UniPhys: Unified Planner and Controller with Diffusion for Flexible Physics-Based Character Control

Appendix

In this appendix, we include further implementation details for both training and inference in Sec.A. In Sec.B, we detail the loss design for tasks utilizing loss-based guided sampling. Sec. C presents the user study design, interface, and complete results on text-to-motion alignment evaluation. Finally, we discuss the limitations of our approach and potential directions for future work.

A. Implementation Details

Architecture. The diffusion model is build with a 12-layer causal transformer decoder with a hidden size of 768. The input is a sequence with 32 frames, and the per-frame input feature includes the 32-dim latent action embedding and the 366-dim state representation.

Training details. During training, we divide motion sequences into 32-frame clips with a stride of 8. If a clip contains multiple text annotations, we randomly select one for training. To improve transition smoothness between different skills, we preprocess the annotations by removing "transition to" and assigning the annotation of transition-phase motion to the target motion.

We train the model with a batch size of 1024, a learning rate of 1.5×10^{-4}, 10k warm-up steps, and cosine learning rate decay. The model undergoes training with 50 denoising steps, taking approximately 10 GPU days on a single RTX A100 over 15k epochs. Despite only a minor decrease in loss as training goes on, we still observe continuous improvements in policy stability and motion-semantic fidelity.

Inference details. At inference time, we use DDIM sampling with 5 steps and apply the stabilization trick across all applications.

(a) Text-Driven Control Policy: We empirically find that a small stabilization noise level (1, 2, or 3) is sufficient for achieving stable long-horizon control, whereas increasing it further to 5 degrades stability. Therefore, we use a stabilization noise level of 3 for all text-driven control experiments.

(b) Loss-Based Guided Applications: For challenging tasks that utilize loss-based guidance, we observe that increasing the stabilization noise level helps stabilize the guided denoising process. Intuitively, a strong task-specific guidance signal may cause the denoised states to drift slightly out of distribution, and a higher stabilization noise level mitigates this effect.

Moreover, we employ Monte Carlo guidance by estimating the gradient from multiple samples to reduce gradient variance and stabilize the guided optimization process. Without Monte Carlo guidance, the optimization tends to be unstable, resulting in a low task success rate.

MC Samples	N=1	N=3	N=5
Succ. Rate	26%	82%	98%
FPS	9.2	8.9	8.7

Table A.1. Ablation on the effect of Monte-Carlo Guidance (MCG) on loss-based guided sampling for goal reaching task.

We analyze the effect of Monte Carlo guidance on the goal-reaching task in Table A.1. With just 2 Monte Carlo samples, the success rate significantly improves from 26% to 82%. Increasing the number of samples to 5 further enhances performance, though at the cost of slightly reduced planning efficiency.

B. Loss-guided sampling design details

Goal reaching. To facilitate this goal reaching process, we design a loss function that encourages the predicted joint position to be close to the target goal. Furthermore, to expedite goal achievement, we incorporate an orientation loss that encourages the character to orient itself toward the goal. Specifically, the loss function is defined as follows:

