
Humanoid Horizon: Extending Task Horizon in Whole-Body Loco-Manipulation via Parallel Training, Dynamic Starting, and Reward Gating

Haozhuo Zhang
The University of Manchester

Qiang Zhang
University of Science and
Technology of China

Jian Tang
X-Humanoid

Mingzhe Ni
The University of Manchester

Michele Caprio
University of Warwick

Angelo Cangelosi*
The University of Manchester

Wei Pan*
Newcastle University

Abstract

Cluttered indoor environments, where large and heavy objects are scattered across diverse surfaces, require humanoid robots to sequentially navigate, grasp, transport, and accurately place each item at its target location within a single uninterrupted episode. This long-horizon, whole-body loco-manipulation task remains a significant challenge for current methods. Previous approaches often suffer from two main issues: easy-reward bias, where training overemphasizes early transport stages at the expense of later ones, and catastrophic forgetting, where focusing on later stages leads to a decline in earlier-stage performance. In this work, we introduce Humanoid Horizon, a unified policy framework designed to overcome these limitations through three interrelated mechanisms. The Parallel Training Strategy organizes N scenes into S concurrent stage streams governed by a shared policy, ensuring all transport stages receive continuous gradient updates and removing the bottleneck of sequential optimization. The Dynamic Starting Mechanism updates each environment’s initial state with terminal states from upstream rollouts, gradually broadening transition coverage and enhancing robustness at stage boundaries. Reward Gating sets the reward to zero for the rest of the episode in later-stage streams when the immediately preceding object is displaced beyond a set threshold, so the shared policy learns not to disturb a just-placed object and earlier placements are preserved throughout the episode. Collectively, these strategies achieve per-stage success rates exceeding 80% on the two-object LHM-Humanoid benchmark (350 training scenes, 66 held-out scenes). As the number of sequentially transported objects grows beyond two, success declines with the horizon, but the degradation is graceful relative to the sharp drop seen in all baselines. Additionally, we demonstrate that the RL teacher policy can be distilled into a Vision-Language-Action student via DAGger. When provided only with egocentric RGB or depth observations and natural language instructions, the student successfully replicates the teacher’s long-horizon, multi-object behaviors, highlighting the potential for perception-driven deployment on real humanoid robots. Videos and results are available at <https://haozhuo-zhang.github.io/Humanoid-Horizon-project-page/>.

*Corresponding authors.

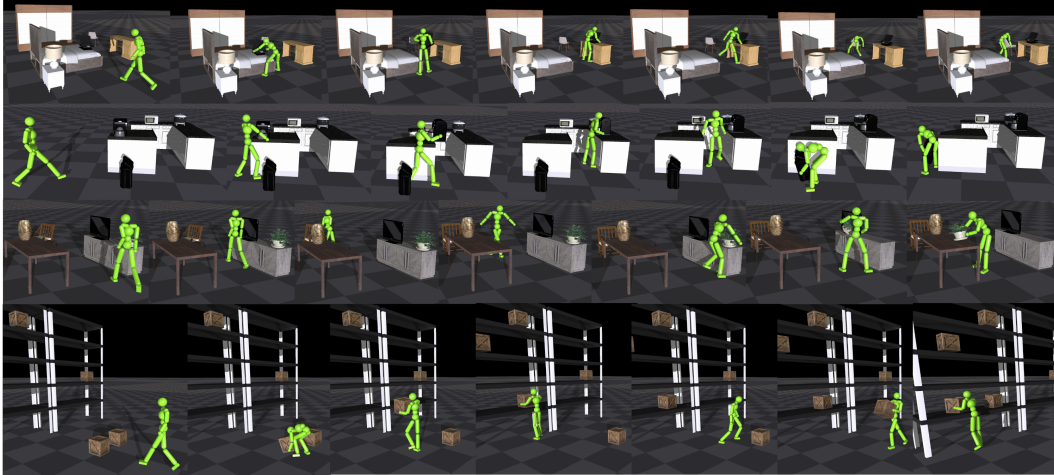


Figure 1: Humanoid Horizon uses a single policy for long-horizon whole-body loco-manipulation across cluttered bedroom, kitchen, living room, and warehouse scenes without resets. Each row shows one uninterrupted multi-object transport episode with sequential navigation, grasping, carrying, and placement.

1 Introduction

Humanoid agents that imitate human motion, perceive cluttered scenes, and manipulate objects are a central yet difficult research target. Prior work has advanced locomotion and control [1, 2, 3, 4], scene interaction [5, 6, 7, 8], and manipulation [9, 10, 11, 12]. A key gap remains: long-horizon whole-body loco-manipulation, i.e., continuously transporting multiple large objects over long task chains without environment resets.

This gap is fundamental. Compared with arms or quadrupeds, humanoids are less dynamically stable and must coordinate balance, locomotion, and manipulation throughout long episodes. As a result, long-horizon pipelines developed for more stable platforms are hard to transfer directly. Despite progress in interface and perception conditioning [9, 13], most current methods still focus on short-horizon single-object skills [14, 15, 16], often relying on fixed starts, and thus struggle with continuous multi-object loco-manipulation. When these methods are extended to longer task chains, two failure modes become prominent: easy-reward bias (over-optimizing early-stage transport while under-exploring later-stage behaviors) and catastrophic forgetting. In our context, catastrophic forgetting specifically means that a policy already capable of single-object transport is further trained from post-first-object terminal states to transport the next object, yet after this shift it loses the ability to re-execute first-object transport-and-release, even under substantial robot-initial-state randomization. LHM-Humanoid [17] provides a concrete instance of this behavior (e.g., teacher model 2), which then motivates dual-teacher distillation into a student and, in turn, constrains scalability to longer multi-object horizons.

We present Humanoid Horizon, a framework for long-horizon whole-body humanoid loco-manipulation that uses one policy to perform continuous long-horizon loco-manipulation across multiple objects, without resets, auxiliary models, or handcrafted task-specific modules; it has three coupled components: Parallel Training Strategy, which jointly optimizes all transport stages in one policy to avoid easy-reward bias in sequential curricula and forgetting risks in decoupled stage-wise training (including distillation-heavy pipelines such as LHM-Humanoid) [17], enabling scalable learning as object count grows; Dynamic Starting Mechanism, which writes each stage terminal state to the corresponding downstream environment buffer and initializes the next stage from that buffer in the next epoch, so coverage of transition states expands through temporal accumulation across epochs; and Reward Gating, which zeros the remaining reward in a later stage once the immediately preceding object is displaced beyond a tolerance, so the policy learns not to disturb a just-placed object and earlier placements are preserved over the episode.

We evaluate on the LHM-Humanoid benchmark [17]: 350 training/evaluation scenes and 66 unseen scenes for generalization, and we follow LHM-Humanoid to test longer-horizon settings with more than two sequentially transported objects. Combining these mechanisms yields per-stage success above 80% in two-object sequential transport. As more objects are carried in one episode, success declines with the horizon, but the degradation is graceful relative to the sharp drop seen in prior baselines, and a single policy still completes multi-object rearrangement across diverse scenes. As an exploratory deployment study, we distill the teacher policy into a vision-language-action (VLA) student using DAgger [18], and with egocentric RGB or depth inputs plus language instructions, the student preserves long-horizon multi-object behaviors learned by the teacher.

Our contributions are summarized as follows:

- We present **Humanoid Horizon**, a single-policy framework for long-horizon whole-body humanoid loco-manipulation that supports continuous multi-object rearrangement without resets, auxiliary models, or handcrafted task modules.
- We introduce a **Parallel Training Strategy** that jointly optimizes all transport stages, reducing easy-reward bias and catastrophic forgetting, and improving scalability to longer task horizons.
- We propose a **Dynamic Starting Mechanism** and **Reward Gating** that together improve inter-stage transition robustness and preserve previously completed placements throughout long-horizon execution.
- We demonstrate strong long-horizon performance on 350 benchmark scenes plus 66 unseen scenes, and show that DAgger-based VLA distillation transfers the learned behavior to egocentric RGB/depth inputs.

