Zero-HLM: Generalizable Humanoid Loco-Manipulation without Real Robot Data

Dianyi Yang[1,2,*], Mengya Liu[2,*], Zhumei Wang[1,2], Yu Liu[2,3], Xiaochen Zhang[1,2], Yixuan Li[1], Le Ma[2], Yuhang Yang[2,3], Weiqi Huang[1,2], Wei Liang[1,†], Baoxiong Jia[2,†,‡], Siyuan Huang[2,4,†]

[1] Beijing Institute of Technology [3] Tsinghua University [4] Delta Intelligence
[2] State Key Laboratory of General Artificial Intelligence, BIGAI

[*] Equal contribution [†] Corresponding authors [‡] Project Lead

Project page: https://young-bit.github.io/zero-hlm/



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.

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. From 45 source demonstrations teleoperated in simulation, **Zero-HLM** generates **4,800** physically validated demonstrations, 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 a **20-percentage-point** 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.

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 [1–8]. However, learning generalizable behaviors requires diverse interaction data that remain costly and difficult to collect on a physical humanoid [9–12].

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 [13]. 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 [14–20]. However, extending this paradigm to scene-scale humanoid interaction is considerably more demanding, as meshes extracted from Gaussian reconstructions can contain contact artifacts, and articulated mechanisms require additional modeling effort [15, 21].

Beyond reconstructing such environments, generating humanoid interaction data at scale remains challenging. Recent methods [10, 22, 23] 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 [1, 2]. However, scaling data generation under unified tracking presents two challenges: (i) *Whole-body reference motion synthesis*. Adapting source demonstrations to varied spatial configurations requires coordinated changes throughout the body while preserving task-relative interaction [24]. (ii) *Execution mismatch*. Whole-body controllers must reconcile reference tracking with balance and contact constraints, so precise state-by-state replay cannot be assumed [25]. Even kinematically valid reference motions may produce tracking deviations in the robot base and hands, disrupting intended object interactions.

To address these challenges, we propose Zero-HLM, a real-to-sim-to-real framework that combines visually faithful, physics-ready real-to-sim reconstruction with scalable demonstration generation under unified whole-body control. First, inspired by recent MLLM-based agentic frameworks that produce faithful digital cousin reconstructions [26, 27], 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 our simulated data transfer directly to the real world and generalize across diverse scene configurations. Our contributions can be 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, physics-ready 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**× 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 a **20-percentage-point gain** over real-data training under matched configurations. Across five extended generalization settings, pooled real-world success increases from **23% to 69%** without additional real demonstrations.
- We establish a **humanoid loco-manipulation benchmark** comprising three long-horizon tasks and nine skill stages, enabling systematic comparisons across diverse policy families under unified training and execution protocols.

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 source 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 [23]	✗	✗	✓	✗	✓	✗	✓
VIRAL [28]	✗	✗	✗	✗	✗	✗	✓
FetchMan [29]	✗	✗	✗	✗	✗	✗	✓
HumanoidMimicGen [22]	✗	✗	✓	✗	✗	✗	✗
LEGS [10]	✗	✗	✗	✓	✗	✗	✓
OASIS [9]	✓	✗	✗	✓	✗	✗	✓
VLK [30]	✓	✗	✗	✓	✗	✗	✓
Ours	✓	✓	✓	✓	✓	✓	✓

2 Related Work

Humanoid Loco-Manipulation. Humanoid loco-manipulation depends on controllers that coordinate mobility with contact-rich interaction. Decoupled systems such as HOMIE [31] combine lower-body control with separately commanded arms and hands. Unified whole-body trackers, including SONIC [1] and OmniClone [32], instead track coordinated reference motions through a shared control interface, enabling human-motion-driven teleoperation and demonstration collection. Although teleoperation provides embodiment-aligned interaction data, expanding its task and scene coverage requires additional operator time and robot rollouts.

Robot-free interfaces such as HuMI [33] and egocentric approaches such as EgoHumanoid [7] reduce dependence on a physical robot during demonstration collection, but still rely on human-recorded interactions to cover new settings. In parallel, vision-language-action and world-action models have advanced generalizable robot learning [3, 34–36], but still depend on diverse interaction data. In this work, we address this data bottleneck through scalable whole-body data generation in simulation, enabling zero-shot real-world deployment with generalization to unseen scene configurations.