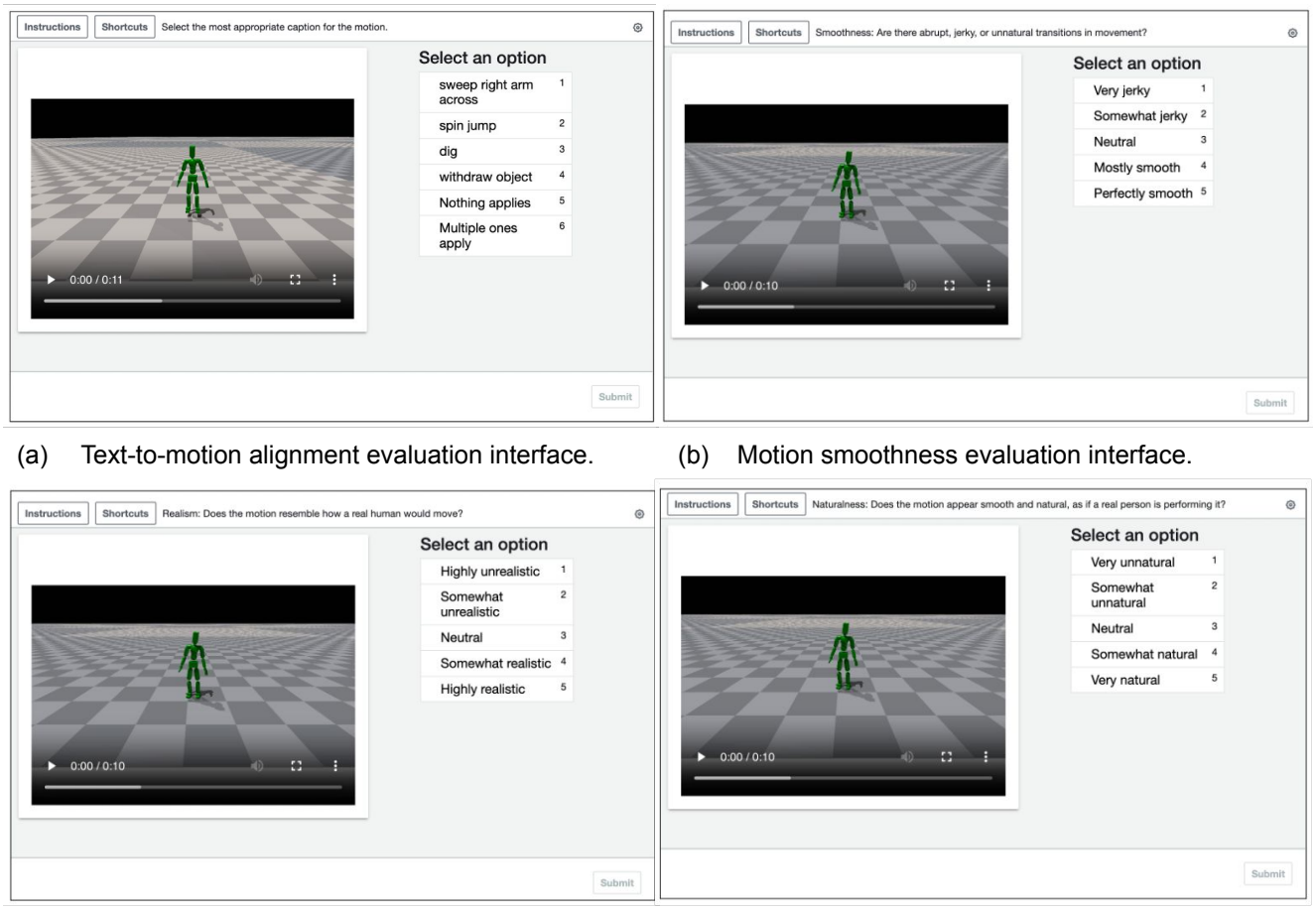
$$\mathcal{G}(\hat{\mathbf{X}}) = \sum_{i=t+1}^{t+H} (w_1 * |\hat{\mathbf{p}}_i - \mathbf{p}^g| + w_2 * (1 - cos < \hat{\phi}_i, \mathbf{p}^g - \hat{\mathbf{p}}_i >)) \quad (1)$$

where $\mathbf{p}$ and $\mathbf{p}^g$ are the joint position and goal position, respectively, and ϕ is the character root orientation, and w_1, w_2 adjust the strength of position guidance and orientation guidance.

Velocity Control. For velocity control, we apply losses the speed magnitude, the steering direction and also the orientation direction to align the character's orientation with the target velocity. The loss function is formulated as follows:

$$\mathcal{G}(\hat{\mathbf{X}}) = \sum_{i=t+1}^{t+H} (w_1 \left| \|\mathbf{v}_t\| - \|\mathbf{v}^g\| \right|^2 + w_2 \left(1 - \cos\theta_v\right) + w_3 \left(1 - \cos\theta_o\right)), \quad (2)$$

where $\mathbf{v}_t, \mathbf{v}^g$ is the predicted velocity and the target velocity respectively, and θ_v is the angle between $\mathbf{v}_t$ and $\mathbf{v}^g$, and θ_o is



(a) Text-to-motion alignment evaluation interface.