2 RELATED WORK

2.1 Physically-Based Motion Synthesis and Human-Scene Interaction

Motion synthesis spans computer graphics, vision, and robotics, and is commonly divided into kinematics-based methods[19, 20] and physics-based methods[2, 4, 1, 21]. Kinematic approaches emphasize semantic coherence, while physics-based approaches learn dynamically feasible control policies in simulation. Foundational systems such as AMP[1] and DeepMimic[21] have enabled later extensions with adversarial priors, latent skill representations, and language conditioning. In human-scene interaction, InterScene[8] and InterPhys[6] move beyond earlier inverse-kinematics or hand-designed controllers toward dynamic object interaction, and RL-based methods show strong manipulation skills[15, 14]. In parallel, embodied AI studies integrated navigation and manipulation[22, 23], including commonsense-guided policies[24]. However, most systems still use simplified embodiments (e.g., mobile bases with parallel grippers) and remain short-horizon and single-object. HumanVLA[9] advances humanoid manipulation of heavier objects but remains limited to single-object tasks with fixed initial states. LHM-Humanoid[17] is an important step toward continuous multi-object vision-language loco-manipulation, yet its sequential training and task-specific designs limit scalability: as horizons grow, catastrophic forgetting and easy-reward bias become pronounced. Our method addresses these issues with a unified policy using Parallel Training Strategy, Dynamic Starting Mechanism, and Reward Gating, enabling robust, scalable long-horizon multi-object loco-manipulation without environment resets.

2.2 Long-Horizon Loco-Manipulation

Long-horizon execution has been studied in fixed-base arms [25, 26, 27], mobile manipulation [28, 29, 30], and aerial manipulation, but remains difficult because long action chains accumulate physical constraints [31, 32, 33, 34]. Typical strategies decompose tasks via subgoals [35, 36], world-state prediction [33], trajectory prediction [32], or historical observations [37], yet often depend on task-specific priors and generalize poorly. Progress is more limited in humanoids: most teleoperation systems remain short-horizon and open-loop [38, 39, 40] because reliable real-time bipedal state estimation is hard, and although legged odometry has advanced [41, 42, 43], transfer to humanoids is still limited (CLONE [44] only partially mitigates drift with LiDAR odometry). Autonomous long-horizon whole-body humanoid loco-manipulation is therefore still nascent, with representative attempts including LessMimic [45] and LHM-Humanoid [17]. As horizons grow, these methods

still face the same two bottlenecks discussed above: catastrophic forgetting and easy-reward bias. Motivated by this gap, Humanoid Horizon targets horizon scalability by maintaining stage-wise competence while preserving previously completed placements as task length increases. Learning those stage transitions also depends on where each stage begins. Reverse-curriculum learning [46] grows the set of start states backward from the goal, and so needs the agent to be reset into hand-specified near-goal configurations. Reset-free autonomous RL [47] avoids manual resets by training a separate reset or backward policy, with its own reward. Curricula taken from expert demonstrations [48] start the agent from demonstrated states. Dynamic Starting (Sec. 3.3) is aimed at the handoff between stages instead: the next stage begins from the terminal state the preceding stage has just produced, with no reset policy, demonstration source, or extra reward.

3 Method

This section presents the Humanoid Horizon training pipeline with a focus on the two central goals introduced earlier: extending effective task horizon and mitigating forgetting in long sequential whole-body manipulation. We begin with the task formulation, then present our Parallel Training Strategy, which re-structures temporally sequential object transport into a unified optimization problem that preserves earlier-stage competence while learning later stages. We next describe the Dynamic Starting Mechanism, which improves stage-to-stage continuity by initializing each stage from naturally occurring intermediate states rather than fixed resets. We then introduce the Reward Gating mechanism to ensure that later-stage transport does not disrupt objects placed in earlier stages. Finally, as an extension, we introduce our distillation pipeline that transfers the reinforcement learning-trained teacher into a vision-language-action (VLA) student policy operating solely from egocentric RGB/depth observations and natural language instructions.

3.1 Task Formulation

We define continuous multi-object rearrangement over a set of scenes $\mathcal{N} = \{1, 2, \dots, N\}$. Each scene $n \in \mathcal{N}$ has its own layout, object configuration, and task specification, and contains an ordered object set $\mathcal{O}_n = \{o_1^n, o_2^n, \dots, o_{S_n}^n\}$. The humanoid must move each object from initial positions $\{p_{\text{init},1}^n, \dots, p_{\text{init},S_n}^n\}$ to target positions $\{p_{\text{goal},1}^n, \dots, p_{\text{goal},S_n}^n\}$. For each object o_i^n , execution has four stages: (1) navigate to the object, (2) grasp and lift, (3) transport toward the goal, and (4) place at the designated target. After finishing o_i^n , the humanoid transitions immediately to o_{i+1}^n without environment reset, so the policy must handle turning, obstacle avoidance, and re-navigation under evolving intermediate states. The benchmark spans diverse placements and semantics: objects and goals may lie on shelves, beds, floors, tables, cabinets, desks, or designated floor regions, and object categories include bedside tables, laptops, storage boxes, coffee machines, and vases. Humanoid Horizon aims to learn a single policy $\pi_\theta : \mathcal{X} \rightarrow \mathcal{A}$ that generalizes across all N scenes and completes each full sequence $\{o_1^n, o_2^n, \dots, o_{S_n}^n\}$ under one control framework, without policy switching or external intervention. Here, n indexes scenes ($n \in \mathcal{N}$), while $\mathcal{X}$ and $\mathcal{A}$ denote the shared robot–environment state space and whole-body control action space, respectively.

3.2 Parallel Training Strategy

Prior long-horizon humanoid rearrangement methods—including end-to-end RL, curriculum RL, and stage-isolated sequential training—are constrained by serial optimization, which tends to induce easy-reward bias toward early transport stages or catastrophic forgetting when later stages are added. To remove this bottleneck, we adopt a Parallel Training Strategy that jointly optimizes all transport stages within one policy. We first pretrain on single-object transport to establish basic loco-manipulation competence (navigate, grasp/lift, and carry/place from fixed starts), then switch to parallel training over N scenes with S sequential objects each (Sec. 3.1). These scenes are organized into S stage streams that share parameters θ . One pool of parallel environments is divided evenly across the streams, and the baselines use the same pool (Appendix A.4).

For stream i , training starts from buffer $\mathcal{B}_i$: $\mathcal{B}_1$ uses the original initial-state distribution, while $\mathcal{B}_i$ ($i > 1$) approximates post-stage- $(i-1)$ intermediate states. Initially, each environment buffer in $\mathcal{B}_i$ ($i > 1$) is set to a simple proxy state (robot near object- $(i-1)$ goal with default pose), and these coarse starts are later replaced by rollout-derived states via the Dynamic Starting mechanism (Sec. 3.3). Keeping all S streams active throughout training provides persistent gradients for every

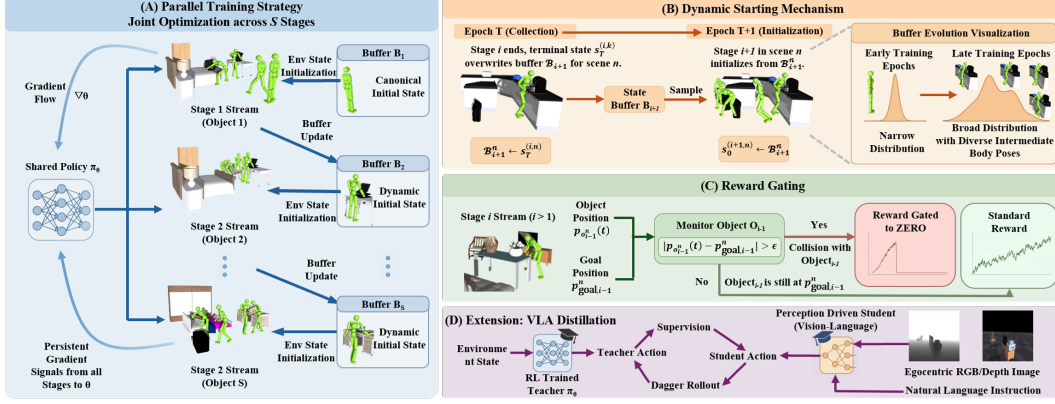


Figure 2: Overview of the Humanoid Horizon training pipeline. (A) **Parallel Training:** S stage streams share one policy π_θ and are optimized jointly, each initialized from its state buffer (B_1 – B_S). (B) **Dynamic Starting:** for each scene n , the terminal state from stage i overwrites the corresponding downstream buffer B_{i+1}^n (one state per environment) to initialize stage $i+1$ in the next epoch; repeated overwrites across epochs broaden transition-state coverage over time. (C) **Reward Gating:** for stage- i streams ($i > 1$), reward is zeroed if object o_{i-1} moves beyond tolerance ϵ , preserving previously completed placements. (D) **VLA Distillation:** the RL teacher is distilled via DAGger into a vision-language-action student conditioned on egocentric RGB/depth observations and language instructions.

stage, suppressing easy-reward bias and forgetting; importantly, stage- i streams ($i > 1$) also force learning of transition-specific behaviors absent in stage 1 (release, reorientation, obstacle-aware maneuvering, and walking from non-canonical poses). Overall, this strategy yields stable optimization for long-horizon continuous rearrangement under a single unified policy.

