VIRAL: Visual Sim-to-Real at Scale for Humanoid Loco-Manipulation

Tairan He[1,2*] Zi Wang[1*] Haoru Xue[1,3*] Qingwei Ben[1,4*]
Zhengyi Luo[1] Wenli Xiao[1,2] Ye Yuan[1] Xingye Da[1] Fernando Castañeda[1]
Shankar Sastry[3] Changliu Liu[2] Guanya Shi[2] Linxi "Jim" Fan[1†] Yuke Zhu[1†]
[1]NVIDIA [2]CMU [3]UC Berkeley [4]CUHK [*]Equal Contribution [†]Project Leads
https://viral-humanoid.github.io

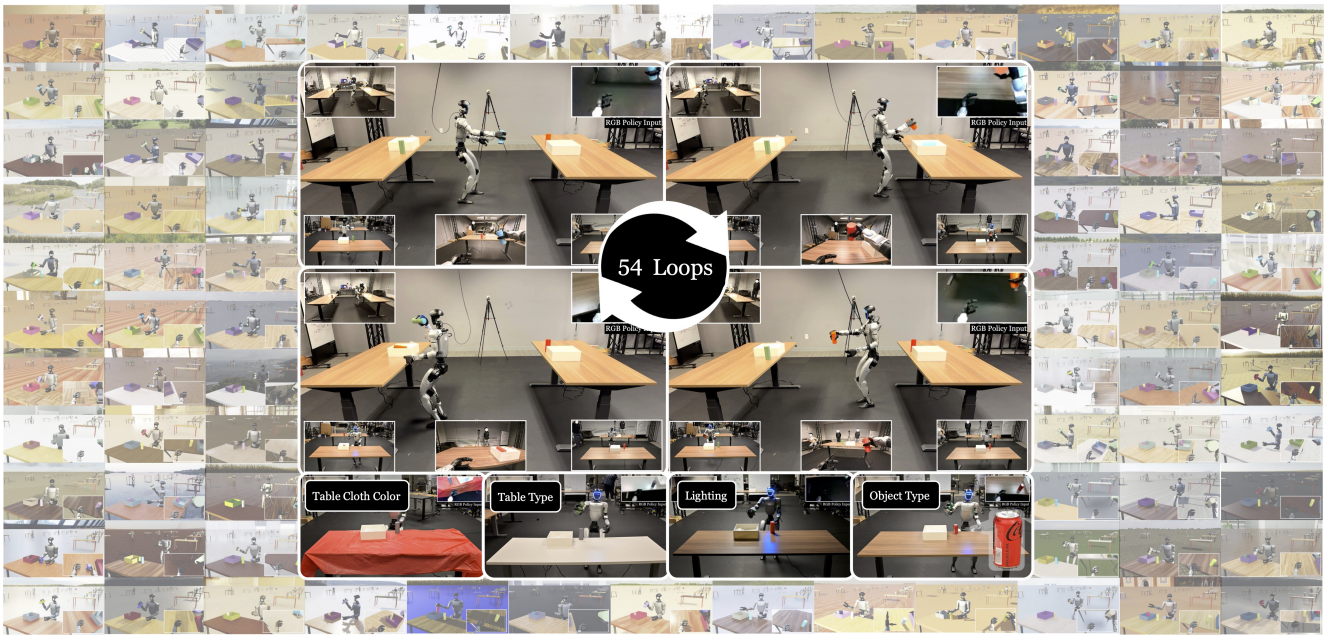

Figure 1. **Center**: Unitree G1 humanoid performing loco-manipulation, walking between tables to place and pick objects for 54 loops with our RGB-based sim-to-real policy. **Surrounding**: diverse simulated scenes used for training. **Website**: https://viral-humanoid.github.io

Abstract

A key barrier to the real-world deployment of humanoid robots is the lack of autonomous loco-manipulation skills. We introduce VIRAL, a visual sim-to-real framework that learns humanoid loco-manipulation entirely in simulation and deploys it zero-shot to real hardware. VIRAL follows a teacher-student design: a privileged RL teacher, operating on full state, learns long-horizon loco-manipulation using a delta action space and reference state initialization. A vision-based student policy is then distilled from the teacher via large-scale simulation with tiled rendering, trained with a mixture of online DAgger and behavior cloning. We find that compute scale is critical: scaling simulation to tens of GPUs (up to 64) makes both teacher and student training reliable, while low-compute regimes often fail. To bridge the sim-to-real gap, VIRAL combines large-scale visual domain randomization over lighting, materials, camera parameters, image quality, and sensor delays—with real-to-sim alignment of the dexterous hands and cameras. Deployed on a Unitree G1 humanoid, the resulting RGB-based policy performs continuous loco-manipulation for up to 54 cycles, generalizing to diverse spatial and appearance variations without any real-world fine-tuning, and approaching expert-level teleoperation performance. Extensive ablations dissect the key design choices required to make RGB-based humanoid loco-manipulation work in practice.

1. Introduction

Humanoid robots are often framed as the natural embodiment of general-purpose physical intelligence: machines that could ultimately take on a large fraction of physical work for society. Yet, despite rapid progress in hardware and control, current humanoids have delivered limited real, sustained productivity outside of carefully engineered demos [21]. A core missing piece is autonomous *loco-manipulation*—tight coordination of locomotion and manipulation under onboard perception—over long horizons

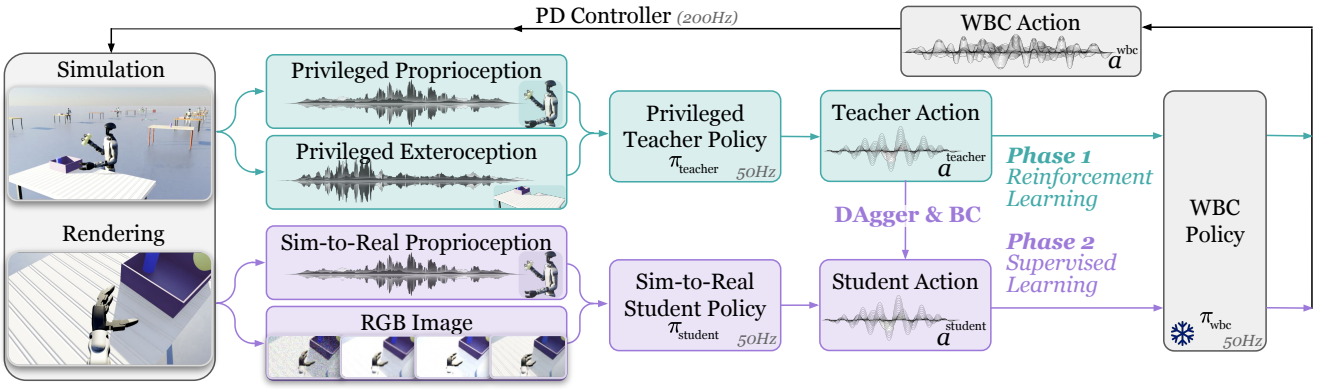

Figure 2. **VIRAL teacher-student pipeline.** Phase 1: In simulation, a privileged RL teacher policy π_{teacher} receives full-state proprioception and exteroception of the task information and outputs WBC commands. Phase 2: A vision-based student policy π_{student} observes only RGB images and sim-to-real proprioception and is trained to imitate the teacher policy via DAgger and behavior cloning.

and across diverse environments to accomplish useful tasks. Most existing humanoid systems either focus on blind locomotion [28, 39, 43, 84], static tabletop manipulation without mobility [41, 45, 77], or rely heavily on human teleoperation [6, 26, 44, 78, 79, 82] or non-onboard sensors [72, 80], and they rarely demonstrate autonomous loco-manipulation with onboard sensors in the real world [15, 65, 74].

Recently, there has been an exciting push to replicate the large language model recipe [1] in robotics, by collecting large-scale real-world datasets and training "robotic foundation models" from real-world teleoperation data [5, 7, 29, 32, 50, 64, 81]. While it remains unclear whether this path alone will suffice for general manipulation, it is clear that mobile manipulation will encounter substantially more variation than fixed tabletop setups and will therefore demand far more data [19, 34, 73]. When the mobile platform is a humanoid, the cost per data point increases even further due to hardware complexity, higher degrees of freedom, safety constraints, and the engineering overhead of the teleoperation stack [16]. In other words, if we treat humanoid mobile manipulation as "just another data problem," the required scale may be prohibitively expensive in practice.

