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

Kaifeng Zhao, Gen Li, Siyu Tang
ETH Zürich
{kaifeng.zhao,gen.li,siyu.tang}@inf.ethz.ch

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 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 **DartC**ontrol, in short **DART**, a **D**iffusion-based **A**utoregressive motion primitive model for **R**eal-time **T**ext-driven motion **C**ontrol. Our model 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 motion 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. Video results and code are available at https://zkf1997.github.io/DART/.

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. These methods fail to accurately generate long and complex motions composed of multiple action segments, where each segment is conditioned on distinct action descriptions. FlowMDM (Barquero et al., 2024) is the state-of-the-art temporal motion composition method, capable of generating complex, continuous motions by composing desired actions with precise adherence to their specified 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 a joint trajectory, or interacting with objects, has 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 effectively balancing spatial control, motion quality, and semantic alignment with text. Moreover, these approaches are typically restricted to controlling isolated short motions in an offline setting.

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

To address these limitations, we propose **DART**, a diffusion-based autoregressive motion primitive model for real-time text-driven motion control. Moreover, the compact and expressive motion space of DART provides a foundation for integrating precise spatial control through latent space optimization or reinforcement learning (RL)-based policies. DART features three key components.

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

Next, DART learns a text-conditioned autoregressive motion generation model from large-scale data using a latent diffusion architecture, which contains a variational autoencoder for learning a compact latent motion primitive space, and a denoiser network for generating motion primitives conditioned on texts and history. Leveraging the trained denoiser and decoder models, DART employs an autoregressive rollout to synthesize motion sequences from real-time text inputs, enabling the efficient generation of motions of arbitrary length. Compared with the offline temporal motion composition method FlowMDM, DART provides real-time response and 10x generation speed.

Lastly, we introduce a latent space control framework based on DART for spatially controllable motion synthesis, leveraging its learned space of realistic human motions to ensure high-quality generation. We present effective optimization and learning algorithms that explore the latent diffusion noise space to synthesize motion sequences that precisely follow both textual and spatial constraints. We evaluate DART across various motion synthesis tasks, including generating long, continuous sequences from sequential text prompts, in-between motion generation, scene-conditioned motion, and goal-reaching synthesis. The experimental results show that DART is a simple, unified and highly effective motion model, consistently outperforming or matching the performance of baselines.

2 Related Works

Conditional Motion Generation. Generating realistic and diverse human motions is a long-standing challenge in computer vision and graphics. Apart from generating highly realistic human motions (Kovar et al., 2008; Holden et al., 2020; Clavet et al., 2016; Zinno, 2019), conditional generation is another important factor that aligns motion generation with human intentions and various application constraints. Text-conditioned motion generation (Tevet et al., 2023; Zhang et al., 2022a; Petrovich et al., 2022; Guo et al., 2022; Jiang et al., 2024a; Zhang et al., 2023a) has become increasingly popular since it allows users to modulate motion generation with flexible natural languages. Audio and speech-driven motion synthesis methods (Alexanderson et al., 2023; Tseng et al., 2023; Siyao et al., 2022; Ao et al., 2022; 2023) have also made significant progress recently. Moreover, there exist many applications that require spatial awareness and precise control in motion generation, such as interactive character control (Ling et al., 2020; Peng et al., 2022; Starke et al., 2022; Luo et al., 2024; Starke et al., 2024), human-scene interactions(Hassan et al., 2021; Starke et al., 2019; Zhao et al., 2022; 2023; Li et al., 2024a; Xu et al., 2023; Jiang et al., 2024b; Wang et al., 2024; Liu et al., 2024; Li et al., 2024b; Zhang et al., 2022b; 2025), and human-human(noid) interactions (Liang et al., 2024; Zhang et al., 2023c; Christen et al., 2023; Cheng et al., 2024; Shan et al., 2024). Synthesizing high-quality motions with precise spatial control remains challenging, and DART is a step toward a general and efficient motion model that supports precise control tasks.

Diffusion Generative Models. Denoising Diffusion Models (Ho et al., 2020; Song et al., 2021a;b) are generative models that learn to predict clean data samples by iteratively 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 of short, isolated motion sequences while neglecting the autoregressive nature of human motions (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). 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 explicit motion representations, 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 generation and capture (Xu et al., 2023; Jiang et al., 2024b; Van Wouwe et al., 2024; Han et al., 2024). 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 intuitive interface for users to control character behaviors. In contrast, DART is an efficient and general motion model that scales effectively to large motion-text datasets. DART supports natural language interfacing and provides a versatile foundation for various motion generation tasks with spatial control. Concurrent work, CloSD (Tevet et al., 2025), trains autoregressive motion diffusion models conditioned on target joint locations to guide the human motion towards these goals. This approach relies on paired training data of control signals and human motions. In contrast, DART learns a latent motion space from motion-only data and introduces latent space control methods to achieve flexible control goals without the need for paired training data.

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 $\mathbf{H}_{seed} = [\mathbf{h}^1, ..., \mathbf{h}^H]$, a sequence of N text prompts $C = [c^1, ..., c^N]$, and spatial goals g, the objective is to autoregressively generate continuous and realistic human motion sequences $\mathbf{M} = [\mathbf{H}_{seed}, \mathbf{X}^1, ..., \mathbf{X}^N]$, 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. Each motion primitive $\mathbf{P}^i = [\mathbf{H}^i, \mathbf{X}^i]$ is a short motion clip containing H frames of history motion $\mathbf{H}^i = [\mathbf{h}^{i,1}, ..., \mathbf{h}^{i,H}]$ that overlap with the previous motion primitive, and F frames of future motion $\mathbf{X}^i = [\mathbf{x}^{i,1}, ..., \mathbf{x}^{i,F}]$. 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 $\mathbf{M} = [\mathbf{H}_{seed}, \mathbf{X}^1, ..., \mathbf{X}^N]$. To represent human bodies in each motion frame, we use an overparameterized representation based on the SMPL-X (Pavlakos et al., 2019) parametric human body model. Each frame is represented as a $D = 276$ dimensional vector including the body root translation $\mathbf{t}$, root orientation $\mathbf{R}$, local joint rotations $\boldsymbol{\theta}$, joint locations $\mathbf{J}$, and the temporal difference features of locations and rotations. Each motion primitive is canonicalized in a local coordinate frame centered at the first-frame body pelvis. 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. For brevity, we omit the primitive index superscript i when discussing in the context of a single primitive.

