Spherical Latent Motion Prior for Physics-Based Simulated Humanoid Control

Jing Tan [*1] **Weisheng Xu** [*1] **Xiangrui Jiang** [*1] **Jiaxi Zhang** [1] **Kun Yang** [1] **Kai Wu** [1] **Jiaqi Xiong** [2]
Shiting Chen [1] **Yangfan Li** [1] **Yixiao Feng** [1] **Yuetong Fang** [1] **Yujia Zou** [1] **Yiqun Song** [1] **Renjing Xu** [†1]

Project page: https://colin-jing.github.io/SLMP/

Abstract

Learning motion priors for physics-based humanoid control is an active research topic. Existing approaches mainly include variational autoencoders (VAE) and adversarial motion priors (AMP). VAE introduces information loss, and random latent sampling may sometimes produce invalid behaviors. AMP suffers from mode collapse and struggles to capture diverse motion skills. We present the Spherical Latent Motion Prior (SLMP), a two-stage method for learning motion priors. In the first stage, we train a high-quality motion tracking controller. In the second stage, we distill the tracking controller into a spherical latent space. A combination of distillation, a discriminator, and a discriminator-guided local semantic consistency constraint shapes a structured latent action space, allowing stable random sampling without information loss. To evaluate SLMP, we collect a two-hour human combat motion capture dataset and show that SLMP preserves fine motion detail without information loss, and random sampling yields semantically valid and stable behaviors. When applied to a two-agent physics-based combat task, SLMP produces human-like and physically plausible combat behaviors only using simple rule-based rewards. Furthermore, SLMP generalizes across different humanoid robot morphologies, demonstrating its transferability beyond a single simulated avatar.

1. Introduction

Physics-based simulated humanoid control has made notable progress in recent years, enabling simulated humanoids to imitate large-scale motion capture datasets, perform agile movements (Peng et al., 2018; Luo et al., 2021; 2023; Tessler et al., 2024; Zhang et al., 2025; Jing et al., 2025), and interact with the environment (Gao et al., 2024; Pan et al., 2025; Wang et al., 2025). These advances benefit applications in graphics, gaming, virtual reality, and robotics, where physically plausible human motion is essential. Despite these gains, controlling high-dimensional humanoids through raw joint torques remains challenging due to unstable dynamics and the sensitivity of low-level control to reward design (Luo et al., 2024a). To mitigate these issues, many recent methods introduce motion priors that constrain actions to physically plausible and coordinated motion manifolds, thereby simplifying policy learning for downstream tasks (Peng et al., 2022; Tessler et al., 2023; Luo et al., 2024a;b; Tessler et al., 2024).

A key challenge in learning motion priors is achieving both diversity and validity. VAE–based (Kingma & Welling, 2013) approaches (Won et al., 2021; Zhu et al., 2023; Luo et al., 2024a) compress full-body motion trajectories into latent codes, but the bottleneck often discards important motor details, and random latent sampling can produce implausible behaviors or falls. AMP-based (Peng et al., 2021) approaches (Peng et al., 2022; Tessler et al., 2023) incorporate adversarial reward shaping, where a discriminator (Goodfellow et al., 2020) distinguishes real from simulated motion and nudges the policy toward the motion data distribution. However, these methods often suffer from mode collapse and limited semantic coverage, making them difficult to scale to diverse motion datasets. As a result, existing priors struggle to support broad motion repertoires, stable random sampling, and reliable hierarchical reuse.

These limitations arise from structural properties of the underlying formulations. VAE-based priors instead compress high-dimensional, multi-modal human motions through a Gaussian latent bottleneck, which encourages a unimodal latent distribution and blurs fine-grained variations, while leaving large regions of latent space with little training signal so that random sampling may produce implausible actions. AMP-based priors rely on reinforcement learning under adversarial rewards, which couples high-variance policy gradients with discriminator feedback that emphasizes a few

[*]Equal contribution [1]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China [2]University of Oxford, Oxford, United Kingdom. Correspondence to: Renjing Xu <renjingxu@hkust-gz.edu.cn>.

Preprint. March 1, 2026.

high-reward behaviors. This often drives the policy toward a small set of modes rather than a well-covered motion repertoire. Since humanoid motion repertoires are inherently multi-modal and state-dependent, a suitable latent space should reflect meaningful local structure for control while avoiding excessive information compression and unbounded low-density regions that destabilize random sampling.

Motivated by these requirements, we propose the Spherical Latent Motion Prior (SLMP), a two-stage method that constructs a structured latent action space for physics-based humanoids. In the first stage, we train a high-quality motion tracking controller. In the second stage, we distill this controller into a unit-sphere latent space by conditioning the policy on latent codes. The distillation objective combines three components: imitation distillation from the expert controller, a discriminator loss that distinguishes expert from non-expert actions, and a discriminator-guided local semantic consistency loss that shapes coherent neighborhoods on the sphere. This formulation preserves motor detail without a reconstruction bottleneck while enabling stable and diverse random sampling.

To evaluate SLMP, we collect a two-hour combat motion capture dataset spanning strikes, evasions, transitions, and footwork. Experiments show that SLMP preserves fine motion detail, exhibits stronger random sampling performance than VAE-based and AMP-based baselines, and produces state-dependent latent neighborhoods that encode feasible action sets. When used as a prior in a two-agent physics-based combat task, SLMP enables simple high-level policies to generate physically plausible and diverse behaviors using lightweight reward signals. Finally, we demonstrate that the same pipeline generalizes to real humanoid robot platforms. Our contributions are summarized as follows:

- We collect and will release a two-hour human combat motion capture dataset, featuring diverse striking, evasion, and footwork behaviors suitable for physics-based humanoid control research.
- We propose SLMP, a two-stage distillation framework that constructs a structured spherical latent action space for humanoids, enabling stable and diverse latent-conditioned control.
- Through extensive experiments, we demonstrate that SLMP enables physically plausible two-agent combat using simple rule-based reward functions and generalizes across different humanoid robot morphologies.

2. Related Work

2.1. Physics-based Humanoid Motion tracking.

Early physics-based systems such as DeepMimic (Peng et al., 2018) demonstrated that reinforcement learning can track reference motion capture clips within a simulated environment. UHC (Luo et al., 2021) enabled general-purpose full-body tracking across AMASS (Mahmood et al., 2019) sequences, while PHC (Luo et al., 2023) removed external force modules and achieved more stable tracking on large-scale datasets. PHC+ (Luo et al., 2024a) further refined training to obtain near-perfect tracking for everyday motions, and MaskedMimic (Tessler et al., 2024) introduced masked motion inpainting to support versatile conditioning modes in a single tracking model. Most recently, FARM (Jing et al., 2025) focused on high-dynamic motions and combined frame-accelerated augmentation with a residual mixture-of-experts architecture to better handle rapid pose transitions in physics-based tracking.

