HoRD: Robust Humanoid Control via History-Conditioned Reinforcement Learning and Online Distillation

Anonymous authors
Paper under double-blind review

Abstract

Humanoid robots hold great promise for general-purpose embodied intelligence, but their control policies often suffer significant performance drops under small changes in robot and environmental dynamics. Dense full-body references simplify tracking but impose a restrictive deployment command, whereas sparse targets leave uncommanded joints underdetermined. We present HoRD, a two-stage learning framework for whole-body humanoid control under partial observability and varying dynamics. A teacher first learns dense-reference tracking under episode-level domain randomization. Its control behavior is then transferred through online distillation to a student that receives partial observations and Standardized Sparse-Joint Representation (SSJR) commands over a fixed set of future key-joint targets. Both policies use a History-Conditioned Dynamics Representation (HCDR), a history encoder that summarizes recent state–action interactions into a control-oriented representation. HoRD achieves State-Of-The-Art (SOTA) performance across four conditions spanning Isaac Lab and Genesis, including zero-shot cross-simulator transfer under fixed and randomized dynamics. Ablations show that episode-level domain randomization and HCDR play complementary roles under dynamics shift. Qualitative Unitree G1 rollouts further illustrate real-world deployment without fine-tuning.

1 Introduction

Whole-body humanoid motion tracking seeks to convert motion intent into stable, coordinated behavior over diverse locomotion and expressive skills. Physics-based imitation and large motion corpora have enabled humanoid controllers to reproduce increasingly broad motion repertoires (Peng et al., 2018; 2021; Mahmood et al., 2019; Liao et al., 2025). Yet strong tracking in the training environment does not by itself provide a practical deployment controller. Such a controller must accept compact motion commands while remaining stable when robot dynamics, contact properties, control delay, or the underlying simulator differ from those encountered during training.

This setting creates two coupled challenges. First, dense full-body references reduce motion ambiguity by specifying targets for every tracked body, but requiring a complete future pose at deployment imposes a restrictive command representation. Sparse body targets retain only selected motion intent and leave the coordination of uncommanded joints underdetermined. Second, changes in robot and environmental dynamics can cause the same joint-position target to produce different state transitions, while an instantaneous proprioceptive observation does not directly reveal the underlying dynamics context. A deployable controller must therefore recover coordinated whole-body motion from sparse future commands while adjusting its control decisions to dynamics that are not directly observable.

Existing methods address these challenges through different mechanisms. Masked motion completion, sparse teleoperation interfaces, and multi-mode motion commands reduce dependence on dense full-body references (Tessler et al., 2024; He et al., 2025b;c; Ji et al., 2024; Zhao et al., 2025). For dynamics robustness, domain randomization exposes policies to varied physical parameters during training (Tobin et al., 2017; Campanaro et al., 2024), while target-domain adaptation can compensate for residual mismatch through additional interaction in the deployment environment (Chebotar et al., 2019; He et al., 2025a). History-based adaptation methods such as UP-OSI and RMA instead use

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

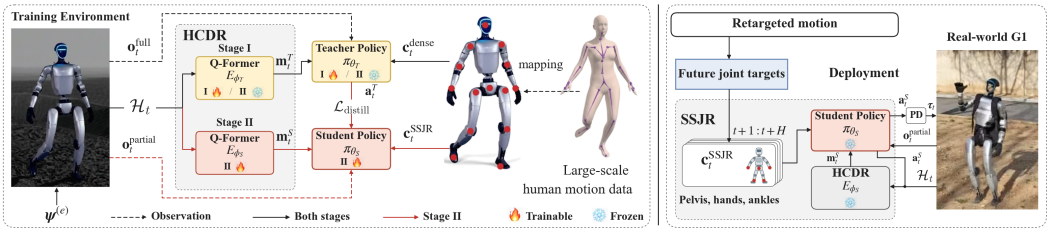

Figure 1: **Framework overview.** Stage I trains a teacher using privileged full-state observations and dense motion commands under episode-level domain randomization. Stage II distills the teacher into a student that uses partial observations and SSJR commands. Both policies condition on recent state–action history through separately parameterized HCDR encoders, while only the student is retained at deployment.

recent observations and actions as evidence about latent dynamics (Yu et al., 2017; Kumar et al., 2021). However, these works largely address sparse motion specification and dynamics adaptation in isolation. Jointly addressing both challenges in a deployable whole-body control policy, particularly for zero-shot transfer, remains less directly studied.

To jointly address these challenges, we present HoRD, a two-stage learning framework based on the insight that training supervision and deployment commands need not contain the same motion information. Dense full-body references provide detailed supervision for learning coordinated motion, whereas a deployed policy can express motion intent through selected future body targets. Meanwhile, recent state–action history provides evidence of how the robot has responded to previous commands. These two sources of information play complementary roles: teacher supervision conveys the whole-body coordination omitted from sparse commands, while history representation provides action–response information unavailable from an instantaneous observation.

Figure 1 shows this design in two stages. Stage I trains a teacher policy with privileged full-state observations and dense future motion commands under episode-level domain randomization. In Stage II, the frozen teacher provides action targets at the states visited by the student. The student receives partial observations and Standardized Sparse-Joint Representation (SSJR) commands specifying future trajectories for the pelvis, hands, and ankles. Both policies condition their actions on a History-Conditioned Dynamics Representation (HCDR), computed from recent state–action history using separately parameterized Q-Former encoders. At deployment, only the student controller is retained. It predicts target joint positions executed by the robot's PD controllers and requires neither privileged state information nor dense full-body references. Our contributions are:

- We present HoRD, a two-stage learning framework that transfers dense-reference whole-body control behavior to a deployable student operating from partial observations and sparse future-motion commands under dynamics variation.
- We define SSJR as a fixed future-conditioned command representation over the pelvis, hands, and ankles, and use HCDR to provide implicit history-conditioned adaptation from recent state–action interactions. Together, they separate the motion information required at deployment from the dense supervision and dynamics diversity available during training.
- HoRD achieves the strongest success rate and pose accuracy among the evaluated methods in all four Isaac Lab and Genesis conditions, including zero-shot transfer under fixed and randomized dynamics. Ablations establish the complementary roles of episode-level domain randomization and HCDR, while simulated terrain and perturbation tests and qualitative Unitree G1 rollouts evaluate robustness and deployment feasibility.