Real-to-Sim-to-Real for Robot Learning. Real-to-sim-to-real learning leverages reconstructed environments to generate simulation data for real-world policy transfer. Gaussian-based methods preserve captured appearance to reduce the visual sim-to-real gap [14, 15, 17], while recent agentic approaches facilitate physical scene construction through geometry modeling, physical parameter inference, and simulator configuration [37, 38]. Our approach brings these capabilities together by grounding agentic scene construction in Gaussian geometry and matched-view renderings, aligning physical interaction geometry with the reconstructed visual scene.

For demonstration generation, MimicGen adapts source demonstrations to new object configurations through simulation rollouts [39], whereas DemoGen performs demonstration and point-cloud transformations without simulation [40]. Extending data generation to humanoids, VIRAL and FetchMan rely on task-specific rewards [28, 29], while other approaches decouple locomotion from upper-body control [10, 22, 23]. VLK adopts unified whole-body control but focuses primarily on single-object transport [30]. In contrast, we adapt source interaction skills without specialized motion generators and correct tracking deviations through motion–environment co-adaptation, validating the resulting motions under the deployment controller. Tab. 1 compares these approaches.

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).

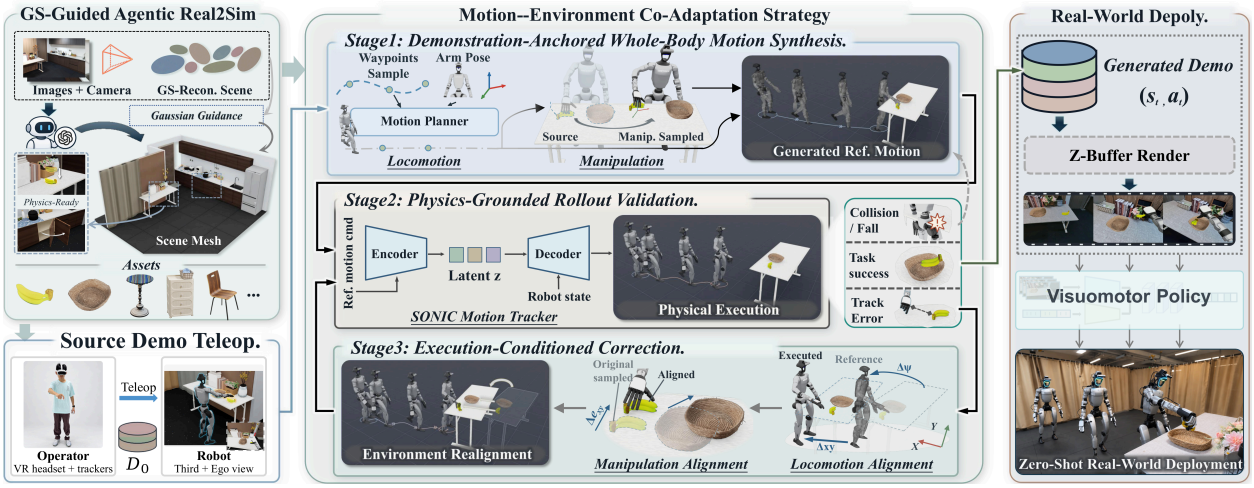


Figure 2. Overview of Zero-HLM. *Left:* Gaussian-guided agentic real-to-sim reconstruction aligns the physical scene with captured appearance and geometry. *Middle:* Motion–environment co-adaptation generates whole-body data from source demonstrations. *Right:* successful rollouts are paired with egocentric 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 [41], 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 from dedicated video captures to preserve the local geometry required for grasping, placement, and physical contact. Camera poses are estimated using COLMAP, while checkerboard calibration establishes metric scale for asset reconstruction. Each asset is then aligned and placed within the scene by the agent according to the task configuration. In IsaacLab, the agent further configures collision geometry, articulated joints, and physical parameters including mass, friction, and center of mass, then refines them through interaction checks. For example, cabinet opening tests articulated motion, while object release tests gravity-driven motion and collision.

3.2 Motion–Environment Co-Adaptation for Scalable Data Generation

In this section, we aim to generate physically executable whole-body demonstrations across diverse spatial configurations from a small set of source demonstrations. We first collect 15 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. For each sampled configuration, we construct a *whole-body reference motion* $\mathcal{R} = (r_1, \ldots, r_T)$ from $\mathcal{D}_0$, where each pose $r_t = (p_t, R_t, q_t)$ specifies the root position, orientation, and body joints. A simulated *rollout* executes this motion through SONIC [1], 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.