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. Moreover, the autoregressive and simple nature of the prim-

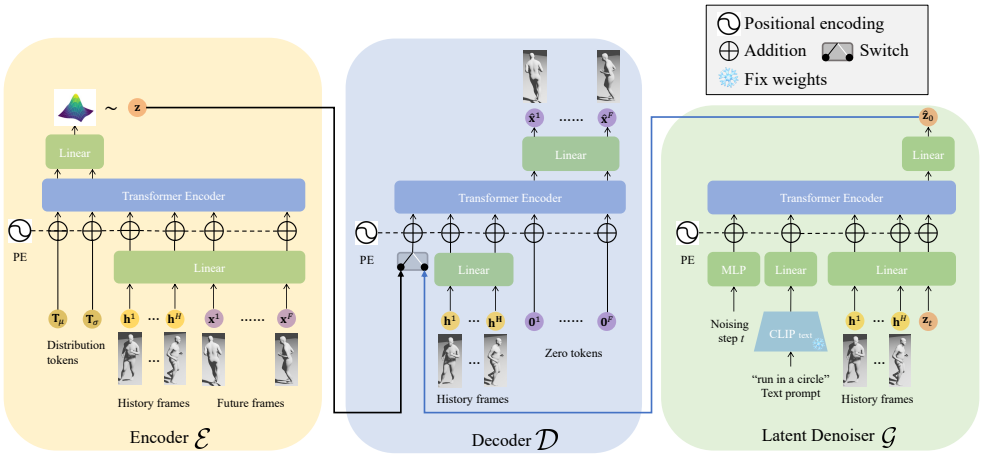


Figure 1: Architecture illustration of DART. The encoder network compresses the future frames $\mathbf{X} = [\mathbf{x}^1, ..., \mathbf{x}^F]$ into a latent variable, conditioned on the history frames $\mathbf{H} = [\mathbf{h}^1, ..., \mathbf{h}^H]$. The decoder network reconstructs the future frames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

itive 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.

3.2 DART: A Diffusion-Based Autoregressive Motion Primitive Model

We propose a latent diffusion model (Rombach et al., 2022; Chen et al., 2023b) designed for seamless autoregressive motion generation, 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 noise, conditioned on text prompts and motion history.

Learning the 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 (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 $\mathbf{T}_\mu$ and $\mathbf{T}_\sigma$, which are responsible for predicting the distribution mean and variance. The latent sample $\mathbf{z}$ 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 $\mathbf{z}$ 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, a small Kullback-Leibler (KL) regularization term L_{KL}, and additional SMPL-based reconstruction and regularization terms. We refer to Appendix C for further details about the motion primitive VAE.

Algorithm 1 Autoregressive rollout generation using latent motion primitive model

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

Latent Motion Primitive Diffusion Model. Building on the compact latent motion primitive space, we formulate text-conditioned autoregressive motion generation via a probabilistic distribution $q(\mathbf{z}|\mathbf{H}, c)$, and train a latent diffusion model $\mathcal{G}$ to approximate it. Diffusion models (Ho et al., 2020) are generative models that learn data distributions by progressively reversing a forward diffusion process, which iteratively adds Gaussian noise to data samples until they approach pure noise $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Given a motion primitive sampled from the dataset and its latent representation $\mathbf{z}_0$ obtained using the encoder $\mathcal{E}$, the forward diffusion produces a sequence of increasingly noisy samples $\mathbf{z}_1, \ldots, \mathbf{z}_T$ by iteratively adding noises as: $q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I})$, where β_t are noise schedule hyper parameters. The denoiser model learns the reverse process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t, t, \mathbf{H}, c) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ for generating motion primitives conditioned on the motion history $\mathbf{H}$ and text label c of the motion primitive. The vairance $\boldsymbol{\Sigma}_t$ of the reverse process distribution is scheduled using hyper parameters, while the mean $\boldsymbol{\mu}_t$ is modeled using a denoiser neural network. We design the denoiser model $\mathcal{G}$ to predict the clean latent variable $\hat{\mathbf{z}}_0 = \mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, c)$, from which $\boldsymbol{\mu}_t$ can be derived as follows:

$$\boldsymbol{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, c) + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{z}_t, \tag{1}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. During generation, we initialize with $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and use the denoiser model to predict the clean variable $\hat{\mathbf{z}}_0$, which is subsequently diffused to a lower noise level $\mathbf{z}_{T-1}$.This denoising process iterates until a clean sample $\mathbf{z}_0$ is obtained.

The denoiser model architecture is shown in Fig. 1. The diffusion step t is embedded using a small MLP, while the text prompt c 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. The cleaned latent variable $\hat{\mathbf{z}}_0$ can be converted back to the future frames using the frozen decoder $\mathcal{D}$: $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{H}, \hat{\mathbf{z}}_0)$. We train the latent denoiser $\mathcal{G}$ using the simple objective (Ho et al., 2020). Additionally, we apply the reconstruction loss L_{rec} and auxiliary losses L_{aux} on the decoded future frames $\hat{\mathbf{X}}$. Notably, we use only 10 diffusion steps for both training and inference. This small number suffices for realistic sample generation due to the simplicity of our motion primitive representation, enabling 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: $\mathcal{G}_w(\mathbf{z}_t, t, \mathbf{H}, c) = \mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, \emptyset) + w \cdot (\mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, c) - \mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, \emptyset))$. 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.