Simulation offers an alternative path. Modern GPU-accelerated, photorealistic simulators can generate orders of magnitude more data at low marginal cost compared with human teleoperation [4, 46, 48]. Sim-to-real has become the de facto approach for legged locomotion [57, 63], where policies trained in simulation routinely transfer to hardware [13, 27, 38]. In contrast, manipulation is still largely dominated by imitation learning from real-world data, with sim-to-real successes typically restricted to tabletop settings and narrow tasks [2, 10, 24, 41, 62]. Moreover, sim-to-real locomotion and manipulation are usually studied in isolation: locomotion work often ignores manipulation, and manipulation work typically assumes a fixed base. In this paper, we aim to answer: *Can visual sim-to-real enable useful humanoid loco-manipulation with onboard perception?*

Visual sim-to-real for robotics is not a new idea [2, 3, 17, 25, 30, 33, 42, 62, 65, 75, 76, 83], but we revisit it in the context of humanoid loco-manipulation and push the system to modern scales in simulation fidelity, GPU compute, and humanoid hardware. Our goal is not to propose yet another novel RL or sim-to-real algorithm, but to provide a technical recipe on the full stack required to make RGB-based humanoid loco-manipulation work in practice: what designs matter, where they fail, and how they interact.

To enable efficient sim-to-real training, we adopt a teacher-student framework as shown in Figure 2. We first train an RL *teacher policy* in simulation with full access to privileged state, operating on top of a pretrained whole-body control (WBC) policy [6]. We then distill this teacher into a vision-based *student policy* that observes only RGB images and proprioception accessible on the real robot. The student is trained with large-scale visual distillation using a mixture of online DAgger [56] and behavior cloning. We find that scaling up GPU compute for simulation training is essential for reliable learning of loco-manipulation skills.

To facilitate visual sim-to-real transfer, on the simulation side, we scale up visual randomization variations, including scene assets, lighting, materials, and camera parameters, with high-fidelity tiled rendering; on the hardware side, we align the simulator and real hardware to best match each other, including system identification (SysID) on high-gear-ratio dexterous hands and the alignment of cameras. Together, these technical elements yield an end-to-end RGB-based student policy that transfers zero-shot to the real humanoid robot and executes continuous loco-manipulation—walking, placing, grasping, and object transport—over long horizons.

In real-world experiments, VIRAL shows not only the robustness of the high success rate that is near the human expert teleoperation performance, but also generalization to various spatial and scene variations. In simulation experiments, scaling studies, and ablations reveal which key com-

ponents of the VIRAL framework are most critical for the full stack to work in practice. Overall, our results suggest that large-scale visual sim-to-real provides a practical path toward autonomous humanoid loco-manipulation.

2. Key Elements of VIRAL

Framework Overview To achieve efficient visual simulation training, the VIRAL controller is trained via teacher-student privileged learning [9] as shown in Figure 2. We first train a privileged RL *teacher* policy with full access to the *privileged state-based inputs* and run the *simulation without the compute burden of visual rendering* on two 8-GPU L40S nodes (*16 GPUs in total*). During this stage, we carefully design stage-based rewards and initialize environments from demonstrations to boost RL training. Instead of training low-level motor skills from scratch, we integrate the pre-trained WBC policy [6] and make the command for WBC the action space for the teacher policy. Details of teacher training are provided in Section 2.1.

After the teacher discovers strong behavior under privileged information, we distill it into a *student* policy that only receives the observations available on the real robot (*i.e.,* proprioception and RGB images). Visual distillation is performed using large-scale simulation on eight 8-GPU L40S nodes (*64 GPUs in total*) with tiled rendering in Isaac Lab [48]. The student is trained by a combination of online DAgger [56] and behavior cloning to predict the teacher's action given only access to proprioception and RGB image. More details of student training are provided in Section 2.2.

To facilitate sim-to-real transfer of the RGB-based student policy, we randomize simulation assets, materials, dome lighting, image effects, camera extrinsics, and sensor delays during student training. We also perform real-to-sim alignment through system identification (SysID) of the Unitree 3-fingered dexterous hand and calibration of camera extrinsics. Finally, we deploy the student policy on the real robot without any fine-tuning. Using onboard sensor observations, the student executes continuous loco-manipulation behaviors—including walking, placing, grasping, and object transport—on the Unitree G1 humanoid. Details of sim-to-real transfer are provided in Section 2.3.

2.1. Key Elements of Teacher Training

Teacher Formulation We formulate the teacher as a goal-conditioned RL policy. At time step t, the teacher $\pi_{\text{teacher}}(a_t|o_t^{\text{priv}})$ outputs a high-level command for the low-level WBC policy given privileged observation. Specifically, the teacher policy outputs $a_t = (\Delta\mathbf{v}_t, \Delta\omega_t^{\text{yaw}}, \Delta q_t^{\text{arm}}, \Delta q_t^{\text{finger}})$ as the command for the WBC policy [6], where $\Delta\mathbf{v}_t$, $\Delta\omega_t^{\text{yaw}}$ are delta linear (x, y) and angular (yaw) velocity commands and Δq_t^{arm}, $\Delta q_t^{\text{finger}}$ are delta joint targets for arm and finger motors. These commands are passed to the WBC policy [6]. The privileged observation $o_t^{\text{priv}} = [o_t^{\text{prop-priv}}, o_t^{\text{exte-priv}}]$ includes privileged proprioception and exteroception. Proprioception consists of $o_t^{\text{prop-priv}} = [\mathbf{v}_t, \boldsymbol{\omega}_t, \mathbf{g}_t, a_{t-1}, q_t, \dot{q}_t, \mathbf{f}_t^{\text{finger}}]$ where $\mathbf{v}_t, \boldsymbol{\omega}_t$ are base linear and angular velocities, $\mathbf{g}_t$ is base projected gravity, a_{t-1} is last action, $q_t, \dot{q}_t$ are joint positions and velocities, $\mathbf{f}_t^{\text{finger}}$ are fingertip forces. As for privileged exteroception, we have $o_t^{\text{exte-priv}} = [e_t, T_t, O_t]$ where e_t is the current stage, T_t is the placement and lift target, O_t is the relative transforms of objects and tables to the robot. All observation terms are specified in Section 7.1. The teacher is trained with PPO [59] with a custom implementation of TRL [68] to train across GPUs in a distributed manner.

Teacher Element #1: Reward Design To design rewards for humanoid loco-manipulation, we segment the task into a sequence of walking, placing, grasping, and turning. Therefore, we define four key rewards:

1. *Walking toward the objects*: $r_{\text{walk}} = \exp(-4\,(\|p_{\text{robot}} - p_{\text{GraspObj}}\| - 0.45)^2)$;
2. *Placing objects when near tray*: $r_{\text{place}} = -\|\mathbf{f}_{\text{PlaceObj}}\| * \mathbb{1}(\|p_{\text{PlaceObj}} - p_{\text{tray}}\| < 0.3)$ where $\mathbf{f}_{\text{PlaceObj}}$ is the force between robot finger and the object to be placed;
3. *Grasping objects*: $r_{\text{grasp-z}} = \min(h_{\text{GraspObj}} - h_{\text{table}}, 0.15)$ and $r_{\text{grasp-goal}} = \exp(-10||p_{\text{GraspObj}} - p_{\text{goal}}||^2)$;
4. *Turning*: $r_{\text{turn}} = -|\text{y}_{\text{robot}} - \text{y}_{\text{desired}}|$ where y is the base yaw heading angle.

Full reward definitions are provided in Section 7.2.

Teacher Element #2: Delta Action Space Rather than outputting absolute joint targets—as is common in legged RL locomotion [57]—we adopt a *delta* action space. The policy outputs increments that are accumulated into the WBC command. In practice, this delta representation significantly accelerates and stabilizes RL training. An ablation of this choice is shown in Figure 9.

Teacher Element #3: WBC command as API To reduce reward engineering burden and enable reliable real-world deployment, the teacher in VIRAL produces high-level WBC commands rather than learning low-level motor skills from scratch. We use HOMIE [6] as the underlying WBC controller, which provides stable lower-body locomotion and diverse upper-body poses. The command space of HOMIE involves velocity and height tracking commands for locomotion and upper-body joint commands. We extend this command interface by incorporating finger actions, yielding the full action space for VIRAL. Note that VIRAL framework does not have designs overfitting to specific WBC policy, and can be extended to other humanoid WBC controllers [44, 78]. With a stable and robust WBC policy as an API layer, the action space of VIRAL policy is limited to a safe and reliable region of humanoid motions, improving deployability.