2 Related Work

2.1 Whole-Body Humanoid Motion Tracking

Physics-based imitation learning has enabled simulated characters and humanoid robots to reproduce reference motions under contact-rich dynamics, from locomotion skills to diverse clip-level

behaviors (Peng et al., 2017; 2018). Large-scale motion corpora such as AMASS provide broad human-motion supervision for learning general tracking policies (Mahmood et al., 2019). Subsequent work has expanded motion coverage through adversarial motion priors, scalable multi-skill controllers, and large-scale motion tracking (Peng et al., 2021; Won et al., 2020; Luo et al., 2025). Teacher–student frameworks further transfer tracking behavior from privileged policies to deployable humanoid controllers (He et al., 2025b;c; Ji et al., 2024). HoRD builds on these advances but focuses on whole-body tracking from sparse future-motion commands under variations in robot and environmental dynamics.

2.2 Sparse Future-Motion Commands

The motion command determines how much reference information is available to a tracking policy. MaskedMimic formulates control as masked motion inpainting and supports partial motion descriptions, including sparse target keyframes (Tessler et al., 2024). OmniH2O uses kinematic poses as a common command representation and distills a privileged teacher into a policy operating from sparse sensing (He et al., 2025b). HOVER consolidates multiple command modes through policy distillation, while ExBody2 combines keypoint tracking and velocity control in a teacher–student framework (He et al., 2025c; Ji et al., 2024). These methods provide flexible alternatives to dense full-body motion commands, but differ in which body targets, command modes, and future information are retained during deployment. In HoRD, SSJR uses a fixed set of future trajectories for the pelvis, hands, and ankles throughout student training and deployment, while dense full-body commands are available only to the teacher during distillation.

2.3 History-Conditioned Adaptation under Dynamics Variation

Dynamics variation can change the state transition produced by the same control action. Domain randomization broadens the physical conditions encountered during training and is widely used to improve transfer of learned locomotion and humanoid controllers (Tobin et al., 2017; Campanaro et al., 2024; He et al., 2025c; Ji et al., 2024; He et al., 2025b). Target-domain adaptation can further reduce residual dynamics mismatch, but requires interaction or data from the deployment domain (Chebotar et al., 2019; He et al., 2025a). UP-OSI explicitly predicts dynamics parameters from recent state–action history (Yu et al., 2017), whereas RMA learns an adaptation module that conditions control on recent interactions (Kumar et al., 2021). HoRD instead uses HCDR to learn a control-oriented representation from recent state–action history jointly with each policy, providing implicit history-conditioned adaptation for both the teacher and student without directly estimating physical parameters.

3 Method

We consider whole-body humanoid tracking from sparse future motion commands under variations in robot and environment dynamics. HoRD is a two-stage learning framework: a teacher policy is first trained with dense reference motions under episode-level domain randomization. A student policy is then trained through online imitation of the teacher while receiving only sparse future-body commands. The deployed student additionally conditions its actions on recent state-action history, allowing its control decisions to depend on the robot's observed response to previous commands.

3.1 Problem Formulation

We model whole-body control of a humanoid robot under variations in robot and environmental dynamics as a Contextual Partially Observable Markov Decision Process (C-POMDP) (Hallak et al., 2015). Following the contextual MDP formulation, each episode is indexed by an unobserved dynamics context $\psi^{(e)} \sim p_{\text{train}}(\Psi)$. This context specifies physical properties such as body-mass scaling, actuator gains, control delay, and contact parameters and parameterizes the transition kernel

$$\mathbf{x}_{t+1} \sim p_{\psi^{(e)}}\left(\cdot \mid \mathbf{x}_t, \mathbf{a}_t\right), \tag{1}$$

where $\mathbf{x}_t$ denotes the physical state and $\mathbf{a}_t$ denotes the control action. The dynamics context is fixed within a training episode but is not provided to the policy. At each control step, the policy receives a

proprioceptive observation $\mathbf{o}_t = \Omega(\mathbf{x}_t)$ and a motion command $\mathbf{c}_t$ that specifies the desired motion over a short future horizon. The policy uses a history of recent observations and actions,

$$\mathcal{H}_t = \{(\mathbf{o}_{t-k}, \mathbf{a}_{t-k})\}_{k=1}^{K}, \tag{2}$$

where K is the history length. A control policy, therefore, takes the general form

$$\mathbf{a}_t \sim \pi_\theta\left(\cdot \mid \mathbf{o}_t, \mathbf{c}_t, \mathcal{H}_t\right). \tag{3}$$

Because $\boldsymbol{\psi}^{(e)}$ is unobserved, the instantaneous observation does not fully determine the transition dynamics. The history $\mathcal{H}_t$ provides additional information about the robot's previous responses to control actions, making the problem partially observable from the policy's perspective.

3.2 Framework Overview

HoRD is a two-stage learning framework for whole-body humanoid control under partial observability and variations in robot and environmental dynamics, as illustrated in Figure 1. In Stage I, a teacher policy is trained under episode-level domain randomization to track dense full-body motion references. In Stage II, the teacher's control behavior is distilled into a deployable student policy that operates using proprioceptive feedback, sparse body-motion commands, and recent state–action history.

Stage I: Dense-Reference Teacher Training. We first train a teacher policy π_{θ_T} in simulation using Proximal Policy Optimization (PPO) (Schulman et al., 2017). At each control step, the teacher receives the current privileged full-state robot observation $\mathbf{o}_t^{\text{full}}$, a dense future motion command $\mathbf{c}_t^{\text{dense}}$, and a representation $\mathbf{m}_t^T$ of the recent state–action history $\mathcal{H}_t$. The dense command specifies the target motion of the tracked robot bodies over a short future horizon. The teacher policy and its history representation are given by

$$\mathbf{a}_t^T \sim \pi_{\theta_T}\left(\cdot \mid \mathbf{o}_t^{\text{full}}, \mathbf{c}_t^{\text{dense}}, \mathbf{m}_t^T\right), \qquad \mathbf{m}_t^T = E_{\phi_T}(\mathcal{H}_t), \tag{4}$$

where $\mathbf{a}_t^T$ denotes the teacher action and E_{ϕ_T} denotes the teacher history encoder. We use $\pi_{\theta,\phi}$ to denote the controller formed jointly by a policy and its history encoder. The teacher policy and history encoder are optimized over domain-randomized episodes according to

$$(\theta_T^\star, \phi_T^\star) = \arg\max_{\theta_T, \phi_T} \mathbb{E}_{\substack{\boldsymbol{\psi}^{(e)} \sim p_{\text{train}}(\Psi), \\ \tau \sim \pi_{\theta_T, \phi_T}}} \left[\sum_{t=0}^{T} \gamma^t r_t\right]. \tag{5}$$

The dynamics context $\boldsymbol{\psi}^{(e)}$ is sampled once per episode and is not provided to the teacher. Training across randomized dynamics produces the reference control behavior used to supervise the student in Stage II.

