ZERO-HLM: GENERALIZABLE HUMANOID LOCO-MANIPULATION WITHOUT REAL ROBOT DATA

Anonymous authors
Paper under double-blind review



Figure 1: *Top:* **Zero-HLM** generates humanoid loco-manipulation data covering diverse whole-body skills in simulation. *Bottom:* Policies trained solely on these data achieve direct sim-to-real transfer on a Unitree G1 under BASE and generalize across EXPANDED configurations.

ABSTRACT

Learning generalizable humanoid loco-manipulation remains bottlenecked by costly real-world demonstrations. Simulation offers a scalable alternative, but extending it to humanoids requires solving two coupled challenges: constructing scene-scale environments that are both visually faithful and physics-ready, and generating successful demonstrations under unified whole-body control. We present **Zero-HLM**, a real-to-sim-to-real framework that transforms smartphone captures and simulated source demonstrations into scalable training data for humanoid loco-manipulation. We couple Gaussian visual reconstruction with agentic physical scene construction to build interaction-ready digital twins with aligned visual and physical geometry. Within the environment, our motion–environment co-adaptation strategy synthesizes unified whole-body references and uses rollout feedback to restore interaction geometry when tracking deviations disrupt execution. We successfully generate 4,800 physically validated demonstrations in simulation using **Zero-HLM**, averaging a 1.8× speedup over real-world data collection. Policies trained solely on these data transfer zero-shot to a Unitree G1 across three long-horizon tasks. With GR00T N1.7, simulation-only training achieves 66.7% mean full-task success with an absolute 20% gain over real-data training under matched configurations. Across five generalization settings, pooled real-world success increases from 23% to 69% without additional real demonstrations. To support systematic evaluation, we establish a benchmark with nine skill stages and shared protocols for comparing models and studying data scaling.

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

1 INTRODUCTION

Humanoid robots combine mobility and dexterity with a morphology naturally suited to human environments. Recent advances in whole-body control and visuomotor policies have enabled diverse loco-manipulation behaviors (Luo et al., 2026; Wu et al., 2026a; NVIDIA et al., 2025; Black et al., 2024; Li et al., 2026c; Jiang et al., 2025; Shi et al., 2026; Fu et al., 2024). However, learning generalizable behaviors requires diverse interaction data that remain costly and difficult to collect on a physical humanoid (Yu et al., 2026; Kim et al., 2026; Zhong et al., 2025; Seo et al., 2023).

Simulation-based data generation offers a scalable alternative to costly physical collection, but its practical value depends on constructing simulation environments that support reliable sim-to-real transfer (Xiang et al., 2026). For humanoids, locomotion expands the workspace beyond a local manipulation area, while a moving egocentric camera observes the environment from changing positions and heights. This calls for scene-level reconstruction that preserves both visual fidelity and the physical geometry and dynamics required for interaction. Existing Gaussian-based real-to-sim-to-real methods have shown promising transfer in tabletop settings (Yang et al., 2025; Han et al., 2026; Yu et al., 2025b; Qureshi et al., 2025; Barcellona et al., 2025; Li et al., 2024; Jiang et al., 2026). However, extending this paradigm to scene-scale humanoid interaction is considerably more demanding, as extracted meshes from Gaussian model can contain contact artifacts, and articulated mechanisms require additional modeling effort (Han et al., 2026; Lee et al., 2026).

Beyond reconstructing such environments, generating humanoid interaction data at scale remains challenging. Recent methods (Kim et al., 2026; Lin et al., 2026; Fu et al., 2026) simplify data generation by specifying locomotion and upper-body commands separately. This separation can limit motion naturalness and support for leg-critical interactions, motivating unified whole-body reference motion tracking (Luo et al., 2026; Wu et al., 2026a). However, scaling data generation under unified tracking presents two challenges: (i) *Whole-body reference motion synthesis*. Adapting demonstrations to varied spatial configurations requires coordinated changes throughout the body while preserving task-relative interaction (Yang et al., 2026). (ii) *Execution mismatch*. Whole-body controllers must reconcile reference tracking with balance and contact constraints, so precise state-by-state replay cannot be assumed (Liao et al., 2026). Even kinematically valid reference motions can lead to base and hand deviations that invalidate planned grasps or placements.

To address these challenges, we propose **Zero-HLM**, a real-to-sim-to-real framework for generalizable humanoid loco-manipulation. First, inspired by recent MLLM-based agentic frameworks that produce faithful digital cousin reconstructions (Dai et al., 2024; Yu et al., 2025a), we propose **Gaussian-Guided Agentic Real-to-Sim**, which builds a scene-level digital twin by coupling a high-fidelity Gaussian reconstruction with an agentic pipeline that iteratively aligns the digital cousins to the real scene, using the Gaussian reconstruction as the alignment reference. We further composite the Gaussian rendering with the simulator's rendering of the aligned scene, yielding a hybrid rendering that narrows the visual sim-to-real gap. To tackle the mismatch between synthesized reference motions and their tracked execution, we introduce a motion–environment co-adaptation strategy for scalable data generation under unified whole-body control. **Demonstration-Anchored Whole-Body Motion Synthesis** combines locomotion planning with object-relative adaptation of source demonstrations, producing continuous whole-body reference motions for varied spatial configurations. **Execution-Conditioned Correction** then uses rollout feedback to adjust scene configurations and reference motions, restoring interactions disrupted by tracking deviations. We pair the resulting physically validated trajectories with photorealistic egocentric observations for policy training. Policies trained solely on this simulated data transfer directly to the real world and generalize across spatial configurations. Our contributions are summarized as follows:

- We propose **Zero-HLM**, a real-to-sim-to-real framework for unified humanoid loco-manipulation. It combines Gaussian-guided agentic reconstruction with whole-body data generation in visually faithful, physically interactive scenes.
- We introduce **motion–environment co-adaptation**, which couples demonstration-anchored motion synthesis with execution-conditioned correction to compensate for tracking deviations, yielding **4,800** successful trajectories across three tasks at a **1.8**$\times$ speedup over real-world collection.
- We demonstrate **direct sim-to-real transfer and real-world generalization** through simulation-only training. With GR00T N1.7, simulation-only training yields an **absolute gain of 20%** over real-data training under matched configurations. Across five generalization settings, pooled real-world success increases from **23% to 69%** without additional real demonstrations.

Table 1: Comparison of humanoid loco-manipulation data-generation methods. **Unified Policy**: unified whole-body action prediction. **Physics-Ready Real-to-Sim**: automated scene-level reconstruction with articulated components for physical simulation. **Few-Demo Skill Adapt.**: task-specific skill adaptation from few demonstrations. **Leg-Critical Interaction**: direct leg/foot–object interaction, excluding locomotion and squatting.

Method	Policy	Data Generation Pipeline			Capability & Deployment		
	Unified Policy	Physics-Ready Real-to-Sim	Few-Demo Skill Adapt.	Photorealistic Render	Articulated Task	Leg-Critical Interaction	Zero-Shot Deployment
DemoHLM (Fu et al., 2026)	✗	✗	✓	✗	✓	✗	✓
VIRAL (He et al., 2026)	✗	✗	✗	✗	✗	✗	✓
FetchMan (Rayyan et al., 2026)	✗	✗	✗	✗	✗	✗	✓
HumanoidMimicGen (Lin et al., 2026)	✗	✗	✓	✗	✗	✗	✗
LEGS (Kim et al., 2026)	✗	✗	✗	✓	✗	✗	✓
OASIS (Yu et al., 2026)	✓	✗	✗	✓	✗	✗	✓
VLK (Wang et al., 2026)	✓	✗	✗	✓	✗	✗	✓
Ours	✓	✓	✓	✓	✓	✓	✓

- We establish a **humanoid loco-manipulation benchmark** with three long-horizon tasks and nine skill stages, comparing diverse policy families under shared training and execution protocols.

2 Related Work

Humanoid Loco-Manipulation. Humanoid loco-manipulation requires coordinated motion to interact with the environment while maintaining balance. Decoupled systems control locomotion and upper-body manipulation separately (Ben et al., 2025; Lu et al., 2025), whereas unified whole-body trackers follow coordinated reference motions through a shared control interface (Luo et al., 2026; Li et al., 2026b; Zhang et al., 2026). Building on these advances, existing data collection systems still require substantial human labor (Ze et al., 2025a; He et al., 2025; Nai et al., 2026; Shi et al., 2026; Ze et al., 2025b; Zhao et al., 2025). Meanwhile, vision-language-action and world-action models rely on diverse interaction data to learn generalizable behaviors (Physical Intelligence et al., 2025; NVIDIA et al., 2025; Ye et al., 2026; Ma et al., 2026; Bai et al., 2026; Wu et al., 2026b). We address this bottleneck through scalable simulation data generation under unified whole-body control.

Real-to-Sim-to-Real for Robot Learning. Real-to-sim-to-real learning combines scene reconstruction with simulation-based data generation for real-world policy transfer. *For scene reconstruction*, Gaussian-based methods recover captured appearance to narrow the visual sim-to-real gap (Yang et al., 2025; Han et al., 2026; Qureshi et al., 2025). Recent agentic systems seek to improve physical fidelity by coordinating geometry modeling, physical parameter inference, and simulator configuration (Chen et al., 2026; Li et al., 2026a). Our approach grounds agentic scene construction in Gaussian geometry and matched-view renderings, constraining the initial layout and local refinement to remain consistent with the captured environment. *For demonstration generation*, source-based methods expand data coverage through spatial adaptation, using either simulation rollouts or direct observation–action transformations (Mandlekar et al., 2023; Xue et al., 2025). For humanoids, RL-based methods learn loco-manipulation with task-specific rewards (He et al., 2026; Rayyan et al., 2026). Others generate demonstrations for imitation learning through separately specified locomotion and upper-body commands (Fu et al., 2026; Lin et al., 2026; Kim et al., 2026). VLK adopts a unified whole-body interface, but its demonstrated interactions remain focused on single-object transport (Wang et al., 2026). Our approach generates whole-body interaction data through demonstration-anchored motion synthesis and uses execution feedback to correct tracking-induced interaction errors. Tab. 1 compares humanoid data-generation pipelines and their capabilities.

3 The ZERO-HLM Framework

Our goal is to reconstruct a visually faithful, physics-ready simulation environment from smartphone captures and generate training data at scale for generalizable humanoid loco-manipulation. Fig. 2 illustrates the overall framework. The following sections describe agentic real-to-sim reconstruction (Sec. 3.1) that supports both physical interaction and egocentric rendering, motion–environment co-adaptation for scalable data generation (Sec. 3.2), and real-world deployment (Sec. 3.3).