Teacher Element #4: Reference State Initialization Learning long-horizon walking-

Figure 3. **Visual randomization** on image, lighting, material, and camera-extrinsics randomization for sim-to-real robustness.

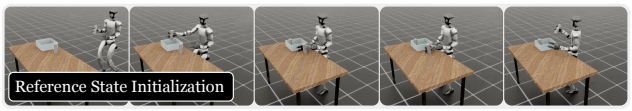

Figure 4. Frames of reference state initialization for teacher RL.

placing-grasping-turning skills for high-DoF humanoids with RL typically demands heavy reward engineering still often yields suboptimal or poor sim-to-real transfer. To mitigate this, we collect 200 teleoperated *simulation* demonstrations and use them as a state-initialization buffer for RL (Figure 4). At every episode reset, we sample a demonstration snapshot and initialize the scene—robot, objects, and tables—accordingly, exposing the policy to diverse rewarding states long before it is capable of reaching them from scratch. This reference-biased exploration greatly reduces reliance on brittle reward tuning and improves sim-to-real transfer, as human-provided grasping and placement poses offer strong priors. Similar ideas have appeared in humanoid control [52, 60] and manipulation [41, 49]. We find this reset strategy to be essential for training humanoid loco-manipulation, as shown in the ablation in Figure 9.

2.2. Key Elements of Student Training

Student Element #1: DAgger&BC Mixture. We train the RGB-based student policy by distilling from the privileged teacher through a hybrid of online DAgger [56] and behavior cloning (BC). Both procedures share the same MSE objective, computed over a mixture of teacher- and student-induced observation distributions:

$$\rho^o \triangleq \alpha\, \rho^o_{\pi_{\text{teacher}}} + (1-\alpha)\, \rho^o_{\pi_{\text{student}}},$$
$$\mathcal{L}_{\text{distill}} = \mathbb{E}_{o_t \sim \rho^o}\left[\left\|\pi_{\text{teacher}}(o_t^{\text{teacher}}) - \pi_{\text{student}}(o_t^{\text{student}})\right\|_2^2\right],$$

where $\rho^o_{\pi_{\text{teacher}}}$ and $\rho^o_{\pi_{\text{student}}}$ denote the observation distributions induced by the teacher and student rollouts, respectively. The distinction between DAgger and BC lies solely in the source of observations: teacher rollouts provide clean, near-optimal demonstrations that rapidly imprint strong priors on the student, while student rollouts expose the learner to states outside the teacher's ideal distribution, which is critical for improving error-correction robustness and preventing compounding error during deployment. This mixed-policy rollout combines the fast initialization of BC with the state-coverage benefits of DAgger, producing a more resilient vision-based controller. Ablation of the mixture coefficient α is provided in Figure 11.

Student Element #2: Network Backbone. For the student's vision backbone, we adopt a state-of-the-art image encoder [61] to extract high-quality RGB features, which are fused with proprioceptive to the policy head. The resulting student observation o^{student} therefore integrates both visual embeddings and the proprioception available on real hardware, enabling the policy to reason over rich visual cues while maintaining grounded low-level awareness. We also evaluate choices for the student policy head, including a single-step MLP and a history-aware architecture that incorporates temporal context. Ablations of the vision backbone and history architecture are shown in Figure 10 and Figure 12, respectively.

Student Element #3: Distributed Simulation Learning System. Large-scale visual simulation is substantially more expensive than rendering-free physics, typically operating at least an order of magnitude slower in terms of simulation throughput. To scale up visual simulation training throughput, we implement a customized version of TRL [68] with support of Accelerate [22] for efficient scaling across multiple GPUs and compute nodes. This implementation preserves the simplicity of single-GPU training while enabling near-linear scaling to large clusters for high-throughput visual sim-to-real learning. We identify scaling up GPUs for both teacher and student training as critical in our ablation studies in Figure 14 and Figure 15.

2.3. Key Elements of Sim-to-Real Transfer

Sim-to-Real Element #1: SysID for Dexterous Hand. While modern humanoids increasingly use low–gear ratio motors—reducing the need for motor-level SysID—the Unitree G1's 3-fingered dexterous hand employs high gear ratios, resulting in a substantial sim-to-real mismatch. To address this, we define a real-world grasp–release primitive and replay the identical action sequence in simulation. We then perform SysID over finger armature, stiffness, and damping parameters to align simulated joint trajectories with real measurements. As shown in Figure 5, SysID significantly improves the correspondence between simulated and real joint positions.

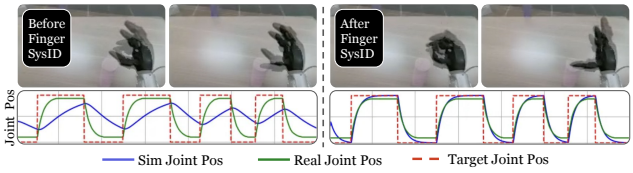

Figure 5. **System identification of the dexterous hand.** Real–sim overlays (top) and joint position trajectories (bottom) before and after SysID, showing markedly improved alignment.

Figure 6. **Real-to-sim camera extrinsics alignment.** Real view versus simulated views before and after alignment.

Sim-to-Real Element #2: FOV Alignment and Randomization. We match the simulator's camera intrinsics (focal length, focus distance, and sensor apertures) to the manufacturer's specifications. However, the camera extrinsics of Unitree G1 robots vary across units due to mechanical tolerances and can even drift over time on the same robot. To better align simulated and real visual observations, we perform a lightweight real-to-sim extrinsics calibration by visually matching rendered and real images (Figure 6). We further apply extrinsics randomization during training (Figure 3) to ensure that the student remains robust to hardware-induced viewpoint differences.

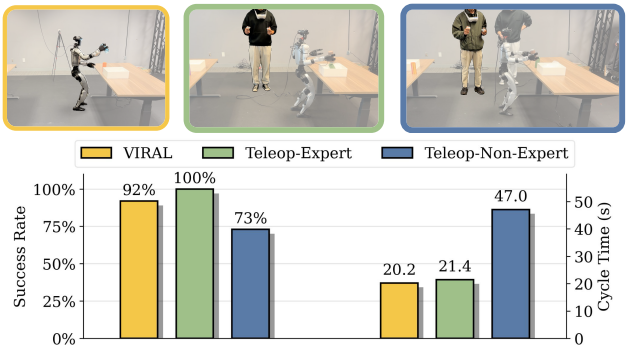

Figure 7. **Real-world performance comparison:** VIRAL matches expert-level reliability, outperforms non-experts, and operates faster than the expert teleoperator.

Sim-to-Real Element #3: Visual and Simulation Randomization. To enhance robustness and improve sim-to-real transfer, we apply extensive visual and physical randomization during training (Figure 3). We randomize image quality (brightness, contrast, hue, saturation, Gaussian noise, and blur), camera extrinsics to account for small pose shifts, and camera latency to model transmission delays. We additionally vary global illumination using dome-light environments and randomize material and color properties of floors, tables, objects, and robot components. These perturbations significantly improve the policy's transferability by preventing overfitting to any specific simulated appearance or lighting condition. Ablation of this randomization is provided in Figure 13.

3. Real-World Results of VIRAL

In this section, we present real-world humanoid loco-manipulation results achieved by VIRAL. The following section (Section 4) analyzes the contribution of each design choice. Our experiments deploy a 29-DoF Unitree G1 humanoid equipped with 7-DoF three-finger dexterous hands. Perception is provided by an Intel RealSense D435i, and all policy inference is performed on a desktop workstation with an Intel i9-14900K CPU and an NVIDIA RTX 4090 GPU.

3.1. Robustness

We evaluate the robustness of the learned student policy on a continuous loco-manipulation task in which the humanoid repeatedly walks between two tables, places an object, grasps a new object, and turns around. Across 59 consecutive real-world trials, VIRAL succeeds in 54, demonstrating strong reliability under extended deployment.