Source Annotation and Selection. We identify interaction segments using task-specific event predicates based on hand closure, object proximity, displacement, and task success conditions. Only demonstrations with valid interaction events and successful physical replay are retained. For each sampled target pose $T_o = (p_o, R_o)$, we rank source demonstrations by spatial proximity:

$$d_i = w_p \|p_o - p_o^{\mathrm{src},i}\|_2 + w_R d_R(R_o, R_o^{\mathrm{src},i}), \tag{1}$$

where d_R measures the relative rotation angle, and w_p, w_R weight positional and rotational differences. During data generation, we randomly select one of the K_{src} nearest valid demonstrations and adapt its interaction segments to the sampled target.

Stage 1: Demonstration-Anchored Whole-Body Motion Synthesis. For each generation attempt, we sample a task configuration around the source setup, varying the robot interaction pose and relevant object or placement-target poses. The initial walking position is sampled from navigable space. We then synthesize a continuous whole-body reference by replanning locomotion and adapting source interaction motions to the sampled configuration.

For locomotion segments, A* plans collision-free paths between the required robot poses. The SONIC motion planner converts these paths into whole-body reference motions using a walking speed sampled once per trajectory from $[0.5, 0.7]$ m/s and a four-frame motion context. For vertical adjustments, such as squatting, we use the planner to generate root-height transitions. When a sustained upper-body posture is required, such as carrying an object, we preserve the corresponding joint angles in both the planning context and generated reference. The resulting reference updates the motion context for subsequent planning. *For leg-critical interactions*, we retain the demonstrated lower-body motion and combine it with the preceding segment's terminal upper-body posture.

For manipulation segments, we adapt source hand trajectories to the sampled interaction targets while keeping the reference root pose fixed. Let $T_o^{\text{src}}, T_o \in SE(3)$ denote the source and sampled poses of an interaction object or placement target, and $T_{e,t}^{\text{src}} \in SE(3)$ the source hand pose, all expressed in the world frame. The adapted hand target is

$$\widetilde{T}_{e,t} = T_o (T_o^{\text{src}})^{-1} T_{e,t}^{\text{src}}. \tag{2}$$

This transformation preserves the demonstrated hand–target relation under the new configuration. For placement, T_o represents the receptacle target; when its orientation is unchanged, adaptation reduces to a positional shift while retaining the source hand orientation. Given a baseline reference pose r_t^{base} from the selected source or preceding segment, arm inverse kinematics (IK) updates the active arm joints:

$$\widetilde{q}_t^{\text{arm}} = \text{IK}_{\text{arm}}(\widetilde{T}_{e,t}; r_t^{\text{base}}). \tag{3}$$

The root pose and inactive joints remain unchanged during reference construction, while their executed states are determined by the subsequent whole-body rollout.

To ensure continuity between segments, we align their boundary poses and adjust transition timing. Each segment inherits the final reference pose of its predecessor. We connect this inherited pose to the retargeted manipulation trajectory using a gradually decaying hand-position offset:

$$p_{e,t}^{\text{blend}} = \widetilde{p}_{e,t} + \lambda_t \big(p_e^{\text{in}} - \widetilde{p}_{e,t_0}\big). \tag{4}$$

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 marks the transition start. As λ_t decreases smoothly from one to zero, the hand moves from its inherited position onto the adapted trajectory. Since retargeting may increase the displacement between segments, we adjust transition durations according to the source hand's linear and angular speeds, keeping grasp and release commands synchronized with the retimed motion. This timing adjustment is independent of the sampled walking speed. Initial hand perturbations are defined in the robot-local frame and incorporated through arm IK. After grasping, connecting motions lift the object before moving toward the next target to reduce contact with the supporting surface.

Stage 2: Physics-Grounded Rollout Validation. We execute each synthesized reference motion in simulation to evaluate its feasibility under the deployment controller. At each step, the SONIC encoder maps a reference-motion window $\mathcal{R}_t$ to a 64-dimensional latent command $z_t = E(\mathcal{R}_t)$. The decoder combines this command with proprioceptive history h_t to produce joint targets $u_t = D(z_t, h_t)$ for low-level PD control. Two hand commands $g_t \in [0, 1]^2$ are executed separately. We record the resulting actions $a_t = (z_t, g_t) \in \mathbb{R}^{66}$ together with the executed robot and object states. Each rollout is evaluated against task success, stability, and collision criteria. Successful executions are added to $\mathcal{D}$, while failed rollouts are examined for recoverable tracking errors. Deviations in the realized root, hand, or object poses, such as locomotion-endpoint or grasp-height errors, provide feedback for Stage 3. Falls, invalid collisions, and failures beyond the correction scope instead trigger resampling in Stage 1.

