcal{F}(\Pi(\mathbf{M}), g) + \text{cons}(\mathbf{M}) \quad (2)$$

where g is the task-dependent spatial goal, e.g., g is a keyframe body for the motion in-between task or the goal location for the navigation task. $\Pi(\cdot)$ is the goal-dependent projection function that maps motion sequence to the goal observation space, $\text{cons}(\cdot)$ denotes physical constraints such as avoiding collisions into scenes.

Directly solving the motion control task in the raw motion space often results in unrealistic motions since most samples in the raw motion space do not represent plausible motions. To improve the generated motion quality, many previous methods tackle such motion control tasks in a latent motion space, where samples can be mapped to plausible motions (Karunratanakul et al., 2024b; Ling et al., 2020; Peng et al., 2022; Holden et al., 2015). DART offers a powerful text-conditioned latent motion space for such latent space control. DART learns a powerful generative model that can generate diverse and realistic motions from standard Gaussian samples Z_T . Using the deterministic DDIM (Song et al., 2021a) sampler, we adapt DART sampling to function as a deterministic mapping from latent noises Z_T to plausible motions. Using the latent noises Z_T as the latent space, we transform the motion control task in Eq. 2 to latent space control as follows: Given the initial motion history \mathbf{H}_{seed} , a sequence of text prompts C , and a deterministic diffusion sampler \mathcal{S} , the rollout function in Alg. 1 can deterministically map a list of motion primitive latent noises $Z_T = [z_T^1, \dots, z_T^N]$ to a motion sequence \mathbf{M} by :

$$\mathbf{M} = \text{ROLLOUT}(Z_T, \mathcal{D}, \mathcal{G}, \mathbf{H}_{seed}, C, T, w, \mathcal{S}) \quad (3)$$

We abbreviate it as $\text{ROLLOUT}(Z_T, \dots)$ for short since Z_T is the only variable in latent control. The minimization objective is converted as :

$$Z_T^* = \operatorname{argmin}_{Z_T} \mathcal{F}(\Pi(\text{ROLLOUT}(Z_T, \dots)), g) + \text{cons}(\text{ROLLOUT}(Z_T, \dots)) \quad (4)$$

Note that we do not use DDIM to skip diffusion steps at sampling, which we observe to cause artifacts in generated motion. We then propose two solutions to this latent space motion control problem, one is to directly optimize the latent noises using gradient descent, and the other is to model the control task as a Markov process and use reinforcement learning to learn control policies.

Motion Control via Latent Diffusion Noise Optimization. One straightforward solution to this minimization problem (Eq. 4) is to directly optimize the latent noises Z_T given the criterion function using gradient descent methods (Kingma & Ba, 2015; Karunratanakul et al., 2024a). The latent noise optimization is illustrated in Alg. 2. This optimization-based control framework is general and applicable for various spatial control tasks. We instantiate the latent noise optimization method in two example control scenarios: in-between motion generation and human-scene interaction generation.

First, we address the motion in-between task that aims to generate the motion frames transition between given history and goal keyframes g that is f frames away conditioned on the text prompt c . We use the distance between the f -th frame of the generated motion and the goal keyframe as

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

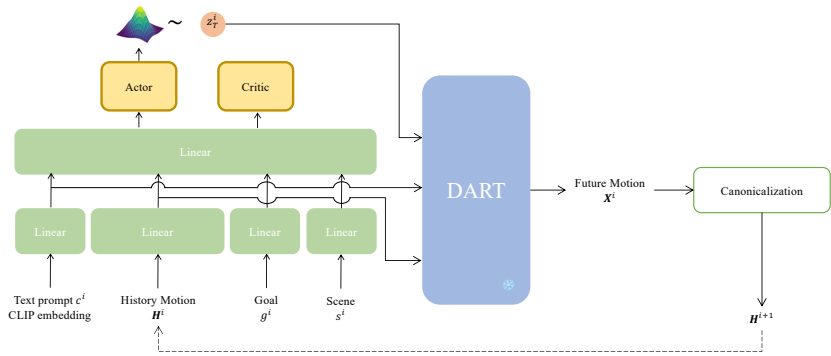


Figure 2: Architecture of the reinforcement learning-based control policy. The frozen DART diffusion denoiser and decoder models transform the latent actions into motion frames. The last predicted frames are canonicalized and provided to the policy model as the next step history condition.

the optimization objective. Second, we show that physical and scene constraints $cons(\cdot)$ can be incorporated to synthesize human motions in a contextual environment. Given **input 3D scenes**, text prompts C , and spatial goals g of the interaction anchor joint locations, e.g., locations of the pelvis when sitting, the objective is to generate motions that not only perform the desired interaction but also achieve the goal joint positions while adhering to scene constraints. During the optimization, **the 3D scenes are represented as signed distance fields (SDF) to compute body-scene distances, which serve as the basis for deriving human-scene contact and collision metrics that encourage foot-floor contact and scene collision avoidance**, as detailed in Appendix. F.

Motion Control via Reinforcement Learning. Although the proposed latent noise optimization is effective for general control tasks, the optimization can be computationally expensive. To address this, the autoregressive primitive-based motion representation of DART allows for another efficient control mechanism using reinforcement learning (RL) (Sutton & Barto, 1998). We model the latent motion control as a Markov decision process with a latent action space and use RL to learn policy models to achieve the goals. We model digital humans as agents to interact with an environment according to a policy to maximize the expected discounted return. At each time step i , the agent observes the state s^i of the system, samples an action a^i from the learned policy, with the system transitioning to the next state s^{i+1} due to the performed action a^i , and receives a reward $r^i = r(s^i, a^i, s^{i+1})$.

Our latent motion primitive model naturally fits in the Markov decision process due to its autoregressive nature. We use the latent noises z_T as the latent action space and train goal-conditioned policy models as controllers. The policy architecture is shown in Fig. 2. Our policy model employs the actor-critic (Sutton & Barto, 1998) architecture and is trained with the PPO (Schulman et al., 2017) algorithm. The state s^i includes the history motion observation H^i , goal observation g^i , scene observation s^i , and the CLIP embedding of the text prompt c^i . The policy model takes in $[H^i, g^i, s^i, c^i]$ to predict the latent noise z_T^i as the action. The latent noise z_T^i is mapped to the future motion frames X^i using the frozen latent denoiser \mathcal{G} and motion primitive decoder \mathcal{D} . The new history motion is extracted from the last H predicted frames and fed to the policy network in the next step. We reformulate the minimization problem in Eq. 4 as reward maximization to train the policy.

We instantiate the reinforcement learning control with the text-conditioned goal-reaching task. Given a text prompt c and a 2D goal location g , we aim to control the human to reach the goal location using the action specified by the text. The goal location is converted to a local observation consisting of its distance to the **last history frame** human pelvis, and its local direction in the human-centric coordinates frame. We consider a simple flat scene and the scene observation is the relative floor height to the **first history frame** human pelvis. The policy is trained with distance rewards encouraging the human pelvis to reach the goal location and scene constraint rewards penalizing foot skating and floor penetration. Further details can be found in appendix G. With the trained control policies, we can efficiently control a human to reach dynamic goals using specified skills like walking or hopping.

4 EXPERIMENTS

We provide extensive experiments showing how DART can serve as a general model for text-conditioned temporal motion composition (4.1) and various motion generation tasks requiring precise spatial control via latent noise optimization (4.2) and reinforcement learning policy (4.3). The qualitative results and comparisons are attached in the **supplementary video**.

Our DART is trained on motion-text data from the BABEL (Punnakkal et al., 2021) dataset in our experiments if not otherwise stated. BABEL contains motion capture sequences with frame-aligned text labels that annotate the fine-grained semantics of actions. Fine-grained text labels in BABEL allow models to learn precise human action controls and natural transitions among actions. However, DART can also learn using motions with coarse sequence-level labels such as the HML3D (Guo et al., 2022) dataset, as in the optimization-based motion in-between experiments in Sec. 4.2.

4.1 TEXT-CONDITIONED TEMPORAL MOTION COMPOSITION

Text-conditioned temporal motion composition aims to generate realistic motion sequences that faithfully align with a list of action segments, each defined by a specific text prompt and duration. We evaluate the motion composition task on the BABEL(Punnakkal et al., 2021) dataset consisting of motion capture sequences with human-annotated per-frame action descriptions, which facilitate the evaluation of precise action controls and natural transitions in motion composition. Since BABEL does not release the test set, we compare our DART with baseline methods on the BABEL validation set. We extract the list of action segments described by tuples of text prompts and durations from each data sequence and feed the action lists as conditions for motion composition.