We also compare VIRAL with two human teleoperators: an expert with over 1000 hours of G1 teleoperation experience and a non-expert teleoperator with approximately one hour of experience. All conditions use the

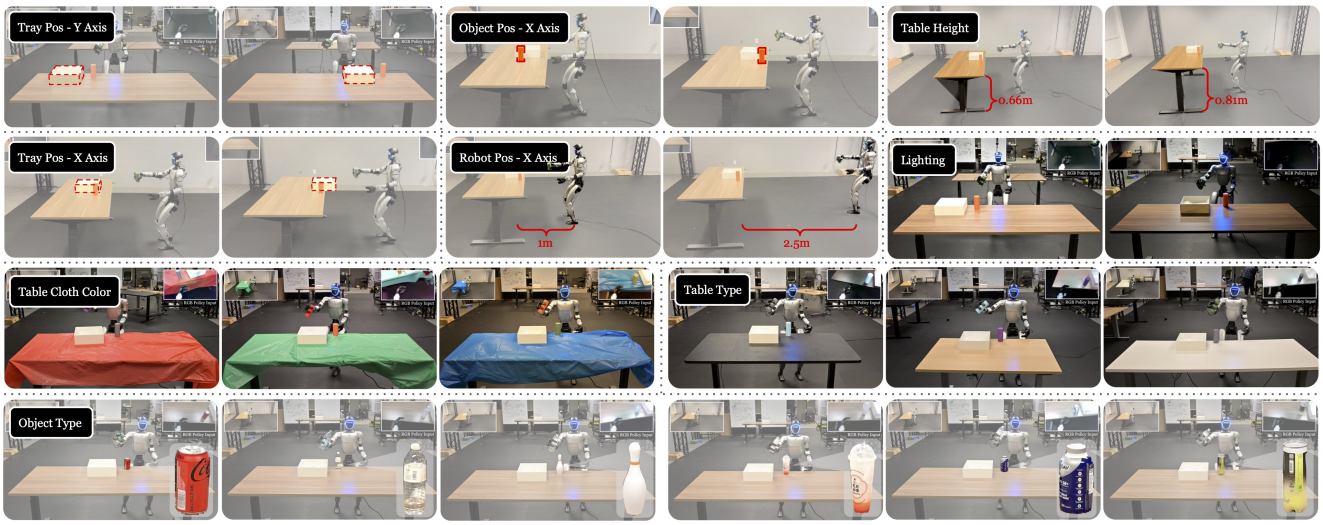

Figure 8. **Real-world generalization of VIRAL RGB-based policy** under variations in tray and object position, robot start pose, table height and type, tablecloth color, lighting, and object category. Videos are provided in https://viral-humanoid.github.io.

same HOMIE policy, yielding a near-apple-to-apple comparison. As shown in Figure 7, the expert attains a 100% success rate with a 21.4 s cycle time, slightly higher than the 20.2 s cycle time of VIRAL. Meanwhile, the non-expert reaches only 73% success with significantly slower execution. These results show that although expert-level success remains challenging, VIRAL achieves *near–expert success performance while being faster than the expert*, and it substantially outperforms non-experts in both reliability and efficiency—highlighting its potential to reduce human workload in assisted teleoperation settings.

3.2. Generalization

We assess real-world generalization by systematically varying the environment along multiple factors, including tray start position, robot start pose, table height, lighting, table cloth, table type and color, and object category (Figure 8). Across these variations, VIRAL consistently completes the task without additional tuning, indicating strong robustness. We attribute this behavior to the domain randomization used during simulation training and the robustness of RL, which exposes the policy to diverse visual and spatial conditions. Videos are provided in https://viral-humanoid.github.io.

4. Experiments

In this section, we evaluate the contribution of each design component of VIRAL, corresponding to the key elements introduced in Section 2.

4.1. Reference State Initialization for RL

Figure 9 compares training curves with and without reference state initialization (RSI) [52] from teleoperated demonstrations. Without RSI, the teacher policy quickly plateaus with a success rate below 10%, whereas the full VIRAL teacher with RSI reaches nearly 95% success. RSI improves exploration by resetting episodes to diverse intermediate states along the task trajectory, so the policy can practice all stages of the task from the outset rather than discovering subgoals sequentially.

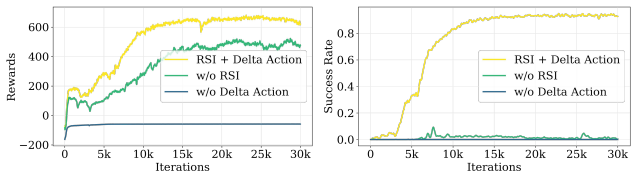

Figure 9. **Ablations of teacher policy training.** Training rewards (left) and success rates (right) for the full method (RSI + delta action), without demonstration resets, and without delta action space, showing that both components are critical for final success.

4.2. Delta versus Absolute Action Space

We compare delta and absolute joint action spaces for the teacher policy. Unlike much of the legged locomotion RL literature, which commonly uses absolute joint targets, we find that a delta action space is crucial for humanoid loco-manipulation: as shown in Figure 9, only the delta-action teacher reliably solves the task, while the absolute-action variant fails to reach high success.

4.3. Vision Backbone

Figure 10 reports the student's training loss and success rate. We see that state-of-the-art vision backbones (DINOv3 [61]) yield stronger visual representations and greater capacity, enabling faster convergence and higher task success—i.e., the policy learns the target behaviors more reliably with better visual features.

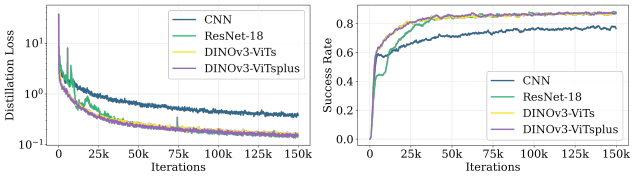

Figure 10. Ablation of vision backbone for student policy.

4.4. DAgger versus BC Visual Distillation

We ablate the DAgger-BC mixture by varying the rollout ratio α, defined as the fraction of environments that follow the teacher policy during data collection ($\alpha = 0$ corresponds to pure DAgger on student rollouts, $\alpha = 1$ to pure BC). As shown in Figure 11, BC ($\alpha = 1$) yields fast loss reduction but produces a brittle policy that fails to correct its own mistakes and performs poorly in Isaac-to-MuJoCo [48, 67] and real-world evaluations. Introducing student rollouts ($\alpha = 0.5$) slows optimization slightly but substantially improves deployment success rate, so we adopt $\alpha = 0.5$ as our default DAgger-BC ratio.

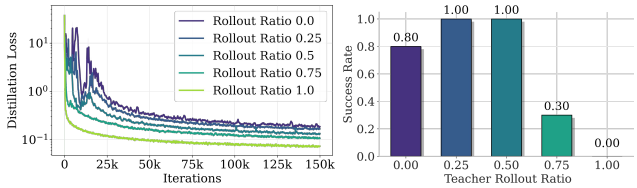

Figure 11. Ablation of ratio of DAgger/BC of student policy.

4.5. History Architecture

Figure 12 compares a single-step baseline, a feed-forward history model, and an LSTM under different history lengths. History-aware models consistently outperform the single-step baseline, and longer temporal windows provide additional gains when resources allow.

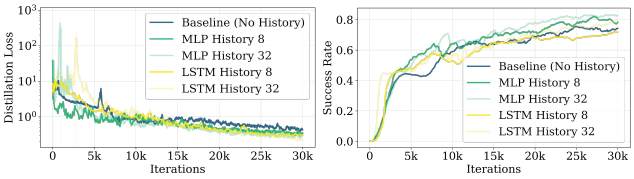

Figure 12. Ablation of the history architecture of student policy.

4.6. Visual Randomization

Figure 13 presents an ablation of our visual domain randomization. We focus on three dominant components: material randomization for table/floor/robot (M), dome-light randomization (D), and camera-extrinsics randomization (E), while other factors (image quality, object color, camera delay) provide smaller gains. All policies are evaluated in IsaacSim [48] with *all* randomizations enabled; training variants differ by removing one component at a time (w/o-M, w/o-D, w/o-E) or using no randomization at all. Success rates are normalized by the model trained with all randomizations (set to 1.0) and averaged over 200 episodes. Two trends emerge: (i) turning off all randomization causes a large drop in performance (down to 0.649, a 35.1% decrease), and (ii) removing any single component also degrades performance, indicating that the randomizations are complementary and together form a crucial pipeline for robust sim-to-real transfer.