2.2. Physics-based Humanoid Motion Latent Space

Latent motion spaces for physics-based control mainly follow two design families. AMP-based methods such as ASE (Peng et al., 2022) learn adversarial skill embeddings on a spherical manifold for downstream control, while CALM (Tessler et al., 2023) learns conditional adversarial latent models for controllable character manipulation. These approaches support skill composition but often exhibit limited latent coverage and sampling stability on highly diverse datasets. VAE-based methods instead learn reconstruction-driven motion priors. PULSE (Luo et al., 2024a) and MaskedMimic (Tessler et al., 2024) adopt VAE-style latent action models for character control, while Neural Categorical Priors (Zhu et al., 2023) use VQ-VAE-style (Van Den Oord et al., 2017) discrete latents to encode behavioral modes. However, reconstruction-based objectives can introduce information bottlenecks and reduce sampling reliability. In contrast, our method learns a continuous spherical latent space through controller distillation, focusing on stable and semantically meaningful sampling for physics-based humanoid control.

3. Method

Our method follows the pipeline shown in Figure 1. We first collect a two-hour human combat mocap dataset and convert it into SMPL (Loper et al., 2015) motion clips. Based on these clips, we train a goal-conditioned expert controller π_{track} via motion tracking (Section 3.1). We then distill this expert into a spherical latent space, using a discriminator and our discriminator-guided local semantic consistency loss to shape the latent manifold (Section 3.2). Finally, we use SLMP as a structured low-level prior in a two-agent combat task. A high-level policy outputs latent codes that drive the SLMP controller, while self-play (Silver et al., 2017; Berner et al., 2019) optimizes these latent decisions using simple rule-based rewards (Section 3.3).

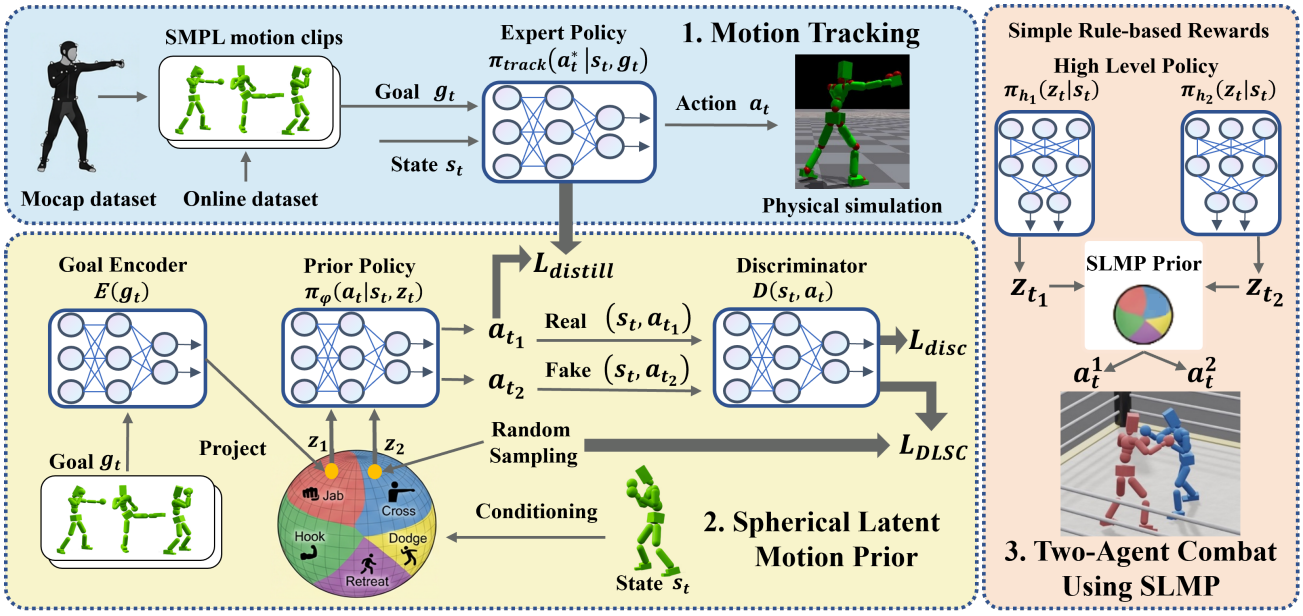


Figure 1. Overview of the Spherical Latent Motion Prior (SLMP). We collect a two-hour combat motion capture dataset and convert it into SMPL motion clips for training. We train a goal-conditioned motion tracking controller, then distill it into a unit-sphere latent space using three losses: imitation, discriminator, and our discriminator-guided local semantic consistency loss (L_{DLSC}). SLMP supports meaningful random sampling and drives downstream tasks such as two-agent combat via simple rewards.

3.1. Motion Tracking via Goal-Conditioned Reinforcement Learning

We train a goal-conditioned motion tracking controller π_{track} that maps the humanoid's proprioceptive state and a reference-driven goal to per-joint target poses. This formulation follows the goal-conditioned reinforcement learning (GCRL) framework widely used in large-scale humanoid controllers. At each timestep t, the policy observes the humanoid state s_t together with a goal g_t derived from the SMPL reference motion. The policy outputs PD target angles a_t for each actuated joint, and joint torques are applied by the simulator through a PD controller. We optimize π_{track} using PPO (Schulman et al., 2017) to maximize a tracking reward that encourages physical imitation of the reference motion. The reward includes exponential matching terms for joint positions, joint orientations, linear velocities, and angular velocities, together with an energy penalty that discourages excessive torque expenditure. After training, π_{track} achieves reliable tracking performance on our combat motion dataset and serves as the expert controller for our latent distillation stage.

3.2. Spherical Latent Motion Prior (SLMP)

We learn a latent-conditioned motion prior that distills π_{track} into a spherical latent space. The training procedure of SLMP is summarized in Algorithm 1. The prior policy $\pi_\varphi(s_t, z_t)$ takes the humanoid state s_t and a latent code $z_t \in \mathbb{R}^d$ with $\|z_t\|_2 = 1$ and outputs a PD action. In each training iteration, we roll out the prior to obtain s_t, query the expert tracking controller to obtain the supervision action $a_t^* = \pi_{\text{track}}(s_t, g_t)$ using a reference goal g_t drawn from the dataset, and encode the goal as

$$z_1 = E(g_t), \quad \|z_1\|_2 = 1. \tag{1}$$

We then generate the prior action

$$a_{t1} = \pi_\varphi(s_t, z_1). \tag{2}$$

To populate the spherical latent space, we draw $\epsilon \sim \mathcal{N}(0, I)$, normalize it onto the unit sphere, and treat it as a random latent

$$z_2 = \frac{\epsilon}{\|\epsilon\|_2}, \quad a_{t2} = \pi_\varphi(s_t, z_2). \tag{3}$$

We optimize π_φ using three losses. The imitation distillation loss matches a_{t1} to the expert action:

$$L_{\text{distill}} = \|a_{t1} - a_t^*\|_2^2. \tag{4}$$

To provide an in-distribution signal, we introduce a discriminator $D(s_t, a_t)$ that distinguishes expert actions from non-expert actions. We train D with the binary cross-entropy objective where a_{t1} is treated as positive samples and a_{t2} as negative samples:

$$L_{\text{disc}} = -\mathbb{E}\big[\log D(s_t, a_{t1}) + \log\big(1 - D(s_t, a_{t2})\big)\big]. \tag{5}$$

The discriminator minimizes L_{disc}, while π_φ does not optimize this adversarial loss directly.