Stage 3: Execution-Conditioned Correction. Even when reference motions are kinematically valid, tracking deviations may alter the robot–object relations required for successful interaction. We therefore use rollout feedback to

jointly correct the environment configuration and reference motion. Specifically, we compensate for planar tracking errors by adjusting relevant scene assets or interaction targets, while vertical grasp errors are corrected through reference-motion adaptation.

Robot-Pose-Guided $SE(2)$ Scene Correction. Locomotion tracking errors can shift the executed robot pose away from its intended interaction configuration. We apply an $SE(2)$ transformation to relevant scene assets to restore their intended spatial relations with the executed robot. Let (x, ψ) and (x', ψ') denote the intended and executed planar root poses, where $x \in \mathbb{R}^2$ and ψ denotes heading. For a scene asset at (x_j, ψ_j) in the same world frame, we apply

$$x_j^+ = x' + R_2(\psi' - \psi)(x_j - x), \qquad \psi_j^+ = \psi_j + \psi' - \psi. \tag{5}$$

Here, $R_2(\cdot)$ denotes planar rotation, and the superscript + indicates the corrected asset pose. This transformation preserves the intended robot–asset relationship relative to the executed root pose.

Object $SE(2)$ Correction. Scene-level alignment cannot fully correct local tracking errors during grasping or placement. We therefore adjust the interaction object or placement target using residual error relative to a successful source rollout:

$$\Delta e_{xy} = e_{xy}^{\text{cand}} - e_{xy}^{\text{src}}. \tag{6}$$

Here, e_{xy}^{cand} and e_{xy}^{src} are the planar tracking errors of the candidate and successful source rollouts, measured at the corresponding grasp or release event. Each error is defined as the executed-minus-reference position of the interacting hand or carried-object anchor in a common planar frame. Their difference isolates the additional tracking deviation introduced in the candidate execution. We compensate by translating the interaction object or placement target by Δe_{xy} and adjusting its yaw according to task-specific rules. Robot-relative targets retain their source yaw offset from the executed robot heading, while fixed-orientation targets preserve their configured yaw.

Z-axis Compensation. Planar adjustments cannot resolve vertical wrist–object misalignment during grasping. We compare the relative wrist heights at grasp closure in the candidate and successful source rollouts. The resulting discrepancy adjusts the reference wrist height to restore the source-relative grasp geometry. This correction is applied smoothly around grasp closure and converted to arm joint targets through IK, while keeping the object height fixed.

After correction, the updated environment and reference motion are re-executed in Stage 2. Only rollouts satisfying the same task success, stability, and collision criteria are accepted. These validated executions provide synchronized robot and object states, together with action labels, for subsequent rendering and policy learning.

3.3 Visual Observations and Real-World Deployment

We generate visual observations from physically validated executions rather than planned reference motions. By replaying the recorded robot and object states, we obtain synchronized egocentric views. We then combine Gaussian background renderings with simulator-rendered robots and objects through mask-guided z-buffer compositing to produce the final egocentric images.

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. See Section 4.1 for deployment details.

4 Experiments

In this section, we aim to answer four questions through real-robot experiments and simulation evaluation:

- **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?
- **Q4: Zero-HLM-Bench.** (Sec. 4.5). How do different policy families perform under shared evaluation protocols, and how does performance scale with simulation data?

4.1 Experimental Setup

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 visuomotor 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) [42] 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. The policy and SONIC run on an RTX 4090; simulation uses the same robot and hand configuration.

Real-world demonstrations are collected through PICO whole-body teleoperation with the SONIC controller [1]. Simulation data generation runs on eight RTX 4090 GPUs, and all policies are trained on eight A100 GPUs. Fig. 3 summarizes the deployment pipeline.