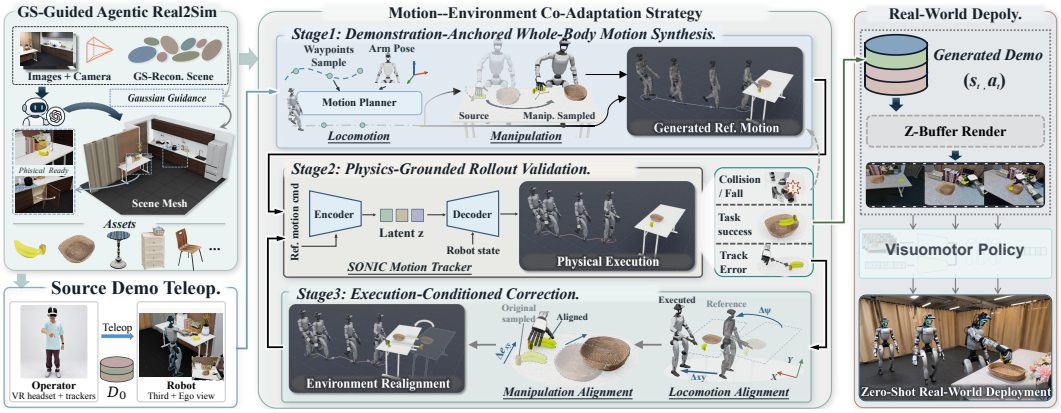


Figure 2: **Overview of Zero-HLM.** *Left:* Gaussian-guided agentic real-to-sim reconstruction aligns the physical scene with captured appearance and geometry. *Middle:* Motion–environment co-adaptation generates whole-body data from source demonstrations. *Right:* successful rollouts are paired with egocentric renderings to train policies for zero-shot real-world deployment.

3.1 Gaussian-Guided Agentic Real-to-Sim Reconstruction

To narrow the sim-to-real gap, we aim to construct a visually faithful, physics-ready simulation environment with aligned visual and physical representations. We build this environment $\mathcal{E}$ from hybrid representations: a Gaussian reconstruction $\mathcal{G}$ for appearance and spatial reference, and an editable mesh-based physical scene for collision and articulation. We reconstruct $\mathcal{G}$ from smartphone images using MAtCha (Guédon et al., 2025), with checkerboards establishing metric scale and ground alignment. A multimodal agent then uses $\mathcal{G}$ and a mesh extracted from it to construct and configure the physical scene using Blender MCP and IsaacLab. The extracted mesh guides the initial layout and geometry, while $\mathcal{G}$ provides an active-view rendering interface for local refinement. The agent therefore selects inspection viewpoints and compares Gaussian and mesh renderings at matched camera poses to iteratively refine local geometry and placement. The resulting hybrid scene $\mathcal{E}$ integrates a mesh-based physics backend with a Gaussian visual frontend in a shared metric frame, aligning physical interactions with visual observations. More details appear in Appendix A.1.

Interaction-critical assets are reconstructed separately to preserve the local geometry required for physical contact. In IsaacLab, the agent configures collision geometry, joints, and physical parameters, then refines them through interaction checks. For example, cabinet opening tests articulated motion, while object release tests gravity-driven motion and supporting contacts.

3.2 Motion–Environment Co-Adaptation for Scalable Data Generation

We collect a small set of successful simulation teleoperation demonstrations per task as the source pool $\mathcal{D}_0$. Each demonstration is segmented into locomotion, manipulation, and leg-critical interaction using event annotations (Appendix A.2). From $\mathcal{D}_0$, we construct a *whole-body reference motion* $\mathcal{R} = (r_1, \ldots, r_T)$ for each sampled configuration. Each reference pose $r_t = (p_t, R_t, q_t)$ specifies root position, orientation, and body joints. A simulated *rollout* executes this motion through SONIC (Luo et al., 2026), with separate hand commands. Successful rollouts are retained in the dataset $\mathcal{D}$. Here, *stage* denotes a generation step and *segment* a temporal portion of the motion.

Stage 1: Demonstration-Anchored Whole-Body Motion Synthesis. For each attempt, we sample a task configuration around the source configuration, including the robot interaction pose and relevant object or placement-target poses. The initial walking position is sampled from navigable space. Given the sampled configuration, locomotion segments are replanned with A*, while manipulation segments are adapted relative to the sampled interaction targets. The SONIC motion planner converts each planned path and sampled walking speed into a locomotion reference-motion segment. When an upper-body posture must be retained, such as when carrying an object while walking, we fix the corresponding joint angles in the planner's input context throughout motion generation. For leg-critical segments, we combine the source lower-body motion with the terminal upper-body posture inherited from the preceding segment.

For manipulation, we hold the reference root pose fixed and adapt the arm motion while preserving the source hand trajectory relative to its interaction object or placement target. Let $T_o^{\mathrm{src}}, T_o \in SE(3)$ denote the source and sampled poses of that object or target, and $T_{e,t}^{\mathrm{src}} \in SE(3)$ the source hand pose, all in the world frame. The adapted target is

$$\widetilde{T}_{e,t} = T_o (T_o^{\mathrm{src}})^{-1} T_{e,t}^{\mathrm{src}}. \tag{1}$$

This transformation adapts the hand trajectory to the sampled target while preserving the demonstrated hand–target geometry. Arm inverse kinematics (IK) updates the active joints, while the root pose and inactive joints retain their source or inherited values. To maintain continuity, adjacent reference-motion segments are aligned in both boundary pose and timing. Each segment inherits the preceding segment's final reference pose. To connect it to a retargeted manipulation trajectory, we add a hand-position offset that gradually vanishes:

$$p_{e,t}^{\mathrm{blend}} = \widetilde{p}_{e,t} + \lambda_t \big(p_e^{\mathrm{in}} - \widetilde{p}_{e,t_0} \big). \tag{2}$$

Here, $\widetilde{p}_{e,t}$ is the position component of $\widetilde{T}_{e,t}$, p_e^{in} is the inherited hand position, and t_0 denotes the start of the transition. As λ_t smoothly decreases from one to zero, the blended trajectory begins at the inherited position and joins the retargeted source path. Retargeting can increase the translational or rotational displacement across segment boundaries. We therefore extend the transition duration according to the source hand's linear and angular speeds while keeping grasp and release commands temporally aligned with the retimed motion. Appendix A.3 details target construction, spatial blending, and timing conventions.

Stage 2: Physics-Grounded Rollout Validation. We execute each synthesized reference motion in simulation to verify its feasibility under balance, contact, and tracking constraints. At each step, the SONIC encoder maps a reference-motion window $\mathcal{R}_t$ drawn from $\mathcal{R}$ to a 64-dimensional latent command $z_t = E(\mathcal{R}_t)$. Its decoder combines z_t with proprioceptive history h_t to produce body joint targets $u_t = D(z_t, h_t)$ for PD control. Two hand-opening commands $g_t \in [0,1]^2$ are executed separately. We record actions $a_t = (z_t, g_t) \in \mathbb{R}^{66}$ together with the executed robot and object states. The realized root, hand, and object poses also provide feedback for Stage 3. Rollouts satisfying the task, stability, and collision criteria are retained in $\mathcal{D}$. Recoverable tracking mismatches, such as locomotion-endpoint or grasp-height errors, are routed to Stage 3 for correction. Falls, invalid collisions, and failures outside the correction scope trigger resampling in Stage 1.

Stage 3: Execution-Conditioned Correction. Tracking deviations can disrupt the relative poses required for interaction even when the robot remains balanced. We use execution feedback from Stage 2 to restore the required interaction geometry: planar errors are corrected by adjusting scene or object configurations, while grasp-height errors are corrected by adjusting the reference motion.

Robot $SE(2)$ correction. Locomotion tracking error changes the realized robot pose relative to the subsequent interaction target. We compensate by transforming the relevant scene assets to restore their intended relation to the executed root pose. Let (x, ψ) and (x', ψ') be the intended and executed planar root poses, where $x \in \mathbb{R}^2$ is position and ψ is yaw. For an affected asset with pose (x_j, ψ_j) in the same world frame, we apply

$$x_j^+ = x' + R_2(\psi' - \psi)(x_j - x), \qquad \psi_j^+ = \psi_j + \psi' - \psi. \tag{3}$$

Here, $R_2(\cdot)$ denotes planar rotation and the superscript $+$ denotes the corrected pose.

Object $SE(2)$ correction. Root alignment alone does not remove local errors at grasp and placement. We adjust the interaction object or placement target using the additional planar tracking error relative to a successful source rollout:

$$\Delta e_{xy} = e_{xy}^{\mathrm{cand}} - e_{xy}^{\mathrm{src}}. \tag{4}$$

Here, e_{xy}^{cand} and e_{xy}^{src} denote the planar tracking errors in the candidate and successful source rollouts, respectively. Each error is the executed-minus-reference position of the interacting hand or carried-object anchor at the corresponding grasp or release event, expressed in the same planar frame. Their difference Δe_{xy} therefore isolates the additional tracking deviation relative to a physically successful source execution. We translate the interaction object or placement target by this residual and update its yaw according to the task-specific placement rule. Robot-relative targets preserve their source yaw offset from the executed robot heading, whereas fixed-orientation targets retain their configured yaw. Object heights remain unchanged.

Z-axis compensation. Planar correction does not address vertical wrist–object misalignment at grasp closure. Around grasp closure, we compare the executed wrist height relative to the object between the candidate and successful source rollouts and adjust the reference wrist height to recover the source-relative vertical relation. The correction is applied smoothly and converted to arm joint targets through IK, while object height remains fixed.

3.3 Visual Observations and Real-World Deployment

Visual observations are rendered from accepted executions rather than planned references. We replay the recorded robot and object states to obtain synchronized egocentric views. The robot and assets are rendered separately from the Gaussian background in IsaacLab, and mask-guided z-buffer compositing produces the final image at the matched camera pose.

Each rendered image I_t and the corresponding proprioceptive state ρ_t form an observation $o_t = (I_t, \rho_t)$, paired with the task instruction ℓ and recorded action $a_t = (z_t, g_t)$. We train a visuomotor policy to predict H-step action chunks, $(\hat{z}_{t:t+H-1}, \hat{g}_{t:t+H-1}) = \pi_\theta(o_t, \ell)$. At deployment, predicted latents drive the SONIC decoder, while hand commands are executed separately through the same interface used during data generation. System and deployment details appear in Appendix B.1.

4 Experiments

In this section, we aim to answer three questions through extensive real-robot experiments:

- **Q1: Data Generation Quality and Efficiency.** (Sec. 4.2). How efficiently does **Zero-HLM** generate successful demonstrations and how does each key component affect data quality?
- **Q2: Direct Sim-to-Real Transfer and Generalization.** (Sec. 4.3). Can simulation data alone support direct sim-to-real transfer, and how do the resulting policies compare with those trained on real demonstrations? How do they scale and generalize across real-world configurations?
- **Q3: Understanding Sim-to-Real Transfer.** (Sec. 4.4). Can simulation reflect real-world performance? What gaps remain in actions and observations? Does mixing real and simulation data improve learning?