(b) Motion smoothness evaluation interface.

(c) Motion realism evaluation interface.

(d) Motion naturalness evaluation interface.

Figure C.1. User study interface on the Amazon Mechanical Turk (AMT).

the angle between the character's orientation and $\mathbf{v}^g$, ensuring the agent faces the movement direction, and w_1, w_2, w_3 balances the guidance strength of each term.

Dynamic Obstacle Avoidance. We employ a smooth SDF-based loss with softplus smoothing, and for SDF computing simplicity, we adopt for the sphere-like obstacle, and the guidance loss is designed as follows,

$$\mathcal{G}(\hat{\mathbf{X}}) = \sum_{i=t+1}^{t+H} \log(1 + e^{-(d_i - r - 1)}) \tag{3}$$

where d_i is the distance between the character's root and obstacle's center in XY plane, and r is the radius of the obstacle.

C. User study interface and more results

We conduct two user studies on Amazon Mechanical Turk to evaluate motion semantic fidelity and motion quality separately.

For motion semantic fidelity, we follow the evaluation protocol from SuperPADL [1]. Raters are presented with four options per motion (three distractors and one ground truth) and can also select "Nothing applies" or "Multiple ones apply" to account for annotation ambiguity. To ensure fair comparisons between our method and baselines, we use the same text prompts for motion generation and provide identical answer choices for each motion.

For motion quality, we ask raters to assess naturalness, smoothness, and realism to make the results more interpretable. All motions are initialized with a standing pose, and we ask 3 independent raters to rate each motion. The user study interface is shown in Fig. C.1, and in Table C.1, we present the complete user study results on the text-to-motion alignment evaluation.

User Response	Ours	CLoSD	MM
Correct	56.3%	61.6%	42.9%
Wrong	14.2%	8.6%	16.6%
Nothing applies	23.8%	21.7%	35.1%
Multi apply	5.6%	7.9%	5.3%
Any Correct	92.7%	94.5%	79.3%
Majority Correct	52.0%	65.3%	34.6%
All Correct	24.0%	25.1%	14.6%

Table C.1. Complete user study results on the text-to-motion semantic alignment evaluation.

D. Limitations and future work

Inference inefficiency is a common limitation of diffusion-based frameworks, making our method less efficient than RL-based policies. For text-driven control, our framework operates at approximately 10 FPS with autoregressive denoising and 18 FPS with gradual denoising. However, improving inference efficiency was not the primary focus of this work. Recent advancements in diffusion-based kinematic motion generation [2, 3] have demonstrated real-time interactive motion generation. We believe that further optimizations in diffusion model inference could enable our framework to be applied to high-frequency, real-time control tasks.

While our model demonstrates robust control, balance loss still occurs, particularly during highly dynamic actions or due to poor timing in changing text instructions, leading to skill transition failures and falls. Completely avoiding falls is unrealistic due to the inherent challenges in bipedal control tasks. Moving forward, we plan to incorporate a fall recovery skill by collecting expert demonstrations on getting up from the ground and leveraging an RL policy specifically trained for this task to enhance the expert demonstration data collection.

Another interesting capability for physics-based character control is traversing different terrains, which is crucial for real-world applications, such as robotics. Due to the lack of terrain-specific data, achieving this under a behavior cloning framework is not immediately feasible. However, reinforcement learning-based policies can serve as a valuable data generator for unseen scenarios, making it possible to explore the potential of behavior cloning in this context.

Lastly, our current approach does not incorporate dexterous hand control for the character, limiting its application in tasks like human-object interaction. However, our framework can be extended to full-body character control, including hand dexterity.

References

[1] Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. Superpadl: Scaling language-directed physics-based control with progressive supervised distillation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[2] 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. *arXiv preprint arXiv:2410.03441*, 2024. 3

[3] Kaifeng Zhao, Gen Li, and Siyu Tang. A diffusion-based autoregressive motion model for real-time text-driven motion control. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*, 2025. 3