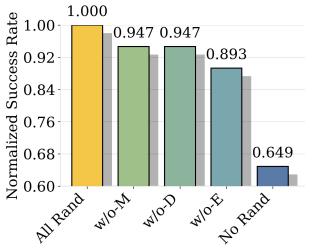

Figure 13. Ablation of visual randomization.

4.7. Scaling Compute for Teacher Training

Figure 14 highlights the impact of scaling GPU resources from 1 to 16 during teacher training. Increasing the number of GPUs substantially accelerates learning: larger batches of parallel environments broaden state-space coverage per unit wall time, enabling the policy to discover rewarding behaviors far more quickly. Early training even shows better-than-linear speedup—for example, reaching a modest success rate of $\sim$0.2 with 4 GPUs takes well under half the time required with 2 GPUs—reflecting richer on-policy experience and more diverse rollouts. Beyond speed, scaling has a pronounced effect on *asymptotic performance*. With insufficient compute (1–2 GPUs), the teacher plateaus far below the desired performance range and never reaches high success rates. In contrast, using 8–16 GPUs consistently drives the policy above 90% success, revealing that large-scale simulation is not only beneficial but often *necessary* for learning long-horizon humanoid loco-manipulation.

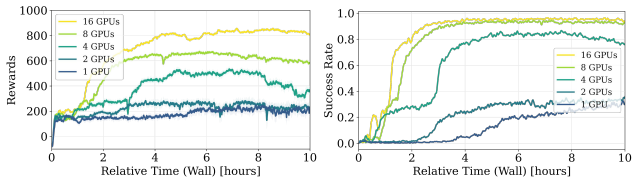

Figure 14. **Scaling compute for teacher training.** Rewards (left) and success rates (right) for 1–16 GPUs. More GPUs yield faster convergence and better asymptotic performance.

4.8. Scaling Compute for Student Training

We observe a clear scaling trend for the student policy as well. Figure 15 plots distillation (DAgger) loss and downstream success rate as we increase the number of GPUs from 1 to 64. Larger-scale training consistently accelerates convergence: the same loss threshold is reached dramatically sooner, and the success curve rises much more steeply. Beyond speed, scaling also improves training *stability*: policies trained with more GPUs exhibit smoother

loss curves and less variance in success rate, especially during the early stages when the student is most sensitive to distribution shift. Interestingly, higher-GPU runs also achieve slightly higher final success, suggesting that large-scale experience collection yields richer and more diverse state coverage, which in turn improves robustness. Overall, these results indicate that substantial computing is not merely a convenience but a practical requirement for reliable visual loco-manipulation distillation.

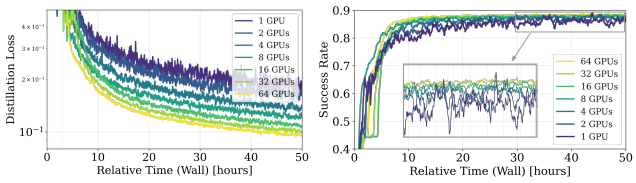

Figure 15. **Scaling compute for student policy training.** Distillation loss (left) and success rate (right) when training with 1–64 GPUs. Larger GPU counts provide significantly faster convergence, smoother optimization dynamics, and higher final performance, highlighting the importance of large-scale parallel simulation for vision-based loco-manipulation.

4.9. Object generalization

We study object-level generalization on the grasping subtask under two training regimes: (i) single-object training on a cylinder only and (ii) multi-object training on ten distinct objects. At test time, we evaluate on the same ten objects and report normalized success rates. As shown in Figure 16, training with multiple objects yields substantially better generalization—the multi-object policy attains higher success on every category than the cylinder-only baseline.

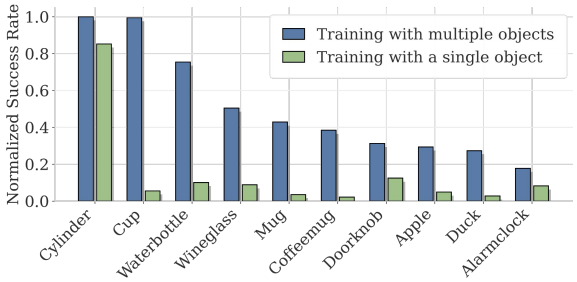

Figure 16. Ablation of object generalization of teacher policy.

5. Related Work

Sim-to-Real for Locomotion Sim-to-real techniques have enabled blind locomotion policies trained in simulation to be deployed zero-shot on real legged robots [6, 11, 18, 20, 31, 35–38, 40, 63]. These proprioceptive policies are robust and agile but lack environmental awareness, making them insufficient for navigation in cluttered or goal-directed settings. To compensate, some works incorporate depth image or LiDAR-based elevation map to model terrain geometry [13, 43, 47, 55, 69, 84], improving foot placement but offering limited semantic understanding. Some works [8, 12, 71] combine RGB vision and language instructions with sim-to-real locomotion policies for semantic navigation, but rely on high-latency vision-language-action (VLA) models. VIRAL instead distills compact, RGB-driven visuomotor policies trained in randomized simulation to support real-time, goal-conditioned locomotion without sacrificing sim-to-real scalability.

Sim-to-Real for Manipulation Visual sim-to-real has been a key driver of progress in manipulation. A central tool is domain randomization: by varying rendering properties in simulation, policies trained on RGB can transfer to the real world [53, 58, 66]. This strategy enabled end-to-end RL for challenging skills—e.g., OpenAI's Dactyl reoriented objects with a five-finger hand using massive randomization and curriculum learning [2, 3], and later works scaled to tasks like Rubik's Cube solving and high-speed rotation [2, 24]. Early attempts that relied on high-fidelity RGB/depth/point-cloud simulation still struggled with the reality gap [2, 3, 25, 30, 33, 75, 76]. Recent methods improve robustness and scalability via a teacher–student paradigm: a privileged-state teacher is trained first, then a student is distilled from RGB with randomization [17, 42, 62]. However, prior work largely targets tabletop settings. We extend this paradigm to humanoid loco-manipulation.

Sim-to-Real for Loco-Manipulation Loco-manipulation requires a humanoid robot to move through an environment while simultaneously interacting with objects. It poses unique challenges for sim-to-real learning. Recent work explores sim-to-real learning for low-level loco-manipulation control, using either modular architectures that decouple leg and arm [6, 14, 42] or end-to-end policies that coordinate full-body motion [23, 26, 51]. On top of these controllers, some systems achieve task-level loco-manipulation via imitation learning or vision–language–action models [23, 54, 70], but these approaches require large real-world datasets and often lack robustness. VIRAL bridges both layers by training an RGB-driven, end-to-end policy entirely in simulation, enabling zero-shot deployment for humanoid goal-conditioned loco-manipulation without real-world demonstrations or large models.

6. Limitations and Discussions

While *sim-to-real* has demonstrated remarkable success in isolated capabilities—robust locomotion, geometric perception, and rigid-body manipulation—scaling these methods to general-purpose loco-manipulation (*"locomote anywhere, perceive anything, manipulate everything"*) exposes four critical coverage gaps that current paradigms have yet to bridge.

`Physics Coverage:`

The Physical Diversity Gap Modern simulators theoretically possess the capability to model complex dynamics, including fluid-structure interactions and deformable bodies. The fundamental bottleneck is not the lack of simulation features, but the *scalability of engineering effort* required to ground these features in reality. We can, with sufficient effort, engineer specific environments to simulate scooping rice, grasping noodles with tongs, cutting garlic, hand-crafting sushi, or feeding beans into a coffee machine. However, each of these scenarios requires bespoke tuning of material properties and boundary conditions to align with the real world. The challenge lies in scaling this effort to the open-ended diversity of daily life: modeling the damping of every cardboard box, the stiffness of every garment, the friction of specific oil stains, or the granular mechanics of food items. The barrier is not that we *cannot* simulate these interactions, but that the engineering cost to accurately instantiate them for the long tail of real-world physics arguably exceeds the complexity of collecting real-world data itself.

`Task Coverage:`

The Long-Tail of Task Generation Even if physics could be perfectly simulated, the diversity of tasks remains an unresolved challenge. Constructing a simulation environment for a single task (e.g., dishwashing) requires modeling not just object geometries, but their functional affordances, varied states (dirty vs. clean), and interaction logic. Scaling this to the thousands of distinct chores in a household environment presents a massive content generation bottleneck. Furthermore, simulation is limited by human imagination; we cannot simulate "unknown unknowns"—edge cases and task variants that only emerge during real-world deployment (e.g., adapting to a pet's interference or accommodating a human user with mobility constraints). Current asset taxonomies and generative procedural pipelines fail to capture this functional breadth.