Algorithm 2 Latent noises optimization

Input: Latent noises $\mathbf{Z}_T = [\mathbf{z}_T^1, ..., \mathbf{z}_T^N]$, Optimizer $\mathcal{O}$, learning rate η, and goal g. (For brevity, we do not reiterate the inputs of the rollout function defined in Alg. 1 and criterion terms in Eq. 2)
Output: a motion sequence $\mathbf{M}$.
for $i \leftarrow 1$ **to** optimization steps **do**
  $\mathbf{M} \leftarrow \text{ROLLOUT}(\mathbf{Z}_T, \mathbf{H}_{seed}, C)$
  $\nabla \leftarrow \nabla_{\mathbf{Z}_T}(\mathcal{F}(\Pi(\mathbf{M}), g) + cons(\mathbf{M}))$
  $\mathbf{Z}_T \leftarrow \mathcal{O}(\mathbf{Z}_T, \nabla/\|\nabla\|, \eta)$ ▷ update using normalized gradient
end for
return $\mathbf{M} \leftarrow \text{ROLLOUT}(\mathbf{Z}_T, \mathbf{H}_{seed}, C)$

3.3 Spatially 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. 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 $cons(\cdot)$:

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

where g is the task-dependent spatial goal (e.g., a keyframe body for motion in-between tasks or a target location for navigation tasks), $\Pi(\cdot)$ is the projection function that extracts goal-relevant features from motion sequences and maps them into the task-aligned observation space, and $cons(\cdot)$ denotes physical and scene constraints, such as preventing scene collisions and floating bodies.

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, as it learns a generative model capable of producing diverse and realistic motions from standard Gaussian samples. Using the deterministic DDIM (Song et al., 2021a) sampler, we adapt DART sampling to function as a deterministic mapping from latent noises $\mathbf{Z}_T$ to plausible motions. This allows us to reformulate the motion control task in Eq. 2 as a latent space control problem as follows: Given the initial motion history $\mathbf{H}_{seed}$, a sequence of text prompts C, the pretrained DART models, and a deterministic diffusion sampler $\mathcal{S}$, the rollout function in Alg. 1 can deterministically map a list of motion primitive latent noises $\mathbf{Z}_T = [\mathbf{z}_T^1, .., \mathbf{z}_T^N]$ to a motion sequence conditioned on the history motion seed and text prompts: $\mathbf{M} = \text{ROLLOUT}(\mathbf{Z}_T, \mathbf{H}_{seed}, C)$. The minimization objective is converted as:

$$\mathbf{Z}_T{}^* = \text{argmin}_{\mathbf{Z}_T} \mathcal{F}(\Pi(\text{ROLLOUT}(\mathbf{Z}_T, \mathbf{H}_{seed}, C)), g) + cons(\text{ROLLOUT}(\mathbf{Z}_T, \mathbf{H}_{seed}, C)) \tag{3}$$

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. 3) is to directly optimize the latent noises $\mathbf{Z}_T$ 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.

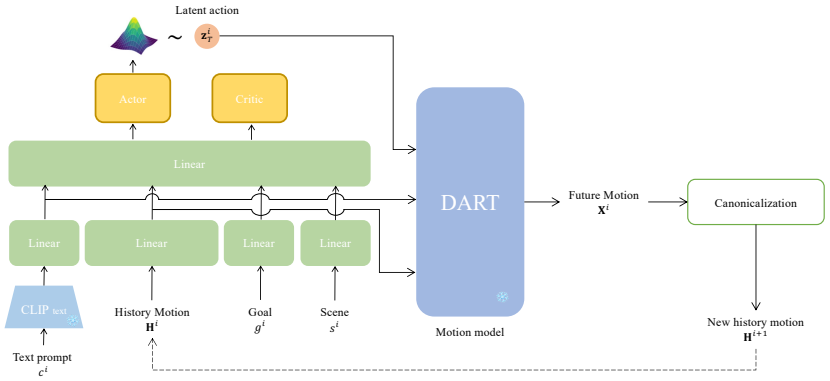


Figure 2: Architecture of the reinforcement learning-based control policy. The pretrained DART diffusion denoiser and decoder models transform the latent actions into motion frames. The last predicted frames are canonicalized and provided to the policy model as the next step history condition.

We use the distance between the f-th frame of the generated motion and the goal keyframe as 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 $\mathbf{s}^i$ of the system, samples an action $\mathbf{a}^i$ from the learned policy, with the system transitioning to the next state $\mathbf{s}^{i+1}$ due to the performed action $\mathbf{a}^i$, and receives a reward $r^i = r(\mathbf{s}^i, \mathbf{a}^i, \mathbf{s}^{i+1})$.

Our latent motion primitive model naturally fits in the Markov decision process due to its autoregressive nature. We use the latent noises $\mathbf{z}_T$ 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 $\mathbf{s}^i$ includes the history motion observation $\mathbf{H}^i$, goal observation g^i, scene observation s^i, and the CLIP embedding of the text prompt c^i. The policy model takes in [$\mathbf{H}^i$, g^i, s^i, c^i] to predict the latent noise $\mathbf{z}_T^i$ as the action. The latent noise $\mathbf{z}_T^i$ is mapped to the future motion frames $\mathbf{X}^i$ 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. 3 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 transformed into a local observation, which includes its distance to the body pelvis at the last history frame and its local direction within the human-centric coordinate frame. We consider a simple flat scene and the scene observation is the relative floor height to the body pelvis at the first history frame. 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). Qualitative results and comparisons are available in the supplementary videos on the project page.