4.1 Experimental Setup

Hardware Setup. We use a Unitree G1 equipped with a head-mounted RealSense D455 camera and two Inspire Robots RH56E2 hands, with the same robot configuration in simulation. Real-world demonstrations are collected through PICO whole-body teleoperation with the SONIC controller (Luo et al., 2026). Simulation data generation runs on eight RTX 4090 GPUs, and all policies are trained on eight A100 GPUs. During deployment, we run the trained policy with SONIC controller on a single RTX 4090. Appendix B.1 provides additional system setup details.

Tasks. We evaluate loco-manipulation tasks covering diverse whole-body skills: **Task A**, *Mobile Pick-and-Place*; **Task B**, *Low-Reach Hat Placement*; and **Task C**, *Leg-Assisted Basket Transport*. Task sequences are illustrated in Fig. 5, and task procedures are described in Appendix B.1.

Data Settings and Evaluation Protocol. To evaluate whether simulation-generated data extend policy generalization beyond the configurations covered by real-world demonstrations, we define Base and Expanded settings. Base matches the configurations used for real-world data collection, whereas Expanded introduces additional robot-start, object-location, and object–goal variations without corresponding real-world demonstrations as shown in Fig. 5(b). For each task, we collect 100 real-world Base demonstrations, denoted Real-Only(100), and 15 simulation Base demonstrations as the source pool for data generation. We introduce GPT-6 Astra as an agentic simulation-generation baseline; given the same source demonstrations and full simulator state, it generate 100 demonstrations for Tasks A and B, denoted Sim-Astra(100). For **Zero-HLM**, we generate 800 Base demonstrations per task, denoted Sim-Ours, and 800 demonstrations per task over Expanded configurations for Tasks A and B, denoted Sim-Ours*. Together with 800 additional object–goal-shift demonstrations, this results in 4,800 generated demonstrations in total.

We fine-tune pretrained GR00T-N1.7 (NVIDIA et al., 2025) under each data setting and execute the resulting policies with SONIC. We evaluate each policy over 20 real-world trials per task in Base and five trials per configuration in Expanded, reporting stage-wise and full-task success rates.

Table 2: **Collection time.** Hours for 50 successful demonstrations; setup excluded for Real.

Task	Real	**Sim-Ours**	Sim-Astra
Task A	1.35	**0.56**	5.00
Task B	1.02	**0.63**	5.83
Task C	1.68	**1.19**	$>$ 8.00, fail

Table 3: **Geometry accuracy.** Symmetric CD against LiDAR scanned ground truth.

Method	CD (cm) ↓
GS-Mesh	5.894
Ours (w/o GS)	11.689
Ours	5.517

Table 4: **Mesh vs. GS Rendering.** Real-world success on Task A.

Stage	Mesh	GS
Walk	18/20	**20/20**
Pick	3/20	**10/20**
Place	3/20	**7/20**

4.2 Data Generation Quality and Efficiency

Generation yield and collection time. To answer **Q1**, we compare the time required to collect 50 successful demonstrations across different data collection methods in Tab. 2. Sim-Ours achieves $2.4\times$, $1.6\times$, and $1.4\times$ speedups over real-world teleoperation on Tasks A, B, and C, respectively, averaging $1.8\times$ across tasks. In contrast, Sim-Astra requires substantially more time on Tasks A and B and produces no successful Task C demonstrations after more than eight hours. Detailed generation outcomes and failure breakdowns are provided in Appendix B.4.1.

Key components for data quality and efficiency. We first assess Gaussian guidance in agentic reconstruction. As shown in Tab. 3, it reduces the symmetric Chamfer distance to the LiDAR-scanned ground truth from 11.689 to 5.517 cm, improving geometric alignment with the real scene. Evaluation details and qualitative comparisons appear in Appendix B.2. We then evaluate key component choices of our motion–environment co-adaptation strategy on Task A using configurations sampled from Base, targeting 100 successful trajectories per variant (Tab. 5). Removing robot $SE(2)$ alignment causes the largest drop in generation success, from 61% to 16%, highlighting the importance of execution-conditioned correction for preserving task feasibility under tracking deviations. To assess Gaussian splatting as visual supervision, we compare policies trained on Gaussian and mesh renderings (Tab. 4). Gaussian rendering improves full-task success from 3/20 to 7/20. Appendix B.4.2 provides qualitative comparisons (Fig. 15).

Table 5: **Data-generation ablations.** Task A configurations sampled from Base; 100 successful trajectories per variant.

Variant	Success (%)
Full	**61.0**
w/o Velocity-Adaptive	55.0
w/o Z Compensation	51.0
w/o Object $SE(2)$	49.0
w/o Robot $SE(2)$	16.0

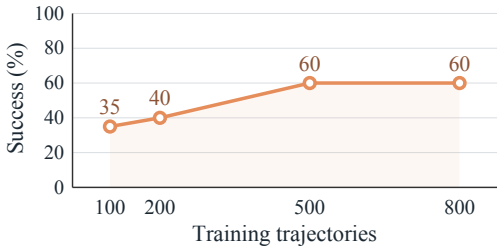


Figure 3: **Sim-Data Scaling on Task A in Real-World Evaluation.**

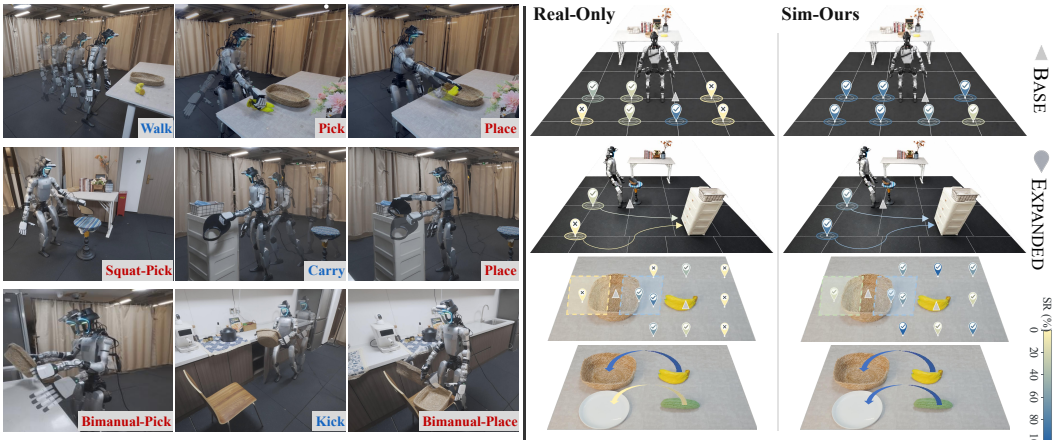
4.3 Direct Sim-to-Real Transfer and Generalization

Direct Sim-to-Real Transfer. To answer **Q2**, we compare GR00T N1.7 fine-tuned on Sim-Ours(100) and Sim-Ours(800) with Real-Only(100) and Sim-Astra(100) under the Base configuration. We evaluate all three tasks, except that Sim-Astra has no usable Task C data. As shown in Tab. 6, Sim-Ours(100) achieves mean full-task success comparable to Real-Only(100) (43.3% vs. 46.7%), suggesting that our method provides training supervision comparable to real-world demonstrations. Scaling to 800 simulation demonstrations per task raises mean success across three tasks from 46.7% with Real-Only(100) 66.7%. Fig. 3 further details how real-world performance scales with the amount of our generated data on Task A.

Generalization across Configurations. To assess whether expanded simulation coverage transfers to real-world configurations beyond those covered by real demonstrations, we compare Sim-Ours(800)* with Real-Only(100). As shown in Tab. 7, Sim-Ours(800)* consistently outperforms Real-Only(100) across all five Expanded settings, increasing pooled success from 23% to 69%. The gains span robot-start shifts, picking and placement locations, and the alternative object–goal pair, demonstrating that simulation coverage transfers robustly across diverse configuration changes. Notably, success on the object–goal shift increases from 20% to 100%. These results show that

Table 6: **Direct Sim-to-Real Transfer under BASE.** We report stage-wise success, with locomotion stage in blue and manipulation stage in red. Place and Bi-Place report full-task success. Avg. averages full-task success over Tasks A–C, except Sim-Astra (A–B only; Task C unavailable).

Training Data	Task A			Task B			Task C			Avg.
	Walk	Pick	Place	Sq-Pick	Carry	Place	Bi-Pick	Kick	Bi-Place	
Real-Only(100)	**20/20**	10/20	8/20	16/20	16/20	11/20	15/20	13/20	9/20	46.7
Sim-Astra(100)	17/20	4/20	4/20	5/20	0/20	0/20	—	—	—	10.0
Sim-Ours(100)	**20/20**	10/20	7/20	17/20	16/20	10/20	16/20	11/20	9/20	43.3
Sim-Ours(800)	**20/20**	**15/20**	**12/20**	**19/20**	**19/20**	**15/20**	**20/20**	**19/20**	**13/20**	**66.7**



(a) Task A/B/C Visualization (b) Zero-shot Generalization Performance

Figure 5: **Tasks and Generalization Evaluation.** (a) Stage-wise sequences for Tasks A–C. (b) From top to bottom: robot starting positions for Tasks A and B, banana picking and basket placement locations, and an alternative object–goal pair. Triangles and pins mark BASE and EXPANDED configurations, respectively. Colors indicate successes over five real-world trials per configuration, from yellow (0/5) to blue (5/5).

Table 7: **Zero-shot transfer across configurations.** Entries are success rates (%) excluding Base. Avg. pools successful trials across all five settings ($n = 100$).

Training Data	Robot Start		Tabletop Location		Object–Goal	Avg.
	Task A	Task B	Pick	Place		
Real-Only(100)	22.9	10.0	25.0	30.0	20.0	23.0
Sim-Ours(800)*	**71.4**	**70.0**	**67.5**	**50.0**	**100.0**	**69.0**



Figure 4: **Sim-Data Scaling in Our Benchmark.**

scalable simulation data can substantially broaden real-world policy coverage without requiring additional physical demonstrations. Detailed per-configuration results are provided in Appendix B.5.

4.4 UNDERSTANDING SIM-TO-REAL TRANSFER

Can simulation reflect real-world performance? We organize the three task environments into a **humanoid loco-manipulation benchmark** with shared execution and evaluation protocols. To assess sim–real consistency, we evaluate the Task A policies trained with Real-Only(100) and Sim-Ours(100) in both simulation and the real world under BASE. As shown in Tab. 9, both policies achieve comparable success rates across simulation and the real world, suggesting that the benchmark provides a reasonable approximation of real-world performance under matched configurations. We further evaluate data scaling and different policy families in the benchmark. Fig. 4 shows consistent gains with increasing simulation data, while Tab. 8 compares four policies under the same evaluation protocol. Additional details are provided in Appendix C.

Table 8: **Policy Evaluation in the Benchmark.** Task success rates (%) over 100 evaluation trials for every model–task pair.