Stage II: Partial-Observation Student Distillation. We then freeze the teacher and distill its control behavior into a student policy π_{θ_S}. The student receives the current partial robot observation $\mathbf{o}_t^{\text{partial}}$ and replaces the dense full-body command with an SSJR command $\mathbf{c}_t^{\text{SSJR}}$ containing future targets for a selected set of body joints. It also conditions its action on a history representation $\mathbf{m}_t^S$:

$$\mathbf{a}_t^S = \pi_{\theta_S}\left(\mathbf{o}_t^{\text{partial}}, \mathbf{c}_t^{\text{SSJR}}, \mathbf{m}_t^S\right), \qquad \mathbf{m}_t^S = E_{\phi_S}(\mathcal{H}_t), \tag{6}$$

where E_{ϕ_S} denotes the student history encoder. The teacher and student history encoders are separately parameterized and jointly trained with their respective policies. During online distillation, trajectories are collected with the student, while the frozen teacher is queried at the states visited by the student using the corresponding dense motion command. The student minimizes the action discrepancy

$$\mathcal{L}_{\text{distill}}(\theta_S, \phi_S) = \mathbb{E}_{\substack{\boldsymbol{\psi}^{(e)} \sim p_{\text{train}}(\Psi), \\ \tau \sim \pi_{\theta_S, \phi_S}}} \left[\sum_{t=0}^{T} \left\|\mathbf{a}_t^S - \mathbf{a}_t^T\right\|_2^2\right]. \tag{7}$$

This stage transfers the teacher's dense-reference tracking behavior to a student that operates using partial observations and sparse motion commands. At deployment, only the student controller is retained. Because it requires neither privileged state information nor dense full-body references, HoRD provides a practical control interface for real-world humanoid deployment.

3.3 STANDARDIZED SPARSE-JOINT REPRESENTATION

Dense full-body references provide detailed targets for motion tracking, but requiring a complete reference pose as the deployment command is restrictive. Sparse body targets require less reference information, yet leave the motion of uncommanded body parts underdetermined. We introduce the Standardized Sparse-Joint Representation (SSJR) to address this tension. During distillation, SSJR restricts the student to a fixed set of future body targets, while the dense full-body reference is provided only to the teacher. SSJR therefore provides a structured sparse command representation: ***it preserves the future targets of selected key joints while requiring the student to recover the remaining whole-body coordination through distillation***.

Let $\mathcal{J}$ denote a fixed set of key joints shared across motion sequences, and let H denote the look-ahead horizon. At time t, SSJR represents the future motion command as

$$\mathbf{c}_t^{\mathrm{SSJR}} = \left\{ \mathbf{q}_{j,t+h}^{\mathrm{cmd}} \right\}_{j \in \mathcal{J},\, h=1,\ldots,H}, \tag{8}$$

where $\mathbf{q}_{j,t+h}^{\mathrm{cmd}}$ denotes the relative target configuration of key joint j at future timestep $t+h$. In contrast to the dense command $\mathbf{c}_t^{\mathrm{dense}}$, which specifies future targets for the full set of tracked robot joints, SSJR retains only a sparse set of task-relevant joints.

In this work, we focus on the Unitree G1 and define $\mathcal{J}$ as the pelvis, left and right hands, and left and right ankles. These key joints compactly represent root motion, upper-body intent, and foot-placement targets. Conditioned on the SSJR command, proprioceptive observations, and recent state–action history, the student predicts target joint positions that are executed by the robot's PD controllers.

Unlike variable-mask motion-inpainting interfaces or multi-mode command spaces (Tessler et al., 2024; He et al., 2025c), SSJR uses a fixed, future-conditioned key-joint representation throughout student training and deployment. Whereas three-point teleoperation interfaces primarily specify head and hand motion (He et al., 2025b), SSJR specifies hand, pelvis, and ankle trajectories to represent upper-body intent, root motion, and foot-placement evolution. This fixed sparse command representation defines the information retained at deployment and enables dense-to-sparse transfer of the teacher's whole-body control behavior.

3.4 HISTORY-CONDITIONED DYNAMICS REPRESENTATION

Variations in robot and environmental dynamics can cause the same control action to produce different state transitions. Because the underlying dynamics context is not observable at deployment, HoRD conditions both the teacher and student policies on recent interaction history through a History-Conditioned Dynamics Representation (HCDR). Rather than explicitly estimating physical parameters, HCDR learns a control-oriented representation of how the robot has responded to recent actions.

Episode-Level Dynamics Randomization. At the beginning of each training episode, we sample a dynamics context $\boldsymbol{\psi}^{(e)} \sim p_{\mathrm{train}}(\Psi)$ and hold it fixed throughout the episode. The context determines physical properties such as body-mass scaling, actuator gains, control delay, and contact parameters. Because the observations and actions within a history window are generated under the same transition dynamics, the history provides temporally coherent information about the robot's response under the current dynamics context. The sampled context $\boldsymbol{\psi}^{(e)}$ is never provided directly to either policy. The specific randomization parameters and ranges are reported in Appendix C.2.

History Representation. HCDR maintains a set of learnable query tokens $\mathbf{m}_0 \in \mathbb{R}^{N \times d_m}$ and uses them to extract a representation from the recent state–action history. Let $\mathbf{H}_t = \mathrm{Embed}(\mathcal{H}_t) \in \mathbb{R}^{K \times d_m}$ denote the encoded history tokens. A cross-attention layer computes

$$\mathbf{A}_t = \mathrm{softmax}\left(\frac{(\mathbf{m}_0 \mathbf{W}_Q)(\mathbf{H}_t \mathbf{W}_K)^\top}{\sqrt{d_h}} \right), \qquad \widetilde{\mathbf{m}}_t = \mathbf{A}_t (\mathbf{H}_t \mathbf{W}_V), \tag{9}$$

where $\mathbf{A}_t \in \mathbb{R}^{N \times K}$ contains the attention weights between the learnable queries and the history tokens. The attended features are subsequently processed by residual, normalization, and feed-forward layers:

$$\mathbf{m}_t = \text{Q-FormerBlock}\left(\mathbf{m}_0, \widetilde{\mathbf{m}}_t\right). \tag{10}$$

The resulting tokens $\mathbf{m}_t$ form a fixed-size summary of the recent interaction history and are supplied to the policy as its history representation.

Policy Conditioning. The resulting history representations $\mathbf{m}_t^T$ and $\mathbf{m}_t^S$ condition the teacher and student policies, respectively, as specified in Eqs. 4 and 6. At each control step, the newest observation–action pair is added to the history window, and HCDR recomputes the corresponding representation. This allows each policy to select its joint-position targets based on recent action–response patterns that are not fully captured by the instantaneous observation. Because the randomized dynamics context $\boldsymbol{\psi}^{(e)}$ is never provided to the policies, HCDR realizes implicit history-conditioned adaptation rather than explicit dynamics-parameter estimation.