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 human preference studies to evaluate 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 inlcuding 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 preference study results in Tab. 2. 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 '$\rightarrow$' denotes that closer to the dataset reference is better and '$\pm$' indicates the 95% confidence interval. **Bold** and blue texts indicate the best and second best results excluding the dataset, respectively.

	Segment				Transition				Profiling		
	FID$\downarrow$	R-prec$\uparrow$	DIV $\rightarrow$	MM-Dist$\downarrow$	FID$\downarrow$	DIV $\rightarrow$	PJ$\rightarrow$	AUJ $\downarrow$	Speed(frame/s)$\uparrow$	Latency(s)$\downarrow$	Mem.(MiB)$\downarrow$
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: Human preference 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 excluding the dataset are in **bold** and '$\pm$' indicates the 95% confidence interval.

	History error (cm)$\downarrow$	Goal error (cm)$\downarrow$	Skate (cm/s)$\downarrow$	Jerk$\downarrow$
Dataset	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	2.27 $\pm$ 0.00	0.74 $\pm$ 0.00
OmniControl	21.22 $\pm$ 2.86	7.79 $\pm$ 1.91	4.97 $\pm$ 1.31	1.41 $\pm$ 0.08
DNO	1.20 $\pm$ 0.20	4.24 $\pm$ 1.34	5.38 $\pm$ 0.70	0.65 $\pm$ 0.06
Ours	**0.00** $\pm$ 0.00	**0.59** $\pm$ 0.01	**2.98** $\pm$ 0.32	**0.61** $\pm$ 0.01

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.

(a) Walk, turn left, sit on the chair (b) Walk upstaris (c) Walk downstaris

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

By integrating DART with reinforcement learning-based control, we train text-conditioned goal-reaching policy models capable of three locomotion styles: 'walk', 'run', and 'hop on the left leg'. We evaluate goal-reaching on paths consisting of sequences of waypoints. We compare our method to a baseline GAMMA (Zhang & Tang, 2022), which also trains goal-reaching policies with a learned motion action space. Unlike DART, GAMMA lacks text conditioning and is limited to generating walking motions. 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 policy 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. Moreover, our text-conditioned goal-reaching policy achieves a generation speed of **240 frames per second**. These results demonstrate the potential of DART as a foundational human motion model, upon which versatile control models for various tasks can be learned through reinforcement learning.

Table 4: Quantitative evaluation of text-conditioned goal-reaching controller. The best results are in **bold** and '$\pm$' 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'	17.08 ± 0.05	$\mathbf{1.0} \pm 0.0$	2.67 ± 0.12	$\mathbf{2.24} \pm 0.02$
Ours 'run'	$\mathbf{10.55} \pm 0.06$	$\mathbf{1.0} \pm 0.0$	3.23 ± 0.24	3.86 ± 0.05
Ours 'hop on left leg'	20.50 ± 0.24	$\mathbf{1.0} \pm 0.0$	$\mathbf{2.22} \pm 0.12$	4.11 ± 0.07

5 Limitations and Conclusions

DART relies on motion sequences with frame-level aligned text annotations, as 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 causes semantic misalignment and ambiguity. We aim to explore hierarchical latent spaces to effectively tackle both fine-grained and global sequence-level semantics (Stoffl et al., 2024) in the future.

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.

Acknowledgements. We sincerely acknowledge the anonymous reviewers for their insightful feedback. We thank Korrawe Karunratanakul for helpful discussion and suggestions, and Siwei Zhang for meticulous proofreading. This work is supported by the SNSF project grant 200021 204840 and SDSC PhD fellowship.

REFERENCES

Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42 (4):1–20, 2023.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? a large-scale study. In *International Conference on Learning Representations*, 2021.

Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transactions on Graphics (TOG)*, 41(6):1–19, 2022.

Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.

Nikos Athanasiou, Mathis Petrovich, Michael J Black, and Gül Varol. TEACH: Temporal action composition for 3d humans. In *International Conference on 3D Vision (3DV)*, pp. 414–423, 2022.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 457–469, 2024.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. HumanMAC: Masked motion completion for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9544–9555, 2023a.

Rui Chen, Mingyi Shi, Shaoli Huang, Ping Tan, Taku Komura, and Xuelin Chen. Taming diffusion probabilistic models for character control. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10, 2024.

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

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive Whole-Body Control for Humanoid Robots. In *Proceedings of Robotics: Science and Systems*, 2024.

Sammy Christen, Wei Yang, Claudia Pérez-D'Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9654–9664, 2023.

Simon Clavet et al. Motion matching and the road to next-gen animation. In *Proc. of GDC*, volume 2, pp. 4, 2016.

Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–9, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Conference on Neural Information Processing Systems*, 2023.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. MotionLCM: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pp. 390–408, 2025.

Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, 2015.

Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa*, and Jitendra Malik*. Humans in 4D: Reconstructing and tracking humans with transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating Diverse and Natural 3D Human Motions from Text. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151, 2022.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. MoMask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024.

Bo Han, Hao Peng, Minjing Dong, Yi Ren, Yixuan Shen, and Chang Xu. AMD: Autoregressive motion diffusion. *AAAI*, 2024.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11374–11384, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*, pp. 1–4, 2015.

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

Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36, 2024a.

Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1737–1747, 2024b.

Roy Kapon, Guy Tevet, Daniel Cohen-Or, and Amit H Bermano. MAS: Multi-view ancestral sampling for 3d motion generation using 2d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1965–1974, 2024.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. GMD: Controllable human motion synthesis via guided diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, pp. 1–10, 2008.

Gen Li, Kaifeng Zhao, Siwei Zhang, Xiaozhong Lyu, Mihai Dusmanu, Yan Zhang, Marc Pollefeys, and Siyu Tang. EgoGen: An egocentric synthetic data generator. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a.

Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023.

Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision*, 2024b.

Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, pp. 1–21, 2024.