Model	Family	Task A	Task B	Task C	Avg.
Diffusion Policy	Task-specific IL	3	4	1	2.7
GR00T-N1.7	Pretrained VLA	**66**	74	**69**	**69.7**
$\pi_{0.5}$	Pretrained VLA	60	60	57	59.0
DiT4DiT	Video Action Model	58	**80**	54	64.0

Table 9: **Sim–Real Consistency on Task A.**

Test Domain	Data	SR
Real	Real(100)	8/20
	Sim(100)	7/20
Sim	Real(100)	31/100
	Sim(100)	39/100

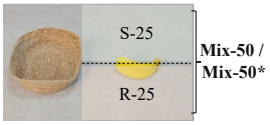


Figure 6: **Spatial Coverage of Data.** S-25 covers the upper region of the table, while R-25 covers the lower region. "Mix" combines both regions.

Table 10: **Different Data Compositions.** "R" denotes real-world data, and "S" denotes simulation data.

Data	Vision	Action	SR_{Up}	SR_{Low}
R-25	R	R	0/5	2/5
S-25	S	S	4/5	0/5
Mix-50	S+R	S+R	2/5	4/5
Mix-50*	R	S+R	**5/5**	**5/5**

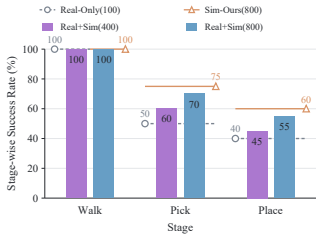


Figure 7: **Co-training Performance on Task A.**

What gaps remain in actions and observations? Task-level agreement does not imply that action labels and visual observations transfer equally well. To isolate these factors, we construct four training sets for Task A. As illustrated in Fig. 6, R-25 contains 25 real demonstrations from the lower table region, while S-25 contains 25 simulation demonstrations from the upper region. Mix-50 combines both sets. For Mix-50*, we replay the S-25 trajectories on the physical robot, replace their rendered observations with recorded real images while retaining the original simulation action labels, and combine them with R-25. As shown in Tab. 10, R-25 and S-25 fail to generalize to the unseen region, while Mix-50 achieves 2/5 and 4/5 success in the upper and lower regions, respectively. Replacing simulated observations with real images raises both to 5/5 in Mix-50*. These results indicate that the generated action labels transfer effectively to the real robot, while visual observations remain a major source of the sim–real gap.

Does mixing real and simulation data improve learning? We next study sim–real co-training under the BASE configuration, where real and simulation data follow matched spatial and trajectory distributions. Starting from 100 real demonstrations, we add either 400 or 800 simulation trajectories and evaluate each setting over 20 real-world trials per task. As shown in Fig. 7, increasing the amount of simulation data improves Task A full-task success from 45% to 55%, but both co-training settings remain below Sim-Ours(800). Results for Tasks B and C show the same trend and are reported in Appendix B.6. Thus, simulation data benefits co-training, yet combining real and simulation data does not necessarily outperform simulation-only training. Together with Tab. 10, these results suggest that the remaining visual-domain gap may limit effective sim–real co-training.

5 CONCLUSIONS

We presented **Zero-HLM**, a simulation-based data-generation framework that provides a practical path toward generalizable real-world humanoid loco-manipulation. Across three long-horizon tasks, policies trained solely on our simulation data transfer zero-shot to the real world and achieve higher mean full-task success than their real-data-trained counterparts. Expanding simulation coverage further improves transfer across spatial configurations without additional real demonstrations, showing that scalable synthetic coverage can translate into broader real-world policy generalization. Our transfer analysis supports the effectiveness of generated action labels while revealing a remaining visual-domain gap. The co-training results show that mixing real and simulation data does not necessarily outperform simulation-only training, underscoring the importance of visual alignment for effective co-training. Future work will extend the framework to additional unified whole-body controllers and examine how tracking accuracy and motion capability affect demonstration quality. Another important direction is to extend the benchmark to richer interaction skills and multiple reconstructed scenes, enabling a more systematic study of cross-scene policy generalization.

AI USE STATEMENT

We use multimodal agents for scene reconstruction and simulation configuration, and GPT-6 Astra for the Sim-Astra data-generation baseline (Section 3.1; Appendices A.1 and B.3). Generated scenes and trajectories undergo geometric comparisons, simulated interaction checks, and rollout validation. AI tools also assist with literature search, manuscript drafting, and language refinement. The authors review AI-assisted text against the implemented method and experimental records and take full responsibility for the final manuscript and research artifacts.

ETHICS STATEMENT

This work aims to reduce reliance on physical demonstrations through simulation-based data generation. Real-world humanoid deployment requires human supervision and task-specific safety precautions to address potential falls, unintended contact, and dropped objects. Scene captures and reconstructed assets should be handled and shared with appropriate privacy protections and permissions, while respecting third-party licenses.

REPRODUCIBILITY STATEMENT

Our reconstruction and data-generation procedures are described in Section 3 and Appendix A, including source annotation and reference-motion construction. Configuration sampling is detailed in Appendix B.1. Appendix B documents the robot platform, deployment interface, and real-world evaluation protocols. Appendix C provides the simulation benchmark setup, model training configurations, and evaluation criteria.

REFERENCES

Shuanghao Bai, Meng Li, Xinyuan Lv, Jiawei Wang, Xinhua Wang, Fei Liao, Chengkai Hou, Langzhe Gu, Wanqi Zhou, Kun Wu, Ziluo Ding, Zhiyuan Xu, Lei Sun, Shanghang Zhang, Zhengping Che, Jian Tang, and Badong Chen. HEX: Humanoid-aligned experts for cross-embodiment whole-body manipulation, 2026. URL https://arxiv.org/abs/2604.07993. 3

Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=3RSLW9YSgk. 2

Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. HOMIE: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 3

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164. 2

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.07339. 18

Guanxiong Chen, Qianjun Xia, Jiawei Peng, Heng Zhang, Pengyu Jing, Bole Ma, Justin Qian, Yixian Cheng, Ziyi Jiao, Bingyang Zhou, Yiduo Qu, Luoxin Ye, Kaifeng Zhang, Kunyi Wang, Weijia Zeng, Yunuo Chen, Pengzhi Yang, Ziqiu Zeng, Siyuan Luo, Huamin Wang, Chao Liu, Alan Yuille, Fan Shi, Changxi Zheng, Yunzhu Li, Chenfanfu Jiang, and Peter Yichen Chen. Agentic Real2Sim: Physics-based world modeling with vision-language agents, 2026. URL https://arxiv.org/abs/2607.19190. 3

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. 25

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024. 2

Yuhui Fu, Feiyang Xie, Chaoyi Xu, Jing Xiong, Haoqi Yuan, and Zongqing Lu. DemoHLM: From one demonstration to generalizable humanoid loco-manipulation. *IEEE Robotics and Automation Letters*, 2026. 2, 3

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. HumanPlus: Humanoid shadowing and imitation from humans. In *Conference on Robot Learning (CoRL)*, 2024. URL https://humanoid-ai.github.io/. 2

Antoine Guédon, Tomoki Ichikawa, Kohei Yamashita, and Ko Nishino. MAtCha Gaussians: Atlas of charts for high-quality geometry and photorealism from sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6001–6011, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Guedon_MAtCha_Gaussians_Atlas_of_Charts_for_High-Quality_Geometry_and_Photorealism_CVPR_2025_paper.html. 4

Xiaoshen Han, Junqiu Yu, Minghuan Liu, Yilun Chen, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^{3}Sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2502.08645. 2, 3

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1516–1540. PMLR, 2025. URL https://proceedings.mlr.press/v270/he25b.html. 3

Tairan He, Zi Wang, Haoru Xue, Qingwei Ben, Zhengyi Luo, Wenli Xiao, Ye Yuan, Xingye Da, Fernando Castañeda, Shankar Sastry, Changliu Liu, Guanya Shi, Linxi Fan, and Yuke Zhu. VIRAL: Visual sim-to-real at scale for humanoid loco-manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13430–13441, June 2026. 3

Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. GSWorld: Closed-loop photo-realistic simulation suite for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2510.20813. 2

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, and Hongyang Li. WholeBodyVLA: Towards unified latent VLA for whole-body loco-manipulation control. *arXiv preprint arXiv:2512.11047*, 2025. URL https://arxiv.org/abs/2512.11047. 2

Hojune Kim, Timothy Chen, Jiankai Sun, Lars W. Osterberg, Qianzhong Chen, Ke Wang, and Mac Schwager. LEGS: Fine-tuning teleop-free VLAs for humanoid loco-manipulation in an embodied gaussian splatting world. *arXiv preprint arXiv:2606.01458*, 2026. 2, 3

Hyesung Lee, Youngseon Lee, Kyutae Lee, Dongjun Lee, and Yongseok Lee. RORA: Realistic object reconstruction with articulation, 2026. URL https://arxiv.org/abs/2608.04842. 2

Bowei Li, Yuner Zhang, and Changliu Liu. ARSTAG: An agentic real2sim2real system for task-specific robot data generation, 2026a. URL https://arxiv.org/abs/2609.24563. 3

Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. RoboGSim: A Real2Sim2Real robotic gaussian splatting simulator, 2024. URL https://arxiv.org/abs/2411.11839. 2

Yixuan Li, Le Ma, Yutang Lin, Yushi Du, Mengya Liu, Kaizhe Hu, Jieming Cui, Yixin Zhu, Wei Liang, Baoxiong Jia, and Siyuan Huang. OmniClone: Engineering a robust, all-rounder whole-body humanoid teleoperation system. *arXiv preprint arXiv:2603.14327*, 2026b. 3

Zhe Li, Zhenzhe Zhang, Yangyang Wei, Wenjie Zhang, Xichen Yuan, Peiyuan Zhi, Gen Li, Xinying Guo, Fengjie Gao, Jianfei Yang, and Shanghang Zhang. ω-0: A latent predictive world action model for concurrent humanoid loco-manipulation. *arXiv preprint arXiv:2608.06375*, 2026c. URL https://arxiv.org/abs/2608.06375. 2

Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C. Karen Liu. BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117):eadx8924, 2026. doi: 10.1126/scirobotics.adx8924. URL https://doi.org/10.1126/scirobotics.adx8924. 2

Kevin Lin, Ajay Mandlekar, Caelan Reed Garrett, Nikita Chernyadev, Yu Fang, Runyu Ding, Yuqi Xie, Justin Tran, Linxi Fan, and Yuke Zhu. HumanoidMimicGen: Data generation for loco-manipulation via whole-body planning. *arXiv preprint arXiv:2605.27724*, 2026. 2, 3

Chenhao Lu, Xuxin Cheng, Jialong Li, Shiqi Yang, Mazeyu Ji, Chengjing Yuan, Ge Yang, Sha Yi, and Xiaolong Wang. Mobile-TeleVision: Predictive motion priors for humanoid whole-body control. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2412.07773. 3

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026. 2, 3, 4, 6

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026. 3, 25

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1820–1864. PMLR, 2023. 3

Ruiqian Nai, Boyuan Zheng, Junming Zhao, Haodong Zhu, Sicong Dai, Zunhao Chen, Yihang Hu, Yingdong Hu, Tong Zhang, Chuan Wen, and Yang Gao. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations. *arXiv preprint arXiv:2602.06643*, 2026. 3