We evaluate the generation results using metrics proposed in Guo et al. (2022) and Barquero et al. (2024). For each action segment, we evaluate the similarity between generation and dataset (FID), motion-text semantic alignment (R-prec, MM-DIST), and generation diversity (DIV). To evaluate smooth transitions between two segments, we measure the jerk (the derivative of acceleration) of the 30-frame transition clip centered at the splitting point of two action segments, reporting the peak jerk (PJ) and Area Under the Jerk (AUJ). Moreover, we profile all methods in a benchmark of generating one 5000-frame-long sequence and report the generation speed, the latency of getting the first generated frames, and memory usage. We also conduct binary user studies to evaluate the motion realism and motion-text semantic alignment of generation results, during which the participants are given generation results from two different methods and are asked to select the generation that is perceptually more realistic or better aligns with the action text stream in subtitles. We compare DART with baselines including TEACH (Athanasiou et al., 2022), DoubleTake (Shafir et al., 2024), a history-conditioned modification of T2M-GPT (Zhang et al., 2023a)(denoted as T2M-GPT*), and the state-of-the-art offline motion composition method FlowMDM (Barquero et al., 2024).

We present the quantitative results in Tab. 1 and Tab. 2. DART achieves the best FID in both the segment and transition evaluation, indicating the highest similarity to the dataset and best motion realism. DART also displays second-best jerk metrics indicating smooth action transitions. We observe that DART performs slightly worse than FlowMDM in motion-text semantic alignment (R-prec and MM-Dist) because of the online generation nature of DART. Natural action transitions require time to transit to the new action after receiving the new action prompt, leading to a delay in the emergence of the new action semantics. For instance, a human cannot immediately transition from kicking a leg in the air to stepping backward without first recovering to a standing pose. This inherent delay in transitions results in a motion embedding shift that impacts the R-prec metric of DART. In contrast, the offline baseline FlowMDM generates the entire sequence as a whole and requires oracle information of the full timeline of action segments to modulate compatibility between subsequent segments. The slight but natural action transition delay of DART is perceived as natural by humans, as shown in the user study results in Tab. 2 where DART is preferred over all the baselines, including FlowMDM, for both motion realism and motion-text semantic alignment in human evaluations.

DART requires significantly less memory than the offline baseline FlowMDM and achieves approximately 10x faster generation, with a frame rate exceeding 300 and a latency of 0.02s, enabling real-time text-conditioned motion composition (see supplementary video). We refer to Appendix E for details of experiments, user studies, and ablation studies about model architecture and hyperparameters, and refer to the supplementary video for qualitative comparisons.

Table 1: Quantitative evaluation results on text-conditioned temporal motion composition. The first row includes the metrics of the dataset for reference. Symbol ‘→’ denotes that closer to the dataset reference is better and ‘±’ indicates the 95% confidence interval. **Bold** and **blue** texts indicate the best and second best results excluding the dataset, respectively.

Dataset	Segment				Transition				Profiling		
	FID↓	R-prec↑	DIV →	MM-Dist↓	FID↓	DIV →	PJ→	AUJ ↓	Speed(frame/s)↑	Latency(s)↓	Mem.(MiB)↓
Dataset	0.00±0.00	0.72±0.00	8.42±0.15	3.36±0.00	0.00±0.00	6.20±0.06	0.02±0.00	0.00±0.00			
TEACH	17.58±0.04	0.66±0.00	10.02±0.06	5.86±0.00	3.89±0.05	5.44±0.07	1.39±0.01	5.86±0.02	3880±144	0.05±0.00	2251
DoubleTake	7.92±0.13	0.60±0.01	8.29±0.16	5.59±0.01	3.56±0.05	6.08±0.06	0.32±0.00	1.23±0.01	85±1	59.11±0.76	1474
T2M-GPT*	7.71±0.55	0.49±0.01	8.89±0.21	6.69±0.08	2.53±0.04	6.61±0.02	1.44±0.03	4.10±0.09	885±12	0.23±0.00	2172
FlowMDM	5.81±0.10	0.67±0.00	8.90±0.06	5.08±0.02	2.39±0.01	6.63±0.08	0.04±0.00	0.11±0.00	31±0	161.29±0.24	11892
Ours	3.79±0.06	0.62±0.01	8.05±0.10	5.27±0.01	1.86±0.05	6.70±0.03	0.06±0.00	0.21±0.00	334±2	0.02±0.00	2394

Table 2: Binary user study results comparing our method against baselines in generation realism and motion-text semantic alignment on text-conditioned temporal motion composition. We report the percentage of each method being voted better than the other (Ours vs. Baselines).

	Realism (%)	Semantic (%)
Ours vs. TEACH	66.7 vs. 33.3	66.0 vs. 34.0
Ours vs. DoubleTake	66.4 vs. 33.6	66.1 vs. 33.9
Ours vs. T2M-GPT*	61.3 vs. 38.7	66.7 vs. 33.3
Ours vs. FlowMDM	53.3 vs. 46.7	51.3 vs. 48.7

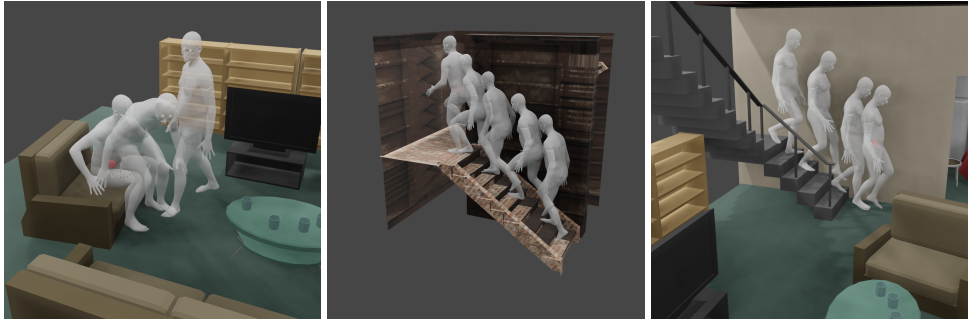
4.2 LATENT DIFFUSION NOISE OPTIMIZATION-BASED CONTROL USING DART

Text-conditioned motion in-between. Motion in-betweening aims to generate realistic motion frames that smoothly transition between a pair of history and goal keyframes. We consider a text-conditioned variant where an additional text prompt is inputted to specify the action semantics of the frames in between. We compare our method with DNO (Karunratanakul et al., 2024b) and OmniControl (Xie et al., 2024). For a fair comparison, we train DART on the HML3D dataset same as the baselines. We evaluate using test sequences covering diverse actions, with the sequence lengths ranging from 2 to 4 seconds. The quantitative evaluations are shown in Tab. 3. We report the L_2 norm errors between the generated motion and the history motion and goal keyframe. We also evaluate the motion realism with the skate and jerk metrics. The skate metric (Ling et al., 2020; Zhang et al., 2018) calculates a scaled foot skating when in contact with the floor: $s = disp \cdot (2 - 2^{h/0.03})$, where $disp$ is the foot displacement in two consecutive frames, h denotes the higher foot height in consecutive frames and 0.03m is the threshold value for contact. We do not calculate skate metric for sequences where the feet are not on a flat floor, such as crawling and climbing down stairs. Our method can generate the motions closest to the keyframe and show fewer skating and jerk artifacts. Our method effectively preserves the semantics specified by the text prompts, while the baseline DNO occasionally ignores the text prompts to reach the goal keyframe, as illustrated in the examples of pacing in circles and dancing in the supplementary video. This highlights the superior capability of our latent motion primitive-based DART in harmonizing spatial control and text semantic alignment.

Table 3: Quantitative evaluation of noise optimization-based In-between. The best results except dataset are in **bold** and ‘±’ indicates the 95% confidence interval.