Jing Lin, Ailing Zeng, Shunlin Lu, Yuanhao Cai, Ruimao Zhang, Haoqian Wang, and Lei Zhang. Motion-X: A large-scale 3d expressive whole-body human motion dataset. *Advances in Neural Information Processing Systems*, 2023.

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character Controllers Using Motion VAEs. *ACM Transactions on Graphics*, 39(4), 2020.

Xinpeng Liu, Haowen Hou, Yanchao Yang, Yong-Lu Li, and Cewu Lu. Revisit human-scene interaction via space occupancy. In *European Conference on Computer Vision*, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, 2015.

Zhengyi Luo, Jinkun Cao, Alexander W. Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *International Conference on Computer Vision (ICCV)*, 2023.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *International Conference on Learning Representations*, 2024.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, 2019.

Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2891–2900, 2017.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10975–10985, 2019.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Trans. Graph.*, 41(4), 2022.

Mathis Petrovich, Michael J Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision*, pp. 480–497, 2022.

Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 722–731, 2021.

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

Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. HuMoR: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11488–11499, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Mengyi Shan, Lu Dong, Yutao Han, Yuan Yao, Tao Liu, Ifeoma Nwogu, Guo-Jun Qi, and Mitch Hill. Towards open domain text-driven synthesis of multi-person motions. In *European Conference on Computer Vision*, 2024.

Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. Interactive character control with auto-regressive motion diffusion models. *ACM Trans. Graph.*, 43(4), 2024. ISSN 0730-0301.

Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11050–11059, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019.

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

Sebastian Starke, Paul Starke, Nicky He, Taku Komura, and Yuting Ye. Categorical codebook matching for embodied character controllers. *ACM Transactions on Graphics (TOG)*, 43(4):1–14, 2024.

Lucas Stoffl, Andy Bonnetto, Stéphane d'Ascoli, and Alexander Mathis. Elucidating the hierarchical nature of behavior with masked autoencoders. In *European conference on computer vision*, 2024.

Richard S Sutton and Andrew G Barto. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

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

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H Bermano, and Michiel van de Panne. CLoSD: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, 2025.

Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 448–458, 2023.

Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott Delp, and C Karen Liu. DiffusionPoser: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2513–2523, 2024.

Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say, interact as you can: Language-guided human motion generation with scene affordance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. OmniControl: Control any joint at any time for human motion generation. In *International Conference on Learning Representations*, 2024.

Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14928–14940, 2023.

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics*, 37(4):1–11, 2018.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. T2M-GPT: Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023b.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:4115–4128, 2022a.

Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. COUCH: Towards controllable human-chair interactions. In *European Conference on Computer Vision (ECCV)*, 2022b.

Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Ilya A. Petrov, Vladimir Guzov, Helisa Dhamo, Eduardo Pérez Pellitero, and Gerard Pons-Moll. FORCE: Dataset and method for intuitive physics guided human-object interaction. In *International Conference on 3D Vision (3DV)*, 2025.

Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20481–20491, 2022.

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023c.

Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *European Conference on Computer Vision*, pp. 311–327, 2022.

Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14738–14749, 2023.

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

Fabio Zinno. Ml tutorial day: From motion matching to motion synthesis, and all the hurdles in between. *Proc. of GDC 2019*, 2, 2019.

A MOTION PRIMITIVE REPRESENTATION

Representation. We represent each frame of the motion primitive as a tuple of $(\mathbf{t}, \mathbf{R}, \boldsymbol{\theta}, \mathbf{J}, \mathrm{d}\mathbf{t}, \mathrm{d}\mathbf{R}, \mathrm{d}\mathbf{J})$, where $\mathbf{t} \in \mathbb{R}^3$ denotes the global body translation, $\mathbf{R} \in \mathbb{R}^6$ denotes the 6D rotation representation (Zhou et al., 2019) of the global body orientation, $\boldsymbol{\theta} \in \mathbb{R}^{21\times6}$ is the 6D representation of 21 joint rotations, $\mathbf{J} \in \mathbb{R}^{22\times3}$ denotes the 22 joints locations, $\mathrm{d}\mathbf{t} \in \mathbb{R}^3$ denotes the temporal difference with previous frame's translation, $\mathrm{d}\mathbf{R} \in \mathbb{R}^6$ denotes the 6D representation of the relative rotation between current and previous frame's body orientation, and $\mathrm{d}\mathbf{J} \in \mathbb{R}^{22\times3}$ denotes the temporal diffence between current and previous frame's joint locations.

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

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

Algorithm 3 Motion primitive rotation canonicalization

```
Input: right_hip ∈ ℝ^3, left_hip ∈ ℝ^3
x_axis = right_hip - left_hip
x_axis[2] = 0                                  ▷ Project to the xy plane
normalize(x_axis)
z_axis = [0, 0, 1]                             ▷ Inverse gravity direction
y_axis = cross_porudct(z_axis, x_axis)
normalize(y_axis)
return [x_axis, y_axis, z_axis]
```

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

B DATASETS

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

The BABEL dataset 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. We use the motion data at a framerate of 30 frames per second the same as prior works (Barquero et al., 2024; Athanasiou et al., 2022). During training, motion primitives are randomly sampled from data sequences and the text label is randomly sampled from all the action segments that overlap with the primitive. To alleviate the action imbalance in the BABEL dataset, we use the provided action labels to perform importance sampling during training so that the motion data of each action category has roughly equal sampling chances despite their varying frequency in the original dataset. To maintain compatibility with prior works, we fix the human gender to male and set the body shape parameter to zero.