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 2, 3, 6, 25

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. 3, 25

Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. SplatSim: Zero-shot sim2real transfer of RGB manipulation policies using gaussian splatting. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025. URL https://arxiv.org/abs/2409.10161. 2, 3

Omar Rayyan, Zhi Li, Max Argus, Yuxin Jiang, Chang Yu, Chenfanfu Jiang, and Yuchen Cui. FetchMan: Learning visual humanoid loco-manipulation policies from simulated experiences. *arXiv preprint arXiv:2608.17027*, 2026. 3

Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2023. URL https://arxiv.org/abs/2309.01952. 2

Modi Shi, Shijia Peng, Jin Chen, Haoran Jiang, Tianyu Li, Di Huang, Ping Luo, Hongyang Li, and Li Chen. EgoHumanoid: Unlocking in-the-wild loco-manipulation with robot-free egocentric demonstration. *arXiv preprint arXiv:2602.10106*, 2026. 2, 3

Yen-Jen Wang, Jiaman Li, Sirui Chen, Takara E. Truong, Pei Xu, Pieter Abbeel, Rocky Duan, Koushil Sreenath, Angjoo Kanazawa, Carmelo Sferrazza, Guanya Shi, and C. Karen Liu. VLK: Learning humanoid loco-manipulation from synthetic interactions in reconstructed scenes. *arXiv preprint arXiv:2606.30645*, 2026. 3

Bingqian Wu, Zicheng Xu, Xianghui Fan, Dayu Li, and Xiangru Huang. Teleopit: A full-embodiment humanoid teleoperation system. *arXiv preprint arXiv:2608.01834*, 2026a. URL https://arxiv.org/abs/2608.01834. 2

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, Yibin Liu, Shuai Yang, Tianxiang Zhou, Kejia Zhang, Lei Zhou, Cheng Su, Nan Xue, Bin Tan, Han Zhang, Youchao Zhang, Fei Liao, Xing Zhu, Yujun Shen, and Kecheng Zheng. From foundation to application: Improving VLA models in practice, 2026b. URL https://arxiv.org/abs/2607.06403. 3

Tianyi Xiang, Xupeng Xie, Jiahang Cao, Andrew F. Luo, Haoang Li, and Jun Ma. Function-preserving data generation for zero-shot real-to-sim-to-real manipulation, 2026. URL https://arxiv.org/abs/2609.18293. 2

Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. DemoGen: Synthetic demonstration generation for data-efficient visuomotor policy learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. 3

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C. Karen Liu, Rocky Duan, and Guanya Shi. OmniRetarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2509.26633. 2

Sizhe Yang, Wenye Yu, Jia Zeng, Jun Lv, Kerui Ren, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Novel demonstration generation with gaussian splatting enables robust one-shot manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2504.13175. 2, 3

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026. 3

Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Song-Chun Zhu, et al. Metascenes: Towards automated replica creation for real-world 3d scans. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1667–1679. IEEE, 2025a. 2

Justin Yu, Letian Fu, Huang Huang, Karim El-Refai, Rares Andrei Ambrus, Richard Cheng, Muhammad Zubair Irshad, and Ken Goldberg. Real2Render2Real: Scaling robot data without dynamics simulation or robot hardware. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 547–577. PMLR, 2025b. URL https://proceedings.mlr.press/v305/yu25a.html. 2

Zehao Yu, Jiakun Zheng, Weiji Xie, Jiyuan Shi, Chenyun Zhang, Chenjia Bai, and Xuelong Li. OASIS: From simulation data collection to real-world humanoid loco-manipulation. *arXiv preprint arXiv:2606.08548*, 2026. 2, 3

Yanjie Ze, Zixuan Chen, Joao Pedro Araujo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and Karen Liu. TWIST: Teleoperated whole-body imitation system. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2143–2154. PMLR, 2025a. URL https://proceedings.mlr.press/v305/ze25a.html. 3

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, 2025b. URL https://arxiv.org/abs/2511.02832. 3

Juntong Zhang, Chun Gu, and Li Zhang. X-WBC: A cross-embodiment foundation model for humanoid whole-body control, 2026. URL https://arxiv.org/abs/2609.15213. 3

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation, 2025. URL https://arxiv.org/abs/2510.08807. 3

Rui Zhong, Yizhe Sun, Junjie Wen, Jinming Li, Chuang Cheng, Wei Dai, Zhiwen Zeng, Huimin Lu, Yichen Zhu, and Yi Xu. HumanoidExo: Scalable whole-body humanoid manipulation via wearable exoskeleton. *arXiv preprint arXiv:2510.03022*, 2025. URL https://arxiv.org/abs/2510.03022. 2

Appendix Contents

- **A. Method Implementation Details**
  - A.1 Agentic Reconstruction: Implementation Details
  - A.2 Source Annotation and Selection
  - A.3 Reference-Motion Construction and Transitions
- **B. Experimental Details**
  - B.1 System Setup, Tasks, and Evaluation
  - B.2 Agentic Reconstruction Evaluation
  - B.3 GPT-6 Astra Baseline
  - B.4 Data-Generation Evaluation Details
  - B.5 Generalization Evaluation Details
  - B.6 Additional Co-Training Results
- **C. Benchmark Details**
- **D. Additional Qualitative Results**

A Method Implementation Details

We provide implementation details that complement the method described in Sec. 3. Specifically, we detail the agentic reconstruction pipeline, source annotation and selection, and reference-motion construction and transitions.

A.1 Agentic Reconstruction: Implementation Details

This section provides implementation details for the reconstruction pipeline in Sec. 3.1. Fig. 8 illustrates the workflow from captured inputs to a physics-ready simulation environment, while Tab. 11 reports the reconstruction time and cost by stage.



Figure 8: **Agentic reconstruction and verification.** (a) Captured RGB views, estimated camera poses, SfM geometry, and Gaussian reconstruction supply the inputs. (b) Overview and close-up views illustrate mesh-scene refinement in Blender. (c) Cabinet opening and object-release tests check articulation and supporting contacts in IsaacLab.

Scene Reconstruction and Refinement. From smartphone RGB captures, we estimate camera poses, reconstruct a 2D Gaussian scene, and extract an initial mesh. Checkerboards establish metric scale and ground alignment. Using these calibrated inputs, the agent constructs a textured mesh scene in Blender. It then compares mesh and Gaussian renderings from matched camera views and iteratively refines scene geometry and object placement. For the scene shown in Fig. 8, this process uses 151 refinement iterations and 461 active-view rendering calls.

Interaction-Object Reconstruction. Interaction-critical objects are reconstructed separately from the background scene to preserve local geometry required for grasping, placement, and contact. Each object is captured independently and reconstructed as a standalone asset. The resulting asset is aligned to the scene scale and imported into the simulation with corresponding visual and collision geometry. Its pose is then instantiated according to the task configuration.

Physical Scene Construction. The reconstructed scene and interaction assets are imported into IsaacLab, where the agent configures collision geometry, articulated joints, and physical properties including mass, friction, and center of mass. Physical checks validate interaction-critical behavior: cabinet-opening tests verify articulation, while object-release tests verify gravity-driven motion and supporting contacts. Task objects are further calibrated in the assembled environment to ensure stable interaction under simulation.

Reconstruction Cost. Tab. 11 reports input-to-Blender scene construction and refinement, Blender-to-IsaacLab physical configuration and validation, and the combined reconstruction time and cost for all seven interaction assets.

Table 11: **Agentic reconstruction resource usage.** Time and allocated subscription cost by stage.

Stage	Time	Cost (USD)
Input → Blender	1h 47m	6.46
Blender → IsaacLab	1h 21m	5.08
Object reconstruction (7 assets)	4h 05m	11.28

A.2 Source Annotation and Selection

We use 15 successful source demonstrations per task to form the source pool $\mathcal{D}_0$. Each demonstration is replayed in simulation and divided into temporal segments corresponding to locomotion, manipulation, or leg-critical interaction.

Source Annotation. We manually specify task-specific event predicates and detect the corresponding segment boundaries from source replay. Grasp events are identified using hand closure, hand–object proximity, and object displacement, while release and placement events use task-specific success conditions. Only demonstrations with valid interaction events and successful physical replay are retained in the source pool.

Source Selection. For each newly sampled interaction configuration, we select a source demonstration whose interaction target is spatially close to the sampled target. Given a sampled target pose $T_o = (p_o, R_o)$, source i is ranked by

$$d_i = w_p \|p_o - p_o^{\mathrm{src},i}\|_2 + w_R d_R(R_o, R_o^{\mathrm{src},i}), \tag{5}$$

where d_R denotes the relative rotation angle and w_p, w_R weight translation and rotation. We randomly select one of the K_{src} nearest valid sources and use its interaction segment for the object-relative adaptation in Eq. (1).

A.3 Reference-Motion Construction and Transitions

This section provides the implementation details for constructing and connecting the reference-motion segments introduced in Sec. 3.2. We describe locomotion reference generation, manipulation retargeting, and the spatial and temporal transitions used to assemble a continuous whole-body reference motion.

Locomotion Reference Generation. Initial walking positions are sampled from navigable space, and a walking speed is sampled once per trajectory from $[0.5, 0.7]$ m/s. The SONIC planner uses a four-frame motion context. When an upper-body posture must be preserved, such as during object transport, the corresponding joint values are imposed on both the planner context and the generated reference motion, and the updated reference motion is used to construct the next planning context.

Manipulation Targets and Arm IK. The object-relative retargeting rule is given in Eq. (1). Source hand poses and source and target object poses are expressed in the world frame. For placement, the target pose corresponds to the receptacle placement target; when its orientation is unchanged, retargeting reduces to a positional shift while preserving the source hand orientation.

The baseline reference pose r_t^{base} provides the root pose and inactive joints from the selected source or preceding segment. The active arm joints are obtained by

$$\widetilde{q}_t^{\text{arm}} = \text{IK}_{\text{arm}}(\widetilde{T}_{e,t}; r_t^{\text{base}}), \tag{6}$$

and replace only the corresponding entries in the baseline joint vector. Thus, the root and inactive joints remain fixed during reference-motion construction, while their realized motion is determined by the subsequent whole-body rollout.

Segment Transitions and Timing. Each segment inherits the terminal reference pose of the preceding segment. For manipulation, the inherited hand position and retargeted trajectory are compared in the world frame and connected using the blending rule in Eq. (2). Initial hand perturbations used within source interaction segments, in contrast, are defined in the robot-local frame and incorporated through arm IK.

When retargeting increases the translational or rotational distance between adjacent segments, we extend the transition according to the characteristic linear and angular hand speeds of the source motion. The resulting timing is independent of the sampled walking speed. Annotated grasp and release events are shifted consistently with the retimed reference motion. After grasping, connecting motions first lift the object before moving toward the next target to reduce contact with the supporting surface.

B EXPERIMENTAL DETAILS

This section supplements the experiments in Sec. 4 with system setup, evaluation protocols, and detailed results.