	History error (cm)↓	Goal error (cm)↓	Skate (cm/s)↓	Jerk↓
Dataset	-	-	2.83 ± 0.00	0.99 ± 0.00
OmniControl	17.22 ± 0.80	5.88 ± 0.57	5.48 ± 0.90	1.26 ± 0.10
DNO	2.15 ± 0.24	5.52 ± 1.86	5.12 ± 0.55	0.72 ± 0.07
DART Opt.(ours)	0.00 ± 0.00	0.54 ± 0.03	3.97 ± 0.96	0.71 ± 0.03

Human-scene interaction. We qualitatively show that our latent noise optimization control can be applied to human-scene interaction synthesis, where the goal is to control the human to interact naturally with the surrounding environment. Given an **input 3D scene** and the text prompts specifying the actions and durations, we use latent noise optimization to control the human to reach the goal joint location while adhering to the scene contact and collision constraints. **The input scenes are represented as signed distance fields for evaluating human-scene collision and contact constraints as detailed in Appendix F.** We present generated interactions of climbing stairs and walking to sit on a chair in Fig. 3 and the supplementary video.

486
487
488
489
490
491
492
493
494
495
496
497

(a) Walk, turn left, sit on the chair (b) Walk upstairs (c) Walk downstairs

498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516

Figure 3: Illustrations of human-scene interaction generation given text prompts and goal pelvis joint location (visualized as a red sphere). Best viewed in the supplementary video.

4.3 REINFORCENT LEARNING-BASED CONTROL USING DART

Combining DART with reinforcement learning-based control, we train text-conditioned goal-reaching policies for three actions including ‘walk’, ‘run’, and ‘hop on the left leg’. We evaluate the controllers on paths consisting of sequences of waypoints. We compare our method to a baseline GAMMA (Zhang & Tang, 2022), which learns a motion model using the same motion source of AMASS (Mahmood et al., 2019) dataset as ours and trains RL controllers to reach goal locations. However, GAMMA does not support text conditioning and can only generate walking motion. The evaluation metrics include the reach time, the success rate of reaching the final goal waypoint, foot skating, and foot-floor distance. The evaluation results are shown in Tab. 4. Our controller consistently reaches all goals within a reasonable timeframe, while GAMMA occasionally fails to meet the final goal and may float off the floor beyond the contact threshold. Our text-conditioned goal-reaching policy achieves a generation speed of **240 frames per second**. We believe our results on text-conditioned goal-reaching controllers demonstrate the potential of DART as a foundational human motion model, upon which control models for multiple tasks can be learned without collecting separate datasets for different tasks.

517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539Table 4: Quantitative evaluation of text-conditioned goal-reaching controller. The best results are in **bold** and ‘±’ indicates the 95% confidence interval.

	Time (s)↓	Success rate↑	Skate (cm/s) ↓	Floor distance (cm)↓
GAMMA walk	31.44 ± 2.58	0.95 ± 0.03	5.14 ± 1.58	5.55 ± 0.84
Ours ‘walk’	13.82 ± 0.10	1.0 ± 0.0	5.07 ± 0.18	1.87 ± 0.02
Ours ‘run’	12.16 ± 0.06	1.0 ± 0.0	4.70 ± 0.13	2.02 ± 0.02
Ours ‘hop on left leg’	13.89 ± 0.12	1.0 ± 0.0	41.18 ± 0.61	1.43 ± 0.02

5 LIMITATIONS AND CONCLUSIONS

Our method relies on motion sequences with frame-level aligned text annotations, as provided in BABEL, to achieve precise text-motion alignment and natural transition between actions. When trained on the coarse sentence-level motion labels from HML3D, the text-motion alignment degenerates for texts describing multiple actions, resulting in motions randomly switching between the described actions in a random order. This occurs because each short motion primitive inherently matches only a portion of the sequence’s semantics. Using a coarse sentence-level description as the text label for a primitive introduces semantic misalignment and ambiguity. Inspired by Stoffl et al. (2024), we aim to explore hierarchical latent spaces to effectively tackle both fine-grained and global sequence-level semantics in future work.

Our method DART effectively learns a text-conditioned motion primitive space that enables real-time online motion generation driven by natural languages. Additionally, the learned powerful motion primitive space allows for precise spatial motion control via latent noise optimization or reinforcement learning policies. Experiments demonstrate the superiority of DART in harmonizing spatial control with motion text-semantic alignment in the generated motions.

REFERENCES

- 540
541
542 Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action!
543 audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42
544 (4):1–20, 2023.
- 545 Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier,
546 Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier
547 Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International
548 Conference on Learning Representations*, 2021.
- 549 Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator:
550 Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transac-
551 tions on Graphics (TOG)*, 41(6):1–19, 2022.
- 552 Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents.
553 *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.
- 555 Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. TEACH: Temporal action
556 composition for 3d humans. In *2022 International Conference on 3D Vision (3DV)*, pp. 414–423.
557 IEEE, 2022.
- 558 German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with
559 blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision
560 and Pattern Recognition*, pp. 457–469, 2024.
- 562 Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence
563 prediction with recurrent neural networks. *Advances in neural information processing systems*, 28,
564 2015.
- 565 Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. HumanMAC:
566 Masked motion completion for human motion prediction. In *Proceedings of the IEEE/CVF
567 International Conference on Computer Vision*, pp. 9544–9555, 2023a.
- 568 Rui Chen, Mingyi Shi, Shaoli Huang, Ping Tan, Taku Komura, and Xuelin Chen. Taming diffusion
569 probabilistic models for character control. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10,
570 2024.
- 571 Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your
572 commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on
573 Computer Vision and Pattern Recognition*, pp. 18000–18010, 2023b.
- 575 Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive
576 Whole-Body Control for Humanoid Robots. In *Proceedings of Robotics: Science and Systems*,
577 Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.107.
- 578 Sammy Christen, Wei Yang, Claudia Pérez-D’Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei
579 Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF
580 Conference on Computer Vision and Pattern Recognition*, pp. 9654–9664, 2023.
- 582 Simon Clavet et al. Motion matching and the road to next-gen animation. In *Proc. of GDC*, volume 2,
583 pp. 4, 2016.
- 584 Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion
585 in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–9,
586 2024.
- 587 Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li,
588 Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models
589 with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*,
590 2023.
- 592 Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. MotionLCM:
593 Real-time controllable motion generation via latent consistency model. In *European Conference
on Computer Vision*, pp. 390–408. Springer, 2025.

- 594 Ross Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*,
595 pp. 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.
- 596
- 597 Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa*, and Jitendra Malik*.
598 Humans in 4D: Reconstructing and tracking humans with transformers. In *International Conference*
599 *on Computer Vision (ICCV)*, 2023.
- 600 Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating
601 Diverse and Natural 3D Human Motions from Text. In *2022 IEEE/CVF Conference on Computer*
602 *Vision and Pattern Recognition (CVPR)*, pp. 5142–5151, June 2022.
- 603
- 604 Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative
605 masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer*
606 *Vision and Pattern Recognition*, pp. 1900–1910, 2024.
- 607 Bo Han, Hao Peng, Minjing Dong, Yi Ren, Yixuan Shen, and Chang Xu. AMD: Autoregressive
608 motion diffusion. *AAAI*, 2024.
- 609
- 610 Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J
611 Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International*
612 *Conference on Computer Vision*, pp. 11374–11384, 2021.
- 613 Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on*
614 *Deep Generative Models and Downstream Applications*, 2021.
- 615
- 616 Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in*
617 *neural information processing systems*, 33:6840–6851, 2020.
- 618 Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J
619 Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646,
620 2022.
- 621 Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with
622 convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*, pp. 1–4, 2015.
- 623
- 624 Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control.
625 *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.
- 626
- 627 Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching.
628 *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020. ISSN 0730-0301. Publisher: ACM New
629 York, NY, USA.
- 630 Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as
631 a foreign language. *Advances in Neural Information Processing Systems*, 36, 2024a.
- 632
- 633 Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin
634 Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings*
635 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1737–1747, 2024b.
- 636
- 637 Roy Kapon, Guy Tevet, Daniel Cohen-Or, and Amit H Bermano. MAS: Multi-view ancestral sampling
638 for 3d motion generation using 2d diffusion. In *Proceedings of the IEEE/CVF Conference on*
Computer Vision and Pattern Recognition, pp. 1965–1974, 2024.
- 639
- 640 Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided
641 motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF*
International Conference on Computer Vision, pp. 2151–2162, 2023.
- 642
- 643 Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn,
644 and Siyu Tang. GMD: Controllable human motion synthesis via guided diffusion models. In *IEEE*
645 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.
- 646
- 647 Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn,
and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of*
the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1334–1345, 2024b.