3.3 Dynamic Starting Mechanism

While Parallel Training addresses easy-reward bias and catastrophic forgetting, it does not by itself solve stage-to-stage transition robustness in long-horizon execution. As described in Sec. 3.2, each stage- i stream ($i > 1$) starts from a fixed proxy buffer B_i ; this is sufficient for bootstrapping isolated transport skills, but it creates a distribution mismatch because fixed proxies represent only a narrow, idealized subset of post-stage- $(i-1)$ states. The mismatch worsens with horizon length: even successful rollouts end in diverse handoff configurations (orientation, foot placement, residual momentum, and goal-relative offset), so stage- $(i+1)$ policies trained from fixed starts become brittle when preceding stages terminate in unfamiliar states.

We therefore introduce Dynamic Starting, which progressively replaces proxy starts with rollout-derived terminal states. At the end of each epoch, for each scene n , we overwrite the corresponding downstream stage buffer (which stores one state per environment) with the stage- i terminal state:

$$B_{i+1}^n \leftarrow s_T^{(i,n)}, \quad (1)$$

where $s_T^{(i,n)}$ is the terminal state of scene n in the stage- i rollout of the current epoch. In the next epoch, scene n in stage- $(i+1)$ is initialized by directly reading this updated buffer:

$$s_0^{(i+1,n)} \leftarrow B_{i+1}^n. \quad (2)$$

Although each scene-stage buffer stores only one state at any instant, repeated epoch-wise overwrites make the policy train on a temporally accumulated sequence of handoff states. As training improves, these recorded terminal states become both more competent and more diverse over time, creating a positive feedback loop: better stage- i behavior yields better downstream starts, which improves stage- $(i+1)$ robustness and subsequently its own terminal-state quality. Compared with fixed proxy initialization, Dynamic Starting better covers natural inter-stage transitions without extra supervision, handcrafted sampling, or scene-specific trajectories. It is complementary to Parallel Training: Parallel

Training supplies persistent gradients to all stages, while Dynamic Starting aligns each stage’s initial-state sequence with the true handoff dynamics from the preceding stage. Together, they improve stability, robustness, and scalability in long-horizon continuous loco-manipulation.

3.4 Reward Gating for Physical Consistency

A unique challenge in continuous multi-object rearrangement is that success requires every transported object to remain at its goal throughout the entire episode. We define the Success-All criterion: an episode succeeds only if all S objects are placed at their respective goals and remain undisturbed, contrasting with per-stage metrics that ignore collateral disturbance of previously placed objects.

When transitioning from transporting object o_{i-1}^n to object o_i^n , the humanoid must release, reorient, and navigate within a cluttered environment where already-placed objects may obstruct the path. Without explicit supervision, the policy may inadvertently displace previous objects during this transition, violating Success-All even if each individual transport succeeds.

To address this, we introduce Reward Gating for stage- i streams with $i > 1$. While all streams share the same reward function, the stage- i stream monitors object o_{i-1}^n throughout rollout. If the robot displaces o_{i-1}^n from its goal (by collision, pushing, or tipping), reward is immediately zeroed for the remainder of the episode:

$$r_t^{(i)} \leftarrow 0 \quad \text{if } \|p_{o_{i-1}^n}^n(t) - p_{\text{goal},i-1}^n\| > \epsilon, \quad (3)$$

where $p_{o_{i-1}^n}^n(t)$ denotes the position of object o_{i-1}^n at time t and ϵ is a tolerance threshold. Stage $i > 1$ therefore gates only on o_{i-1}^n . The stages share one policy π_θ , which learns a single index-independent behavior: executing stage i keeps

$$\|p_{o_{i-1}^n}^n(t) - p_{\text{goal},i-1}^n\| \leq \epsilon. \quad (4)$$

At test time the same policy runs stages $i = 2, \dots, S$ in order. Substituting $i = k + 1$ at each boundary yields

$$\|p_{o_k^n}^n - p_{\text{goal},k}^n\| \leq \epsilon, \quad k = 1, \dots, S - 1. \quad (5)$$

Eq. (5) is Eq. (4) applied once per stage. Each object is guarded exactly when it is the immediately preceding one, and preservation of $\{o_1^n, \dots, o_{S-1}^n\}$ emerges from that recursion alone.

3.5 Extension: Vision-Language-Action Policy Distillation

As an exploratory step toward real-world deployment, we distill the RL-trained teacher policy into a Vision-Language-Action (VLA) student model using DAGger [18], which iteratively collects on-policy student rollouts and queries the teacher for corrective supervision, mitigating the compounding distribution shift that would otherwise arise from behavior cloning over long horizons. We train two student variants: one conditioned on egocentric RGB images and one on egocentric depth maps, both conditioned on per-stage natural language instructions encoded by a frozen BERT-base-uncased [49] model and issued interactively by a human user: after each transport stage is completed, a new instruction specifying the next object and its goal is provided (e.g., “*carry the laptop to the desk*”, then after completion, “*move the trash bin to the corner*”). The robot must execute each instruction online without access to any privileged state information. Despite operating under realistic perceptual constraints, both VLA variants retain the long-horizon multi-object loco-manipulation capability of the teacher, suggesting that the behaviors learned by Humanoid Horizon are transferable to perception-driven settings and represent a meaningful step toward deployment on physical humanoid platforms.

4 Experiments

4.1 Dataset and Evaluation Metrics

Our experiments use the dataset from LHM-Humanoid [17], spanning four room categories—bedroom, living room, kitchen, and warehouse—with 350 diverse cluttered scene layouts and corresponding long-horizon loco-manipulation episodes. The scenes include 79 object categories, among which 25 are movable placement targets. In each episode, the agent must finish two consecutive

fetch-carry-place cycles within one uninterrupted rollout (i.e., without resets). For generalization evaluation, we also use the 66 unseen tasks from LHM-Humanoid [17], which serve as an out-of-distribution benchmark with novel layouts, objects, and instructions.

We evaluate performance using **Succ1** and **Succ2** (success rates for each transport cycle), **SuccAll** (episodes where both cycles are completed), and **Dist 1/Dist 2** (final object-to-goal distances in meters). A cycle is considered successful if the object root lies within 0.2 m of its target location. Better performance corresponds to higher success metrics and smaller distance errors.

4.2 Baseline clarifications

If parallel training is removed, our Humanoid Horizon effectively becomes Curriculum RL; in that case, dynamic starting and reward gating are no longer well-defined because both mechanisms are designed on top of the parallel-training framework. Hierarchical RL is implemented with five low-level controllers and a privileged FSM for oracle stage transitions: walk-to-target, reach-and-grasp, lift, carry with obstacle avoidance, and place-and-release. For HumanVLA [9], we decompose the 350 long-horizon tasks into 700 short-horizon subtasks to match its original protocol, since HumanVLA [9] is built for single-object transport; reward design and training procedures are kept the same, and a privileged FSM is still used at test time to transition between consecutive transport stages. InterMimic [15] is run according to its standard pipeline, i.e., AMP-motion imitation followed by policy rollout evaluation. TokenHSI [16] uses the TokenHSI [16] architecture while being trained on the same five skills as Hierarchical RL. The LHM-Humanoid [17] baseline is likewise trained strictly under its original pipeline, and only its end-to-end student model is used for comparison. Except for the VLA table, all compared methods use identical observation and action interfaces, so differences primarily come from the learning strategies rather than mismatched I/O settings.