B.1 SYSTEM SETUP, TASKS, AND EVALUATION

Fig. 9 summarizes the hardware and policy execution pipeline.

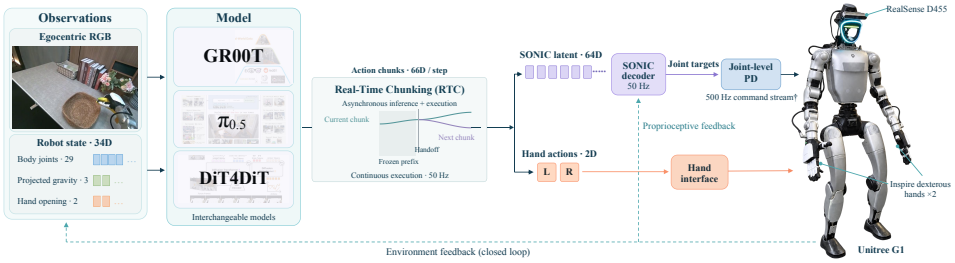


Figure 9: **System setup and deployment.** The policy maps egocentric RGB observations and robot state to SONIC motion latents and hand-opening commands. The frozen SONIC decoder produces body joint targets, while the two hands are controlled through a separate interface.

System Setup. We use a Unitree G1 with 29 body degrees of freedom and two Inspire Robots RH56E2 hands, each controlled through a single scalar opening command. A head-mounted RealSense D455 provides egocentric RGB observations with automatic exposure disabled to reduce visual variation. The policy takes RGB observations and a 34-dimensional robot state consisting of 29 body joint positions, three projected-gravity components, and two hand-opening values. It predicts 66-dimensional actions comprising a 64-dimensional SONIC motion latent and two hand-opening commands. We use real-time chunking (RTC) (Black et al., 2025) for asynchronous inference and execution at 50 Hz. The frozen SONIC decoder converts motion latents into body joint targets, while hand commands are executed through a separate interface. Both the policy and SONIC run on an RTX 4090, and simulation uses the same robot and hand configuration.

Camera calibration. We estimate the camera extrinsics from a predefined upper-body trajectory that keeps the arms and hands visible. For selected RGB frames, we annotate 2D arm and hand landmarks and associate them with link-local 3D points from the robot URDF. Forward kinematics maps these points to the camera mounting-link frame using the recorded joint configuration q_t. With fixed camera intrinsics K, we estimate the transform T from the mounting-link frame to the optical frame by minimizing robust reprojection error:

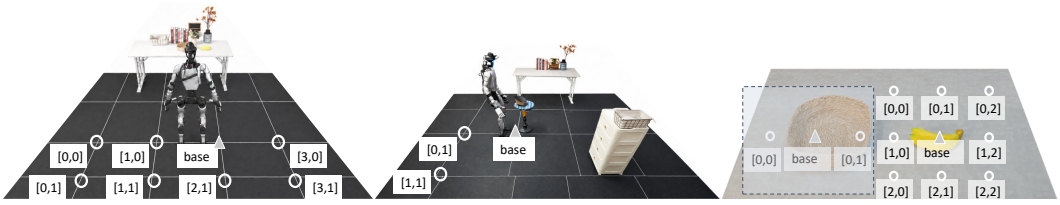
$$T^\star = \arg\min_{T\in \mathrm{SE}(3)} \sum_{(t,i)\in\mathcal{V}} \sum_{k=1}^{2} \rho\Big([\pi_K(T\mathbf{x}_i(q_t)) - \mathbf{u}_{ti}]_k^2 \Big), \tag{7}$$

where $\mathbf{u}_{ti}$ denotes the annotated image location, π_K the camera projection, and ρ a soft-ℓ_1 loss. Per-frame PnP estimates initialize the optimization and filter frames with large reprojection errors, after which a joint nonlinear fit estimates a fixed camera extrinsic transform.

Tasks. The following descriptions summarize the evaluated procedures; they are not verbatim policy prompts.

- **Task A: Mobile Pick-and-Place.** The robot walks to the table, picks up a banana, and places it in a basket. The reported stages are Walk, Pick, and Place, where Place reports full-task success.
- **Task B: Low-Reach Hat Placement.** The robot squats to pick up a hat, stands, carries it to a cabinet, and places it on the cabinet. The reported stages are Squat-Pick, Carry, and Place, where Place reports full-task success.
- **Task C: Leg-Assisted Basket Transport.** The robot carries a fruit basket bimanually, closes an obstructing cabinet door with its foot, and places the basket on a stool. The reported stages are Bi-Pick, Kick, and Bi-Place. Bi-Place reports full-task success.

Evaluation Configurations and Simulation Sampling. BASE denotes the reference configuration used for real-world demonstration collection, while EXPANDED introduces shifted robot starts and interaction locations for Tasks A and B; Task A additionally includes an alternative object–goal pair, whereas Task C uses only BASE. As illustrated in Fig. 10, each BASE or indexed EXPANDED location $[i, j]$ defines the center of a local sampling region. Robot starts are sampled within a 30×30 cm region around each center, while picking and placement locations use a 5×5 cm region. For simulation data generation, Tab. 12 summarizes the task-specific perturbations around these configurations. Notably, in Task B, the hat and cabinet are additionally perturbed by ± 5 cm and ± 40 cm per axis, respectively, with hat yaw varied by $\pm 20°$; the reference root height is lowered by 5–10 cm relative to the source pose. In Task C, the robot and basket are perturbed around the source configuration, while the stool pose is coupled to the robot endpoint. All planar offsets are expressed in the world XY frame; roll and pitch remain fixed unless otherwise specified.



(a) Robot-start sampling
BASE: 30×30 cm
EXPANDED $[i, j]$: 30×30 cm

(b) Robot-start sampling
BASE: 30×30 cm
EXPANDED $[i, j]$: 30×30 cm

(c) Picking / placement sampling
BASE: 5×5 cm
EXPANDED $[i, j]$: 5×5 cm

Figure 10: **Base and Expanded configurations.** (a,b) Robot starting locations for Tasks A and B. (c) Task A picking and placement locations. BASE marks the reference configuration; EXPANDED $[i, j]$ identifies shifted configurations. The annotations below each panel give the full simulation sampling widths around its centers. Task A start indices follow [column, row].

Table 12: **Simulation configuration sampling.** Offsets are relative to the corresponding BASE or EXPANDED $[i, j]$ center. Endpoint widths describe a sampling region, rather than an independent object perturbation.

Task	Entity / configuration	ΔX (cm)	ΔY (cm)	ΔYaw (°)
A	Robot start: BASE / EXPANDED $[i, j]$ (a)	± 15	± 15	± 15
	Picking location: BASE / EXPANDED $[i, j]$ (c)	± 2.5	± 2.5	± 15
	Placement location: BASE / EXPANDED $[i, j]$ (c)	± 2.5	± 2.5	± 20
	Robot interaction pose	± 10	± 15	± 15
B	Robot start: BASE / EXPANDED $[i, j]$ (b)	± 15	± 15	± 15
	Robot pickup pose	± 10	± 10	± 15
	Pickup root-height offset	$\Delta z \in [-10, -5]$ cm		
	Hat (BASE)	± 5	± 5	± 20
	Cabinet (BASE)	± 40	± 40	0
C	Robot interaction pose	± 15	$[-10, 5]$	± 15
	Basket (BASE)	$[-10, 5]$	$[0, 10]$	0
	Stool	Coupled to the robot endpoint		

Zero denotes no additional yaw perturbation. For Task C, the robot endpoint is sampled over a 51.644×39.66 cm region, and the stool preserves its source pose relative to that endpoint.

B.2 Agentic Reconstruction Evaluation

This section provides additional evaluation details for the agentic reconstruction described in Sec. 3.1. We compare three reconstruction variants against a ground-truth scan captured with a handheld Q9000 LiDAR scanner: GS-Mesh, our full agentic reconstruction, and a variant without Gaussian guidance. Reconstruction accuracy is measured using symmetric Chamfer distance (CD) within the table and cabinet interaction regions, where geometric errors directly affect contact and object placement. CD is computed as the average of the two directional nearest-neighbor distances between points sampled from the reconstructed mesh and the LiDAR scan.

As reported in Tab. 3, GS-Mesh achieves a CD of 5.894 cm, while our full reconstruction reduces it to 5.517 cm. Without Gaussian guidance, the error increases to 11.689 cm. This variant retains the captured images and estimated camera poses but removes Gaussian-derived mesh initialization and matched-view Gaussian rendering feedback. Qualitative comparisons in Fig. 11 further show that Gaussian guidance improves furniture dimensions, support-surface geometry, and local object placement.



Figure 11: **Agentic reconstruction comparison.** (a) LiDAR ground truth and interaction evaluation regions. (b) GS-Mesh with schematic collision examples. (c–d) Reconstructions with and without Gaussian guidance, with qualitative mesh–Gaussian detail comparisons.

B.3 GPT-6 Astra Baseline

This section provides implementation details for the Sim-Astra baseline introduced in Sec. 4.1. GPT-6 Astra receives the same reconstructed scenes, robot and object assets, task configurations, and source demonstrations as our method, together with access to the full simulator state. The agent autonomously constructs its data-generation procedure, either by synthesizing complete reference motions or by issuing incremental closed-loop commands. It may use the SONIC planner, inverse kinematics, and whole-body control interfaces. Fig. 12 shows the complete Task B prompt, including the available inputs, execution constraints, success criteria, and recording requirements.

Reported results use the stabilized pipeline after human-assisted debugging and task alignment. We collect 100 successful trajectories per task for policy training on Tasks A and B, while no successful Task C trajectory is obtained. Notably, simulator-side planning and most agent reasoning latency are excluded from each demonstration; only the executed robot trajectories are stored for policy training.

The generated trajectories frequently contain pauses and repeated attempts and often exceed one minute, compared with approximately 30 seconds for real-world demonstrations. Policies trained on these trajectories exhibit qualitatively similar hesitation during real-world deployment.

Task B: Data Generation Prompt

You are an autonomous agent for generating robot demonstration data. You are provided with all robot and scene assets, a physics simulation environment, the SONIC planner and whole-body control interfaces, and source demonstrations covering grasping and standing up, and object placement.

Goal: Generate 100 successful robot manipulation trajectories. Randomize the robot's initial pose, the hat's pose, and furniture positions within the specified bounds. Imitate the source demonstrations' action sequence, whole-body poses, and motion trajectories to perform the complete task: crouch to grasp the hat from a small table, stand up, carry it to a cabinet, place it on the cabinet surface, and release it. Adapt the motions to each randomized scene while preserving the behavioral characteristics of the demonstrations.

Generate whole-body reference motions autonomously. You may call the SONIC planner to plan or supplement reference trajectories, then execute them through the SONIC whole-body controller in physics simulation. Continuously monitor robot states, object states, and contacts, and adjust control during grasping, walking, and placement using closed-loop feedback. Analyze failures and retry until 100 successful trajectories have been collected. Do not achieve success by writing states frame by frame, teleporting objects, or rigidly attaching the hat to the hand.