The HML3D dataset contains short motions with coarse sequence-level sentence descriptions. We only use a subset of HML3D in training since its subset HumanAct12 and left-right mirroring motion sequences only provide joint locations instead of SMPL (Loper et al., 2015; Pavlakos et al., 2019) body sequences that our motion representation requires. The joint rotations used in the original HML3D are calculated using naive inverse kinematics and can not be directly used to animate human motions. We therefore only train using the subset where the SMPL body motion sequences is available. The HML3D motions are sampled at a framerate of 20 frames per second. During training, we randomly sample primitives with uniform probability and the text label is randomly chosen from one of the multiple sentence captions of the overlapping sequence. To maintain compatibility with prior works, we fix the human gender to male and set the body shape parameter to zero.

C MOTION PRIMITIVE VAE

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

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

$$L_{VAE} = L_{rec} + w_{KL} \times L_{KL} + w_{aux} \times L_{aux} + w_{SMPL} \times L_{SMPL}. \tag{4}$$

The reconstruction loss L_{rec} aims to minimize the distance between the reconstructed future frames $\hat{\mathbf{X}}$ and the ground truth future frames $\mathbf{X}$ as follows:

$$L_{rec} = \mathcal{F}(\hat{\mathbf{X}}, \mathbf{X}), \tag{5}$$

where $\mathcal{F}(\cdot, \cdot)$ denotes the distance function and we use the smoothed L1 loss(Girshick, 2015) in implementation.

The Kullback-Leibler divergence (KL) term L_{KL} penalizes the distribution difference between the predicted distribution and a standard Gaussian:

$$\mathcal{L}_{KL} = KL(q(\mathbf{z}|\mathbf{H}) \parallel \mathcal{N}(0, \mathbf{I})), \tag{6}$$

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

The auxiliary loss L_{aux} regularizes the predicted temporal difference features $\hat{\mathrm{d}}\cdot$ of translations, global orientation, and joints to be close to the actual temporal differences $\bar{\mathrm{d}}\cdot$ calculated from the predicted motion features:

$$L_{aux} = \mathcal{F}(\bar{\mathrm{d}}\mathbf{t}, \hat{\mathrm{d}}\mathbf{t}) + \mathcal{F}(\bar{\mathrm{d}}\mathbf{J}, \hat{\mathrm{d}}\mathbf{J}) + \mathcal{F}(\bar{\mathrm{d}}\mathbf{R}, \hat{\mathrm{d}}\mathbf{R}). \tag{7}$$

For instance, the difference of the first two frames of the predicted root translation ($\bar{\mathrm{d}}\mathbf{t}^0 := \hat{\mathbf{t}}^1 - \hat{\mathbf{t}}^0$) should be consistent with the predicted first frame root translation difference feature $\hat{\mathrm{d}}\mathbf{t}^0$. We use $w_{aux} = 100$ in our experiments.

The SMPL losses L_{SMPL} consist of two components: the SMPL-based joint reconstruction loss L_{joint_rec}, and the joint consistency loss $L_{consistency}$. The SMPL losses L_{SMPL} are defined as follows:

$$L_{SMPL} = L_{joint\_rec} + L_{consistency}. \tag{8}$$

The SMPL joint reconstruction loss L_{joint_rec} penalizes the discrepancy between the joints regressed from the predicted body parameters and those regressed from the ground truth body parameters:

$$L_{joint\_rec} = \mathcal{F}(\mathcal{J}(\hat{\mathbf{t}}, \hat{\mathbf{R}}, \hat{\boldsymbol{\theta}}), \mathcal{J}(\mathbf{t}, \mathbf{R}, \boldsymbol{\theta})), \tag{9}$$

where $\mathcal{J}$ denotes the SMPL body joint regressor that predicts joint locations given the body parameters. The SMPL body shape parameter β is assumed to be zero and ignored for simplicity here.

The joint consistency loss $L_{consistency}$ regularizes the predicted joint locations and predicted SMPL body parameters consistently represent the same body joints:

$$L_{consistency} = \mathcal{F}(\hat{\mathbf{J}}, \mathcal{J}(\hat{\mathbf{t}}, \hat{\mathbf{R}}, \hat{\boldsymbol{\theta}})). \tag{10}$$

We use $w_{SMPL} = 100$ in our experiments. We train the motion primitive VAE with the AdamW (Kingma & Ba, 2015) optimizer and the learning rate is set to $1e^{-4}$ with linear annealing.

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

$$L_{denoiser} = L_{simple} + w_{rec} \times L_{rec} + w_{aux} \times L_{aux} \tag{11}$$

$$L_{simple} = \mathbb{E}_{(\mathbf{z}_0, c) \sim q(\mathbf{z}_0, c),\, t \sim [1,T],\, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \mathcal{F}(\mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, c), \mathbf{z}_0), \tag{12}$$

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 $\hat{\mathbf{X}} = \mathcal{D}(\mathcal{G}(\mathbf{z}_t, t, \mathbf{H}, c))$ 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.

Algorithm 4 Calculate rollout probability

Input: current iteration number $iter$, number of train iterations in the first supervised stage I_1, number of train iterations in the second scheduled stage I_2
Output: rollout probability p
function ROLLOUT_PROBABILITY($iter$, I_1, I_2)
    if $iter \leq I_1$ **then** ▷ no rollout in the first supervised stage
        $p \leftarrow 0$
    else if $iter > I_1 + I_2$ **then** ▷ the third rollout stage always use rollout
        $p \leftarrow 1$
    else ▷ linearly scheduled rollout probability in the second scheduled stage
        $p \leftarrow \frac{iter - I_1}{I_2}$
    end if
    return p
end function

We use $w_{rec} = 1$ and $w_{aux} = 10000$ in our experiments. We do not use the SMPL losses L_{SMPL} in training the denoiser model because the SMPL body model inference process slows down the training. 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 (Athanasiou et al., 2022), 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 HUMAN PREFERENCE STUDIES.