4 Experiments

We evaluate HoRD from four complementary perspectives: whole-body tracking under simulator and dynamics shifts, the information retained by SSJR, the role of online distillation, and the contribution of history-conditioned adaptation. We additionally evaluate the learned student policy on a physical Unitree G1. Architectural details, training hyperparameters, and complete evaluation protocols are provided in the appendix.

4.1 Experimental Setup

Robot and Environment. All experiments use a Unitree G1 humanoid with 29 actuated degrees of freedom. The policy operates at 50 Hz and predicts target joint positions that are executed by joint-level PD controllers. Training is performed in Isaac Lab (Mittal et al., 2025), while cross-simulator evaluation is conducted in Genesis (Genesis Authors, 2024) without test-domain retraining or fine-tuning.

Dataset. We use motion sequences from AMASS (Mahmood et al., 2019). The motions are retargeted to the Unitree G1 and divided into training and held-out test sets following the protocol described in Appendix C.4. The teacher receives dense full-body motion references, whereas the student receives SSJR commands specifying future trajectories for the pelvis, hands, and ankles.

Baselines. We compare HoRD with MaskedMimic (Tessler et al., 2024), OmniH2O (He et al., 2025b), ExBody2 (Ji et al., 2024), HOVER (He et al., 2025c), GMT/ResMimic (Zhao et al., 2025), and VisualMimic (Yin et al., 2025). MaskedMimic, OmniH2O, ExBody2, and HOVER are adapted from their released implementations, whereas GMT/ResMimic and VisualMimic are reimplemented in our codebase. All evaluated policies use the same Unitree G1 embodiment, and share the same preprocessing. Implementation adaptations are documented in Appendix A.4.

Evaluation Metrics. We report strict tracking Success Rate (Succ.), Global Mean Per-Joint Position Error $\mathbf{E}_{\text{g-mpjpe}}$, and Local Mean Per-Joint Position Error $\mathbf{E}_{\text{mpjpe}}$. An episode is successful only if its global tracking error remains below 0.5 m throughout the motion. Complete metric definitions and rollout counts are provided in Appendix B.2.

4.2 Main Results on AMASS

All methods in Table 1 are trained in Isaac Lab using the same episode-level domain-randomization distribution. **IsaacLab (ID)** evaluates the policies in the training simulator under a single fixed dynamics configuration shared across all evaluation episodes. In contrast, **IsaacLab + DR (ID)** samples the dynamics parameters from the training distribution at the beginning of each evaluation episode. **Genesis (OOD)** evaluates the same policies zero-shot in the unseen Genesis simulator using a single fixed dynamics configuration, while **Genesis + DR (OOD)** combines the simulator shift with episode-level dynamics randomization. Here, "ID" and "OOD" indicate whether the simulator is observed during training, whereas "+DR" indicates that the dynamics context is resampled for each evaluation episode. All Genesis evaluations use policies trained only in Isaac Lab, without retraining or fine-tuning.

Table 1 reports whole-body tracking performance on 873 held-out AMASS motions. HoRD achieves the highest success rate and the lowest global and local pose errors in all four conditions. Under **IsaacLab (ID)**, HoRD achieves 90.7% success, compared with 89.8% for the strongest baseline,

Method	IsaacLab (ID)			IsaacLab + DR (ID)			Genesis (OOD)			Genesis + DR (OOD)		
	Succ. (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓
VisualMimic	31.5	388	190	<10	721	304	<10	736	316	<10	732	317
MaskedMimic	32.1	376	182	<10	713	283	<10	742	322	<10	738	326
OmniH2O	85.2	266	132	83.2	282	168	72.3	312	165	70.2	335	191
ExBody2	86.6	247	108	85.9	273	128	73.1	305	157	69.4	342	202
Hover	71.2	278	138	67.9	375	196	16.2	722	258	15.5	746	282
GMT / ResMimic	89.8	113	81	87.4	178	119	82.2	288	152	78.4	321	184
HoRD (Ours)	**90.7**	**102**	**76**	**88.4**	**124**	**87**	**86.0**	**162**	**96**	**84.4**	**171**	**108**

Table 1: **Overall performance** on the AMASS test split. Genesis columns are zero-shot (no test-domain retraining). Best results are bold; second-best are underlined.

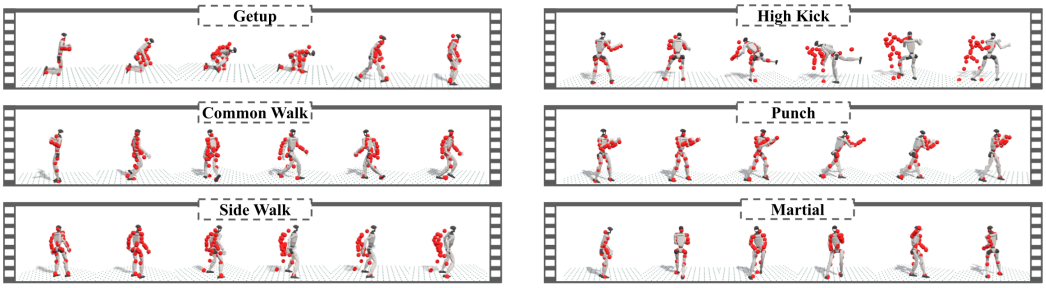

Figure 2: **Representative motion rollouts in Genesis (OOD).** The rendered humanoid shows the executed pose, while the red markers indicate the corresponding reference keypoints.

GMT/ResMimic. The success-rate margin increases under cross-simulator transfer: HoRD outperforms GMT/ResMimic by 3.8 percentage points under **Genesis (OOD)** (86.0% versus 82.2%) and by 6.0 percentage points under **Genesis + DR (OOD)** (84.4% versus 78.4%). In the latter condition, HoRD also reduces global pose error from 321 mm to 171 mm and local pose error from 184 mm to 108 mm.

VisualMimic and MaskedMimic fall below 10% success under randomized or cross-simulator evaluation. OmniH2O, ExBody2, and HOVER also degrade more substantially than HoRD in Genesis.

Figure 2 presents qualitative rollouts of HoRD on six representative behaviors in Genesis (OOD): common walk, side walk, get-up, punch, high kick, and martial arts. These examples span locomotion, ground recovery, upper-body actions, and motions involving rapid center-of-mass shifts and single-leg support. The executed robot poses remain closely aligned with the reference keypoints across these different motion patterns, including contact transitions during get-up and high kick. Full motion-level comparisons with baseline policies are provided in Appendix A.2.