- 648 Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International*
649 *Conference on Learning Representations*, 2015.
- 650 Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference*
651 *on Learning Representations*, 2014.
- 652 Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM SIGGRAPH 2008*
653 *classes*, pp. 1–10, 2008.
- 654 Gen Li, Kaifeng Zhao, Siwei Zhang, Xiaozhong Lyu, Mihai Dusmanu, Yan Zhang, Marc Pollefeys,
655 and Siyu Tang. EgoGen: An egocentric synthetic data generator. In *IEEE Conference on Computer*
656 *Vision and Pattern Recognition (CVPR)*, 2024a.
- 657 Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM*
658 *Transactions on Graphics (TOG)*, 42(6):1–11, 2023.
- 659 Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. Control-
660 lable human-object interaction synthesis. In *European Conference on Computer Vision*, 2024b.
- 661 Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-
662 human motion generation under complex interactions. *International Journal of Computer Vision*,
663 pp. 1–21, 2024.
- 664 Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang.
665 Motion-X: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural*
666 *Information Processing Systems*, 2023.
- 667 Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character Controllers Using
668 Motion VAEs. *ACM Transactions on Graphics*, 39(4), August 2020.
- 669 Xinpeng Liu, Haowen Hou, Yanchao Yang, Yong-Lu Li, and Cewu Lu. Revisit human-scene
670 interaction via space occupancy. In *European Conference on Computer Vision*, 2024.
- 671 Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL:
672 A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):
673 248:1–248:16, October 2015.
- 674 Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid
675 control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*,
676 2023.
- 677 Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M Kitani, and Weipeng
678 Xu. Universal humanoid motion representations for physics-based control. In *The Twelfth*
679 *International Conference on Learning Representations*, 2024.
- 680 Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black.
681 AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer*
682 *Vision*, pp. 5442–5451, October 2019.
- 683 Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent
684 neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,
685 pp. 2891–2900, 2017.
- 686 Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios
687 Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single
688 image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp.
689 10975–10985, 2019.
- 690 Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: large-scale reusable
691 adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 41(4), jul
692 2022.
- 693 Mathis Petrovich, Michael J Black, and Gül Varol. TEMOS: Generating diverse human motions from
694 textual descriptions. In *European Conference on Computer Vision*, pp. 480–497. Springer, 2022.

- 702 Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and
703 Michael J Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the*
704 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 722–731, 2021.
- 705 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
706 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
707 models from natural language supervision. In *International conference on machine learning*, pp.
708 8748–8763. PMLR, 2021.
- 709 Davis Remppe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas.
710 HuMoR: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF*
711 *international conference on computer vision*, pp. 11488–11499, 2021.
- 712 Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-
713 resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF confer-*
714 *ence on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- 715 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
716 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- 717 Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a
718 generative prior. In *The Twelfth International Conference on Learning Representations*, 2024.
- 719 Mengyi Shan, Lu Dong, Yutao Han, Yuan Yao, Tao Liu, Ifeoma Nwogu, Guo-Jun Qi, and Mitch Hill.
720 Towards open domain text-driven synthesis of multi-person motions. In *European Conference on*
721 *Computer Vision*, 2024.
- 722 Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. Interactive character
723 control with auto-regressive motion diffusion models. *ACM Trans. Graph.*, 43(4), jul 2024. ISSN
724 0730-0301.
- 725 Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and
726 Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In
727 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
728 11050–11059, 2022.
- 729 Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Interna-*
730 *tional Conference on Learning Representations*, 2021a.
- 731 Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben
732 Poole. Score-based generative modeling through stochastic differential equations. In *International*
733 *Conference on Learning Representations*, 2021b.
- 734 Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene
735 interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.
- 736 Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning
737 motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022. ISSN 0730-
738 0301. Publisher: ACM New York, NY, USA.
- 739 Sebastian Starke, Paul Starke, Nicky He, Taku Komura, and Yuting Ye. Categorical codebook
740 matching for embodied character controllers. *ACM Transactions on Graphics (TOG)*, 43(4):1–14,
741 2024.
- 742 Lucas Stoffl, Andy Bonnetto, Stéphane d’Ascoli, and Alexander Mathis. Elucidating the hierarchical
743 nature of behavior with masked autoencoders. In *European conference on computer vision*.
744 Springer, 2024.
- 745 Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT
746 press Cambridge, 1998.
- 747 Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano.
748 Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*,
749 2023.

- 756 Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music.
757 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
758 448–458, 2023.
- 759 Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott Delp, and C Karen Liu. DiffusionPoser:
760 Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion.
761 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
762 2513–2523, 2024.
- 763 Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin
764 Zhu, Wei Liang, and Siyuan Huang. Move as you say, interact as you can: Language-guided
765 human motion generation with scene affordance. In *Proceedings of the IEEE/CVF Conference on
766 Computer Vision and Pattern Recognition (CVPR)*, 2024.
- 767 Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control
768 any joint at any time for human motion generation. In *The Twelfth International Conference on
769 Learning Representations*, 2024.
- 770 Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object
771 interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International
772 Conference on Computer Vision*, pp. 14928–14940, 2023.
- 773 He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for
774 quadruped motion control. *ACM Transactions on Graphics*, 37(4):1–11, August 2018.
- 775 Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen,
776 and Ying Shan. T2M-GPT: Generating human motion from textual descriptions with discrete
777 representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern
778 recognition*, pp. 14730–14740, 2023a.
- 779 Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image
780 diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
781 pp. 3836–3847, 2023b.
- 782 Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu.
783 MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions
784 on Pattern Analysis and Machine Intelligence*, 46:4115–4128, 2022.
- 785 Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF
786 Conference on Computer Vision and Pattern Recognition*, pp. 20481–20491, 2022.
- 787 Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simula-
788 tion and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference
789 Proceedings*, pp. 1–11, 2023c.
- 790 Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. Compositional human-scene
791 interaction synthesis with semantic control. In *European Conference on Computer Vision*, pp.
792 311–327. Springer, 2022.
- 793 Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse
794 human motions in 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on
795 Computer Vision*, pp. 14738–14749, 2023.
- 796 Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation
797 representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer
798 vision and pattern recognition*, pp. 5745–5753, 2019.
- 799 Fabio Zinno. MI tutorial day: From motion matching to motion synthesis, and all the hurdles in
800 between. *Proc. of GDC 2019*, 2, 2019.
- 801
802
803
804
805
806
807
808
809

810 A MOTION PRIMITIVE REPRESENTATION

811
812 **Representation.** We represent each frame of the motion primitive as a tuple of
813 $(\mathbf{t}, \mathbf{R}, \theta, \mathbf{J}, d\mathbf{t}, d\mathbf{R}, d\mathbf{J})$, where $\mathbf{t} \in \mathbb{R}^3$ denotes the global body translation, $\mathbf{R} \in \mathbb{R}^6$ denotes the
814 6D rotation representation (Zhou et al., 2019) of the global body orientation, $\theta \in \mathbb{R}^{21 \times 6}$ is the 6D
815 representation of 21 joint rotations, $\mathbf{J} \in \mathbb{R}^{22 \times 3}$ denotes the 22 joints locations, $d\mathbf{t} \in \mathbb{R}^3$ denotes the
816 temporal difference with previous frame’s translation, $d\mathbf{R} \in \mathbb{R}^6$ denotes the 6D representation of the
817 relative rotation between current and previous frame’s body orientation, and $d\mathbf{J} \in \mathbb{R}^{22 \times 3}$ denotes the
818 temporal difference between current and previous frame’s joint locations.

819 Our motion representation is **overparameterized** (Holden et al., 2017; Ling et al., 2020; Rempe
820 et al., 2021; Guo et al., 2022), with multiple benefits. Firstly, the joint rotation components θ
821 are compatible with animation pipelines, saving the time-consuming optimization-based skeleton-
822 to-body conversion required by the commonly used HML3D (Guo et al., 2022) representation in
823 text-to-motion methods. Moreover, including the joint location components \mathbf{J} facilitates solving
824 physical constraints like reducing foot skating and joint trajectory control. Our motion representation
825 also models the first-order kinematics with the temporal difference features to improve the motion
826 naturalness.

827 **Canonicalization.** We represent motion primitives in a human-centric local coordinates frame to
828 canonicalize the primitive features and facilitate model learning. Each motion primitive is canonical-
829 ized in a local coordinates system centered at the first frame body. The origin is located at the pelvis
830 of the first frame body, the X-axis is the horizontal projection of the vector pointing from the left hip
831 to the right hip, and the Z-axis is pointing in the inverse gravity direction. Given the pelvis, left hip,
832 and right hip joints, the origin is located at the pelvis joint, and the local axis system can be derived
833 by:

834 Algorithm 3 Motion primitive rotation canonicalization

```
835
836 Input: right_hip  $\in \mathbb{R}^3$ , left_hip  $\in \mathbb{R}^3$ 
837   x_axis = right_hip - left_hip
838   x_axis[2] = 0                                ▷ Project to the xy plane
839   normalize(x_axis)
840   z_axis = [0, 0, 1]                            ▷ Inverse gravity direction
841   y_axis = cross_product(z_axis, x_axis)
842   normalize(y_axis)
843   return [x_axis, y_axis, z_axis]
```

844 We store the canonicalization transformations and apply their inverse transformation to the generated
845 motion primitives to recover global motions in the world coordinates.