We conduct human preference studies to quantitatively compare our method DART with baselines to provide a more comprehensive and convincing evaluation. We run human preference 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

Algorithm 5 Scheduled training for the latent denoising model

Input: pretrained motion primitive decoder $\mathcal{D}$ and encoder $\mathcal{E}$, latent variable denoiser $\mathcal{G}_\theta$ parameterized by θ, total diffusion steps T, optimizer $\mathcal{O}$, loss criterion $\mathcal{L}$, train dataset $\mathcal{X}$.
Scheduled training parameters: the number of train iterations in the first supervised stage I_1, the number of train iterations in the second scheduled stage I_2, the number of train iterations in the third rollout stage I_3, the maximum number of primitive rollouts N during training.

$I_{total} \leftarrow I_1 + I_2 + I_3$
$iter \leftarrow 0$
while $iter < I_{total}$ **do**
  $[\mathbf{H}_{seed}, \mathbf{X}^1, c^1, ..., \mathbf{X}^N, c^N] \sim \mathcal{X}$
    ▷ sample N consecutive motion primitives with text labels from dataset $\mathcal{X}$
  $\mathbf{H} \leftarrow \mathbf{H}_{seed}$ ▷ initialize motion history
  for $i \leftarrow 1$ **to** N **do** ▷ number of rollouts
    $\mathbf{z}_0^i = \mathcal{E}(\mathbf{H}, \mathbf{X}^i)$ ▷ compress motion primitive into latent space
    $t \sim \mathcal{U}[0, T)$ ▷ sample diffusion step t
    $\mathbf{z}_t^i \leftarrow$ FORWARD_DIFFUSION$(\mathbf{z}_0^i, t)$
    $\hat{\mathbf{z}}_0^i = \mathcal{G}_\theta(\mathbf{z}_t^i, t, \mathbf{H}, c^i)$ ▷ latent denoising model prediction
    $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{H}, \hat{\mathbf{z}}_0^i)$ ▷ decode predicted latent variable to future motion frames
    $\nabla \leftarrow \nabla_\theta \mathcal{L}(\mathbf{z}_0^i, \hat{\mathbf{z}}_0^i, \mathbf{H}, \mathbf{X}^i, \hat{\mathbf{X}}^i)$ ▷ model parameter gradient calculation
    $\theta \leftarrow \mathcal{O}(\theta, \nabla)$ ▷ model update using optimizer

    $p \leftarrow$ ROLLOUT_PROBABILITY$(iter, I_1, I_2)$
      ▷ update history motion using predicted or GT motion by a scheduled probability
    if rand() $< p$ **then** ▷ use predicted rollout history
      $\mathbf{z}_T^i \leftarrow$ FORWARD_DIFFUSION$(\mathbf{z}_0^i, T)$ ▷ maximum noising simulating inference time
      $\hat{\mathbf{z}}_0^i =$ DDPM_SAMPLE$(\mathcal{G}_\Theta, \mathbf{z}_T^i, T, \mathbf{H}, c^i)$ ▷ full DDPM denoising loop
      $\hat{\mathbf{X}} =$ SG$(\mathcal{D}(\mathbf{H}, \hat{\mathbf{z}}_0^i))$ ▷ decode predicted latent variable and stop gradient
      $\mathbf{H} \leftarrow$ CANONICALIZE$(\hat{\mathbf{X}}^{F-H+1:F})$
    else ▷ use ground truth history
      $\mathbf{H} \leftarrow$ CANONICALIZE$(\mathbf{X}^{F-H+1:F})$
    end if

    $iter \leftarrow iter + 1$
  end for
end while

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.

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 (AUJ) 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.

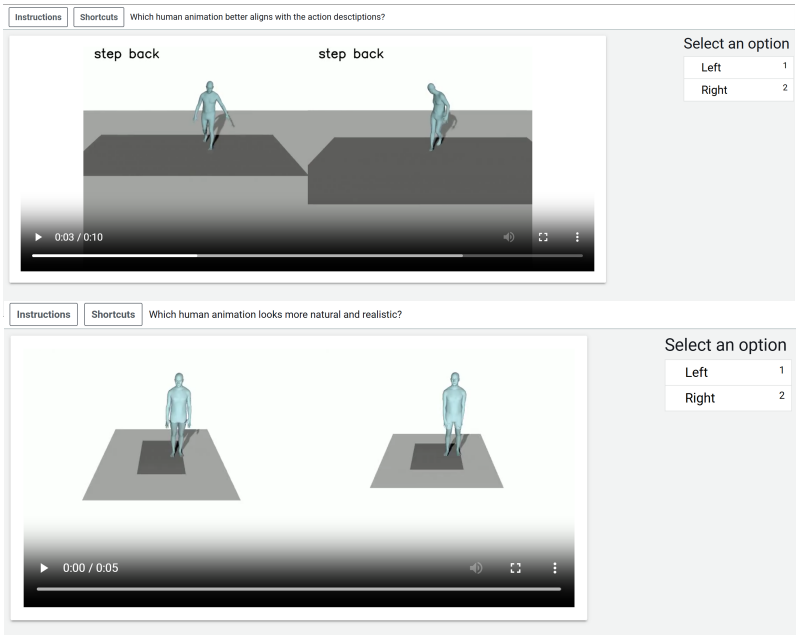


Figure 4: Illustration of the human preference study interface for evaluating motion-text semantic alignment (top) and perceptual realism(bottom). Participants are requested to select the generation that is perceptually more realistic or better aligns with the action descriptions in subtitles (only visible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