4.3 Main results on 350 tasks

Table 1: Results on 350 training tasks.

Method	Succ1 $\uparrow$	Succ2 $\uparrow$	SuccAll $\uparrow$	Dist 1 (m) $\downarrow$	Dist 2 (m) $\downarrow$
End-to-End RL	2.3%	0.0%	0.0%	2.05	1.97
Curriculum RL	88.3%	48.5%	47.2%	0.26	1.01
Hierarchical RL	37.5%	25.0%	20.8%	0.78	1.13
HumanVLA [9]	42.3%	36.7%	29.9%	0.65	0.97
InterMimic [15]	20.6%	9.8%	7.5%	1.32	1.82
TokenHSI [16]	40.5%	31.5%	27.6%	0.58	1.07
LHM-Humanoid [17]	87.4%	72.6%	71.1%	0.28	0.50
Humanoid Horizon w/o DS	92.7%	73.5%	67.7%	0.08	0.46
Humanoid Horizon w/o RG	83.9%	75.6%	69.6%	0.12	0.47
Humanoid Horizon	96.6%	81.3%	80.4%	0.06	0.38

We report results on the 350 training tasks, 66 unseen tasks, and the VLA extension. DS and RG denote dynamic starting and reward gating, and three-seed standard deviations for our method and LHM-Humanoid are given in Appendix C. Table 1 reveals clear failure patterns across baselines. End-to-End RL collapses in the long-horizon regime (SuccAll: 0.0%), showing that sparse rewards do not propagate reliably through multi-stage, contact-rich execution. Curriculum RL exhibits easy-reward bias: despite high Succ1 (88.3%), it under-learns transition behaviors and drops to 47.2% SuccAll. Hierarchical RL, TokenHSI, and InterMimic also generalize poorly to cluttered rearrangement because narrow primitives, motion-derived controllers, or missing scene-specific supervision cause errors to accumulate across stages (SuccAll: 20.8%, 27.6%, and 7.5%; Dist2 for InterMimic: 1.82 m). HumanVLA further degrades to 29.9% SuccAll, consistent with catastrophic forgetting caused by cumulative cross-stage distribution shift. LHM-Humanoid, the strongest prior baseline, still trails our method by about 9% on Succ1/Succ2/SuccAll (87.4/72.6/71.1 vs. 96.6/81.3/80.4), highlighting the limits of a non-parallel design tuned mainly for two-object horizons.

Ablations confirm the complementary roles of DS and RG. Removing DS (*w/o DS*) preserves strong first-stage performance (Succ1: 92.7%) but makes downstream stages brittle because fixed proxy starts mismatch the evolving handoff states from preceding stages, reducing SuccAll to 67.7%. Removing

RG (*w/o* RG) lowers both stability and full-horizon completion (SuccAll: 69.6%) as previously placed objects are disturbed during later navigation. Combining Parallel Training, Dynamic Starting, and Reward Gating yields 96.6%/81.3%/80.4% on Succ1/Succ2/SuccAll, establishing a new state of the art on this benchmark.

4.4 Results on 66 unseen tasks

Table 2: Results on 66 unseen tasks.

Method	Succ1 $\uparrow$	Succ2 $\uparrow$	SuccAll $\uparrow$	Dist 1 (m) $\downarrow$	Dist 2 (m) $\downarrow$
End-to-End RL	1.4%	0.0%	0.0%	2.37	2.28
Curriculum RL	75.5%	40.4%	39.5%	0.36	1.11
Hierarchical RL	31.6%	21.8%	17.3%	0.97	1.39
HumanVLA [9]	36.3%	31.3%	25.1%	0.72	1.16
InterMimic [15]	17.2%	8.2%	5.9%	1.47	2.03
TokenHSI [16]	33.9%	26.1%	22.9%	0.67	1.21
LHM-Humanoid [17]	80.8%	64.9%	61.6%	0.37	0.54
Humanoid Horizon <i>w/o</i> DS	90.4%	69.8%	63.5%	0.10	0.50
Humanoid Horizon <i>w/o</i> RG	89.1%	72.3%	66.4%	0.11	0.49
Humanoid Horizon	97.1%	79.3%	78.0%	0.05	0.47

Table 2 evaluates generalization to 66 out-of-distribution scenes. The trend is consistent with the 350-task setting but with lower absolute values, indicating genuine generalization difficulty rather than memorization. Baselines still degrade substantially (e.g., End-to-End RL: SuccAll 0.0%; Curriculum RL: 39.5%; Hierarchical RL/TokenHSI: 17.3%/22.9%; HumanVLA: 25.1%; InterMimic: 5.9%). The strongest prior baseline, LHM-Humanoid [17], reaches 61.6% SuccAll, while Humanoid Horizon achieves 97.1%/79.3%/78.0% (Succ1/Succ2/SuccAll), outperforming it by about 16% on SuccAll with only a marginal drop from seen tasks. Ablations remain consistent: removing Dynamic Starting (*w/o* DS) reduces SuccAll to 63.5%, while removing Reward Gating (*w/o* RG) yields 89.1%/72.3%/66.4% (Succ1/Succ2/SuccAll), still below the full model.

4.5 VLA extension results

Table 3: VLA extension results (all methods distilled via the same DAgger [18] pipeline).

Method	Succ1 $\uparrow$	Succ2 $\uparrow$	SuccAll $\uparrow$	Dist 1 (m) $\downarrow$	Dist 2 (m) $\downarrow$
End-to-End RL	1.3%	0.0%	0.0%	2.50	2.49
Curriculum RL	71.5%	41.1%	40.7%	0.37	1.18
Hierarchical RL	29.3%	20.7%	16.5%	1.04	1.45
HumanVLA [9]	34.2%	29.8%	23.5%	0.70	1.26
InterMimic [15]	16.3%	8.4%	6.2%	1.64	2.22
TokenHSI [16]	32.0%	27.0%	23.7%	0.72	1.31
LHM-Humanoid [17]	76.5%	65.4%	63.7%	0.42	0.56
Humanoid Horizon <i>w/o</i> DS	77.8%	58.4%	51.4%	0.19	0.68
Humanoid Horizon <i>w/o</i> RG	74.0%	67.3%	58.6%	0.21	0.64
Humanoid Horizon (RGB)	93.5%	76.8%	75.6%	0.07	0.50
Humanoid Horizon (Depth)	93.7%	77.6%	73.7%	0.07	0.52

Table 3 evaluates distillability rather than the absolute capability of VLA models. All methods are distilled with the same DAgger [18] pipeline into VLA students conditioned on egocentric observations and sequential human language instructions, where a new instruction is issued after each transport stage. Distillation degrades all methods because privileged state inputs are replaced by partial egocentric vision, but weaker teacher policies suffer larger drops, while Humanoid Horizon preserves a substantially larger fraction of its original performance. Our RGB and depth variants outperform all distilled baselines on most metrics, showing that the long-horizon behaviors learned through parallel training, dynamic starting, and reward gating transfer effectively to the egocentric VLA setting.

4.6 Extension to more than two objects sequential transport

Table 4: More than two objects sequential transport results. LHM-Humanoid[†] denotes the variant trained with five teacher policies strictly following the original LHM-Humanoid [17] pipeline.