848 B DATASETS

849 We train separate DART models on motion-text data from the BABEL (Punnakkal et al., 2021)
850 and HML3D (Guo et al., 2022) dataset. Both BABEL and HML3D use motion sources from the
851 AMASS (Mahmood et al., 2019) dataset. Their main difference is that BABEL features fine-grained
852 frame-level text annotation while HML3D uses coarse sequence-level annotation.

853 The BABEL dataset contains motion capture sequences with frame-aligned text labels that annotate
854 the fine-grained semantics of actions. Fine-grained text labels in BABEL allow models to learn
855 precise human action controls and natural transitions among actions. We use the motion data at a
856 framerate of 30 frames per second the same as prior works (Barquero et al., 2024; Athanasiou et al.,
857 2022). During training, motion primitives are randomly sampled from data sequences and the text
858 label is randomly sampled from all the action segments that overlap with the primitive. To alleviate
859 the action imbalance in the BABEL dataset, we use the provided action labels to perform importance
860 sampling during training so that the motion data of each action has roughly equal sampling chances
861 despite their varying frequency in the original dataset.

862 The HML3D dataset contains short motions with coarse sequence-level sentence descriptions. We
863 only use a subset of HML3D in training since its subset HumanAct12 and left-right mirroring motion

864 sequences only provide joint locations instead of SMPL (Loper et al., 2015; Pavlakos et al., 2019)
 865 body sequences that our motion representation requires. The joint rotations used in the original
 866 HML3D are calculated using naive inverse kinematics and can not be directly used to animate
 867 human motions. We therefore only train using the subset where the SMPL body motion sequences is
 868 available. The HML3D motions are sampled at a framerate of 20 frames per second. During training,
 869 we randomly sample primitives with uniform probability and the text label is randomly chosen from
 870 one of the multiple sentence captions of the overlapping sequence.

872 C MOTION PRIMITIVE VAE

873
 874 **Architecture.** Our Motion Primitive VAE employs the transformer-based architecture. Both the
 875 encoder and decoder consist of 7 transformer encoder layers with skip connections (Chen et al.,
 876 2023b). The transformer layers use the dropout rate of 0.1, feed-forward dimension of 1024, hidden
 877 dimension of 256, 4 attention heads, and the gelu activation function. The latent space dimension is
 878 256. After finishing training, we follow (Rombach et al., 2022) to calculate the variance of the latent
 879 variables using a data batch. When training the latent denoiser model, the raw latent output from the
 880 encoder is scaled to have a unit standard deviation.

881 **Losses.** The motion primitive VAE is trained with the following losses:

$$882 L_{VAE} = L_{rec} + w_{KL} \times L_{KL} + w_{aux} \times L_{aux} \quad (5)$$

$$883 L_{rec} = \mathcal{F}(\hat{\mathbf{X}}, \mathbf{X}) \quad (6)$$

884 The reconstruction loss L_{rec} aims to minimize the distance between the reconstructed future frames
 885 $\hat{\mathbf{X}}$ and the ground truth future frames \mathbf{X} , where $\mathcal{F}(\cdot, \cdot)$ denotes the distance function and we use the
 886 smoothed L1 loss (Girshick, 2015) in implementation.

$$887 \mathcal{L}_{KL} = KL(q(z|\mathbf{H})||\mathcal{N}(0, I)), \quad (7)$$

888 L_{KL} penalizes the distribution difference between the predicted distribution and a standard Gaussian,
 889 where $KL(\cdot, \cdot)$ denotes the Kullback-Leibler divergence (KL), and $q(z|\mathbf{H})$ denotes the predicted
 890 distribution from the encoder \mathcal{E} . We use a small KL term of $1e^{-6}$ following Rombach et al. (2022)
 891 as we aim to keep the latent space expressive and only use the small KL loss to avoid arbitrarily
 892 high-variance latent spaces.

$$893 L_{aux} = \mathcal{F}(\bar{d}\hat{\mathbf{t}}, \hat{d}\hat{\mathbf{t}}) + \mathcal{F}(\bar{d}\hat{\mathbf{J}}, \hat{d}\hat{\mathbf{J}}) + \mathcal{F}(\bar{d}\hat{\mathbf{R}}, \hat{d}\hat{\mathbf{R}}) \quad (8)$$

894 The auxiliary loss L_{aux} regularizes the predicted temporal difference features $\hat{d}\cdot$ of translations,
 895 global orientation, and joints to be close to the actual temporal differences $\bar{d}\cdot$ calculated from the
 896 predicted motion features. **For instance, the difference of the first two frames of the predicted root
 897 translation ($\bar{d}\hat{\mathbf{t}}[0] := \hat{\mathbf{t}}[1] - \hat{\mathbf{t}}[0]$) should be consistent with the predicted first frame root translation
 898 difference feature $\hat{d}\hat{\mathbf{t}}[0]$.** We use $w_{KL} = 1e^{-6}$ and $w_{aux} = 100$ in our experiments. We train the
 899 motion primitive VAE with the AdamW (Kingma & Ba, 2015) optimizer and the learning rate is set
 900 to $1e^{-4}$ with linear annealing.

901 We conduct ablation studies on the impacts of the losses. We evaluate motion primitive VAEs
 902 trained with different loss weights on autoregressive motion sequence reconstruction error and motion
 903 jittering in reconstructed motions. Using large KL loss weight of $w_{KL} = 1$, reduces the model
 904 expressiveness (the reconstruction error increases from 0.08 to 0.44 compared to $w_{KL} = 1e^{-6}$ on the
 905 test motions) and fails to accurately reconstruct complex sequences like cartwheeling. Using a small
 906 KL loss weight of $w_{KL} = 1e^{-6}$ can maintain the expressiveness of the learned VAE while allowing
 907 the latent distribution to deviate a bit more from a standard Gaussian. Applying the auxiliary losses
 908 with $w_{aux} = 100$ helps to reduce the jittering in the reconstructions and improve the motion quality
 909 compared to using $w_{aux} = 0$, as reflected by a smaller jerk metric of 2.45 when using $w_{aux} = 100$
 910 compared to a jerk of 3.67 when using $w_{aux} = 0$.

D LATENT DENOISER MODEL

D.1 LOSSES

We train the latent denoiser model using DDPM (Ho et al., 2020) with 10 diffusion steps and use a cosine noise scheduler. The denoiser model is trained with the following losses:

$$L_{denoiser} = L_{simple} + w_{rec} \times L_{rec} + w_{aux} \times L_{aux} \quad (9)$$

$$L_{simple} = \mathcal{F}(\mathcal{G}(z_t, t, \mathbf{H}, c), z_0), \quad (10)$$

where $\mathcal{F}(\cdot, \cdot)$ is a distance function and we use the smooth L1 loss (Girshick, 2015) in our implementation. We train the denoiser to predict the clean latent variable with the simple objective L_{simple} , and apply the feature reconstruction loss L_{rec} and auxiliary losses L_{aux} on the decoded motion primitive $\hat{\mathbf{X}} = \mathcal{D}(\mathcal{G}(z_t, t, \mathbf{H}, c))$ to ensure the decoded motion primitives are valid.

D.2 SCHEDULED TRAINING

We use scheduled training to improve the stability of long sequence generation and the text prompt controllability. Long-term prediction stability is a significant challenge in autoregressive generation since the sample distribution can drift and accumulate during autoregressive generation. When the sample drifts out of the distribution covered by the learned model, the generation results can go wild. Our latent motion primitive model also faces the long-term stability challenge as an autoregressive method.