To shape the latent manifold, we introduce a discriminator-guided local semantic consistency loss. For the pair (z_1, z_2), we compute the spherical distance and neighborhood weight

$$d_{12} = \|z_2 - z_1\|_2, \qquad w_d = e^{-\beta\, d_{12}}. \tag{6}$$

and a discriminator-based semantic weight

$$w_c = 1 + \big|\min\big(0,\, D(s_t, a_{t2})\big)\big|. \tag{7}$$

The discriminator-guided local semantic consistency loss is then defined as

$$L_{\text{DLSC}} = w_d\, w_c\, \|a_{t2} - a_t^*\|_2^2. \tag{8}$$

The final SLMP objective minimizes distillation and semantic consistency while the discriminator is updated separately:

$$L_{\text{SLMP}} = \lambda_{\text{distill}} L_{\text{distill}} + \lambda_{\text{DLSC}} L_{\text{DLSC}}. \tag{9}$$

Only L_{disc} updates D, while L_{SLMP} updates π_φ. To maintain training stability, we first train using only the w_d objective until convergence, and then introduce w_c for joint optimization. Note that although π_φ does not optimize L_{disc} directly, the semantic weight w_c couples π_φ with the discriminator, inducing an implicit adversarial interaction in the latent space.

Intuition. Pure distillation preserves expert behaviors but does not support random sampling, since most latent directions on the unit sphere lack semantic meaning and therefore cannot induce valid actions. One could instead apply a standard adversarial generator loss to increase discriminator scores on actions induced by random latents, but such adversarial feedback only provides coarse in-distribution gradients and does not offer the fine-grained supervision required to reproduce precise joint-space behaviors. We therefore employ L_{DLSC} as an implicit discriminator-guided shaping objective. The geometric weight w_d encourages local smoothness, while the discriminator-based weight w_c increases the weight of out-of-distribution samples to impose stronger constraints. Together, these terms carve a structured latent action space on the sphere. This structure emerges conditionally on the current state s_t. States with multiple plausible futures (e.g., standing) yield multi-modal latent regions, while highly constrained motions (e.g., aerial kicks) produce lower semantic variability across the sphere.

3.3. High-Level Task: Two-Agent Combat

Similar to NCP (Zhu et al., 2023) and Smplolympics (Luo et al., 2024b), SLMP is deployed as a structured low-level control prior, while a high-level policy operates in latent space and outputs latent action codes. For each agent $i \in \{1, 2\}$, a high-level policy samples

$$z_{t_i} \sim \pi_{h_i}(z \mid s_{t_i}), \tag{10}$$

Algorithm 1 Spherical Latent Motion Prior (SLMP)

Freeze expert controller π_{track}; initialize prior policy π_φ, encoder E, discriminator D
Set use_$w_c \leftarrow$ **False**
repeat
  Sample reference goals g_t from dataset; Rollout prior to obtain states s_t
  Compute expert actions $a_t^* = \pi_{\text{track}}(s_t, g_t)$
  Encode latent $z_1 = E(g_t)$ and normalize to unit sphere
  Sample random latent $z_2 \sim \mathcal{S}^{d-1}$
  Compute prior actions $a_{t1} = \pi_\varphi(s_t, z_1)$ and $a_{t2} = \pi_\varphi(s_t, z_2)$
  Compute distillation loss L_{distill} (Eq. 4)
  if use_w_c = **False then**
    Compute L_{DLSC} without w_c
    Update (π_φ, E) by minimizing L_{SLMP}
    if converged **then**
      Set use_$w_c \leftarrow$ **True**
    end if
  else
    Compute semantic consistency loss L_{DLSC} (Eq. 8)
    Compute discriminator loss L_{disc} (Eq. 5)
    Update (π_φ, E) by minimizing L_{SLMP} (Eq. 9)
    Update D by minimizing L_{disc} (Eq. 5)
  end if
until convergence

and SLMP produces the final PD action via

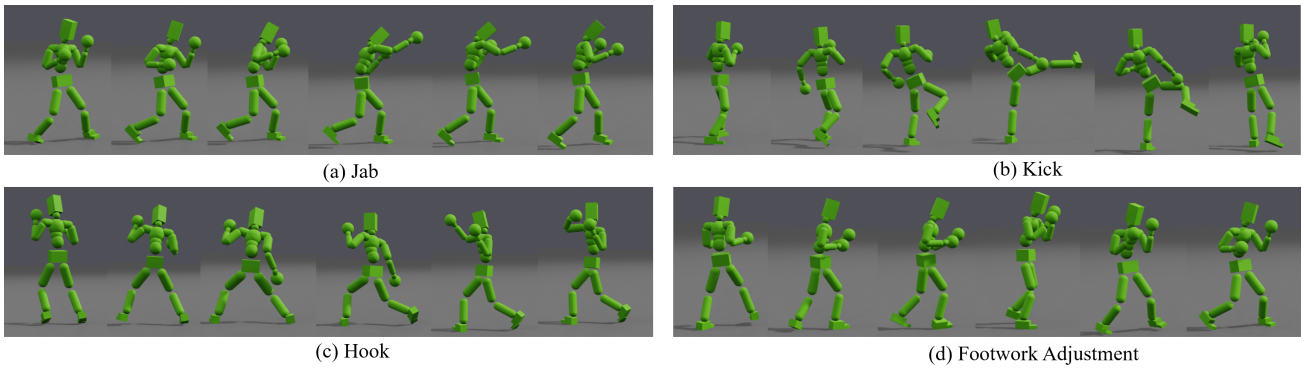
$$a_{t_i} = \pi_\varphi(s_{t_i}, z_{t_i}). \tag{11}$$

This yields a clean separation of roles that the high-level policy selects latent behaviors in a compact space, while SLMP provides physics-consistent full-body execution.

The two-agent combat scenario is trained via self-play, where each agent observes its own proprioceptive state and the opponent's relative pose. The reward is strictly task-based and sparse, consisting only of rule-based hit and knockout events, with no additional reward shaping. Early termination is triggered when the agents maintain a large separation or when any agent falls, which encourages engagement under sparse rewards. Despite the minimal reward design, the high-level policies learn coordinated striking and evasion behaviors, while SLMP maintains physically consistent motor execution and transitions.

4. Experiments

We base all experiments on our two-hour combat motion capture dataset introduced in Section 4.1, which provides the reference motions and tracking targets for our evaluations. We then evaluate the learned latent space in Section 4.2,



(a) Jab (b) Kick (c) Hook (d) Footwork Adjustment

Figure 2. Qualitative examples of random latent-conditioned rollouts generated by SLMP. Uniformly sampled latent codes produce diverse, stable, and physically plausible full-body motions. Additional rollouts are provided in the supplemental video.

where we quantify motion tracking performance, evaluate random rollout stability, assess semantic realism, visualize the spherical latent manifold, and perform ablations over latent representations and loss components. We demonstrate how the learned latent space supports high-level decision-making by running a two-agent combat task with simple rule-based rewards. Lastly, we show that SLMP can also be applied to real humanoid robot platforms (see Appendix A).