4.3 Ablation under Dynamics Shift

Domain Randomization and History Representation. We evaluate two ablated variants to isolate the contributions of episode-level domain randomization and HCDR. *HoRD w/o D* removes domain randomization during training while retaining the history-conditioned policy. *HoRD w/o H* removes the HCDR representation from the policy input while retaining the same domain-randomized training distribution. All other training and evaluation settings remain unchanged.

Table 2 shows that domain randomization and history conditioning play complementary roles. Removing domain randomization reduces success from 90.7% to 79.8% under **IsaacLab (ID)** and to below 10% in all randomized and cross-simulator conditions. This result shows that exposure to diverse dynamics during training is necessary for generalization beyond a single fixed dynamics configuration.

Removing HCDR has a different effect. Under **IsaacLab (ID)**, *HoRD w/o H* achieves 91.3% success, which is comparable to the 90.7% achieved by the complete model. Its performance, however, decreases to 70.5% under **IsaacLab + DR (ID)** and to below 10% in both Genesis conditions.

Method	IsaacLab (ID)			IsaacLab + DR (ID)			Genesis (OOD)			Genesis + DR (OOD)		
	Succ. (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓
HoRD w/o D	79.8	186	136	<10	832	286	<10	838	316	<10	838	287
HoRD w/o H	**91.3**	137	97	70.5	341	182	<10	846	324	<10	433	234
HoRD	90.7	**102**	**76**	**88.4**	**124**	**87**	**86.0**	**162**	**96**	**84.4**	**171**	**108**

Table 2: Ablation study of HoRD across different environments. We analyze the contribution of domain randomization (D) and the HCDR module (H).

History conditioning therefore contributes little under the fixed in-distribution dynamics configuration but becomes important when the dynamics parameters or simulator differ from the fixed Isaac Lab evaluation setting. Together, these results indicate that domain randomization provides the dynamics diversity required during training, while HCDR allows the policy to condition its actions on the robot's recent responses at execution time.

History Encoder Design. To compare encoder designs while holding history availability fixed, we retain the same state–action history and replace only the Q-Former with alternative encoders. All variants use the same history length, latent width, motion-command interface, downstream policy architecture, training budget, and evaluation protocol. We compare the four-layer Q-Former with an RMA-style MLP encoder (Kumar et al., 2021), a four-layer Transformer encoder (Vaswani et al., 2017), a two-layer GRU (Cho et al., 2014), and a two-layer LSTM (Graves, 2012).

Table 3 reports the history-encoder ablation on the six-motion diagnostic subset in Genesis + DR, which is distinct from the full AMASS test split used in Table 1. Among the alternative encoders, the RMA-style MLP performs best, achieving a success rate of 63.3%. The four-layer Q-Former achieves 95.0% success and reduces $\mathbf{E}_{\text{g-mpjpe}}$ from 277 mm to 134 mm and $\mathbf{E}_{\text{mpjpe}}$ from 193 mm to 98 mm. These results identify the four-layer Q-Former as the strongest of the evaluated encoder configurations under this targeted OOD protocol.

Encoder	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓
MLP (RMA-style, 4 layers)	63.3	277	193
GRU (2 layers)	23.3	667	238
LSTM (2 layers)	21.6	655	242
Transformer (4 layers)	<10	728	323
HoRD (Q-Former, 4 layers)	**95.0**	**134**	**98**

Table 3: History encoder ablation on the six-motion Genesis + DR diagnostic subset, using ten rollouts per motion. Best results are bold; second-best results are underlined.

Further implementation details, diagnostic metrics, and training convergence analyses are provided in Appendix A.3, and Appendix C.5.

4.4 Robustness to Terrain Variation and Root-Velocity Perturbations

We further evaluate HoRD under terrain variation and transient root-velocity perturbations.

Terrain Variation. We evaluate zero-shot Genesis transfer on flat ground, a smooth slope, and a rough slope. Success decreases from 86.2% on flat ground to 85.8% and 84.2% on the two slopes (Table 4). Figure 3 shows representative rollouts under these conditions. HoRD maintains stable motion on both inclines while adapting its posture and foot placement to the terrain geometry.

Terrain	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓
Flat ground	86.2	167	102
Smooth slope	85.8	171	106
Rough slope	84.2	180	112

Table 4: Terrain robustness under zero-shot Genesis transfer.

Root-Velocity Perturbations. We perturb the robot's root velocity to approximate an external push. HoRD recovers stable motion in 85.2% of trials, compared with 71.8% for OmniH2O and 72.1% for ExBody2. Figure 4 shows a representative recovery.

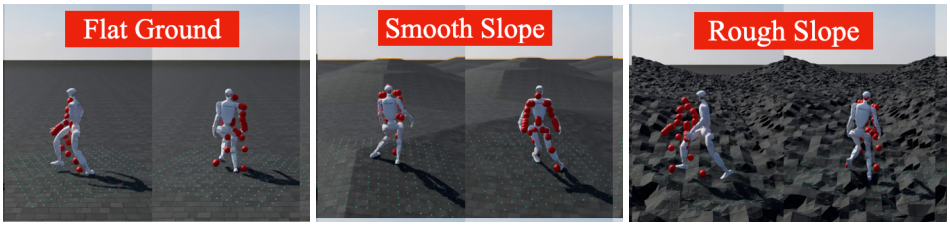

Figure 3: **Terrain robustness under zero-shot Genesis transfer.** Representative frames on flat ground, a smooth incline, and a rough incline, shown from left to right.

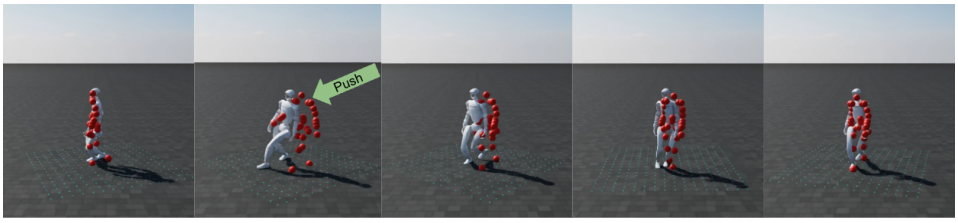

Figure 4: **Recovery from a transient root-velocity perturbation.** HoRD re-stabilizes and resumes reference tracking after the perturbation is introduced in the second frame.

4.5 Real-World Deployment

We deploy the student policy, trained only in IsaacLab with domain randomization, on a physical Unitree G1 at 50 Hz without real-world fine-tuning. The robot receives SSJR keypoint trajectories retargeted from AMASS clips through the same command interface used in simulation. The supplementary video presents qualitative outdoor rollouts on flat hard ground, on flat ground with an added load, and on uneven grass and soil while carrying the same load.