Method	Succ1 $\uparrow$	Succ2 $\uparrow$	Succ3 $\uparrow$	Succ4 $\uparrow$	Succ5 $\uparrow$	SuccAll $\uparrow$
End-to-End RL	1.5%	0.0%	0.0%	0.0%	0.0%	0.0%
Curriculum RL	91.2%	54.2%	37.5%	25.6%	9.3%	3.2%
Hierarchical RL	38.6%	27.3%	16.6%	3.8%	0.0%	0.0%
HumanVLA [9]	45.8%	40.5%	21.7%	5.6%	1.2%	0.1%
InterMimic [15]	21.3%	10.2%	2.5%	0.4%	0.0%	0.0%
TokenHSI [16]	41.7%	32.7%	18.9%	4.9%	1.1%	0.1%
LHM-Humanoid [17]	90.8%	76.1%	61.0%	38.5%	20.9%	18.1%
LHM-Humanoid [†] [17]	92.5%	78.8%	68.1%	48.6%	30.2%	25.8%
Humanoid Horizon w/o DS	90.8%	77.4%	69.2%	61.4%	53.8%	49.8%
Humanoid Horizon w/o RG	87.3%	76.9%	65.4%	54.8%	43.6%	38.9%
Humanoid Horizon	96.5%	85.2%	82.2%	75.4%	66.5%	60.2%

Table 4 evaluates five-object sequential transport (five consecutive fetch-carry-place cycles without reset). Hierarchical RL, InterMimic, and TokenHSI are not retrained for longer horizons and are kept identical to their zero-shot setting, since their low-level primitives are learned independently of episode length; their results therefore reflect architectural limits. End-to-End RL is retrained under the five-object setting but still collapses (Succ1: 1.5%). Curriculum RL is also retrained on five-object episodes, yet easy-reward bias remains and SuccAll drops to 3.2%. LHM-Humanoid[†], trained strictly with the original LHM pipeline using five teacher policies, improves over the two-teacher variant on later stages (Succ3–Succ5: 68.1%/48.6%/30.2% vs. 61.0%/38.5%/20.9%), but still degrades with horizon length.

Humanoid Horizon declines more slowly. Success falls from 82.2% at stage 3 to 66.5% at stage 5, with SuccAll at 60.2%. Relative to the baselines just described, the drop is graceful, and the same policy stays ahead at every stage. As in the two-object setting, both added mechanisms matter: SuccAll falls to 49.8% without Dynamic Starting and to 38.9% without Reward Gating.

5 Failure Analysis

Humanoid Horizon exhibits two recurring failure modes. First, the Succ1-to-Succ2 gap is larger than later per-stage drops. The reason is an inherent task property: the first cycle starts from a fixed canonical pose on open floor, while later cycles begin from arbitrary post-task body configurations in clutter where surrounding objects become nearby obstacles, making reorientation and obstacle-aware navigation more difficult.

Second, frozen-action failures occur when the humanoid reaches the target but fails to complete a stable grasp. A likely contributor is uniform scene allocation during training: harder clutter layouts and object configurations require more exploration, so equal per-scene training budget can leave difficult grasp cases under-optimized. These challenges are not unique to our method and are also reflected in baseline degradation under longer horizons.

6 Conclusion and Future Work

We presented Humanoid Horizon, a unified framework for stable and scalable long-horizon whole-body humanoid loco-manipulation built on three tightly coupled mechanisms: Parallel Training, which maintains simultaneous optimization pressure across stages to mitigate easy-reward bias and catastrophic forgetting; Dynamic Starting, which trains downstream stages from realistic handoff-state distributions rather than fixed proxies; and Reward Gating, which preserves previously completed placements during later-stage transport. Across both in-distribution and out-of-distribution evaluation, this design delivers state-of-the-art performance on seen and unseen tasks, with strong gains over prior baselines on per-stage success and full-episode completion. Beyond the two-object setting, the same policy extends to five-object sequential transport without any change of architecture. Success declines

as the horizon lengthens, but the degradation is graceful relative to the sharp drop of competing methods.

For future work, we plan to address the remaining failure modes (the Succ1-to-Succ2 navigation gap and frozen-action grasping failures) through improved path planning and difficulty-aware training, and to scale beyond five-object horizons toward real-world deployment on physical humanoid platforms.

Acknowledgments and Disclosure of Funding

This work was partially supported by the ERC Advanced project ETALK (UKRI Funded), the Horizon/UKRI project PRIM1 and the US AFOSR project CASPER++.

References

- [1] 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.
- [2] Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.
- [3] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023.
- [4] Qingxu Zhu, He Zhang, Mengting Lan, and Lei Han. Neural categorical priors for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6):1–16, 2023.
- [5] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6):178, 2019.
- [6] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–9, 2023.
- [7] Zeqi Xiao, Tai Wang, Jingbo Wang, Jinkun Cao, Wenwei Zhang, Bo Dai, Dahua Lin, and Jiangmiao Pang. Unified human-scene interaction via prompted chain-of-contacts. *arXiv preprint arXiv:2309.07918*, 2023.
- [8] Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Haofan Wang, Xu Tang, and Yangang Wang. Synthesizing physically plausible human motions in 3d scenes. In *2024 International Conference on 3D Vision (3DV)*, pages 1498–1507. IEEE, 2024.
- [9] Xinyu Xu, Yizheng Zhang, Yong-Lu Li, Lei Han, and Cewu Lu. Humanvla: Towards vision-language directed object rearrangement by physical humanoid. *Advances in Neural Information Processing Systems*, 37:18633–18659, 2024.
- [10] Yinhuai Wang, Jing Lin, Ailing Zeng, Zhengyi Luo, Jian Zhang, and Lei Zhang. Physshoi: Physics-based imitation of dynamic human-object interaction. *arXiv preprint arXiv:2312.04393*, 2023.
- [11] Zhaoming Xie, Jonathan Tseng, Sebastian Starke, Michiel van de Panne, and C Karen Liu. Hierarchical planning and control for box loco-manipulation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(3):1–18, 2023.
- [12] Yuhang Lin, Yijia Xie, Jiahong Xie, Yuehao Huang, Ruoyu Wang, Jiajun Lv, Yukai Ma, and Xingxing Zuo. Simgenhoi: Physically realistic whole-body humanoid-object interaction via generative modeling and reinforcement learning, 2025.
- [13] Pengxiang Ding, Jianfei Ma, Xinyang Tong, Binghong Zou, Xinxin Luo, Yiguo Fan, Ting Wang, Hongchao Lu, Panzhong Mo, Jinxin Liu, et al. Humanoid-vla: Towards universal humanoid control with visual integration. *arXiv preprint arXiv:2502.14795*, 2025.
- [14] Yinhuai Wang, Qihan Zhao, Runyi Yu, Ailing Zeng, Jing Lin, Zhengyi Luo, Hok Wai Tsui, Jiwen Yu, Xiu Li, Qifeng Chen, et al. Skillmimic: Learning reusable basketball skills from demonstrations. *arXiv e-prints*, pages arXiv–2408, 2024.

- [15] Sirui Xu, Hung Yu Ling, Yu-Xiong Wang, and Liang-Yan Gui. Intermimic: Towards universal whole-body control for physics-based human-object interactions. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12266–12277, 2025.
- [16] Liang Pan, Zeshi Yang, Zhiyang Dou, Wenjia Wang, Buzhen Huang, Bo Dai, Taku Komura, and Jingbo Wang. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5379–5391, 2025.
- [17] Haozhuo Zhang, Jingkai Sun, Michele Caprio, Jian Tang, Shanghang Zhang, Qiang Zhang, and Wei Pan. Lhm-humanoid: Learning a unified policy for long-horizon humanoid whole-body loco-manipulation in diverse messy environments, 2026.
- [18] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.
- [19] Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1737–1747, 2024.
- [20] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023.
- [21] 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.
- [22] Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander William Clegg, John Turner, et al. Homerobot: Open-vocabulary mobile manipulation. *arXiv preprint arXiv:2306.11565*, 2023.
- [23] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5922–5931, 2021.
- [24] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.
- [25] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.
- [26] Zhenyang Lin, Yurou Chen, and Zhiyong Liu. Hierarchical human-to-robot imitation learning for long-horizon tasks via cross-domain skill alignment. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2783–2790. IEEE, 2024.
- [27] Zihang Zhao, Yuyang Li, Wanlin Li, Zhenghao Qi, Lecheng Ruan, Yixin Zhu, and Kaspar Althoefer. Tac-man: Tactile-informed prior-free manipulation of articulated objects. *IEEE Transactions on Robotics*, 41:538–557, 2024.
- [28] Ziyuan Jiao, Zeyu Zhang, Weiqi Wang, David Han, Song-Chun Zhu, Yixin Zhu, and Hangxin Liu. Efficient task planning for mobile manipulation: a virtual kinematic chain perspective. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8288–8294. IEEE, 2021.
- [29] Peiyuan Zhi, Zhiyuan Zhang, Yu Zhao, Muzhi Han, Zeyu Zhang, Zhitian Li, Ziyuan Jiao, Baoxiong Jia, and Siyuan Huang. Closed-loop open-vocabulary mobile manipulation with gpt-4v. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4761–4767. IEEE, 2025.
- [30] Peiyuan Zhi, Peiyang Li, Jianqin Yin, Baoxiong Jia, and Siyuan Huang. Learning unified force and position control for legged loco-manipulation. *arXiv e-prints*, pages arXiv–2505, 2025.
- [31] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.
- [32] Shichao Fan, Quantao Yang, Yajie Liu, Kun Wu, Zhengping Che, Qingjie Liu, and Min Wan. Diffusion trajectory-guided policy for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 2025.