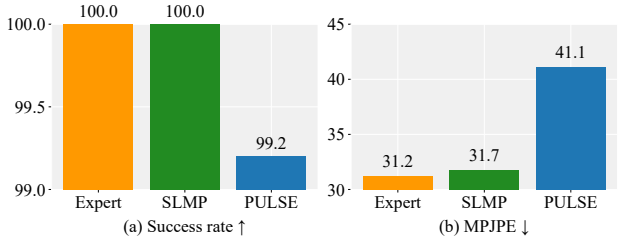


Figure 3. Latent-space motion tracking performance. We evaluate information loss by tracking reference clips through the latent space. SLMP achieves higher success and lower MPJPE than PULSE, and approaches the expert controller.

4.1. Combat Motion Capture Dataset

A key contribution of this work is a new human combat motion capture dataset tailored for physics-based humanoid control. We recruited a volunteer with over three years of experience in Kickboxing. Motion data was recorded using an Xsens MVN Link inertial motion capture suit at 180 Hz (Xsens, 2013).

The participant was instructed to perform a comprehensive range of striking and defensive techniques. The dataset covers stances and footwork (including varying speeds and varying weight distributions), punches (jabs, crosses, hooks, uppercuts, and swings), kicks (front kicks, roundhouse kicks, side kicks, low kicks, and spinning kicks), knees, elbows, and diverse defensive maneuvers (slips, ducks, lean-backs, and checks). Crucially, to support the learning of diverse combat strategies, the actions were captured with explicit variations in speed (slow, normal, and fast/explosive) and target height (head, body, and leg). Additionally, we also selected a small set of boxing motions from the AMASS dataset with a total duration of about 8 minutes. In total, we collected approximately two hours of data, which are segmented into 502 clips with durations of approximately 14 seconds. For all experiments, the dataset is downsampled to 30 Hz. For more details, refer to Appendix B.

4.1.1. Motion Tracking Performance

We evaluate information loss introduced by the latent space by tracking reference clips using latent codes and rolling out motions in the physics simulator. For each reference frame we compute $z_t = E(g_t)$ and reconstruct an action $a_t = \pi_\varphi(s_t, z_t)$, then execute it in simulation to obtain latent-space tracking. We compare against two baselines: (1) the expert tracking controller π_{track} (upper bound), and (2) PULSE (Luo et al., 2024a), a state-of-the-art motion prior based on a VAE latent space. We measure *Success* (fraction of clips tracked without falling or diverging, higher is better) and *MPJPE* (mean per-joint position error between simulated and reference body markers, lower is better). Figures 3 show that SLMP yields higher Success and lower MPJPE than PULSE, approaching the expert upper bound. This indicates that SLMP introduces less information loss during latent reconstruction.

4.2. Latent Space Evaluation

4.2.1. Random Rollout Stability

To evaluate random rollout stability, we uniformly sample latent codes $z \sim \mathcal{S}^{d-1}$ and execute the induced trajectories $a_t = \pi_\varphi(s_t, z)$ in Isaac Gym (Makoviychuk et al., 2021) for fixed horizons, comparing against PULSE. Because PULSE employs a Gaussian latent distribution, the outer regions of its latent space exhibit low semantic density, and random sampling from these regions often produces invalid joint

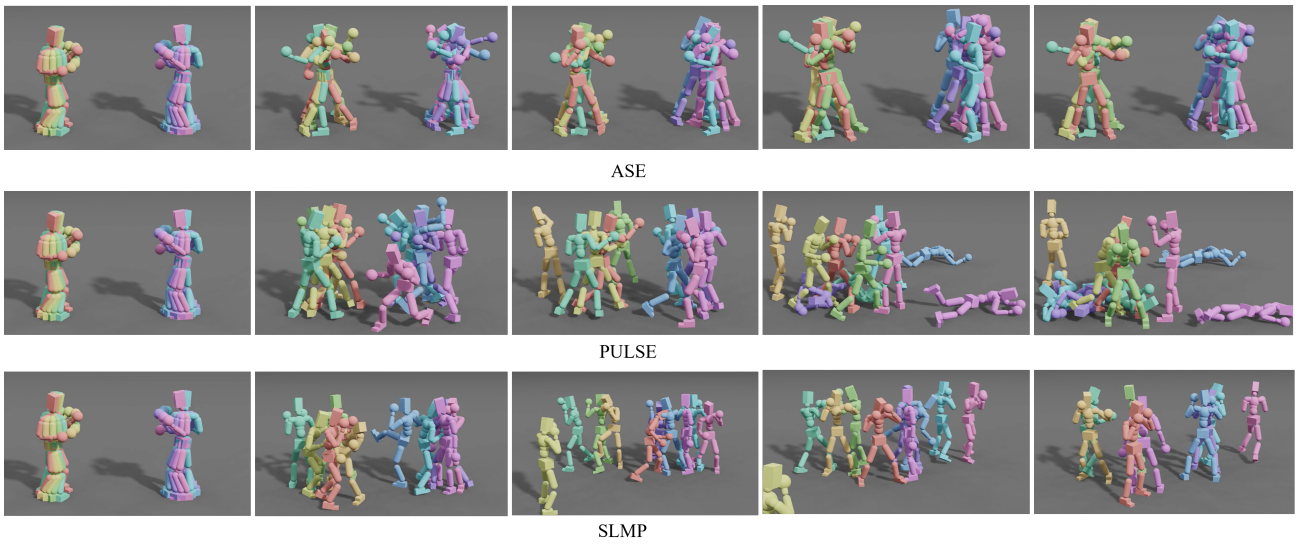


Figure 4. Qualitative comparison of random latent-conditioned rollouts. ASE exhibits repetitive low-diversity behaviors, PULSE collapses occasionally, and SLMP generates diverse and stable motions. See supplemental video for full rollouts.

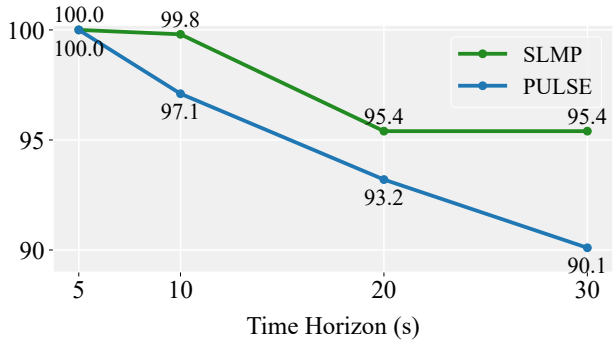


Figure 5. Random latent rollout survival curves over 1000 trials in Isaac Gym. SLMP maintains substantially higher survival rates across long horizons, whereas PULSE survival degrades due to semantic sparsity in the tails of its VAE latent space.

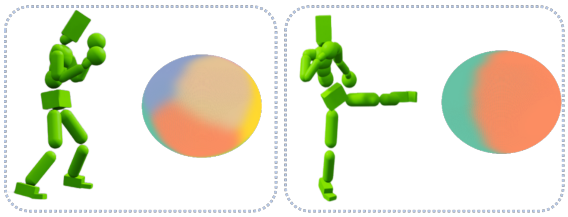


Figure 6. State-dependent spherical latent manifold. left: a guard stance yields diverse feasible follow-ups with multiple regions on the sphere. right: an airborne kick yields only a few feasible continuations, causing the sphere to collapse into one or two regions.

actions that rapidly lead to falls. We perform 1000 random rollouts for each method and report the fraction of trials that do not result in falls after 5s, 10s, 20s, and 30s. As shown in Figure 5, SLMP maintains substantially higher survival rates across all time horizons, whereas PULSE survival declines with rollout duration, indicating that SLMP yields a more uniformly valid latent action space for high-dynamic combat behaviors under random sampling.