To alleviate the out-of-distribution problems, we use the scheduled training (Ling et al., 2020; Bengio et al., 2015; Rempe et al., 2021; Martinez et al., 2017) to progressively introduce the test-time distributions during training. Specifically, we train the latent denoiser model on sequences of N consecutive motion primitives and use the prediction result of the previous motion primitive instead of the ground truth dataset to extract the history motion input \mathbf{H} . We name such history motion extracted from the predicted last primitive as rollout history. Using the rollout history instead of the ground truth history introduces the test-time distribution which can differ from the dataset distribution, e.g., unseen human poses or out-of-distribution combinations of human bodies and text labels. Exposing the model to such test-time distribution at training can improve the long-term generation stability and increase the text controllability when facing novel combinations of history motion and text prompts at generation time.

The scheduled training has three stages to progressively introduce the rollout history. The first stage is fully supervised training where only the ground truth history is used during training. The second scheduled learning stage randomly replaces the ground truth history motion with rollout history motion by a probability linearly increasing from 0 to 1. The third stage of rollout training always uses the rollout history instead of the ground truth history. The scheduled training algorithm for the latent denoising model is shown in Alg. 5.

We use $w_{rec} = 1$ and $w_{aux} = 10000$ in our experiments. The denoiser model is trained using an AdamW optimizer. The learning rate is set to $1e^{-4}$ with linear annealing. Our denoiser model is trained with scheduled training (Ling et al., 2020; Bengio et al., 2015), consisting of a fully supervised stage of 100K iterations, a scheduled stage of 100K iterations, and a rollout stage of 100K iterations. We set the maximum number of rollouts as 4. With the scheduled training, our latent motion primitive model can stably generate long motion sequences and better respond to the text prompt control even at poses that are not paired with the text prompt in the dataset.

E TEXT-CONDITIONED TEMPORAL MOTION COMPOSITION

E.1 EXPERIMENT DETAILS

We apply DART to conduct online motion generation using the rollout algorithm 1 with a default seed motion \mathbf{H}_{seed} of rest standing and using a classifier-free guidance **weight of 5**. We use the released checkpoints of FlowMDM (Barquero et al., 2024), TEACH (Athanasios et al., 2022),

Algorithm 4 Calculate rollout probability

```

972 1: Input: current iteration number  $iter$ , number of train iterations in the first supervised stage  $I_1$ ,
973   number of train iterations in the second scheduled stage  $I_2$ 
974 2: Output: rollout probability  $p$ 
975 3: function ROLLOUT_PROBABILITY( $iter, I_1, I_2$ )
976 4:   if  $iter \leq I_1$  then                                     ▷ no rollout in the first supervised stage
977    $p \leftarrow 0$ 
978 5:   else if  $iter > I_1 + I_2$  then                             ▷ the third rollout stage always use rollout
979    $p \leftarrow 1$ 
980 6:   else                                                       ▷ linearly scheduled rollout probability in the second scheduled stage
981    $p \leftarrow \frac{iter - I_1}{I_2}$ 
982 7:   end if
983 8:   return  $p$ 
984 9: end function

```

Algorithm 5 Scheduled training for the latent denoising model

```

987 1: Input: pretrained motion primitive decoder  $\mathcal{D}$  and encoder  $\mathcal{E}$ , latent variable denoiser  $\mathcal{G}_\Theta$ 
988   parameterized by  $\Theta$ , total diffusion steps  $T$ , optimizer  $\mathcal{O}$ , loss criterion  $\mathcal{L}$ , train dataset  $\mathcal{X}$ .
989 2: Scheduled training parameters: the number of train iterations in the first supervised stage  $I_1$ ,
990   the number of train iterations in the second scheduled stage  $I_2$ , the number of train iterations in
991   the third rollout stage  $I_3$ , the maximum number of primitive rollouts  $N$  during training.
992 3:
993 4:  $I_{total} \leftarrow I_1 + I_2 + I_3$ 
994 5:  $iter \leftarrow 0$ 
995 6: while  $iter < I_{total}$  do
996 7:    $[\mathbf{H}_{seed}, \mathbf{X}^1, c^1, \dots, \mathbf{X}^N, c^N] \sim \mathcal{X}$ 
997   ▷ sample  $N$  consecutive motion primitives with text labels from dataset  $\mathcal{X}$ 
998 8:    $\mathbf{H} \leftarrow \mathbf{H}_{seed}$                                        ▷ initialize motion history
999 9:   for  $i \leftarrow 1$  to  $N$  do                                     ▷ number of rollouts
1000 10:     $z_0^i = \mathcal{E}(\mathbf{H}, \mathbf{X}^i)$                                ▷ compress motion primitive into latent space
1001 11:     $t \sim \mathcal{U}[0, T)$                                        ▷ sample diffusion step  $t$ 
1002 12:     $z_t^i \leftarrow \text{FORWARD\_DIFFUSION}(z_0^i, t)$ 
1003 13:     $\hat{z}_0^i = \mathcal{G}_\Theta(z_t^i, t, \mathbf{H}, c^i)$                        ▷ latent denoising model prediction
1004 14:     $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{H}, \hat{z}_0^i)$                                ▷ decode predicted latent variable to future motion frames
1005 15:     $\nabla \leftarrow \nabla_\Theta \mathcal{L}(z_0^i, \hat{z}_0^i, \mathbf{H}, \mathbf{X}^i, \hat{\mathbf{X}}^i)$    ▷ model parameter gradient calculation
1006 16:     $\Theta \leftarrow \mathcal{O}(\Theta, \nabla)$                              ▷ model update using optimizer
1007 17:
1008 18:     $p \leftarrow \text{ROLLOUT\_PROBABILITY}(iter, I_1, I_2)$ 
1009 19:    ▷ update history motion using predicted or GT motion by a scheduled probability
1010 20:    if  $\text{rand}() < p$  then                                       ▷ use predicted rollout history
1011 21:     $z_T^i \leftarrow \text{FORWARD\_DIFFUSION}(z_0^i, T)$  ▷ maximum noising simulating inference time
1012 22:     $\hat{z}_0^i = \text{DDPM\_SAMPLE}(\mathcal{G}_\Theta, z_T^i, T, \mathbf{H}, c^i)$        ▷ full DDPM denoising loop
1013 23:     $\hat{\mathbf{X}} = \text{sg}(\mathcal{D}(\mathbf{H}, \hat{z}_0^i))$                                ▷ decode predicted latent variable and stop gradient
1014 24:     $\mathbf{H} \leftarrow \text{CANONICALIZE}(\hat{\mathbf{X}}^{F-H+1:F})$ 
1015 25:  else                                                           ▷ use ground truth history
1016 26:     $\mathbf{H} \leftarrow \text{CANONICALIZE}(\mathbf{X}^{F-H+1:F})$ 
1017 27:  end if
1018 28:
1019 29:     $iter \leftarrow iter + 1$ 
1020 30:  end for
1021 31: end while

```

and DoubleTake (Shafir et al., 2024) for baseline comparison. We adjust the handshake size and blending length of DoubleTake to be compatible with the shortest segment length of 15 frames. The history-conditioned modification of T2M-GPT is retrained on the BABEL dataset using the original

hyperparameters. At generation time, the last frames of the previous action segment are encoded into tokens and provided as the first tokens when generating the next action segment to provide history conditioning of the previous action.

We extract the timeline of action segments for evaluation from the BABEL (Punnakkal et al., 2021) valid set. For each sequence in the validation set, we sort the original frame labels provided by BABEL to obtain a list of tuples of text prompts and durations. We randomly sample one text label when a segment is annotated by multiple texts. We skip the ‘transition’ text labels due to the ambiguous semantics and clamp the duration length with a minimum of 15 frames.

E.2 PERCEPTUAL USER STUDIES.

We conduct user studies to quantitatively compare our method DART with baselines to provide a more comprehensive and convincing evaluation. We run binary user studies on Amazon Mechanical Turk (AMT) to evaluate generation realism and text-motion semantic alignment. Participants are given generation results from two different methods and are asked to select the generation that is perceptually more realistic or better aligns with the action text stream in subtitles, as illustrated in Fig. 4. During the realism evaluation, action descriptions were not displayed to eliminate distractions. We sample 256 sequences of action texts and durations from the BABEL dataset and use each method to generate motions given the action timelines. Participants are shown random pairs of results from our method and a baseline, and they are asked to choose the better generation. Each comparison is voted by 3 independent participants, with the video pairs randomly shuffled to ensure that participants do not know the method source.