A successful trajectory must satisfy the specified acceptance criteria: the robot remains upright without prohibited collisions, the hat is not dropped during transport, and the released hat comes to rest within the designated region of the cabinet surface. Save complete executed state trajectories, timestamps, scene configurations, and source-demonstration correspondences. Record the outcome, failure reason where applicable, and generation time for every attempt.

Figure 12: **Verbatim Task B prompt for the GPT-6 Astra baseline.** The prompt gives the agent the shared assets, source demonstrations, planning and control interfaces, execution restrictions, acceptance criteria, and recording requirements.

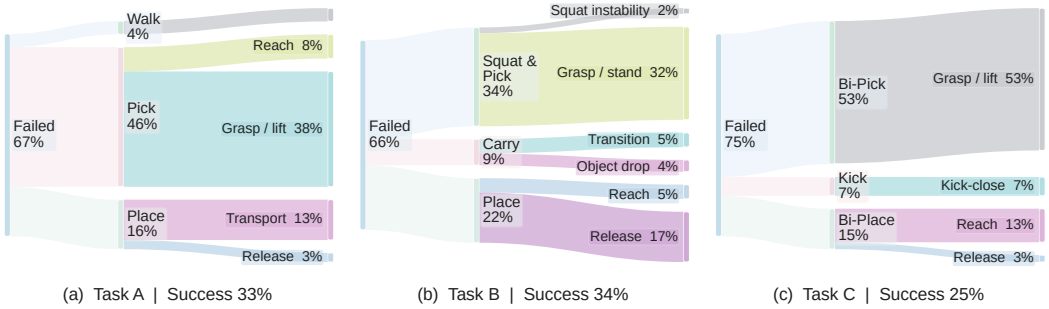
B.4 Data-Generation Evaluation Details

This section provides additional diagnostics for the data-generation quality and efficiency experiments in Sec. 4.2, including rollout failure attribution and qualitative rendering comparisons.

B.4.1 Generation Outcomes and Failure Assignment

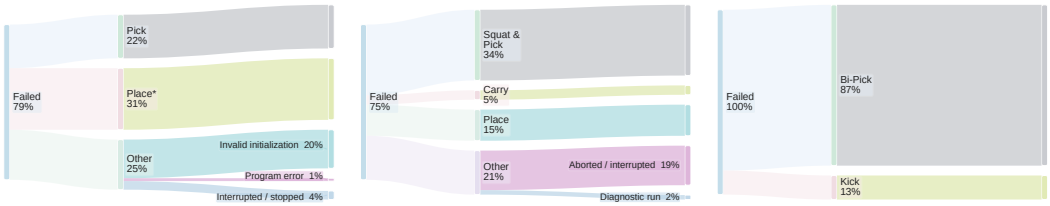
We analyze all recorded rollout attempts using the same task-success, stability, and collision criteria for Sim-Ours and Sim-Astra. Each failed rollout is assigned to its earliest failed task segment and one mutually exclusive failure category. Ribbon widths in Figs. 13 and 14 represent fractions of all attempts, with displayed percentages rounded.

Sim-Ours achieves generation success rates of 33%, 34%, and 25% on Tasks A, B, and C, respectively. Grasp-related failures dominate across the three tasks. Sim-Astra achieves 21.38% and 24.70% success on Tasks A and B, while producing no successful Task C trajectory. Its failures additionally include invalid initialization, program errors, and interrupted runs.



(a) Task A | Success 33% (b) Task B | Success 34% (c) Task C | Success 25%

Figure 13: **Sim-Ours generation outcomes and failure breakdown.** Ribbons partition rollout attempts by task segment and outcome. Generation success is 33%, 34%, and 25% for Tasks A, B, and C; ribbon percentages are rounded.



(a) Task A | Success 21.38% (b) Task B | Success 24.70% (c) Task C | Success 0.00%

Figure 14: **GPT-6 Astra generation outcomes after stabilization.** Success rates are 21.38%, 24.70%, and 0% for Tasks A–C. Task C failures occur at Bi-Pick (87%) and Kick (13%); its total attempt count is unavailable. *Place** groups placement, hand-withdrawal, and acceptance failures.

B.4.2 Rendering Comparison

Fig. 15 compares mesh and Gaussian-splatting observations for the same simulated Task A sequence, together with stage-matched observations from an independently recorded real-world episode. The simulation and real-world rows are not temporally synchronized; the columns indicate comparable task progress rather than pixel-aligned views. The comparison illustrates the visual differences between the two simulation renderers and the observations encountered during real-world deployment.

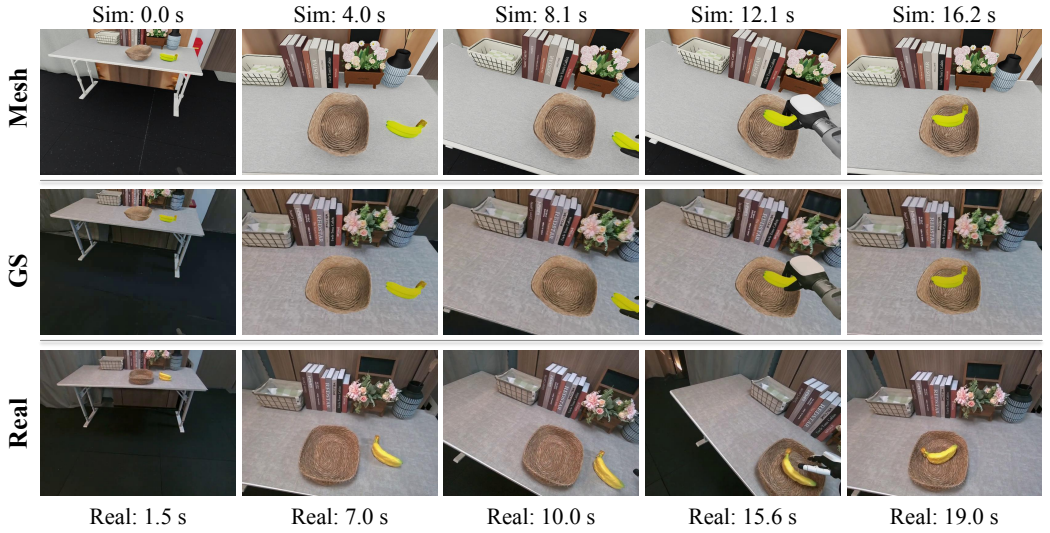


Figure 15: **Mesh, Gaussian-splatting, and real-world observations.** The top two rows show the same Task A simulation sequence rendered with meshes and Gaussian splatting. The bottom row shows stage-matched observations from an independent real-world episode; columns indicate comparable task progress rather than synchronized timestamps or pixel correspondences.

B.5 Generalization Evaluation Details

This section details the five settings summarized in Tab. 7 (Sec. 4.3), using the locations in Fig. 10. Each configuration has five real-world trials. Entries give successes out of five; *Changed* reports percentages excluding Base. We compare Real-Only(100) with Sim-Ours(800)*. The Task A robot-start and tabletop-location evaluations (picking and placement) use the same set of 800 simulation demonstrations. The object–goal evaluation uses a separately generated set of 800 demonstrations, with 400 for each of the two goal types.

- Table 13: **Task A: robot starts.**

Training Data	[0,0]	[1,0]	Base	[3,0]	[0,1]	[1,1]	[2,1]	[3,1]	Changed (%)
Real-Only(100)	2/5	1/5	4/5	0/5	0/5	1/5	4/5	0/5	22.9
Sim-Ours(800)*	4/5	4/5	5/5	4/5	4/5	4/5	3/5	2/5	**71.4**

Seven Expanded starting locations in Fig. 10(a).

- Table 14: **Task B: robot starts.**

Training Data	Base	[0,1]	[1,1]	Changed (%)
Real-Only(100)	2/5	1/5	0/5	10.0
Sim-Ours(800)*	4/5	3/5	4/5	**70.0**

Two Expanded starting locations in Fig. 10(b).

- Table 15: **Task A: picking locations.**

	[0,0]		[0,1]		[0,2]		[1,0]		Base		[1,2]		[2,0]		[2,1]		[2,2]		Changed (%)	
Training Data	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place	Pick	Place
Real-Only(100)	1/5	0/5	2/5	2/5	2/5	0/5	**5/5**	**4/5**	1/5	1/5	0/5	0/5	3/5	3/5	**3/5**	1/5	0/5	0/5	40.0	25.0
Sim-Ours(800)*	**3/5**	**3/5**	**5/5**	**5/5**	**3/5**	**3/5**	3/5	3/5	**2/5**	**2/5**	**2/5**	**2/5**	**5/5**	**5/5**	2/5	**2/5**	**4/5**	**4/5**	**67.5**	**67.5**

The banana varies across eight Expanded locations while the basket remains fixed (Fig. 10(c)).

- Table 16: **Task A: placement locations.**

	[0,0]		Base		[0,1]		Changed (%)	
Training Data	Pick	Place	Pick	Place	Pick	Place	Pick	Place
Real-Only(100)	**4/5**	0/5	**2/5**	**2/5**	3/5	**3/5**	**70.0**	30.0
Sim-Ours(800)*	2/5	**2/5**	**2/5**	**2/5**	**5/5**	**3/5**	**70.0**	**50.0**

The basket varies across two Expanded locations while the banana remains fixed (Fig. 10(c)).

- Table 17: **Task A: object–goal pair.**

Training Data	Banana to Basket (Base)	Bitter Melon to White Plate	Changed (%)
Real-Only(100)	2/5	1/5	20.0
Sim-Ours(800)*	5/5	5/5	100.0

The alternative pair replaces the banana and basket with bitter melon and a white plate.

Pooling the five Expanded settings gives 100 trials per method (35, 10, 40, 10, and 5 by setting). Real-Only(100) and Sim-Ours(800)* achieve 23/100 and 69/100 successes, matching Tab. 7.

B.6 Additional Co-Training Results

Tab. 18 provides the complete stage-wise co-training results for Tasks A–C, complementing the Task A results reported in the main paper. Both settings use 100 real demonstrations per task: Real+Sim(400) adds 400 simulation trajectories, while Real+Sim(800) adds 800. Each policy is evaluated over 20 real-world trials per task.

Table 18: **Detailed real–sim co-training results.** Entries are successful trials out of 20. Avg. is the mean full-task success rate across Tasks A–C. Both settings use 100 real demonstrations per task; the parenthetical number denotes the additional simulation trajectories.

Training Data	Real:Sim	Task A			Task B			Task C			Avg. (%)
		Walk	Pick	Place	Squat-Pick	Carry	Place	Bi-Pick	Kick	Bi-Place	
Real+Sim(400)	1:4	**20/20**	12/20	9/20	**15/20**	**15/20**	11/20	18/20	**13/20**	8/20	46.7
Real+Sim(800)	1:8	**20/20**	**14/20**	**11/20**	14/20	14/20	**12/20**	**20/20**	12/20	**9/20**	**53.3**

Increasing the amount of simulation data improves full-task success from 9/20 to 11/20 on Task A, 11/20 to 12/20 on Task B, and 8/20 to 9/20 on Task C. However, intermediate-stage performance does not improve uniformly, with small decreases in Task B Squat-Pick and Carry and Task C Kick.