4.2.2. Semantic Realism of Random Latent Rollouts

We sample latent codes from the spherical prior and roll out SLMP starting from a neutral stance. Figure 2 shows representative examples. SLMP generates a wide range of physically plausible and semantically interpretable combat motions, including guards, jabs, kicks, and footwork adjustments. We observe no significant artifacts such as excessive foot sliding or unrealistic joint poses. Figure 4 illustrates qualitative results of PULSE and ASE. PULSE loses balance and collapses occasionally, while ASE remains upright but degenerates into repetitive and low-diversity motions such as small arm swings. In contrast, SLMP produces diverse and coherent motions. Refer to the supplemental video for clearer demonstrations of motion realism and diversity.

4.2.3. Spherical Latent Manifold Visualization

We use a state-conditioned visualization to show how SLMP changes the structure of the latent sphere depending on the current pose. For a fixed humanoid state s_t, we uniformly sample latent codes z on the unit sphere, pass each pair (s_t, z) through the prior policy to obtain an action $a_t = \pi_\varphi(s_t, z)$, and cluster the resulting actions in joint space. We then assign a color to each cluster and paint the corresponding latent points on the sphere, which reveals how many distinct motion modes SLMP considers feasible from that state (Figure. 6).

We visualize two representative states. For a neutral guard stance, SLMP produces a wide variety of follow-up actions such as jabs, crosses and evasive steps, and the sphere decomposes into several colored regions with smooth boundaries. This pattern indicates that many qualitatively different but valid motions are available from this posture. In contrast, for an airborne kicking state, most latents decode to very similar landing or recovery motions, and one or two clusters cover almost the entire sphere. This collapse of the sphere into a small number of colors reflects the fact that the character has very few physically plausible choices while mid-air. Together, these visualizations show that SLMP encodes a state-dependent feasible set of actions on the latent sphere, which in turn ensures that random sampling remains stable and physically reasonable across different poses.

4.2.4. Ablation Studies

Table 1. Ablation over latent representations. Success rate and MPJPE measure latent tracking quality. Survival measures random rollout stability under uniform sampling, defined as the percentage of 1000 randomly sampled rollouts that do not result in a fall within 10 seconds.

Method	Success (%) ↑	MPJPE ↓	Survival(%) ↑
VAE	99.8	33.7	10.2
VQ-VAE	99.0	36.9	12.4
Sphere	**100.0**	**31.5**	0.1
SLMP (ours)	**100.0**	31.7	**99.8**

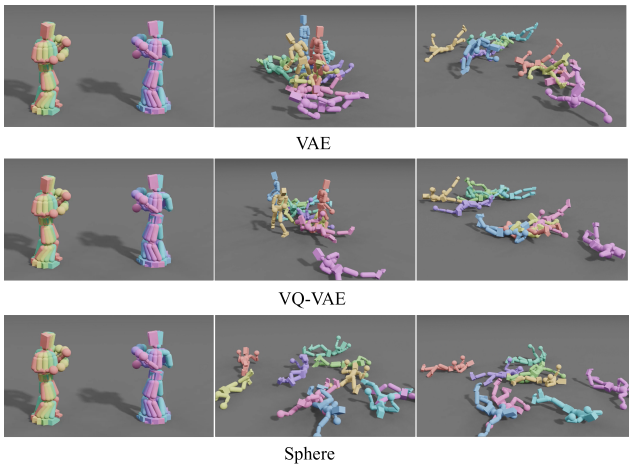


Figure 7. Each row shows a rollout sampled from different latent structures.

Latent Representation. We ablate the effect of latent representations by comparing three variants against SLMP: (i) a VAE latent, (ii) a VQ-VAE latent, and (iii) a unit-sphere latent without our proposed semantic consistency objectives. As summarized in Table 1, VAE and VQ-VAE suffer from reconstruction-induced information loss and show unstable performance under random latent sampling. Although both methods can achieve reasonable tracking accuracy, their survival rates drop drastically, indicating that many randomly sampled latents do not correspond to physically valid or semantically meaningful motions. Imposing a spherical constraint alone removes the Gaussian prior and slightly improves tracking, but it still lacks semantic structure in the latent space. As a result, randomly sampled latents often correspond to incoherent or weakly coordinated behaviors, leading to frequent falls and near-zero survival rates.

In contrast, SLMP maintains a structured spherical latent manifold where nearby latents correspond to semantically consistent actions. This structure preserves tracking fidelity while dramatically improving random-sampling stability, achieving near-perfect survival rates. Figure 7 further visualizes qualitative rollouts. Refer to the supplemental video for clearer demonstrations.

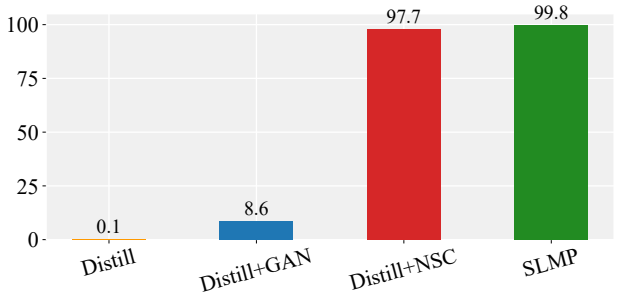


Figure 8. Survival rate at 10 seconds under random latent sampling.

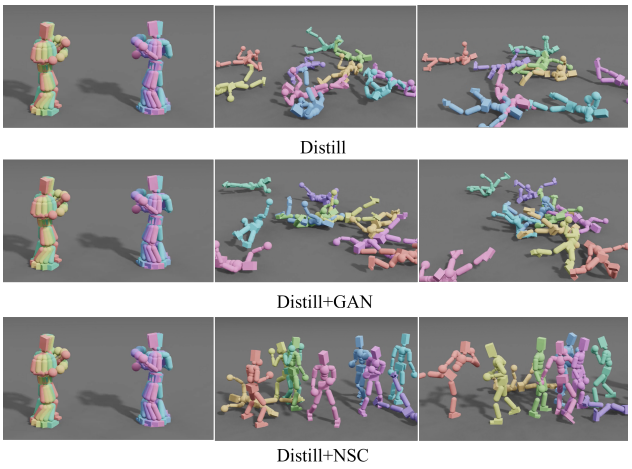
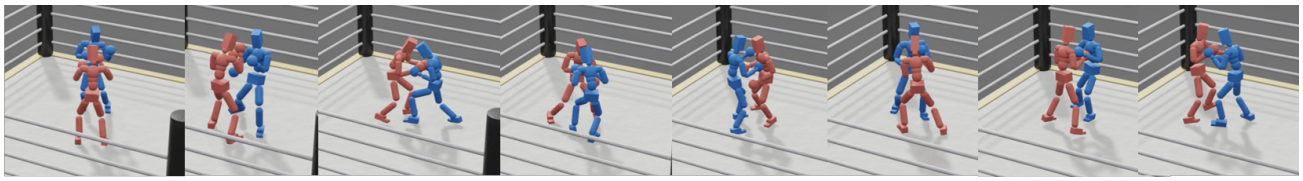
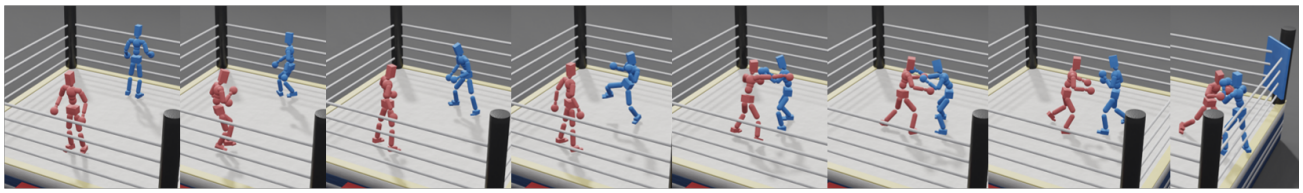


Figure 9. Each row shows a rollout sampled from the latent sphere under a different training loss configuration.