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$\rightarrow$	AUJ $\downarrow$
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
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
DART-VAE	4.23±0.02	0.62±0.01	8.33±0.12	5.29±0.01	1.79±0.02	6.73±0.23	0.20±0.00	0.96±0.00
DART-schedule	8.08±0.09	0.39±0.01	8.05±0.12	6.96±0.03	7.41±0.10	6.58±0.06	0.03±0.00	0.18±0.00
per frame(H=1,F=1)	10.31±0.09	0.29±0.01	6.82±0.13	7.41±0.01	7.82±0.09	6.03±0.08	0.02±0.00	0.08±0.00
H=2,F=16	4.04±0.10	0.66±0.00	8.20±0.06	4.96±0.01	2.22±0.10	6.60±0.20	0.06±0.00	0.18±0.00
steps 2	4.44±0.04	0.60±0.00	8.20±0.15	5.38±0.01	2.24±0.02	6.77±0.07	0.05±0.00	0.20±0.00
steps 5	3.49±0.09	0.63±0.00	8.25±0.15	5.18±0.01	2.11±0.07	6.74±0.11	0.05±0.00	0.20±0.00
steps 8	3.70±0.03	0.62±0.01	8.04±0.13	5.25±0.03	2.15±0.08	6.72±0.15	0.06±0.00	0.20±0.00
steps 10 (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
steps 50	3.82±0.05	0.60±0.00	7.74±0.07	5.30±0.01	2.11±0.10	6.58±0.10	0.06±0.00	0.22±0.00
steps 100	4.16±0.06	0.61±0.00	7.82±0.15	5.32±0.02	2.20±0.05	6.43±0.10	0.06±0.00	0.21±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 $\mathbf{Z}_T$ for 100 to 300 steps. The learning rate is linearly annealed to 0 and the gradient is normalized to stabilize optimization. The latent noises are initialized by sampling from Gaussian distributions. Empirically, we observe that initializing these latent noises with a small variance, such as 0.01, slightly improves the jerk and skate metrics at the cost of reduced motion diversity in the in-between experiment, compared to a unit variance initialization. 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 that collide into scenes and exhibit negative SDF values in each frame:

$$cons\_coll(\mathbf{J}) = -\sum_{k=1}^{22}(\Psi(\mathbf{J}^k) - \boldsymbol{\tau}^k)_- \tag{13}$$

where $\mathbf{J}^k$ denotes the location of the k-th joint in the scene coordinates, $\Psi(\cdot)$ denotes the signed distance function returning the signed distance from the query joint location to the scene, $\boldsymbol{\tau}$ denotes the joint-dependent contact threshold values, which are determined by the joint-skin distance in the rest pose, and $(\cdot)_-$ 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 in every frame and is defined as:

$$cons\_cont(\mathbf{J}) = (\min_{k \in Foot} \Psi(\mathbf{J}^k) - \tau)_+ \tag{14}$$

where $Foot$ denotes the set of foot joint indices, τ denotes the foot-floor contact threshold distance, and $(\cdot)_+$ 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.

$$r_{dist} = d^{i-1} - d^i \tag{15}$$

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.

$$r_{succ} = \begin{cases} 1 & \text{if } d^i < 0.3 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

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.

$$r_{ori} = \frac{\langle \mathbf{p}^i - \mathbf{p}^{i-1}, g - \mathbf{p}^{i-1} \rangle + 1}{2}, \tag{17}$$

The moving orientation reward encourages the moving orientation to be aligned with the goal orientation, where $\mathbf{p}^i$ 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.

$$r_{skate} = -disp \cdot (2 - 2^{h/0.03}), \tag{18}$$

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.

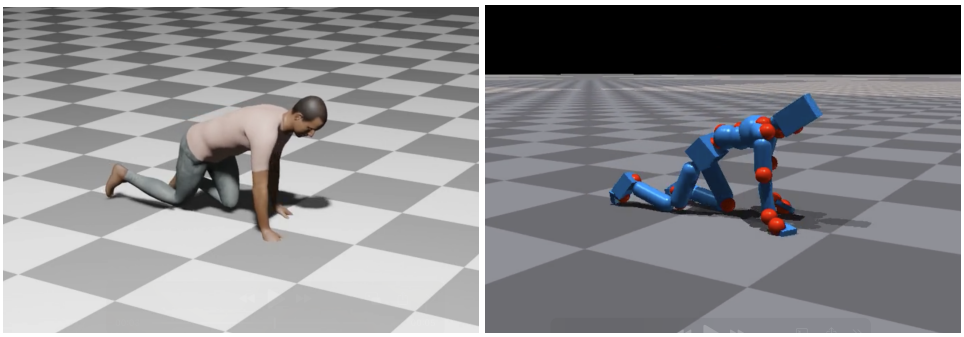
$$r_{floor} = -(|lf| - 0.03)_+, \tag{19}$$

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 $w_{dist} = 1$, $w_{succ} = 1$, $w_{ori} = 0.1$, $w_{skate} = 100$, $w_{floor} = 10$ for hopping and $w_{floor} = 100$ for walking and running.

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 OF LONG ROLLOUT RESULTS GIVEN A SINGLE TEXT PROMPT

Our DART can autoregressively generate perpetual rollouts of actions that are inherently repeatable and extendable. For example, DART can produce minutes-long sequences of continuous human motion, such as jogging in circles, performing cartwheels, or dancing. These actions are inherently repeatable, and such extensions are also represented in the AMASS (Mahmood et al., 2019) dataset. DART can stably generate minutes-long rollouts of the same action, enabled by its autoregressive motion primitive modeling and scheduled training scheme.

Some other actions, however, have inherent boundary states that mark the completion of the action. For instance, "kneel down" reaches a boundary state where the knees achieve contact with the floor. Further extrapolation of "kneel down" beyond this boundary state is not represented in the dataset and is not intuitively anticipatable by humans, as no further motion logically extends within the action semantics. Continuing rollout using the "kneel down" text prompt results in motions exhibiting fluctuations around the boundary state.

In summary, long rollout results given a single text prompt will repeat naturally or fluctuate around a boundary state, depending on the inherent nature of the action and its representation in the dataset. We provide video results of minutes-long rollout generation results on our project website.

J 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).

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.