`Reward and Policy Coverage:`

The Reward Engineering Bottleneck Defining "RL-friendly" reward functions that are both dense enough to guide exploration and sparse enough to prevent specification gaming is a delicate art that does not scale. In practice, we observe a tension between *under-exploration* (where dense, shaped rewards bias the policy toward local optima or simulator exploits) and *over-exploration* (where sparse rewards fail to bootstrap learning in high-dimensional spaces). For a single task, tuning these rewards to find the "Goldilocks" regime is feasible. However, manually designing robust reward functions for thousands of distinct tasks is intractable. This highlights a crucial trade-off: while sim-to-real offers scalable data generation, it demands high upfront engineering effort. In contrast, imitation learning moves the burden to data collection. As it stands, a few days of high-quality teleoperation data can often outperform months of sim-to-real engineering for specific tasks, primarily because the "reward" is implicitly provided by the human demonstrator, bypassing the specification problem entirely.

`Hardware Coverage:`

The Hardware-Simulation Gap Finally, a distinct gap remains between the idealized actuation in simulation and the reality of current humanoid hardware. While quasi-direct drive (QDD) actuators for locomotion are relatively well-modeled, dexterous manipulation hardware often suffers from unmodeled friction, backlash, thermal throttling, and sensor noise. Simulation policies that rely on precise finger positioning or force feedback often fail to transfer to hardware that lacks the requisite reliability and precision, limiting the complexity of tasks that can be genuinely attempted in the real world.

Outlook These four gaps suggest that while sim-to-real will retain a critical role in robotics—particularly for safe, stable evaluation and solving skills with bounded state-spaces—scaling it to solve general-purpose loco-manipulation is likely out of reach for the near future. The field has successfully identified the *sweet spot* for sim-to-real in locomotion: where aggressive randomization of limited parameters (terrain, mass) and carefully-designed reward functions produce robust policies that generalize well. However, the equivalent sweet spot for manipulation remains undiscovered, as the complexity of contact physics and semantic diversity in manipulation vastly exceeds that of locomotion tasks.

We believe the path forward involves redefining the role of simulation within a broader data ecosystem. Rather than forcing simulation to generate the entire distribution of the real world, the next frontier lies in integrating sim-to-real with the rapidly maturing stacks of real-world imitation learning and foundation models. Discovering this synergy—where simulation complements rather than replaces real-world learning—is the most exciting direction for the future of general-purpose loco-manipulation.

Acknowledgement

We thank Jeremy Chimienti, Tri Cao, Jazmin Sanchez, Isabel Zuluaga, Jesse Yang, Caleb Geballe, Haotian Lin, Lingyun Xu, Chaitanya Chawla, Jason Liu, Tony Tao, Ritvik Singh, Ankur Handa, Arthur Allshire, Guanzhi Wang, Yinzhen Xu, Runyu Ding, Xiaowei Jiang, Yuqi Xie, Jimmy Wu, Avnish Narayan, Kaushil Kundalia, Qi Wang, Scott Reed, Ziang Cao, Fengyuan Hu, Sirui Chen, Chenran Li, and Tingwu Wang for their help and support during this project.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 2, 8

[3] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2, 8

[4] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond. *URL https://github.com/Genesis-Embodied-AI/Genesis*, 2024. 2

[5] Jose Barreiros, Andrew Beaulieu, Aditya Bhat, Rick Cory, Eric Cousineau, Hongkai Dai, Ching-Hsin Fang, Kunimatsu Hashimoto, Muhammad Zubair Irshad, Masha Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint arXiv:2507.05331*, 2025. 2

[6] Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025. 2, 3, 8

[7] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 2

[8] Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2505.08712*, 2025. 8

[9] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on robot learning*, pages 66–75. PMLR, 2020. 3

[10] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024. 2

[11] Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, et al. Learning smooth humanoid locomotion through lipschitz-constrained policies. *arXiv preprint arXiv:2410.11825*, 2024. 8

[12] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024. 8

[13] Xuxin Cheng, Kexin Shi, Ananye Agarwal, and Deepak Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023. 2, 8

[14] 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. 8

[15] Jeremy Dao, Helei Duan, and Alan Fern. Sim-to-real learning for humanoid box loco-manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16930–16936. IEEE, 2024. 2

[16] Kourosh Darvish, Luigi Penco, Joao Ramos, Rafael Cisneros, Jerry Pratt, Eiichi Yoshida, Serena Ivaldi, and Daniele Pucci. Teleoperation of humanoid robots: A survey. *IEEE Transactions on Robotics*, 39(3):1706–1727, 2023. 2

[17] Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Wenhao Zhang, et al. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. *arXiv preprint arXiv:2505.03233*, 2025. 2, 8

[18] Helei Duan, Ashish Malik, Mohitvishnu S Gadde, Jeremy Dao, Alan Fern, and Jonathan Hurst. Learning dynamic bipedal walking across stepping stones. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6746–6752. IEEE, 2022. 8

[19] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. 2

[20] Xinyang Gu, Yen-Jen Wang, Xiang Zhu, Chengming Shi, Yanjiang Guo, Yichen Liu, and Jianyu Chen. Advancing humanoid locomotion: Mastering challenging terrains with denoising world model learning. *arXiv preprint arXiv:2408.14472*, 2024. 8

[21] Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrory, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C Karen Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *arXiv preprint arXiv:2501.02116*, 2025. 1

[22] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022. 5

[23] Huy Ha, Yihuai Gao, Zipeng Fu, Jie Tan, and Shuran Song. Umi on legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. *arXiv preprint arXiv:2407.10353*, 2024. 8

[24] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5977–5984. IEEE, 2023. 2, 8

[25] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021. 2, 8

[26] 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. 2, 8

[27] Tairan He, Chong Zhang, Wenli Xiao, Guanqi He, Changliu Liu, and Guanya Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024. 2

[28] 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)*, pages 9989–9996. IEEE, 2025. 2

[29] Yingdong Hu, Fanqi Lin, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024. 2

[30] Yangru Huang, Peixi Peng, Yifan Zhao, Guangyao Chen, and Yonghong Tian. Spectrum random masking for generalization in image-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:20393–20406, 2022. 2, 8

[31] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019. 8

[32] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0. 5: a vision-language-action model with open-world generalization, 2025. *URL https://arxiv. org/abs/2504.16054*, 1(2):3. 2

[33] Yunfan Jiang, Chen Wang, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. *arXiv preprint arXiv:2405.10315*, 2024. 2, 8

[34] Yunfan Jiang, Ruohan Zhang, Josiah Wong, Chen Wang, Yanjie Ze, Hang Yin, Cem Gokmen, Shuran Song, Jiajun Wu, and Li Fei-Fei. Behavior robot suite: Streamlining real-world whole-body manipulation for everyday household activities. *arXiv preprint arXiv:2503.05652*, 2025. 2

[35] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021. 8

[36] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[37] Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

[38] Zhongyu Li, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, 44(5): 840–888, 2025. 2, 8

[39] Qiayuan Liao, Takara E Truong, Xiaoyu Huang, 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. 2

[40] Qiayuan Liao, Bike Zhang, Xuanyu Huang, Xiaoyu Huang, Zhongyu Li, and Koushil Sreenath. Berkeley humanoid: A research platform for learning-based control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2897–2904. IEEE, 2025. 8

[41] Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025. 2, 4

[42] Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ri-Zhao Qiu, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. *arXiv preprint arXiv:2403.16967*, 2024. 2, 8

[43] Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9997–10003. IEEE, 2025. 2, 8

[44] Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025. 2, 3

[45] Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, et al. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. *arXiv preprint arXiv:2503.24361*, 2025. 2

[46] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021. 2

[47] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022. 8

[48] 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. 2, 3, 7

[49] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299. IEEE, 2018. 4

[50] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. 2

[51] Guoping Pan, Qingwei Ben, Zhecheng Yuan, Guangqi Jiang, Yandong Ji, Shoujie Li, Jiangmiao Pang, Houde Liu, and Huazhe Xu. Roboduet: Learning a cooperative policy for whole-body legged loco-manipulation. *IEEE Robotics and Automation Letters*, 2025. 8