Figure 5: **Real-world deployment on a Unitree G1.** Representative outdoor rollouts on uneven grass and soil with an added load: (a) walking, (b) waving, (c) punching, and (d) a martial-arts motion.

Figure 5 shows representative walking, waving, punching, and martial-arts motions in the uneven-ground setting with the added load. These rollouts demonstrate the feasibility of deploying the same student policy on hardware under non-nominal outdoor conditions. Additional deployment details and limitations are provided in Appendix D.

5 Conclusion

We presented HoRD, a two-stage learning framework for whole-body humanoid control under partial observability and variations in robot and environmental dynamics. A dense-reference teacher is trained under episode-level domain randomization, and its control behavior is distilled into a deployable student policy that operates using partial observations and SSJR commands. SSJR provides a structured sparse command representation, while HCDR enables implicit history-conditioned adaptation from recent state–action history. Across 873 held-out AMASS motions, HoRD achieves SOTA performance across all four evaluation conditions, including zero-shot cross-simulator transfer to Genesis under both fixed and randomized dynamics. Ablations show the complementary roles of episode-level domain randomization and HCDR, while qualitative Unitree G1 rollouts illustrate deployment on a physical humanoid.

AI USE STATEMENT

We used generative AI tools solely for language polishing and limited assistance with code generation. All AI-assisted content was reviewed and verified by the authors, and we take full responsibility for the final manuscript and all associated claims, code, and artifacts.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide our implementation code in the supplementary materials, together with demonstration videos of the real-world experiments. Detailed training settings are provided in Appendix C.

REFERENCES

Luigi Campanaro, Siddhant Gangapurwala, Wolfgang Merkt, and Ioannis Havoutis. Learning and deploying robust locomotion policies with minimal dynamics randomization. In *6th Annual Learning for Dynamics & Control Conference*, pp. 578–590. PMLR, 2024.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8973–8979. IEEE, 2019.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Genesis Authors. Genesis: A generative and universal physics engine for robotics and beyond, December 2024. URL `https://github.com/Genesis-Embodied-AI/genesis-world`.

Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025a.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Conference on Robot Learning*, pp. 1516–1540. PMLR, 2025b.

Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9989–9996. IEEE, 2025c.

Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castaneda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

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

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *Acm transactions on graphics (tog)*, 36(4):1–13, 2017.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

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

Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions On Graphics (TOG)*, 43(6):1–21, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 39(4):33–1, 2020.

Shaofeng Yin, Yanjie Ze, Hong-Xing Yu, C Karen Liu, and Jiajun Wu. Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation. *arXiv preprint arXiv:2509.20322*, 2025.

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

Siheng Zhao, Yanjie Ze, Yue Wang, C Karen Liu, Pieter Abbeel, Guanya Shi, and Rocky Duan. Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning. *arXiv preprint arXiv:2510.05070*, 2025.

APPENDIX

A ADDITIONAL RESULTS

A.1 QUALITATIVE COMPARISONS

Figure 6 provides additional execution-level comparisons of HoRD, MaskedMimic, and ExBody2 on six representative motions. The sequences complement the aggregate AMASS results in Table 1 by showing how tracking errors appear over the course of individual executions.



(a) Punch (b) Get Up

(c) High Kick (d) Common Walk

(e) Side Walk (f) Martial Arts

Figure 6: **Qualitative comparisons on six representative motions.** Each panel compares the reference motion with executions produced by HoRD, MaskedMimic, and ExBody2.

A.2 QUALITATIVE COMPARISONS

Figure 6 shows representative rollouts on the six-motion suite from Section 4.2. We compared HoRD with MaskedMimic and ExBody2 to present quantitative trends from a visual perspective at the execution level, the results for the complete baseline are shown in Table 1.

Metric	HoRD	HoRD w/o D	HoRD w/o H
Cartesian Error (m) ↓	0.087	0.286	0.182
Global Rotation Error (rad) ↓	0.369	1.594	1.011
Global Translation Error (m) ↓	0.124	0.832	0.341
DoF Velocity Reward ↑	0.972	0.980	0.960
Key Body Reward ↑	0.907	0.412	0.612
Local Rotation Reward ↑	0.833	0.274	0.507
Root Angular Velocity Reward ↑	0.480	0.722	0.412
Root Velocity Reward ↑	0.954	0.878	0.904

Table 5: **Additional test-set metrics for HoRD and its ablation variants.** All policies are evaluated on held-out AMASS motions in **IsaacLab + DR (ID)**.

	Stress			Switching		
Method	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓	**Succ.** (%)↑	$\mathbf{E}_{\text{g-mpjpe}}$ (mm)↓	$\mathbf{E}_{\text{mpjpe}}$ (mm)↓
MaskedMimic	<10	734	332	<10	753	324
OmniH2O	61.4	361	<u>206</u>	<u>53.2</u>	384	<u>223</u>
ExBody2	<u>62.2</u>	<u>353</u>	214	51.8	<u>381</u>	237
HOVER	13.8	742	292	<10	697	314
HoRD (Ours)	**81.2**	**214**	**134**	**80.6**	**212**	**152**

Table 6: **Stress and mid-episode dynamics-change diagnostics in Genesis.** Results use six motions and ten rollouts per motion. Best results are bold; second-best results are underlined.

A.3 Additional Test-Set Metrics

Table 5 reports additional diagnostic metrics for HoRD and its two ablation variants on held-out AMASS motions in **IsaacLab + DR (ID)**. These quantities complement the success rate and pose errors reported in the main paper.

A.4 Baseline Implementation and Evaluation Protocol

All methods in Table 1 use the Unitree G1 embodiment, the same held-out AMASS split, and the same evaluation initializations, termination criterion, and metrics. Training uses Isaac Lab and the episode-level dynamics-randomization distribution in Table 7; Genesis evaluation is zero-shot, without retraining or fine-tuning in the target simulator. Where applicable, we keep the environment-step budget and dynamics-randomization envelope fixed across methods while retaining each method's own policy architecture and motion-command formulation.

MaskedMimic, OmniH2O, ExBody2, and HOVER are adapted from their released implementations to the common G1 evaluation setup. GMT/ResMimic and VisualMimic are local reimplementations based on their published descriptions. GMT and ResMimic share the same local controller and evaluation wiring in our setup and are therefore reported as one row. Accordingly, the table compares adapted or reimplemented systems under a common evaluation protocol rather than reproducing numbers from their original papers or evaluating official checkpoints unchanged.