- [33] Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. *arXiv preprint arXiv:2502.16707*, 2025.
- [34] Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based vlm plans the next step for vision-and-language navigation. *arXiv preprint arXiv:2402.15852*, 2024.
- [35] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.
- [36] Jessica Borja-Diaz, Oier Mees, Gabriel Kalweit, Lukas Hermann, Joschka Boedecker, and Wolfram Burgard. Affordance learning from play for sample-efficient policy learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6372–6378. IEEE, 2022.
- [37] Xiaofan Wang, Xingyu Gao, Jianlong Fu, Zuolei Li, Dean Fortier, Galen Mullins, Andrey Kolobov, and Baining Guo. Lola: Long horizon latent action learning for general robot manipulation. *arXiv preprint arXiv:2512.20166*, 2025.
- [38] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.
- [39] Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.
- [40] 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.
- [41] David Wisth, Marco Camurri, and Maurice Fallon. Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots. *IEEE Transactions on Robotics*, 39(1):309–326, 2022.
- [42] Guangjun Ou, Dong Li, and Hanmin Li. Leg-kilo: Robust kinematic-inertial-lidar odometry for dynamic legged robots. *IEEE Robotics and Automation Letters*, 9(10):8194–8201, 2024.
- [43] Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.
- [44] Yixuan Li, Yutang Lin, Jieming Cui, Tengyu Liu, Wei Liang, Yixin Zhu, and Siyuan Huang. Clone: Closed-loop whole-body humanoid teleoperation for long-horizon tasks. In *9th Annual Conference on Robot Learning*, 2025.
- [45] Yutang Lin, Jieming Cui, Yixuan Li, Baoxiong Jia, Yixin Zhu, and Siyuan Huang. Lessmimic: Long-horizon humanoid interaction with unified distance field representations. *arXiv preprint arXiv:2602.21723*, 2026.
- [46] Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.
- [47] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. *arXiv preprint arXiv:1711.06782*, 2017.
- [48] Siyu Dai, Andreas Hofmann, and Brian Williams. Automatic curricula via expert demonstrations. *arXiv preprint arXiv:2106.09159*, 2021.
- [49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

A Implementation Details

A.1 Humanoid Embodiment

The humanoid agent has 15 rigid bodies: pelvis, torso, head, right/left upper arm, right/left lower arm, right/left hand, right/left thigh, right/left shin, and right/left foot. The action space consists of 28 degrees of freedom (DoF) controlled via PD position targets. The network outputs $\mathbf{a} \in [-1, 1]^{28}$, which is mapped to PD targets through an affine transformation:

$$q_{\text{target}} = \mathbf{a} \odot \mathbf{s} + \mathbf{o}, \quad (6)$$

where $\mathbf{s}$ (scale) and $\mathbf{o}$ (offset) are per-DoF constants derived from the joint limits. For revolute joints (right/left elbow and knee, 4 DoFs), the range is first shrunk by a factor of 0.7, then $s_j = 0.5 \times (\bar{q}_j^+ - \bar{q}_j^-)$ and $o_j = 0.5 \times (\bar{q}_j^+ + \bar{q}_j^-)$ where $\bar{q}^\pm$ denote the shrunk limits. For spherical joints (8 joints $\times$ 3 axes = 24 DoFs), the per-axis limit is set to $\min(1.2 \times \max |\theta_{\text{limit}}|, \pi)$ symmetrically, then s_j and o_j are computed analogously. The final torques are computed by the PD controller:

$$\tau = k_p \cdot (q_{\text{target}} - q_{\text{current}}) - k_d \cdot \dot{q}_{\text{current}}, \quad (7)$$

where k_p and k_d are the per-joint stiffness and damping gains. The control frequency is 30 Hz (control_freq_inv = 2 at 60 Hz simulation).

A.2 Observation and Action Spaces

Teacher (RL) observation. The teacher policy receives a privileged state vector $\mathbf{o}_{\text{priv}} \in \mathbb{R}^{250}$ composed of: (i) body proprioception for all 15 rigid bodies (position [3], rotation in 6D representation [6], linear velocity [3], angular velocity [3]), totaling $15 \times 15 = 225$ dimensions (minus 2 for the root height redundancy); (ii) current target object position and orientation [9]; (iii) current object pose and velocity [15]; and (iv) navigation guidance waypoint [3]. In addition, a Basis Point Set (BPS) representation $\mathbf{b} \in \mathbb{R}^{200 \times 3}$ encodes the object point cloud from 200 basis points. The BPS is processed by a dedicated MLP $[600 \rightarrow 512 \rightarrow 512 \rightarrow 128]$ with ReLU activations. All components of $\mathbf{o}_{\text{priv}}$ and the BPS features are individually normalized via running mean and standard deviation.

AMP discriminator observation. The adversarial motion prior (AMP) discriminator receives $\mathbf{o}_{\text{AMP}} \in \mathbb{R}^{1310}$, consisting of 10 consecutive reference motion observation frames (num_ref_obs_frames = 10), each of dimension 131. Each frame is composed of: root height [1], root rotation in 6D representation [6], local root linear velocity [3], local root angular velocity [3], joint rotations in 6D representation for all 12 joints [72] (8 spherical joints $\times$ 6D + 4 revolute joints $\times$ 6D), 28 DoF velocities [28], local positions of 5 key bodies (left hand, right hand, left foot, right foot, head) [15], and local object position [3], totaling $1 + 6 + 3 + 3 + 72 + 28 + 15 + 3 = 131$ per frame. The 5 key bodies are specified via amp_body_name.

VLA student observation. The student is trained by DAgger to reproduce the teacher’s actions from onboard sensing, and its observation is separate from $\mathbf{o}_{\text{priv}}$. It receives four signals. (i) Body proprioception $\mathbf{o}_{\text{body}} \in \mathbb{R}^{223}$, the per-body state of all 15 rigid bodies (position, 6D rotation, linear velocity, and angular velocity), with the two redundant root-height dimensions removed. (ii) An egocentric camera image $\mathbf{I} \in \mathbb{R}^{3 \times 256 \times 256}$ (RGB) or $\mathbb{R}^{1 \times 256 \times 256}$ (depth), captured from the head-mounted camera at position $[0.103, 0, 0.175]$ relative to the head link with horizontal FoV of 90° . (iii) A natural language instruction embedding $\mathbf{t} \in \mathbb{R}^{768}$, obtained from a frozen BERT-base-uncased [49] encoder. (iv) The previous action $\mathbf{a}_{t-1} \in \mathbb{R}^{28}$.

Action space. All policies output $\mathbf{a} \in \mathbb{R}^{28}$, corresponding to normalized actions for the 28 humanoid DoFs. Actions are clipped to $[-1, 1]$ and then mapped to PD position targets via the per-DoF affine transformation $q_{\text{target}} = \mathbf{a} \odot \mathbf{s} + \mathbf{o}$ described in Sec. A.1.

A.3 Network Architectures

Teacher (AMP) network. The teacher uses a shared-backbone architecture with separate actor, critic, and discriminator heads:

- **BPS encoder:** MLP $[600 \rightarrow 512 \rightarrow 512 \rightarrow 128]$ with ReLU.