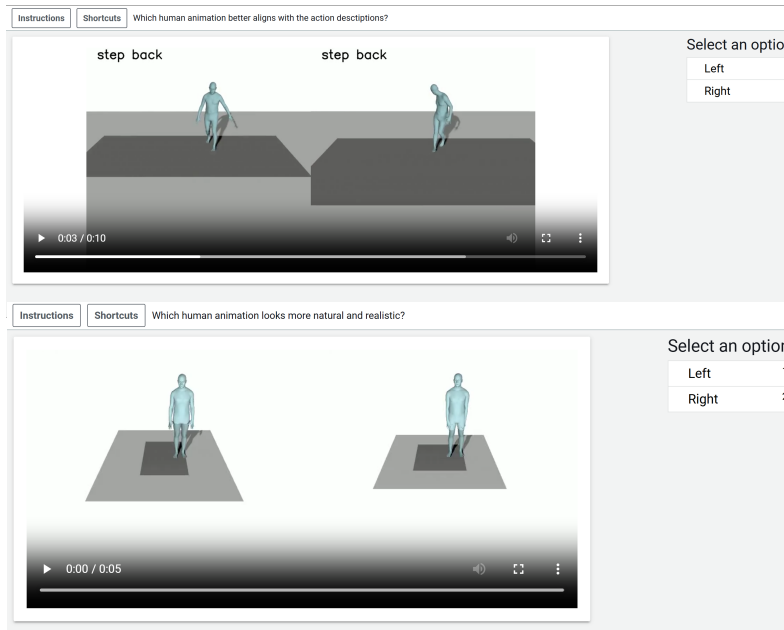


Figure 4: Illustration of the binary user study interface for evaluating motion-text semantic alignment (top) and perceptual realism (bottom). Participants are requested to select the generation that is perceptually more realistic or better aligns with the action descriptions in subtitles (only visible in semantic user study).

E.3 ABLATION STUDIES.

We conduct ablative studies on architecture designs, diffusion steps, primitive representation, and scheduled training on the text-conditioned motion composition task.

Removing the variational autoencoder (DART-VAE) and training diffusion model in the raw motion space results in a significantly higher jittering as reflected by the peak jerk (PJ) and Area Under Jerk

(AUI) metrics. This validates the effectiveness of integrating the variational encoder to compress the high-frequency noise in motion data and improve motion generation quality.

Without the scheduled training (DART-schedule), the model can not effectively respond to text control and have significantly worse R-Prec and FID metrics. This is because, without scheduled training, the model can easily encounter out-of-distribution combinations of history motion and text prompts during autoregressive generation.

We also include the ablation study of training a model to predict the single next frame conditioned on the current frame (per frame) as used in Shi et al. (2024). This is a special case of primitive with history length $H = 1$ and future length $F = 1$. The ablative model has significantly worse R-Prec and FID metrics and cannot respond to text prompts. This indicates using frame-by-frame prediction is less effective than primitives with a reasonable horizon ($H = 2, F = 8$) in learning text-conditioned motion space.

DART can learn high-quality text-conditioned motion primitive models using very few diffusion steps because of the simplicity of the primitive representation. We conduct ablative studies of training DART using different numbers of DDPM diffusion steps. Reducing the diffusion steps from 100 to fewer than 10 does not significantly affect performance. However, an extremely low diffusion step number of 2 leads to a much higher FID, indicating poorer motion quality.

Table 5: Ablation studies results on text-conditioned temporal motion composition. The first row includes the metrics of the dataset for reference. Symbol ‘ \rightarrow ’ denotes that closer to the dataset reference is better and ‘ \pm ’ indicates the 95% confidence interval.

	Segment				Transition			
	FID \downarrow	R-prec \uparrow	DIV \rightarrow	MM-Dist \downarrow	FID \downarrow	DIV \rightarrow	PJ \rightarrow	AUI \downarrow
Dataset	0.00 \pm 0.00	0.72 \pm 0.00	8.42 \pm 0.15	3.36 \pm 0.00	0.00 \pm 0.00	6.20 \pm 0.06	0.02 \pm 0.00	0.00 \pm 0.00
Ours	3.79 \pm 0.06	0.62 \pm 0.01	8.05 \pm 0.10	5.27 \pm 0.01	1.86 \pm 0.05	6.70 \pm 0.03	0.06 \pm 0.00	0.21 \pm 0.00
DART-VAE	4.23 \pm 0.02	0.62 \pm 0.01	8.33 \pm 0.12	5.29 \pm 0.01	1.79 \pm 0.02	6.73 \pm 0.23	0.20 \pm 0.00	0.96 \pm 0.00
DART-schedule	8.08 \pm 0.09	0.39 \pm 0.01	8.05 \pm 0.12	6.96 \pm 0.03	7.41 \pm 0.10	6.58 \pm 0.06	0.03 \pm 0.00	0.18 \pm 0.00
per frame($H=1, F=1$)	10.31 \pm 0.09	0.29 \pm 0.01	6.82 \pm 0.13	7.41 \pm 0.01	7.82 \pm 0.09	6.03 \pm 0.08	0.02 \pm 0.00	0.08 \pm 0.00
$H=2, F=16$	4.04 \pm 0.10	0.66 \pm 0.00	8.20 \pm 0.06	4.96 \pm 0.01	2.22 \pm 0.10	6.60 \pm 0.20	0.06 \pm 0.00	0.18 \pm 0.00
steps 2	4.44 \pm 0.04	0.60 \pm 0.00	8.20 \pm 0.15	5.38 \pm 0.01	2.24 \pm 0.02	6.77 \pm 0.07	0.05 \pm 0.00	0.20 \pm 0.00
steps 5	3.49 \pm 0.09	0.63 \pm 0.00	8.25 \pm 0.15	5.18 \pm 0.01	2.11 \pm 0.07	6.74 \pm 0.11	0.05 \pm 0.00	0.20 \pm 0.00
steps 8	3.70 \pm 0.03	0.62 \pm 0.01	8.04 \pm 0.13	5.25 \pm 0.03	2.15 \pm 0.08	6.72 \pm 0.15	0.06 \pm 0.00	0.20 \pm 0.00
steps 10 (Ours)	3.79 \pm 0.06	0.62 \pm 0.01	8.05 \pm 0.10	5.27 \pm 0.01	1.86 \pm 0.05	6.70 \pm 0.03	0.06 \pm 0.00	0.21 \pm 0.00
steps 50	3.82 \pm 0.05	0.60 \pm 0.00	7.74 \pm 0.07	5.30 \pm 0.01	2.11 \pm 0.10	6.58 \pm 0.10	0.06 \pm 0.00	0.22 \pm 0.00
steps 100	4.16 \pm 0.06	0.61 \pm 0.00	7.82 \pm 0.15	5.32 \pm 0.02	2.20 \pm 0.05	6.43 \pm 0.10	0.06 \pm 0.00	0.21 \pm 0.00

F LATENT DIFFUSION NOISE OPTIMIZATION-BASED CONTROL

Optimization details. We use the Adam (Kingma & Ba, 2015) optimizer with a learning rate of 0.05 to optimize latent noises Z_T for 100 to 300 steps. The learning rate is linearly annealed to 0 and the gradient is normalized to stabilize optimization. The optimization running time is dependent on both the motion sequence length and the number of optimization steps. An example experiment of a target motion sequence of 60 frames and 100 optimization steps costs around 74 seconds.

Human-scene interaction. In human-scene interaction synthesis, we use two scene constraints to avoid human-scene collision and foot-floating artifacts. To reduce human-scene interpenetration, $cons_coll(\cdot)$ penalizes joints colliding into scenes with negative SDF value and can be formulated as:

$$cons_coll(\mathbf{M}) = - \sum_J (\Psi(J(\mathbf{M})) - J(\mathbf{M})_{thresh})_- \quad (11)$$

where $\Psi(\cdot)$ is the signed distance function returning the signed distance from the query point to the closest scene point. $J(\cdot)$ extracts the joint locations from the motion \mathbf{M} . $(\cdot)_{thresh}$ is the joint-dependent contact threshold value, which is determined by the joint-skin distance in the rest pose. $(\cdot)_-$ denotes clipping positive values.

To reduce the occurrence of foot-floating artifacts, $cons_cont(\cdot)$ encourages the foot to be in contact with the scene and is defined as:

$$cons_cont(\mathbf{M}) = \sum_F (\Psi(F(\mathbf{M})) - F(\mathbf{M})_{thresh})_+ \quad (12)$$

where $F(\cdot)$ extracts the lower foot joint location. $(\cdot)_+$ denotes clipping negative values.