A.5 Stress and Mid-Episode Dynamics Changes

We further consider two Genesis diagnostics beyond the four main evaluation conditions. In **Stress**, dynamics parameters are sampled outside the training ranges in Table 7. In **Switching**, the dynamics context is resampled every 3–6 s within an episode. Table 6 reports results on the same six-motion subset used for the history-encoder diagnostic, with ten rollouts per motion.

HoRD obtains the highest success rate and the lowest two pose errors under both diagnostic protocols. These results are reported separately from the full AMASS benchmark because they use a smaller, targeted motion subset.

B Environment Details

B.1 Observations, Actions, and Termination

Observation. The teacher receives privileged full-state robot observations, whereas the student receives partial observations available at deployment. Proprioceptive inputs include base motion, joint positions and velocities, and the previous action. Motion commands are supplied separately: the teacher receives the dense future motion command and the student receives SSJR. HCDR processes the recent observation–action history described in Section 3.4.

Action and actuation model. The Unitree G1 model has 29 actuated degrees of freedom. The policy predicts target joint positions $\boldsymbol{\alpha}_t$, which the joint-level proportional–derivative (PD) controllers convert into torques:

$$\boldsymbol{\tau}_t = \mathbf{K}_p\big(\boldsymbol{\alpha}_t - \mathbf{q}_t\big) - \mathbf{K}_d\,\dot{\mathbf{q}}_t,$$

where $\mathbf{q}_t$ and $\dot{\mathbf{q}}_t$ are the current joint positions and velocities. Stiffness is set to 100 for the hips, 200 for the knees, 400 for the waist, and 60–90 for the shoulders; damping coefficients range from 0.1 to 5.0 across joint groups.

Termination. Each episode follows the duration of its target motion clip and terminates early if the global tracking error exceeds 0.5 m.

B.2 Metrics

We use task-level success and pose-tracking errors as the primary evaluation metrics. Unless otherwise stated, expectations are taken over joints, time, and evaluation episodes.

Strict Tracking Success. Our primary metric is the strict success rate. For an episode i, let $\mathrm{GTE}_t^{(i)}$ denote the global translation error at time t (defined below). The episode is counted as successful if the per-frame error never exceeds the threshold $\varepsilon_{\mathrm{GTE}}$=0.5 m,

$$\text{success}^{(i)} = \mathbb{I}\Big[\max_{t\le T^{(i)}} \mathrm{GTE}_t^{(i)} < \varepsilon_{\mathrm{GTE}}\Big],$$

and the strict success rate over N episodes is

$$\text{success rate} = \frac{1}{N}\sum_{i=1}^{N} \text{success}^{(i)}.$$

Mean Per-Joint Position Error. Local pose error is computed after removing the initial root translation:

$$E_{\mathrm{mpjpe}} = \mathbb{E}\Big[\big\|(\hat{p}_t^{\mathrm{ref}} - \hat{p}_0^{\mathrm{ref}}) - (\hat{p}_t - \hat{p}_0)\big\|_2\Big],$$

where $\hat{p}_t$ and $\hat{p}_t^{\mathrm{ref}}$ are local-frame joint positions for the executed and reference motions, respectively.

Global mean per-joint position error. Global pose error is computed from joint positions in world coordinates:

$$E_{\text{g-mpjpe}} = \mathbb{E}\Big[\big\|p_t^{\mathrm{ref}} - p_t\big\|_2\Big],$$

and GTE_t denotes the corresponding per-frame global tracking error used in the strict success criterion.

Global Rotation Error. The additional rotation error in Table 5 is

$$E_{\mathrm{gr}} = \mathbb{E}\big[\Delta q(gr_t, gr_t^{\mathrm{ref}})\big],$$

where $\Delta q(\cdot,\cdot)$ denotes the angular distance between two unit quaternions.

Aspect	Parameter	Range / distribution
Control delay	d_{ctrl}	$\{0, 1, 2, 3\}$ physics steps
Body mass scale	s_m	$\mathcal{U}[0.9, 1.1]$
Joint parameter scale	s_j	$\mathcal{U}[0.9, 1.1]$
Actuator gain scale	s_a	$\mathcal{U}[0.9, 1.1]$
Gravity (downwards)	g_z	$\mathcal{U}[9.7, 9.9]$ m/s^2
External push interval	Δt_{push}	$\mathcal{U}[5, 10]$ s

Table 7: **Episode-level training randomization.** The same ranges are used during randomized training and the +DR evaluation conditions.

Reward-Aligned Kinematic Terms. For completeness, Table 5 also reports exponentiated tracking terms that form part of the teacher's RL objective. They are not used as primary evaluation metrics:

$$\begin{aligned}
\text{dv_rew} &= \exp\Big(-c_{\text{dv}} \left\|\dot{q}_t - \dot{q}_t^{\text{ref}}\right\|_2^2\Big), \\
\text{kb_rew} &= \exp\Big(-c_{\text{kb}} \left\|b_t - b_t^{\text{ref}}\right\|_2^2\Big), \\
\text{lr_rew} &= \exp\big(-c_{\text{lr}}\, \Delta q(lr_t, lr_t^{\text{ref}})\big), \\
\text{rv_rew} &= \exp\Big(-c_{\text{rv}} \left\|v_t^{\text{root}} - v_t^{\text{root,ref}}\right\|_2^2\Big), \\
\text{rav_rew} &= \exp\Big(-c_{\text{rav}} \left\|\omega_t^{\text{root}} - \omega_t^{\text{root,ref}}\right\|_2^2\Big).
\end{aligned}$$

Here $\dot{q}_t$ denotes joint velocities, b_t denotes key-body positions, lr_t denotes local joint rotations, and v_t^{root} and ω_t^{root} denote root linear and angular velocities. The coefficients $c_{\text{dv}}, c_{\text{kb}}, c_{\text{lr}}, c_{\text{rv}}, c_{\text{rav}}$ are positive scaling constants shared across methods.

C Training Details

C.1 Reward Design

The teacher objective combines tracking rewards, constraint penalties, and a control regularizer:

- **Tracking rewards:** global translation (GT), 0.5; global rotation (GR), 0.4; global velocity (GV), 0.1; global angular velocity (GAV), 0.1; and root height (RH), 0.2.
- **Constraint penalties:** global rotation (GR_c), -10; global translation (GT_c), -100; root tracking (RT_c), -120; and root height (RH_c), -100.
- **Regularization:** power penalty, 5×10^{-6}.

C.2 Domain Randomization

The physical parameters and disturbance schedule in Table 7 are sampled once at the beginning of each randomized episode. The sampled context is not provided directly to either policy.

C.3 Optimization and Implementation Settings

Table 8 summarizes the optimization, control, and simulator settings. Architecture-specific history-encoder variants are described with their corresponding ablation in the main paper.