C Benchmark Details

This section provides additional details for the humanoid loco-manipulation benchmark used in Sec. 4.4, including the shared simulation interface, model-specific training setups, evaluation protocol, and stage-wise results.

C.1 Simulation Setup and Shared Interface

The benchmark instantiates the three task environments in IsaacLab with a simulated Unitree G1. Each policy receives egocentric RGB observations at 640×480 resolution and a 34-dimensional robot state consisting of 29 body joint positions, three projected-gravity components, and two hand states. Language-conditioned models additionally receive a fixed task instruction. Privileged object poses and contact states are used only for evaluation and are not provided to the policy.

All models predict the same 66-dimensional action representation, consisting of a 64-dimensional SONIC motion latent and two hand commands. A frozen SONIC decoder converts the motion latent into body joint targets. Closed-loop rollouts run at 50 Hz for at most 2,000 control steps (40 s), with early termination upon task success or robot fall.

All models are evaluated using the same task-specific initialization distributions. Fig. 16 visualizes the task layouts and sampled configurations used for benchmark evaluation.

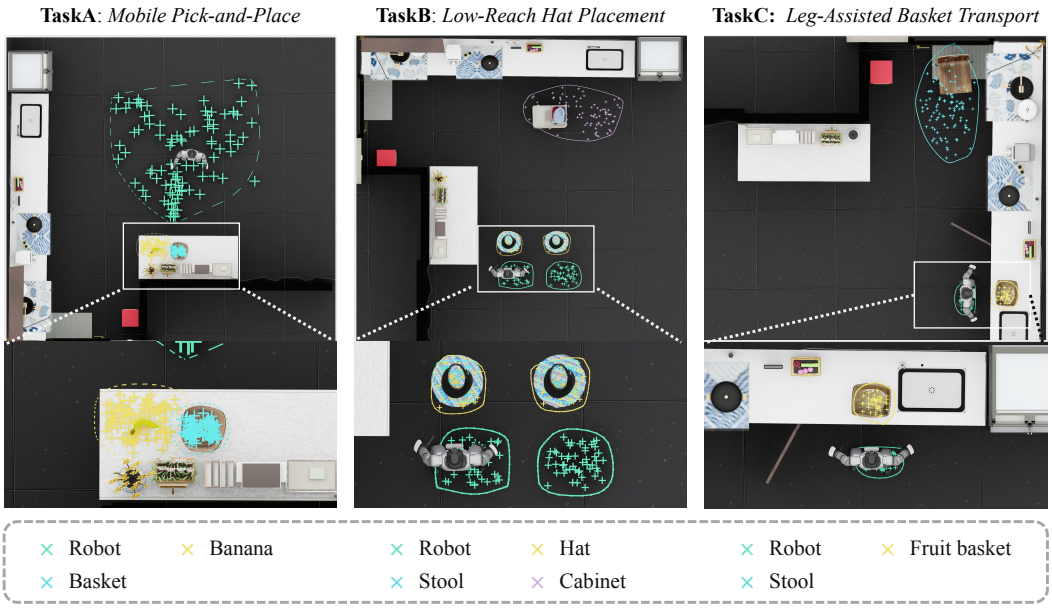


Figure 16: **Benchmark setup.** Top-down layouts and sampled robot and object configurations for Tasks A–C, with enlarged views of the interaction regions.

C.2 Model Training and Execution

All model comparisons use Sim-Ours(800), containing 800 generated trajectories per task. We retain each model's native training recipe while adapting its observation and action interfaces to the shared benchmark format. Image preprocessing, normalization, and padding follow the corresponding pretrained checkpoint.

Diffusion Policy. We use the LeRobot implementation of Diffusion Policy (Chi et al., 2023) for task-specific imitation learning from RGB observations, robot state, and action sequences. Images are resized to 224×224, and the robot state and actions are normalized during training. The policy executes 16 predicted actions before replanning.

$\pi_{0.5}$. We fine-tune the pretrained OpenPI $\pi_{0.5}$ checkpoint (Physical Intelligence et al., 2025) using RGB observations, robot state, and a fixed task instruction. The model predicts a 50-step, 66-dimensional action horizon.

GR00T N1.7. We adapt pretrained GR00T N1.7 (NVIDIA et al., 2025) to the G1–SONIC embodiment. Training updates the action-diffusion module, projection layers, and configured vision–language normalization layers while keeping the visual and language backbones frozen. The model predicts a 40-step action horizon.

DiT4DiT. DiT4DiT (Ma et al., 2026) combines a pretrained Cosmos video backbone with an action-diffusion transformer. We use its native video–action training recipe with a 40-step, 66-dimensional action horizon.

Prediction horizon and execution chunk are treated separately. Diffusion Policy executes 16 actions per update on all tasks. The VLA and video–action models execute 16 actions per update on Task A and 40 actions on Tasks B and C.

C.3 Evaluation Protocol and Metrics

Each model–task pair is evaluated over 100 simulation trials. We report full-task success together with independently measured stage-wise diagnostics. Intermediate stages are evaluated independently and therefore do not form cumulative success prefixes.

Task A. Walk indicates a stable rollout without a fall or execution error. Pick requires lifting the object by at least 3 cm. Full-task success additionally requires the object to be placed inside the basket target region while satisfying the configured height and final-speed conditions.

Task B. Squat-Pick requires lifting the hat by at least 10 cm. Carry requires the hat to enter the target region above the configured minimum height. Full-task success requires the hat to remain within the cabinet-local placement region with sufficiently low linear and angular velocity over the stability window.

Task C. Bi-Pick requires lifting the basket by at least 10 cm. Kick records successful activation of the cabinet-door condition. Bi-Place reports full-task success and requires the door condition to hold while the basket is supported and settled within the stool-seat region over the stability window.

The reported Avg. is the arithmetic mean of the full-task success rates across Tasks A–C; intermediate-stage diagnostics are not included.

Sim–Real Consistency. For the separate comparison in Tab. 9, we evaluate the same Task A policies in simulation and the real world under the Base configuration. Each policy is evaluated over 100 simulation trials and 20 real-world trials.

C.4 Stage-Wise Benchmark Results

Tab. 19 reports the complete stage-wise benchmark results. Intermediate columns provide diagnostic success rates for individual task stages, while Place for Tasks A and B and Bi-Place for Task C denote full-task success.

Table 19: **Stage-wise simulation benchmark.** Success rates (%) over 100 trials per model–task pair. Intermediate stages are evaluated independently. Place for Tasks A and B and Bi-Place for Task C denote full-task success; Avg. is the mean full-task success across the three tasks.

Model	Task A			Task B			Task C			Avg.
	Walk	Pick	Place	Squat-Pick	Carry	Place	Bi-Pick	Kick	Bi-Place	
Diffusion Policy	90	11	3	32	4	4	36	44	1	2.7
GR00T N1.7	100	85	66	92	76	74	93	97	69	69.7
$\pi_{0.5}$	100	82	60	79	60	60	93	98	57	59.0
DiT4DiT	97	76	58	100	84	80	97	99	54	64.0

D ADDITIONAL QUALITATIVE RESULTS

Fig. 17 visualizes generated simulation demonstrations for Tasks A–C. Each example pairs third-person views of a simulated rollout with the corresponding rendered egocentric observations used for policy training.

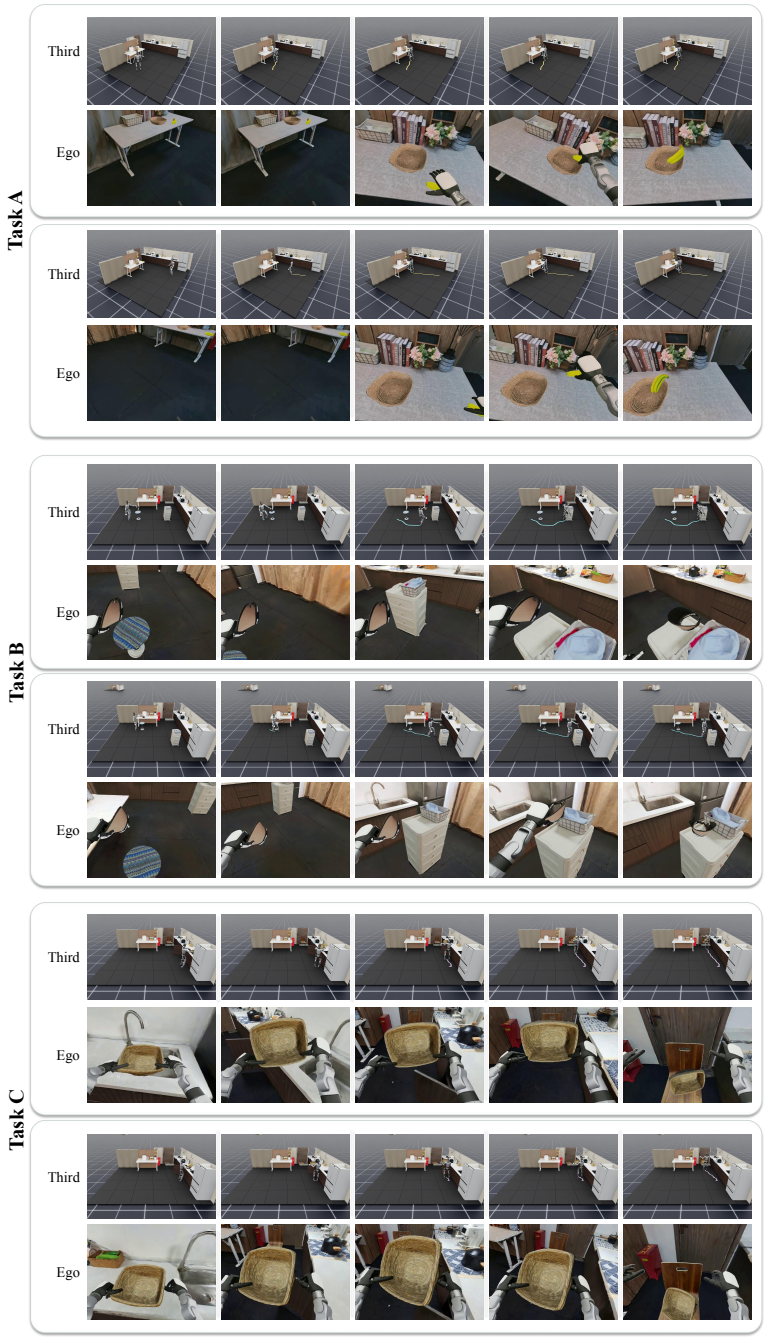


Figure 17: **Generated simulation demonstrations.** Third-person simulation views and corresponding egocentric renderings illustrate locomotion, manipulation, and leg-assisted interactions across Tasks A–C.