- **Actor:** MLP $[\text{obs_dim} \rightarrow 1024 \rightarrow 1024 \rightarrow 512 \rightarrow 28]$ with ReLU, where $\text{obs_dim} = \dim(\mathbf{o}_{\text{priv}}) + 128$.
- **Critic:** MLP $[\text{obs_dim} \rightarrow 1024 \rightarrow 1024 \rightarrow 512 \rightarrow 1]$ with ReLU.
- **Discriminator:** MLP $[1310 \rightarrow 1024 \rightarrow 1024 \rightarrow 512 \rightarrow 1]$ with ReLU; the final layer has no bias and uses $\text{Uniform}(-1, 1)$ weight initialization.
- **Action distribution:** Gaussian $\mathcal{N}(\mu, \sigma^2)$ with σ as a learnable parameter (not conditioned on observation), initialized to $\exp(-2.9) \approx 0.055$.

VLA student network. The VLA student network comprises:

- **Image backbone:** EfficientNet-B0 (classifier head removed), producing a 1280-dimensional feature vector. Images are resized to 256×256 and normalized with mean and std of 0.5.
- **Image pre-MLP:** $[1280 \rightarrow 128]$.
- **Text pre-MLP:** $[768 \rightarrow 512 \rightarrow 128]$, where 768 is the hidden dimension of the frozen BERT-base-uncased encoder.
- **Actor:** $[\dim(\mathbf{o}_{\text{body}}) + 128 + 128 + 28 \rightarrow 1024 \rightarrow 512 \rightarrow 28]$ with residual blocks, from body proprioception, the image embedding, the language embedding, and the previous action.
- The student outputs deterministic actions (no stochastic sampling).

A.4 Reinforcement Learning Hyperparameters

Table 5 lists the key hyperparameters for teacher policy training.

Table 5: Teacher (RL) training hyperparameters.

Hyperparameter	Value
PPO epochs per iteration	4
Horizon length (rollout steps)	32
Number of mini-batches	8
Discount factor γ	0.98
GAE λ	0.95
PPO clip ϵ	0.2
Optimizer	Adam
Learning rate	5×10^{-5}
Adam (β_1, β_2)	(0.9, 0.999)
Weight decay	0.0
Gradient norm clipping	5.0
Auto mixed precision (AMP)	True
Advantage normalization	True
Action clip	1.0
Observation clip	5.0
AMP replay buffer size	400,000
AMP demo fetch batch size	512
AMP replay keep probability	0.03
Parallel environments	16,384 total, divided evenly across the S stage streams
Maximum episode length (single-object)	300 steps
Maximum episode length (multi-object)	600 steps
Maximum training epochs	16,000
Simulation frequency	60 Hz
Control frequency	30 Hz

A.5 Loss Functions and Weights

The total training loss comprises actor, critic, bound, and AMP discriminator losses:

$$\mathcal{L}_{\text{total}} = w_a \mathcal{L}_{\text{actor}} + w_c \mathcal{L}_{\text{critic}} + w_b \mathcal{L}_{\text{bound}} + w_d \mathcal{L}_{\text{disc}}, \quad (8)$$

where:

- $\mathcal{L}_{\text{actor}}$: PPO clipped surrogate objective.
- $\mathcal{L}_{\text{critic}}$: MSE loss on normalized value estimates.
- $\mathcal{L}_{\text{bound}}$: Action bound regularization = $\max(\mu - \text{clip}, 0)^2 + \min(\mu + \text{clip}, 0)^2$.
- $\mathcal{L}_{\text{disc}}$: AMP discriminator loss, composed of:
 - Binary cross-entropy prediction loss (positive=demo, negative=policy).
 - Logit regularization: L_2 norm of the final discriminator layer weights.
 - Gradient penalty: $\|\nabla_{\mathbf{o}_{\text{pos}}} f(\mathbf{o}_{\text{pos}})\|^2$ where f is the discriminator output.
 - Weight decay on all discriminator parameters.

The loss weights are listed in Table 6.

Table 6: Loss weights.

Loss term	Weight
Actor (w_a)	1.0
Critic (w_c)	5.0
Bound mu (w_b)	10.0
Discriminator total (w_d)	5.0
Disc prediction	1.0
Disc logit regularization	0.01
Disc gradient penalty	5.0
Disc weight decay	0.0001

A.6 Reward Design

Each transport stage uses an 8-term shaped reward function. The total task reward for transporting object o is:

$$r_{\text{task}} = \sum_{k=1}^8 w_k \cdot r_k, \quad (9)$$

with individual terms and weights defined in Table 7.

Table 7: Reward function components. All reward terms are bounded in $[0, 1]$. The thresholds $\delta_r = 0.5$ m and $\delta_o = 0.3$ m gate the approach and grasp rewards once the robot/object is sufficiently close. When $d_{\text{obj} \rightarrow \text{goal}} < \delta_o$, the approach, grasp, and height rewards are clamped to 1.0 to avoid interfering with placement.

w_k	Reward term	Name	Formula
0.1	Approach velocity	r_{vel}	$\exp(-2(v^* - v_{\text{proj}})^2)$
0.1	Approach position	r_{pos}	$\exp(-0.5 \cdot d_{\text{robot} \rightarrow \text{obj}})$
0.1	Grasp proximity	r_{grasp}	$\frac{1}{2} \sum_h \exp(-5 \cdot d_{h \rightarrow \text{obj}})$
0.1	Lift height	r_{height}	$\frac{\min(z_{\text{obj}}, z_{\text{target}}) - z_{\text{init}}}{z_{\text{target}} - z_{\text{init}}}$
0.2	Object transport vel.	$r_{\text{obj_vel}}$	$\exp(-2(v_{\text{obj}}^* - v_{\text{obj,proj}})^2)$
0.2	Object waypoint dist.	$r_{\text{obj_wp}}$	$\exp(-1 \cdot d_{\text{obj} \rightarrow \text{waypoint}})$
0.1	Object goal proximity	$r_{\text{obj_goal}}$	$\exp(-5 \cdot d_{\text{obj} \rightarrow \text{goal}})$
0.1	Object goal rotation	$r_{\text{obj_rot}}$	$\exp(-2 \cdot \Delta\theta_{\text{obj} \rightarrow \text{goal}})$

Here $v^* = v_{\text{obj}}^* = 1.5$ m/s is the target navigation/transport speed, v_{proj} is the projection of the robot/object velocity onto the direction toward the target, $d_{h \rightarrow \text{obj}}$ is the minimum distance from hand h to the object point cloud, $z_{\text{target}} = \max(z_{\text{init}}, z_{\text{goal}}) + 0.1$ is the carry height (with a 0.1 m lift margin), and $\Delta\theta$ is the geodesic rotation distance. For objects with both initial and goal heights below 0.1 m (floor-to-floor transport), r_{height} is set to 1.0.

The discriminator reward for style imitation is:

$$r_{\text{disc}} = -\log(\max(1 - \sigma(f(\mathbf{o}_{\text{AMP}})), 0.01)), \quad (10)$$

where f is the AMP discriminator logit and σ is the sigmoid function. The discriminator reward is clamped: $r_{\text{disc}} \leftarrow \text{clamp}(r_{\text{disc}}, 0, \max(r_{\text{task}}, 0.4))$. The final reward combines both:

$$r = 1.0 \cdot r_{\text{disc}} + 1.0 \cdot r_{\text{task}}. \quad (11)$$

A.7 DAgger Distillation Hyperparameters

Table 8 lists the DAgger distillation hyperparameters.

Table 8: DAgger distillation hyperparameters.

Hyperparameter	Value
Initial mixing coefficient β_0	0.998
β decay schedule	Exponential
Number of step iterations per epoch	1
Number of train iterations per epoch	5
Replay buffer size	100,000
Training batch size	600
Test batch size	200
Active rendering weight	0.2
Optimizer	Adam
Learning rate	5×10^{-4}
Maximum training epochs	30,000
Number of parallel environments	558
Student loss	$\mathcal{L}_2 = \ \mathbf{a}_{\text{student}} - \mathbf{a}_{\text{teacher}}\ ^2$

The DAgger training loop operates as follows: at each epoch, (1) the student executes an action in the environment with probability $1 - \beta$ (otherwise the teacher acts), (2) the teacher’s action is recorded as the supervision target, and (3) the $(o_t, a_t^{\text{teacher}})$ pairs are stored in the replay buffer. The student is then trained by sampling mini-batches from the buffer and minimizing the MSE loss. The mixing coefficient β decays exponentially, gradually shifting from teacher-driven to student-driven rollouts.