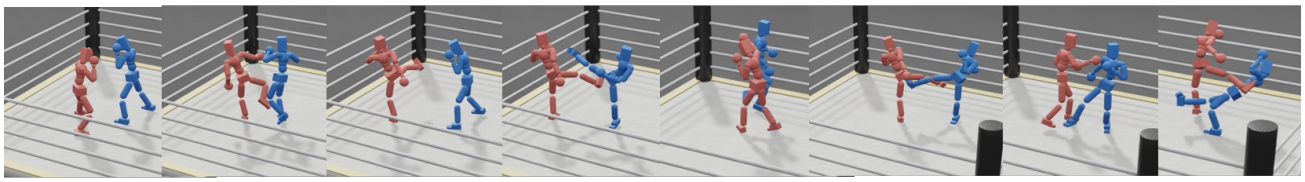
Loss Components. To isolate the contribution of each learning signal, we compare four configurations: (i) Distill, (ii) Distill+GAN, (iii) Distill+NSC (nearest semantic consistency without discriminator semantics), and (iv) SLMP. Since all four configurations achieve near-perfect latent



NPC-Style Reward



NPC-Style + AMP Reward



Simple Rule-based Reward

Figure 10. Qualitative results on the two-agent combat task using SLMP as a motion prior. We compare dense NCP-style rewards, Smplolympics-style rewards (NCP-style rewards augmented with AMP reward), and sparse rule-based rewards. NCP-style rewards lead to degenerate stationary interactions, while Smplolympics-style rewards bias the policy toward repetitive punching behaviors. In contrast, sparse rule-based rewards with SLMP produce diverse and human-like combat behaviors.

tracking accuracy, we focus on random rollout stability and report survival rates at multiple time horizons.

We visualize the survival function at the 10-second horizon in Figure 8, where the bar plot reports the fraction of rollouts that remain fall-free. The gap highlights that SLMP maintains a substantially more stable and uniformly valid latent action space under random sampling.

As shown in Figure 9, all three variants exhibit varying degrees of instability under random sampling. Distill lacks any regularization on the latent manifold and therefore produces unconstrained actions. Distill+GAN only receives coarse distribution-level gradients from adversarial logits, which do not provide actionable guidance for continuous control. Distill+NSC provides meaningful constraints for most samples, yet ambiguity remains when semantically different behaviors compete for neighboring regions on the sphere. SLMP achieves significantly higher survival across all horizons by combining geometric proximity with discriminator-informed semantic cues. Refer to the supplemental video for clearer demonstrations.

4.3. High-Level Two-Agent Combat

We further evaluate how well the learned latent space supports high-level decision-making in a competitive setting. We compare against two reward designs commonly used in prior work. The first follows the combat setup in NCP (Zhu et al., 2023), which relies on dense shaping rewards including orientation facing, proximity, locomotion velocity, and end-effector contact. The second adopts the reward design used in Smplolympics (Luo et al., 2024b), which augments NCP-style shaping rewards with an additional AMP-based adversarial reward to enforce motion realism.

Our setting uses only sparse rule-based combat events. A hit reward when one agent makes contact with the opponent and a knockdown reward when the opponent fall. No imitation, adversarial, or shaping terms are used. Despite the absence of dense shaping, SLMP enables the high-level policy to discover diverse offensive and evasive behaviors through self-play. As shown in Figure 10, policies trained with NCP-style rewards often converge to degenerate strategies where both agents remain in close proximity without executing coherent strikes. Smplolympics-style rewards, while improving motion realism, bias the policy toward a narrow subset of behaviors, resulting in repetitive punching patterns with limited tactical diversity. With SLMP and sparse rule-based rewards, agents instead exhibit structured striking, footwork, and countering behaviors. These results demonstrate that SLMP provides a strong motion prior that reduces the need for complex reward shaping and enables expressive multi-agent behaviors.

5. Conclusion

We introduced the Spherical Latent Motion Prior (SLMP), a two-stage framework for learning high-fidelity and semantically structured motion priors for physics-based humanoid control. By combining expert distillation with a spherical latent manifold and a discriminator-guided semantic consistency objective, SLMP mitigates the information loss of VAE-based priors and the mode collapse tendencies of adversarial methods. Experiments on a large-scale combat motion capture dataset demonstrate that SLMP preserves fine motion detail, supports stable random sampling, and enables the emergence of human-like multi-agent combat behaviors under sparse reward signals. Beyond simulated humanoid avatars, we further validate SLMP on realistic humanoid robot models within physics simulation, demonstrating its compatibility with practical robot morphologies and control constraints. These results highlight the importance of well-structured motion priors for reducing reward engineering and simplifying high-level control in complex humanoid tasks. Future work will explore scaling SLMP to larger and more diverse motion datasets, extending it to high-level tasks that require rich environment interaction, and pursuing real-world validation on physical humanoid robots.

References

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

ENGINEAI Robotics. Engineai pm01 humanoid robot. https://www.engineai.com.cn/product-pm01.html. Accessed: 2026-03-01.

Gao, J., Wang, Z., Xiao, Z., Wang, J., Wang, T., Cao, J., Hu, X., Liu, S., Dai, J., and Pang, J. Coohoi: Learning cooperative human-object interaction with manipulated object dynamics. *Advances in Neural Information Processing Systems*, 37:79741–79763, 2024.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Jing, T., Chen, S., Li, Y., Xu, W., and Xu, R. Farm: Frame-accelerated augmentation and residual mixture-of-experts for physics-based high-dynamic humanoid control. *arXiv preprint arXiv:2508.19926*, 2025.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

Luo, Z., Hachiuma, R., Yuan, Y., and Kitani, K. Dynamics-regulated kinematic policy for egocentric pose estimation. *Advances in Neural Information Processing Systems*, 34: 25019–25032, 2021.

Luo, Z., Cao, J., Kitani, K., Xu, W., et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10895–10904, 2023.

Luo, Z., Cao, J., Merel, J., Winkler, A., Huang, J., Kitani, K. M., and Xu, W. Universal humanoid motion representations for physics-based control. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=OrOd8PxOO2.