Category	Values
Training setup	Environment count: 4,096; maximum steps: 1×10^{9} Simulator FPS: 200; decimation: 4; mixed-terrain curriculum
Optimization	PPO $\gamma = 0.99$, GAE $\lambda = 0.95$, clip $\epsilon = 0.2$ Actor LR: 2×10^{-5}; critic LR: 1×10^{-4} Batch size: 16,384; mini-epochs: 1; gradient clip: 50.0
Reward shaping	Weights: GT=0.5, GR=0.4, GV=0.1, GAV=0.1, RH=0.2 Penalties: $\mathrm{GR}_c = -10$, $\mathrm{GT}_c = -100$, $\mathrm{RT}_c = -120$, $\mathrm{RH}_c = -100$ Power penalty = 5×10^{-6}
Control and robot	Actions: target positions for 29 DoFs; joint-level PD control Stiffness: hips=100, knees=200, waist=400, shoulders=60–90 Damping: 0.1–5.0 depending on joint group
Terrain	Terrain types: [smooth slope, rough slope, stairs up, stairs down, discrete, stepping, poles, flat] Terrain proportions: [0.25, 0.25, 0., 0., 0., 0., 0., 0.5]

Table 8: Key training hyperparameters.

.

C.4 Motion Dataset Processing

We retarget SMPL-format AMASS sequences to the Unitree G1 representation using global translations and joint rotations. We then apply the filtering procedure used by PHC and MaskedMimic and follow the MaskedMimic train–test split.

Motion sequences are segmented into clips of 2–10 seconds. Clips are discarded if retargeting violates joint limits or produces persistent contact penetration in more than 5% of frames. The remaining clips are downsampled to 30 Hz. Dense retargeted trajectories are retained for teacher training, and the selected pelvis, hand, and ankle trajectories are used to construct the student's SSJR commands.

C.5 Training Procedure and Convergence

Optimization and distillation. The teacher policy and its history encoder are optimized jointly with PPO using dense motion references and episode-level domain randomization. After teacher training, both components are frozen. Student trajectories are then collected online, and the frozen teacher supplies action targets at the states visited by the student using the corresponding dense motion command. The student policy and its separately parameterized history encoder are optimized jointly with the action-distillation loss in Eq. 7.

Training resources. Training is performed on a single NVIDIA A100 80 GB GPU using PyTorch 2.3. The policies operate and are evaluated at 50 Hz.

Convergence diagnostics. HoRD and its two ablation variants are trained for 10^{9} simulation steps. Figure 7 reports the strict tracking success rate during training. Policies are evaluated on the training distribution every 500 epochs, corresponding to approximately 65.54 million simulation steps between evaluations. The complete HoRD model generally improves after the initial training phase and reaches the highest success rate among the three configurations at later checkpoints. Both ablations initially follow a similar trend but remain lower during later training.

We additionally record tracking-error and reward-aligned quantities throughout training. At later checkpoints, HoRD attains the lowest Cartesian, global-rotation, and global-translation errors. Its clearest reward advantages appear in key-body, local-rotation, and root-velocity tracking, while the DoF-velocity reward varies within a

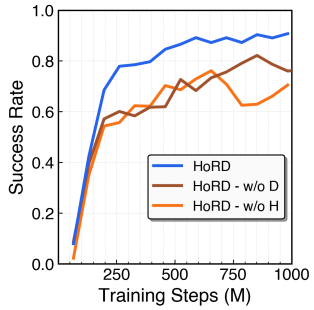

Figure 7: **Strict success during training.** Periodic evaluation on the training distribution.

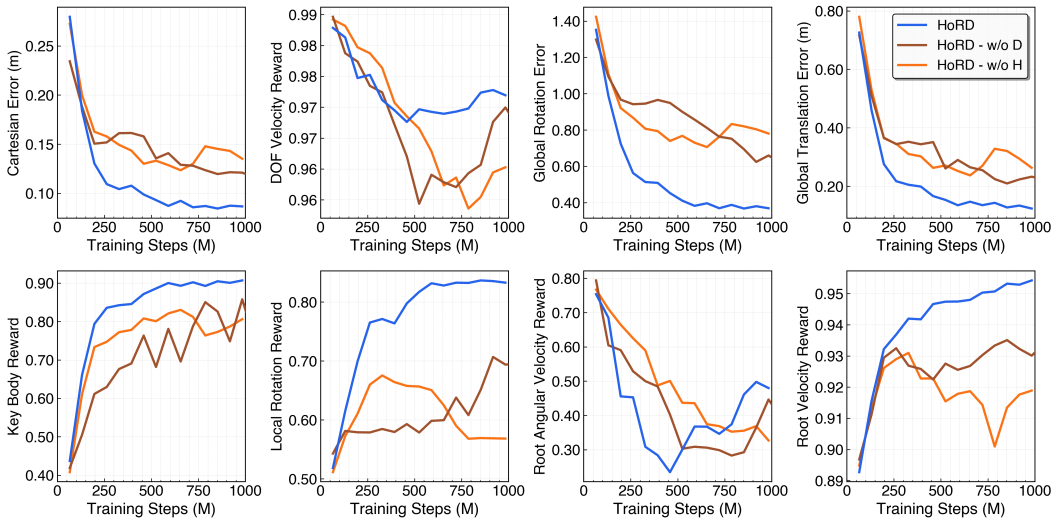

Figure 8: **Training-error and reward diagnostics for HoRD and its ablation variants.** The panels report Cartesian, global-rotation, and global-translation errors (lower is better), together with DoF-velocity, key-body, local-rotation, root-angular-velocity, and root-velocity rewards (higher is better). D denotes domain randomization and H denotes HCDR.

narrower range. The root-angular-velocity reward is non-monotonic for all three configurations, but the complete model recovers to the highest final value. Figure 8 summarizes these optimization diagnostics.

C.6 Release Assets

The supplementary release will include the training and evaluation code, configuration files, motion-split manifests manifests, evaluation commands, the SSJR schema, and the teacher and student checkpoints used for the Unitree G1 experiments. Derived AMASS assets will be distributed only where permitted by the licenses of the source datasets (Mahmood et al., 2019).

D Qualitative Real-World Deployment

The deployed student runs at 50 Hz on the physical Unitree G1 without real-world fine-tuning. It receives the same partial observations and SSJR command format used in simulation. The montage in Figure 5 illustrates executions on hard ground and uneven outdoor terrain, including trials with an added load.

The hardware evidence is qualitative: the payload and terrain conditions are not instrumented, and we do not report repeated-trial success rates, tracking errors, or matched baseline results on the physical robot. We therefore use these rollouts to demonstrate deployment feasibility rather than to support the quantitative robustness claims, which are based on the Isaac Lab and Genesis evaluations.