A.8 Simulation and Physics

All experiments are conducted in NVIDIA IsaacGym with GPU-accelerated PhysX on $2 \times$ NVIDIA A100 (80 GB) GPUs. Key physics settings:

- Simulation timestep: $\Delta t = 1/60$ s (60 Hz) with 2 substeps.
- PhysX solver: TGS (Temporal Gauss-Seidel), 4 position iterations, 0 velocity iterations.
- Contact offset: 0.02 m; max depenetration velocity: 10.0 m/s.
- Ground plane: static friction = 1.0, dynamic friction = 1.0, restitution = 0.0.
- Environment spacing: 5 m between adjacent environments.
- Up axis: z .

A.9 Termination Conditions

Episodes terminate under two conditions:

1. **Fall detection:** A non-ignored body part (excluding hands, lower arms, and feet) has contact force > 0.1 N and height < 0.15 m simultaneously.
2. **Maximum episode length:** 300 steps for single-object, 600 steps for two-object episodes.

B Training Curves across Seeds

To verify the stability and reproducibility of Humanoid Horizon, we train the full pipeline with three independent random seeds (seed 0, 1, 2) on the 350-scene benchmark for 16,000 epochs each. Figure 3 shows the training success curves for all three seeds.

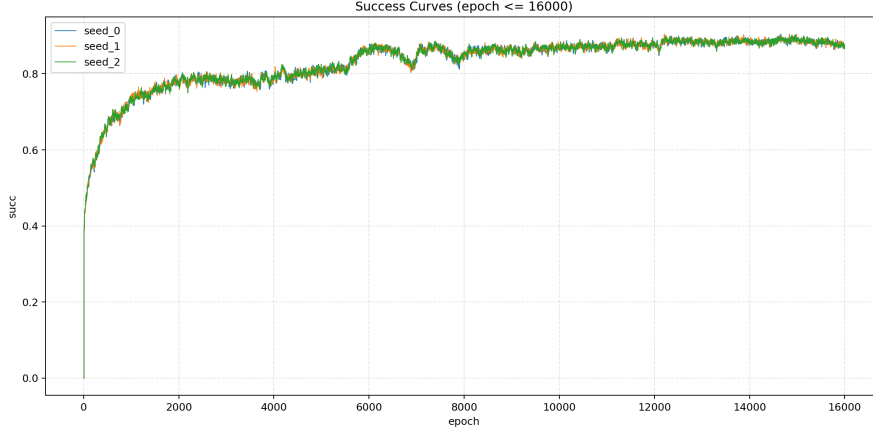


Figure 3: Training success curves across three random seeds (seed 0, 1, 2) over 16,000 epochs. Solid lines show per-seed success rates. All three seeds converge to similar final performance (SuccAll $\approx 87\text{--}89\%$), demonstrating that Humanoid Horizon is robust to random initialization. The initial rapid increase (epochs 0–2,000) corresponds to the pretrained single-object checkpoint’s adaptation to the parallel multi-object regime, followed by steady improvement through the interplay of dynamic starting and reward gating.

Table 9 reports the final converged metrics (averaged over the last 100 epochs) for each seed, along with the mean and standard deviation across seeds.

Table 9: Final training metrics across three random seeds (last 100 epochs). `success` refers to the SuccAll rate logged during training, and `success_1` refers to the Succ1 rate.

Seed	success (SuccAll)	success_1 (Succ1)
Seed 0	87.7%	92.2%
Seed 1	87.3%	92.5%
Seed 2	87.8%	92.8%
Mean $\pm$ Std	$87.6 \pm 0.3\%$	$92.5 \pm 0.3\%$

The low standard deviation ($\leq 0.3\%$) across seeds confirms that the Parallel Training Strategy, combined with Dynamic Starting and Reward Gating, yields stable convergence regardless of random initialization.

Two systematic differences exist between the training-time metrics reported here and the evaluation metrics in Table 1 of the main paper. First, the training-time SuccAll (87.6%) is *higher* than the evaluation SuccAll (80.4%). This is because Parallel Training flattens all S transport stages into $S \times N$ independent concurrent streams; the logged SuccAll is therefore the average of per-stage success rates computed independently, rather than the product of sequential successes within a single uninterrupted episode. In evaluation, stages are executed sequentially—errors in earlier stages propagate and compound into later ones, naturally yielding a lower SuccAll. Second, the training-time Succ1 (92.5%) is *lower* than the evaluation Succ1 (96.6%). This stems from a deliberately tighter success threshold used during training: as described in Appendix A.9, the training-time success criterion requires the object to stay within $0.5 \times \epsilon_{\text{eval}} = 0.1$ m, whereas evaluation counts a placement as successful if the object is within $\epsilon_{\text{eval}} = 0.2$ m. This stricter training threshold encourages more precise object placement, which in turn improves evaluation robustness, but it also causes training-time per-stage success to be conservatively lower than the evaluation counterpart.

C Evaluation Seed Variance

The tables in the main text report point estimates. Table 10 gives the standard deviation of each success rate over the same three seeds, for Humanoid Horizon and LHM-Humanoid, in percentage points. The VLA block is the distilled student; RGB and depth are listed separately in Table 3.

Table 10: Three-seed standard deviations of success rates, in percentage points.

	Succ1	Succ2	SuccAll
<i>Seen tasks</i> (Table 1)			
LHM-Humanoid	1.55	1.76	1.91
Humanoid Horizon	0.94	0.13	0.94
<i>Unseen tasks</i> (Table 2)			
LHM-Humanoid	3.27	3.78	3.78
Humanoid Horizon	0.71	1.43	1.24
<i>VLA</i> (Table 3)			
LHM-Humanoid	5.23	5.73	4.36
Humanoid Horizon	3.79	3.63	4.11

	Succ1	Succ2	Succ3	Succ4	Succ5	SuccAll
<i>Five-object transport</i> (Table 4)						
LHM-Humanoid	1.66	2.15	1.48	3.48	3.16	2.76
Humanoid Horizon	0.97	0.88	0.50	2.21	1.12	1.91

D Broader Impact

This work advances autonomous humanoid manipulation in simulation, with potential positive impacts including assistive robotics (e.g., tidying rooms for elderly or mobility-impaired users) and warehouse automation. Potential risks include the dual-use concern that humanoid agents capable of autonomous object manipulation in cluttered environments could be repurposed for unintended applications. We mitigate this by releasing only the simulation-based framework and trained policies; real-world deployment would require additional safety mechanisms (e.g., force limits, human-in-the-loop supervision). The method does not use personal data, human subjects, or crowd-sourcing.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: The claims in the abstract and introduction are supported by the experiments and ablations in Sec. 4.

Guidelines:

- The answer [\[N/A\]](#) means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [\[No\]](#) or [\[N/A\]](#) answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Limitations are discussed in Sec. 5 (Failure Analysis) and Sec. 6 (Future Work).

Guidelines:

- The answer [\[N/A\]](#) means that the paper has no limitation while the answer [\[No\]](#) means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Full implementation details are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Dataset and code will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Data splits and metrics are described in Sec. 4.1, and all hyperparameters and training details are in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Results are reported as mean $\pm$ standard deviation across three random seeds in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Compute resources (GPUs, environments, epochs) are reported in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research is simulation-only and conforms to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Broader impacts are discussed in Appendix D.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: Released models are simulation-only and pose limited misuse risk; safeguards are briefly noted in Appendix D.

Guidelines:

- The answer [\[N/A\]](#) means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All existing assets used are cited and used under their respective licenses.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Full details are in the appendix.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[N/A\]](#)

Justification: No crowdsourcing or human subjects are involved.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[N/A\]](#)

Justification: No human subjects research is involved.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not part of the core method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.