Luo, Z., Wang, J., Liu, K., Zhang, H., Tessler, C., Wang, J., Yuan, Y., Cao, J., Lin, Z., Wang, F., et al. Smplolympics: Sports environments for physically simulated humanoids. *arXiv preprint arXiv:2407.00187*, 2024b.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5442–5451, 2019.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Pan, L., Yang, Z., Dou, Z., Wang, W., Huang, B., Dai, B., Komura, T., and Wang, J. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5379–5391, 2025.

Peng, X. B., Abbeel, P., Levine, S., and Van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

Peng, X. B., Ma, Z., Abbeel, P., Levine, S., and Kanazawa, A. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

Peng, X. B., Guo, Y., Halper, L., Levine, S., and Fidler, S. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

Ross, S., Gordon, G., and Bagnell, D. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Tessler, C., Kasten, Y., Guo, Y., Mannor, S., Chechik, G., and Peng, X. B. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–9, 2023.

Tessler, C., Guo, Y., Nabati, O., Chechik, G., and Peng, X. B. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6):1–21, 2024.

Unitree Robotics. Unitree g1 humanoid robot. https://www.unitree.com/cn/g1. Accessed: 2026-03-01.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Wang, Y., Zhao, Q., Yu, R., Tsui, H. W., Zeng, A., Lin, J., Luo, Z., Yu, J., Li, X., Chen, Q., et al. Skillmimic: Learning basketball interaction skills from demonstrations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17540–17549, 2025.

Won, J., Gopinath, D., and Hodgins, J. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics (TOG)*, 40(4):1–11, 2021.

Xsens, M. Full 6dof human motion tracking using miniature inertial sensors. *Daniel RoetenbergLuingeHenk*, 2013.

Zhang, Z., Bashkirov, S., Yang, D., Shi, Y., Taylor, M., and Peng, X. B. Physics-based motion imitation with adversarial differential discriminators. In *SIGGRAPH Asia 2025 Conference Papers (SIGGRAPH Asia '25 Conference Papers)*, 2025.

Zhu, Q., Zhang, H., Lan, M., and Han, L. Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6):1–16, 2023.

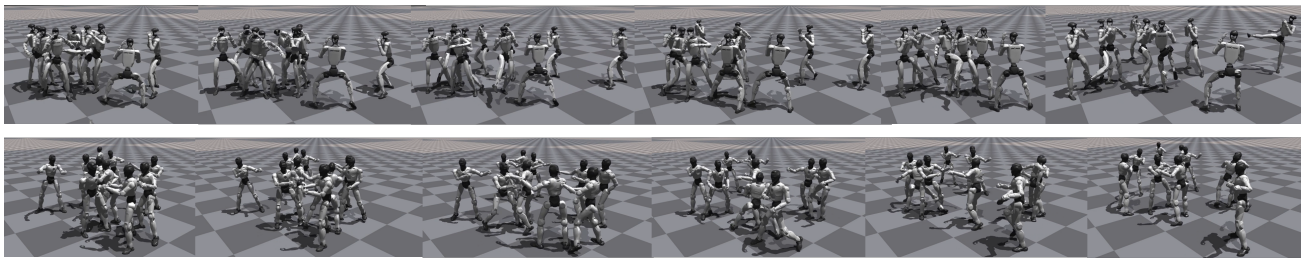


Figure 11. Random latent sampling on realistic humanoid robot models in simulation. Top: Unitree G1. Bottom: ENGINEAI PM01. Both models can stably generate diverse motions under random latent inputs.

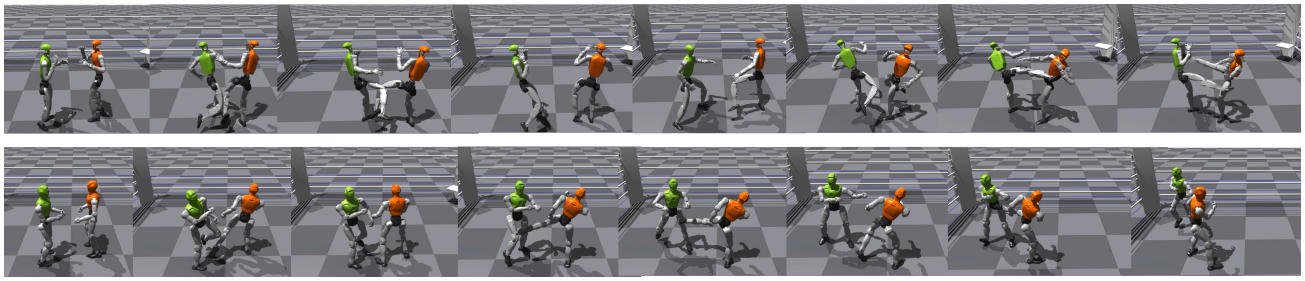


Figure 12. Two-agent combat evaluation on realistic humanoid robot models in simulation. Top: Unitree G1. Bottom: ENGINEAI PM01. The robots generate human-like combat behaviors under the SLMP prior.

A. Validation on Realistic Humanoid Robot Models

To further evaluate the generalization ability of SLMP, we validate the learned motion prior on realistic humanoid robot models within physics simulation. Unlike the SMPL-based avatar used in the main experiments, these models reflect practical robot kinematic structures. Specifically, we conduct experiments on the Unitree G1 (Unitree Robotics) and the ENGINEAI PM01 (ENGINEAI Robotics) humanoid robot models.

A.1. Random Latent Sampling

We first evaluate random latent sampling on both robot models. As shown in Figure 11, the robots are able to stably generate diverse motions under random sampling, indicating that SLMP generalizes across different humanoid morphologies.

A.2. Two-Agent Combat Evaluation

We further evaluate SLMP in a two-agent combat setting using the realistic robot models. As shown in Figure 12, the robots are able to generate human-like combat behaviors, demonstrating that SLMP supports high-level interaction across different humanoid morphologies.

Overall, these results demonstrate that SLMP is not restricted to a single virtual humanoid representation, but can generalize to realistic robot morphologies within physics simulation. This validation suggests strong potential for future deployment on physical humanoid platforms.

B. Dataset

Table 2 summarizes detail statistics of the dataset.

C. Implementation Details

C.1. Network Architectures

All networks are implemented as multilayer perceptrons (MLPs) with component-specific depth and width, as summarized in Table 3. The two high-level combat controllers π_{h_1} and π_{h_2} share the same architecture.

Table 2. Statistics of the combat motion capture dataset (502 clips). The dataset covers diverse speeds and target levels.

Category	Action Types	Attributes & Targets	Clips	Dur.
Stance	Orthodox (Idle, Guard), Weight Shift	Speed: Slow, Normal	20	11s
Footwork	Step, Pivot (90°), Shuffle	Step Size: Small, Large	62	11s
Straight Punches	Jab, Cross	Target: Head, Body, Downward	60	11s
Hooks & Swings	Hook, Swing	Target: Head, Body; Type: Flat/Up	50	11s
Uppercuts	Uppercut (Left/Right)	Target: Head, Body	25	11s
Front Kicks	Front Kick (Left/Right)	Target: Body; Incl. Heavy Bagss	40	11s
Roundhouse	Roundhouse Kick (Left/Right)	Target: Low, Mid, High; Incl. Heavy Bag	55	11s
Other Kicks	Side, Low, Spinning Kick	Target: Leg, Body, Head; Incl. Heavy Bag	55	11s
Close Range	Knee, Elbow	Target: Body, Head; Direction: Multi	30	11s
Defense	Guard, Slip, Duck, Lean-back, Check	Intensity: Low to High	70	11s
Combinations	Jab-Cross-Hook variants, Counters	Seq: 2-5 strikes; Mixed Defense	35	10-20s
Total	**All Categories**	**–**	**502**	**∼2.0h**

C.2. Training Hyperparameters

We use PPO to train the expert tracking controller π_{track}. The latent prior is trained using a DAgger-style (Ross et al., 2011) online imitation procedure, where the student policy rolls out trajectories and the expert provides on-policy supervision. Table 4 summarizes key hyperparameters.