[52] 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. 4, 6

[53] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018. 8

[54] Ri-Zhao Qiu, Yuchen Song, Xuanbin Peng, Sai Aneesh Suryadevara, Ge Yang, Minghuan Liu, Mazeyu Ji, Chengzhe Jia, Ruihan Yang, Xueyan Zou, et al. Wildlma: Long horizon loco-manipulation in the wild. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10011–10019. IEEE, 2025. 8

[55] Junli Ren, Tao Huang, Huayi Wang, Zirui Wang, Qingwei Ben, Junfeng Long, Yanchao Yang, Jiangmiao Pang, and Ping Luo. Vb-com: Learning vision-blind composite humanoid locomotion against deficient perception. *arXiv preprint arXiv:2502.14814*, 2025. 8

[56] 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. 2, 3, 4, 1

[57] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pages 91–100. PMLR, 2022. 2, 3

[58] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016. 8

[59] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3, 1

[60] Dana Sharon and Michiel van de Panne. Synthesis of controllers for stylized planar bipedal walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2387–2392. IEEE, 2005. 4

[61] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 4, 6

[62] Ritvik Singh, Arthur Allshire, Ankur Handa, Nathan Ratliff, and Karl Van Wyk. Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024. 2, 8

[63] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018. 2, 8

[64] Generalist AI Team. Gen-0: Embodied foundation models that scale with physical interaction. *Generalist AI Blog*, 2025. https://generalistai.com/blog/preview-uqlxvb-bb.html. 2

[65] Dhruva Tirumala, Markus Wulfmeier, Ben Moran, Sandy Huang, Jan Humplik, Guy Lever, Tuomas Haarnoja, Leonard Hasenclever, Arunkumar Byravan, Nathan Batchelor, et al. Learning robot soccer from egocentric vision with deep reinforcement learning. *arXiv preprint arXiv:2405.02425*, 2024. 2

[66] 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)*, pages 23–30. IEEE, 2017. 8

[67] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012. 7

[68] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020. 3, 5

[69] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025. 8

[70] Kaijun Wang, Liqin Lu, Mingyu Liu, Jianuo Jiang, Zeju Li, Bolin Zhang, Wancai Zheng, Xinyi Yu, Hao Chen, and Chunhua Shen. Odyssey: Open-world quadrupeds exploration and manipulation for long-horizon tasks. *arXiv preprint arXiv:2508.08240*, 2025. 8

[71] Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025. 8

[72] Haoyang Weng, Yitang Li, Nikhil Sobanbabu, Zihan Wang, Zhengyi Luo, Tairan He, Deva Ramanan, and Guanya Shi. Hdmi: Learning interactive humanoid whole-body control from human videos. *arXiv preprint arXiv:2509.16757*, 2025. 2

[73] Haoyu Xiong, Russell Mendonca, Kenneth Shaw, and Deepak Pathak. Adaptive mobile manipulation for articulated objects in the open world. *arXiv preprint arXiv:2401.14403*, 2024. 2

[74] 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. 2

[75] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024. 2, 8

[76] Yanjie Ze, Nicklas Hansen, Yinbo Chen, Mohit Jain, and Xiaolong Wang. Visual reinforcement learning with self-supervised 3d representations. *IEEE Robotics and Automation Letters*, 8(5):2890–2897, 2023. 2, 8

[77] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024. 2

[78] Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025. 2, 3

[79] Yuanhang Zhang, Yifu Yuan, Prajwal Gurunath, Tairan He, Shayegan Omidshafiei, Ali-akbar Agha-mohammadi, Marcell Vazquez-Chanlatte, Liam Pedersen, and Guanya Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025. 2

[80] 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. 2

[81] Tony Z Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity. *arXiv preprint arXiv:2410.13126*, 2024. 2

[82] Peiyuan Zhi, Peiyang Li, Jianqin Yin, Baoxiong Jia, and Siyuan Huang. Learning unified force and position control for legged loco-manipulation. *arXiv preprint arXiv:2505.20829*, 2025. 2

[83] Yuke Zhu, Ziyu Wang, Josh Merel, Andrei Rusu, Tom Erez, Serkan Cabi, Saran Tunyasuvunakool, János Kramár, Raia Hadsell, Nando de Freitas, et al. Reinforcement and imitation learning for diverse visuomotor skills. *arXiv preprint arXiv:1802.09564*, 2018. 2

[84] Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024. 2, 8

VIRAL: Visual Sim-to-Real at Scale for Humanoid Loco-Manipulation

Supplementary Material

7. Training Details

7.1. Observation Details

Table 1 summarizes the observation terms and their corresponding dimensions.

State term	Dimensions
Base linear velocity	3
Base angular velocity	3
Projected gravity	3
Actions	31
Stage	5
Delta actions	11
DoF position	43
DoF velocity	43
Placement position	2
Table–pelvis transform	9
Finger-tip forces for hold_object	12
Hold_object transform	9
Hold_object–hand transform	9
Target pre-place position	3
Finger-tip forces for grasp_object	12
Grasp_object transform	9
Grasp_object–hand transform	9
Target lift position	3
HOMIE commands	7
Single-step total dim	226

Table 1. Observation dimensions for `teacher`.

Table 2 lists the observation terms and their corresponding dimensions. In addition to these state observations, we feed an RGB image of size 108×192 into the vision encoder. The resulting 128-dimensional visual feature is concatenated with the state observations and then passed to the policy head.

State term	Dimensions
Base angular velocity	3
Projected gravity	3
Actions	31
DoF position (w/o fingers)	29
DoF velocity (w/o fingers)	29
Delta actions	11
HOMIE commands	7
Single-step total dim	113

Table 2. Observation dimensions for `student`.

7.2. Reward Details

A single place–pickup cycle is decomposed into five stages: (1) walking toward the object; (2) moving the arm and hand to a pre-place pose; (3) placing the object; (4) grasping and lifting the next object; and (5) turning. Repeating this sequence produces a long-horizon loco-manipulation loop. At each step, the total reward is a stage-weighted sum

$$r_t = \sum_{i=0}^{4} w_i \mathbb{1}[s_t = i] r_t^{(i)}, \qquad w_i > 0,$$

and stage transitions are governed by stage-specific advancement and completion criteria. Table 3 instantiates $r^{(s)}$ with stage-dependent shaping terms for teacher policy.

7.3. Hyperparameters Details

Table 4 lists the PD gains used for the Unitree G1 robot equipped with 3-finger dexterous hands.

Table 5 lists the hyperparamters for teacher policy trained by PPO [59].

Table 6 lists the hyperparameters for student policy trained by the mixture of DAgger [56] and Behavior Cloning.

7.4. Domain Randomization

Table 7 summarizes all randomizations used during policy training, including image quality, dome lighting, materials, table properties, and camera extrinsics.