G REINFORCEMENT LEARNING-BASED CONTROL

Architecture. Both the actor and critic networks are 4-layer MLPs with residual connections and a hidden dimension of 512. We apply the tanh scaling: $x = 4 \cdot \tanh(x)$ (Ling et al., 2020) to the actor output to clip the action prediction in the range of $[-4, 4]$, avoiding unbounded action predictions. The actor networks are initialized with close to zero weights to boost policy training following (Andrychowicz et al., 2021). The observation input contains a 512D text embedding, a 552D of history motion, a 1D observation of floor height relative to the human pelvis, a 1D observation of the floor plane distance from the pelvis to the goal location, and a 3D unit vector of the goal direction in the human-centric coordinates frames. We clamp the goal distance observation with a maximum value of 5m and the goal direction in the range of a 120-degree field of view to simulate egocentric human perception.

Rewards. The text-conditioned goal-reaching control policies are trained to maximize the discounted expectation of the following rewards (Zhang & Tang, 2022; Li et al., 2024a; Zhao et al., 2023):

We use three distance-related rewards to encourage the human agent to minimize its distance to the goal location.

$$r_{dist} = D^{i-1} - D^i \quad (13)$$

The distance reward r_{dist} encourages the human to get closer to the goal location, where D^i is the 2D distance between the human pelvis and the goal location at step i .

$$r_{succ} = \begin{cases} 1 & \text{if } D^i < 0.3 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The success reward $succ$ gives a sparse but strong reward when the human arrives at the goal, where 0.3m is the success threshold value.

$$r_{ori} = \frac{\langle p^i - p^{i-1}, g - p^{i-1} \rangle + 1}{2}, \quad (15)$$

The moving orientation reward encourages the moving orientation to be aligned with the goal orientation, where p^i is the human pelvis location at step i , and g is the goal location.

Moreover, we apply scene constraints-related rewards to discourage unnatural behaviors such as foot skating and collision with the floor.

$$r_{skate} = -disp \cdot (2 - 2^{h/0.03}), \quad (16)$$

The skate reward r_{skate} penalizes foot displacements when in contact with the floor, where $disp$ is the foot displacement in two consecutive frames, h denotes the higher foot height in two consecutive frames and 0.03m is the threshold value for contact.

$$r_{floor} = -(|lf| - 0.03)_+, \quad (17)$$

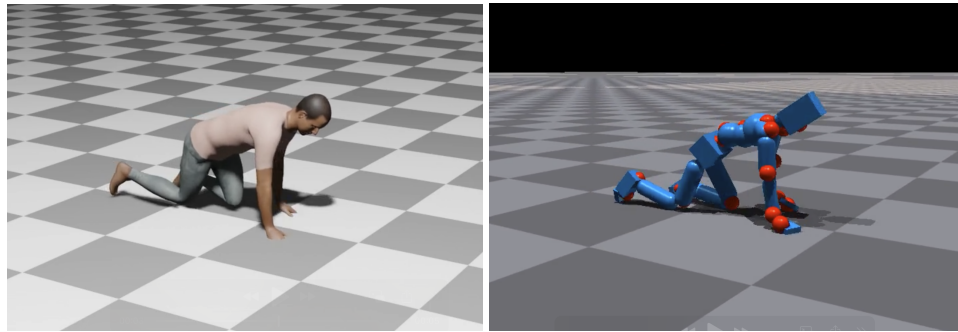
The floor contact reward r_{floor} penalizes when the lower foot distance to the floor is above the threshold of 0.03m, where lf denotes the height of the lower foot, $|\cdot|$ denotes the absolute value operator, and $(\cdot)_+$ is the clipping operator with a minimum of 0.

The rewards are weighted with $w_{dist} = 1$, $w_{succ} = 1$, $w_{ori} = 0.1$, $w_{floor} = 100$, $w_{skate} = 10$ for hopping and 100 for others.

H COMBINATION WITH PHYSICS-BASED MOTION TRACKING

Our method, DART, enables real-time motion generation in response to online text prompts. However, as a kinematic-based approach, DART may produce physically inaccurate motions with artifacts such as skating and floating. To address this, we demonstrate that DART can be integrated with physically simulated motion tracking methods, specifically PHC (Luo et al., 2023), to generate

more physically plausible motions. In Fig. 5, we present an example sequence of a person crawling. The raw generation results from DART exhibit artifacts such as hand-floor penetration. Applying physics-based tracking to refine the raw motion successfully produces more physically plausible results, improving joint-floor contact and eliminating penetration artifacts. This integration combines the versatile text-driven motion generation of DART with the physical accuracy provided by the physics-based simulation. Given the real-time capabilities of both DART and PHC, it is possible to leverage physics to correct the kinematic motion generated by DART on the fly and then use the corrected motions for subsequent online generation.



(a) Crawling sequence generated by DART

(b) Physics-based motion tracking result

Figure 5: We demonstrate an example of integrating DART with the physics-based motion tracking method PHC (Luo et al., 2023) to achieve more physically plausible motions. The left image illustrates a crawling sequence generated by DART, exhibiting artifacts such as hand-floor penetration. The right image displays the physics-based motion tracking outcome applied to the raw generated sequence, which enhances joint-floor contact and resolves the hand-floor penetration issue.

I DISCUSSION ON OPEN-VOCABULARY MOTION GENERATION

Limited vocabulary is a critical limitation and challenge shared by existing text-conditioned motion generation methods. Existing methods, including our approach DART, struggle to generalize to open-vocabulary text prompts due to the scarcity of 3D human motion data with text annotations. The scale of motion data available is several orders of magnitude smaller than that for text-conditioned image and video generation, primarily due to the reliance on marker-based motion capture systems, which are challenging to scale.

To expand the dataset and enable open-vocabulary generation, extracting human motion data from in-the-wild internet videos and generative image/video models (Kapon et al., 2024; Goel et al., 2023; Lin et al., 2023; Shan et al., 2024), is a promising direction. Additionally, the rapid advancement of vision-language models (VLMs) holds promise for automatically providing detailed, frame-aligned motion text labels to facilitate text-to-motion generation (Shan et al., 2024; Dai et al., 2023).

J ADDITIONAL COMPARISON FOR TEXT-CONDITIONED GOAL-REACHING

We include an additional comparison with GMD (Karunratanakul et al., 2023) for the text-conditioned goal-reaching task. While GMD supports text-conditioned trajectory control, it is an offline method and lacks the capability for online dynamic goal-reaching—a key advantage of our reinforcement learning-based approach. Moreover, GMD requires an additional time input for each waypoint to specify when the human should arrive at that point. In addition, GMD is limited to generating motion sequences with a maximum duration of 6 seconds and cannot reach locations beyond a threshold distance of approximately 5 meters. This restriction arises from its use of absolute translation representation and the reliance on motion data consisting of short sequences confined to areas near the origin. As a result, GMD can not be evaluated on the set of test paths used in the evaluations in Sec. 4.3.

Table 6: Comparison with GMD on the text-conditioned goal-reaching task. Arrows indicate whether lower (\downarrow) or higher (\uparrow) values are better.

Method	Time (s) \downarrow	Success Rate \uparrow	Skate (cm/s) \downarrow	Floor Distance (cm) \downarrow
GMD walk	6.00 ± 0.00	0.72 ± 0.11	3.87 ± 0.74	2.59 ± 0.20
Ours walk	9.03 ± 0.09	1.00 ± 0.00	2.21 ± 0.31	2.34 ± 0.10
GMD run	6.00 ± 0.00	0.56 ± 0.12	5.58 ± 1.05	2.67 ± 0.14
Ours run	6.18 ± 0.11	1.00 ± 0.00	4.12 ± 0.53	2.71 ± 0.11
GMD hop	6.00 ± 0.00	0.55 ± 0.12	3.90 ± 1.05	4.29 ± 0.20
Ours hop	7.65 ± 0.60	1.00 ± 0.00	2.47 ± 0.53	5.05 ± 0.22

Therefore, we compare our reinforcement-learning-based control with GMD on a shortened path where all waypoints are close enough to the origin. Additionally, for GMD, we uniformly distribute the waypoints over its 6-second motion duration. The quantitative evaluation results are presented in Tab. 6. GMD fails to consistently reach the final waypoint within the threshold distance of 0.3 meters. GMD also exhibits more foot skating than our method.

K COMPUTING RESOURCES

Our experiments and performance profiling are conducted on a workstation with single RTX 4090 GPU, intel i7-13700K CPU, 64GiB memory. The workstation runs with Ubuntu 22.04.4 LTS system.