Table 3. Network architectures used in our framework. All networks are implemented as multilayer perceptrons (MLPs).

Component	Hidden Units	Activation
π_{track}	[2048, 1536, 1024, 1024, 512, 512]	SiLU
$E(\cdot)$	[512, 256]	ReLU
π_{φ}	[4096, 2048, 1024, 1024, 512, 512]	SiLU
$D(\cdot)$	[4096, 2048, 1024, 512]	ReLU
π_{h_1}, π_{h_2}	[2048, 1024, 512]	SiLU

Table 4. Key hyperparameters used in our experiments.

PPO Hyperparameters					
Batch Size	Env Num	LR	γ	Clip ϵ	
32768	1024	5×10^{-5}	0.99	0.2	
SLMP Hyperparameters					
λ_{distill}	λ_{disc}	Disc LR	λ_{DLSC}	β	d
1.0	1×10^{-4}	5×10^{-5}	1.0	0.1	64

C.3. Software and Hardware Setup

All simulations are implemented in Isaac Gym with a physics timestep of 60 Hz, and PD control runs at the same frequency. Motion data is downsampled to 30 Hz for training. Experiments are conducted on a machine with Ubuntu 22.04, Python 3.8, and PyTorch 2.1. Training is performed on a server equipped with two NVIDIA RTX 4090 GPUs. Training the tracking controller typically takes 24 hours, training the latent prior takes 12 hours, and training the two-agent combat policy requires approximately 8 hours.

D. Detailed Task Formulation

We provide additional implementation details for the three training stages. We focus on state/observation design and reward functions, which are most critical for reproducibility.

D.1. Stage 1: Motion Tracking Controller

We train the expert motion tracking controller following PHC (Luo et al., 2023) under a goal-conditioned RL formulation. The task is an MDP where the policy outputs PD targets for each actuated DoF, and the simulator applies torques via a PD controller.

State. The simulation state is defined as

$$s_t \triangleq (s_t^p, s_t^g), \tag{12}$$

where s_t^p is the humanoid proprioception and s_t^g is a reference-driven goal. The proprioceptive state is

$$s_t^p \triangleq (q_t, \dot{q}_t), \tag{13}$$

where q_t and $\dot{q}_t$ denote the humanoid pose and velocity in simulation.

The goal state uses the discrepancy between the next-step reference quantities and the simulated counterpart:

$$s_t^{g\text{-rot}} \triangleq (\hat{\theta}_{t+1} \ominus \theta_t,\ \hat{p}_{t+1} - p_t,\ \hat{v}_{t+1} - v_t,\ \hat{\omega}_{t+1} - \omega_t,\ \hat{\theta}_{t+1},\ \hat{p}_{t+1}), \tag{14}$$

where $\ominus$ computes the rotation difference.

All quantities in (s_t^p, s_t^g) are normalized w.r.t. the humanoid's current facing direction and root position, as in PHC (Luo et al., 2023).

Reward. The per-step reward is composed of a task imitation term, an AMP-style discriminator term, and an energy penalty:

$$r_t = 0.5\, r_t^g + 0.5\, r_t^{\text{amp}} + r_t^{\text{energy}}. \tag{15}$$

For motion tracking, the task reward takes the form

$$\begin{aligned} r_t^{g\text{-imitation}} = {} & w_{\text{jp}} \exp\big(-100\|\hat{p}_t - p_t\|\big) + w_{\text{jr}} \exp\big(-10\|\hat{q}_t \ominus q_t\|\big) \\ & + w_{\text{jv}} \exp\big(-0.1\|\hat{v}_t - v_t\|\big) + w_{\text{j}\omega} \exp\big(-0.1\|\hat{\omega}_t - \omega_t\|\big), \end{aligned} \tag{16}$$

which measures discrepancies of link translations, rotations, linear velocities, and angular velocities between the simulated motion and the reference.

The energy penalty is defined as

$$r_t^{\text{energy}} = -0.0005 \sum_{j \in \text{joints}} \|\mu_j \omega_j\|^2, \tag{17}$$

where μ_j and ω_j are the joint torque and joint angular velocity, respectively. The style reward r_t^{amp} is computed using an AMP-style discriminator with the same observation design and training objective as PHC (Luo et al., 2023).

D.2. Stage 2: SLMP Latent Prior

State and Goal. The policy input uses the same proprioceptive state definition as Stage 1. The goal g_t is derived from the reference motion and encoded by the goal encoder as described in the main paper. No additional observations are introduced in this stage.

D.3. Stage 3: Two-Agent Combat

Observation. The observation design largely follows the boxing setup in SMPLOlympics (Luo et al., 2024b). Each agent receives both self and opponent information in an egocentric frame, including:

- Self proprioception: joint poses, velocities, and root state,

- Opponent root pose and velocity relative to the agent,
- Relative positions between the agent's striking limbs (hands and feet) and the opponent's scoring regions (head and torso),
- Contact force magnitudes on key body parts.

All quantities are expressed in a heading-aligned local frame.

Reward Function. We use simple rule-based sparse rewards focused on effective striking.

Hit reward. A positive reward is given when an agent's hand or foot comes within $0.3\,\mathrm{m}$ of the opponent's head or torso and the corresponding contact force exceeds 30. The reward magnitude is proportional to the measured contact force.

Hit penalty. A symmetric penalty is applied when the agent's own head or torso is hit under the same conditions.

Knockdown reward. A bonus of $+50$ is given when the opponent falls. Conversely, a penalty of -50 is applied when the agent falls.

Termination Conditions. Episodes terminate under the following conditions:

- Knockdown of either agent,
- pelvis-to-pelvis distance below $0.3\,\mathrm{m}$ for more than $1\,\mathrm{s}$,
- Distance between an attacking limb and the opponent's scoring regions below $0.3\,\mathrm{m}$ for more than $1\,\mathrm{s}$ (to prevent reward farming),
- During early training, if the pelvis distance exceeds $1.2\,\mathrm{m}$.

Self-Play Setup. We adopt an alternating self-play scheme with two separate policy instances. At any time, one policy is designated as the learning agent while the other serves as a fixed opponent. During training, the learning agent is updated using PPO while the opponent policy remains frozen. Every 250 epochs, the roles are swapped, and the previously fixed policy becomes trainable while the other is held fixed.

This alternating update stabilizes competitive learning by preventing both agents from simultaneously drifting, and encourages continual adaptation against a progressively improving opponent. Both policies are initialized identically but evolve independently during training.