Term	Expression	Weight	Stage(s)
Termination / generic penalties			
Termination	$\mathbb{1}_{\{\text{termination}\}}$	-2000.0	0–4
Delta action rate	$\|\Delta a_t\|_2^2$	-0.01	0–4
DoF velocity	$\|\dot{\mathbf{q}}\|_2^2$	-0.5	0–4
DoF acceleration	$\|\ddot{\mathbf{q}}\|_2^2$	-3.0×10^{-6}	0–4
Torque limits	$\|\boldsymbol{\tau}\|_2^2$	-0.001	0–4
Output smoothness	$\|\pi_t - \pi_{t-1}\|_2^2$	-9.0	0–4
Finger primitive limits	$\lvert\text{clip}(u_{\text{finger}}, [l, u]) - u_{\text{finger}}\rvert$	-20.0	0–4
Fast right-arm velocity	$\|\dot{\mathbf{q}}_{\text{right arm}}\|_2^2$	-80.0	0–4
Finger qvel, when contacting ground with single-foot	$\|\dot{\mathbf{q}}_{\text{finger}}\|_2 \, \mathbb{1}_{\text{single-foot}}$	-3000.0	1–3
Arm qvel, when contacting ground with single-foot	$\|\dot{\mathbf{q}}_{\text{right arm}}\|_2 \, \mathbb{1}_{\text{single-foot}}$	-1300.0	1–3
Heading / command shaping			
Heading toward object	$\big((\psi_{\text{GraspObj}} - \psi_{\text{robot}})/\pi\big)^2$	-10000.0	0
Object in view	$\mathbb{1}[y_{\text{right hand}} > y_{\text{GraspObj}} - 0.1] + \mathbb{1}[y_{\text{left hand}} < y_{\text{GraspObj}} + 0.1]$	-1.0	0
Large linear v_x command	$\sum \max\big(0, \lvert v_x^{\text{cmd}}\rvert - 0.5\big)$	-20.0	0–4
Large linear v_y command	$\sum \max\big(0, \lvert v_y^{\text{cmd}}\rvert - 0.5\big)$	-20.0	0–4
Large angular ω command	$\sum \max\big(0, \lvert \omega^{\text{cmd}}\rvert - 0.5\big)$	-20.0	0–4
Large upper-body actions	$\sum \max\big(0, \lvert u_{\text{upper}}\rvert - 2\pi\big)$	-20.0	0–4
Zero linear v_x, linear v_y, angular ω cmd	$\lvert v_x^{\text{cmd}}\rvert + \lvert v_y^{\text{cmd}}\rvert + \lvert \omega^{\text{cmd}}\rvert$	-12.0	1–3
Zero linear v_x, linear v_y cmd	$\lvert v_x^{\text{cmd}}\rvert + \lvert v_y^{\text{cmd}}\rvert$	-4.0	4
Task / object-centric rewards			
Robot-Object distance	$\exp(-4\,(\|p_{\text{robot}} - p_{\text{GraspObj}}\| - 0.45)^2)$	2.0	0–4
Upper-body actions (pose)	$\|\mathbf{q}_{\text{right arm}}\|_2^2$	-1.0	0
Keep hand closed	$\exp\big(-4\,(u_{\text{finger}} - u_{\text{close}})^2\big)$	9.0	0–1, 3–4
Place objects when near tray	$-\|\mathbf{f}_{\text{PlaceObj}}\| * \mathbb{1}\,(\|p_{\text{PlaceObj}} - p_{\text{tray}}\| < 0.3)$	10.0	0–1
Holding object	$\exp\big(-4\|p_{\text{PlaceObj}} - p_{\text{hand}}\|_2\big)$	1.0	0–4
Hand–object distance	$\exp\big(-10\max_k \|p_{\text{finger}}^{(k)} - p_{\text{GraspObj}}\|_2\big)$	20.0	3–4
Grasp based on obj–finger dir	$-\hat{\mathbf{d}}_{\text{thumb}}^{\top}\hat{\mathbf{d}}_{\text{index}}$	5.0	3–4
Grasp force	$\sum \|\mathbf{f}_{\text{GraspObj-hand}}\|$	1.0	3–4
Lift goal distance	$\exp(-10\|\|p_{\text{GraspObj}} - p_{\text{goal}}\|\|^2)$	10.0	3–4
Lift z	$\min(h_{\text{GraspObj}} - h_{\text{table}}, 0.15)$	200.0	3–4
Turn around	$-\lvert \text{y}_{\text{robot}} - \text{y}_{\text{desired}}\rvert$	15.0	4
Right-arm qpos tracking (hold)	$\exp\big(-4\|\mathbf{q}_{\text{right arm}} - \mathbf{q}^*_{\text{Place}}\|_2\big)$	5.0	0–2
Right-arm qpos tracking (front)	$\exp\big(-4\|\mathbf{q}_{\text{right arm}} - \mathbf{q}^*_{\text{Grasp}}\|_2\big)$	25.0	3–4
Finger qvel during right-arm qvel	$\exp\big(-6\|\dot{\mathbf{q}}_{\text{arm}}\|_2\|\dot{\mathbf{q}}_{\text{finger}}\|_2\big)$	15.0	1–4
Object–table contact move	$\|\mathbf{v}_{\text{GraspObj},xy}\| \, \mathbb{1}_{\text{table-contact}}$	-1000.0	1–4
Object relative move (hand–obj v_z)	$\lvert v^z_{\text{GraspObj}} - v^z_{\text{hand}}\rvert \, \mathbb{1}_{\text{in-grasp}}$	-3000.0	1–3
Object lean during pick	$\lvert\phi_{\text{GraspObj}}\rvert + \lvert\theta_{\text{GraspObj}}\rvert$	-500.0	0–3
Object non–z velocity during pick	$\|\mathbf{v}_{\text{GraspObj},xy}\|_2$	-500.0	0–3

Table 3. Reward components, expressions, weights, and the stages (0–4) where each term is applied.

Joint	K_p **[N·m/rad]**	K_d **[N·m·s/rad]**
hip_yaw	150	2.0
hip_roll	150	2.0
hip_pitch	150	2.0
knee	200	4.0
ankle_pitch	40	2.0
ankle_roll	40	2.0
waist_yaw	250	5.0
waist_roll	250	5.0
waist_pitch	250	5.0
shoulder_pitch	100	5.0
shoulder_roll	100	5.0
shoulder_yaw	40	2.0
elbow	40	2.0
wrist_roll	20	2.0
wrist_pitch	20	2.0
wrist_yaw	20	2.0
hand_index	0.5	0.1
hand_middle	0.5	0.1
hand_thumb_1	0.5	0.1
hand_thumb_2	0.5	0.1
hand_thumb_0	2.0	0.1

Table 4. Joint-space PD gains (K_p, K_d) used in the low-level controller.

Hyperparameters	**Values**
Number of environments	32768 (2048*8GPUs*2Nodes)
Discount factor (γ)	0.998
Learning rate	0.00002
Entropy coefficient	0.01
Value loss coefficient	1
Init noise std (RL)	0.5
MLP size	[512, 256, 128]

Table 5. Hyperparameters for teacher policy.

Hyperparameters	**Values**
Number of environments	65535 (1024*8GPUs*8Nodes)
Number of steps per environment	1
Learning rate	0.0002

Table 6. Hyperparameters for student policy.

Table 7. Comprehensive domain randomization parameters during training

Parameter	Probability	Distribution
Image Augmentation		
Brightness	0.25	$\sim \mathcal{U}(0.7, 2)$
Contrast	0.25	$\sim \mathcal{U}(0.5, 1.5)$
Hue	0.5	$\sim \mathcal{U}(-0.1, 0.1)$
Saturation	0.25	$\sim \mathcal{U}(0.5, 2)$
Gaussian Noise Std	0.25	$\sim \mathcal{U}(0.0, 0.15)$
Gaussian Blur Kernel Size	0.25	$\sim \mathcal{U}(3, 5)$
Gaussian Blur Sigma	0.25	$\sim \mathcal{U}(0.1, 1.5)$
Lighting		
Dome Light Intensity	1.0	$\sim \mathcal{U}(800, 2000)$
Dome Light Yaw Rotation	1.0	$\sim \mathcal{U}(-\pi, \pi)$
Dome Light Texture Map	1.0	$\sim \mathcal{U}$(texture_maps) (Indoor, Clear, Cloudy, Night, Studio)
Material Randomization		
Robot Material - Roughness	1.0	$\sim \mathcal{U}(0.0, 0.8)$
Robot Material - Metallic	1.0	$\sim \mathcal{U}(0.0, 0.8)$
Robot Material - Specular	1.0	$\sim \mathcal{U}(0.0, 0.8)$
Floor Material Texture	1.0	$\sim \mathcal{U}$(texture_maps) (Wood, Carpet, Masonry, Metals, Natural, Plastics, Stone, Wall Board)
Table Material Texture	1.0	$\sim \mathcal{U}$(texture_maps) (Wood)
Object Material Texture	1.0	$\sim \mathcal{U}$(texture_maps) (All Base Materials)
Table Physical Properties		
Table Height (m)	1.0	$\sim \mathcal{U}(0.65, 0.6775)$
Table Depth (m)	1.0	$\sim \mathcal{U}(0.7, 0.75)$
Table Width (m)	1.0	$\sim \mathcal{U}(1.4, 1.6)$
Table Thickness (m)	1.0	$\sim \mathcal{U}(0.035, 0.04)$
Camera Extrinsics		
Position Noise - X (m)	1.0	$\sim \mathcal{U}(-0.02, 0.02)$
Position Noise - Y (m)	1.0	$\sim \mathcal{U}(-0.05, 0.05)$
Position Noise - Z (m)	1.0	$\sim \mathcal{U}(-0.02, 0.02)$
Rotation Noise - Roll (rad)	1.0	$\sim \mathcal{U}(-0.05, 0.05)$
Rotation Noise - Pitch (rad)	1.0	$\sim \mathcal{U}(-0.1, 0.1)$
Rotation Noise - Yaw (rad)	1.0	$\sim \mathcal{U}(-0.05, 0.05)$