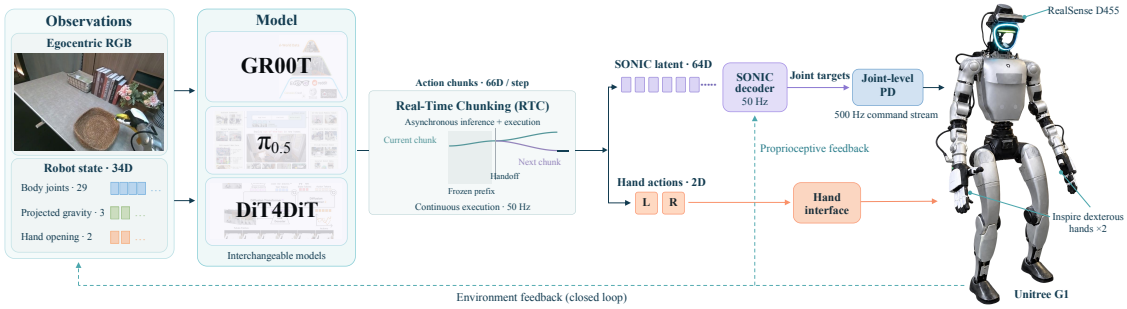


Figure 3. System Setup and Deployment. The policy maps egocentric RGB observations and robot state to SONIC motion latents and hand-opening commands. The frozen SONIC decoder produces body joint targets, while the two hands are controlled through a separate interface.

Tasks. We evaluate three long-horizon loco-manipulation tasks covering diverse whole-body skills:

- **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, where Bi-Place reports full-task success.

Task sequences are illustrated in Fig. 9 (a).

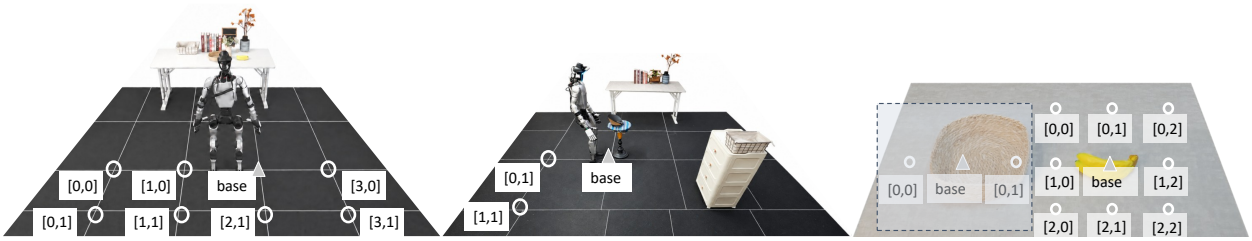
Camera Calibration. We first define an upper-body motion trajectory that maintains visibility of the robot's arms and hands. The physical robot executes this trajectory while recording egocentric RGB video and joint configurations. From selected video frames, we annotate 2D arm and hand landmarks and associate them with corresponding link-local 3D points from the robot URDF. Forward kinematics transforms these points into the camera mounting-link frame using the recorded joint configurations q_t. Given fixed camera intrinsics K, we obtain per-frame extrinsic estimates through PnP and discard frames with large reprojection errors. We then jointly optimize a fixed transform T from the mounting-link frame to the camera optical frame by minimizing the 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 is the camera projection function, and ρ is a soft-ℓ_1 loss.

Evaluation Configurations and Simulation Sampling. To systematically evaluate spatial generalization, we define two types of task configurations: 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. 4, 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. 2 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 ±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 4. 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 2. 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.

Data Settings and Evaluation Protocol. We compare policies trained on real-world demonstrations and simulation-generated data under BASE and EXPANDED configurations, as illustrated in Fig. 9(b). The real-data baseline, Real-Only(100), uses 100 demonstrations per task, while all simulation-generation methods start from the same pool of 15 teleoperated source demonstrations per task. As an agentic baseline, GPT-6 Astra generates 100 demonstrations for Tasks A and B using source demonstrations and full simulator state, denoted Sim-Astra(100). Our method generates 800 BASE demonstrations per task (Sim-Ours) and an additional 800 EXPANDED demonstrations for each of Tasks A and B (Sim-Ours*). Together with 800 object–goal-shift demonstrations, we generate 4,800 successful demonstrations in total.

For policy learning, we fine-tune pretrained GR00T-N1.7 [3] under each data setting and deploy all policies through the same SONIC whole-body controller. Each policy is evaluated on the physical robot over 20 trials per task in BASE and five trials per configuration in EXPANDED. We report stage-wise success rates for individual locomotion and interaction stages, alongside full-task success rates requiring completion of the entire task sequence.

4.2 Data Generation Quality and Efficiency

To answer **Q1**, we evaluate reconstruction quality and cost (Sec. 4.2.1), demonstration generation success and efficiency (Sec. 4.2.2), and the contributions of key components (Sec. 4.2.3).

4.2.1 Reconstruction Cost and Geometric Accuracy

Tab. 3 reports the time and cost of scene construction, physical configuration, and seven interaction-asset reconstructions, totaling 7h 13m and $22.82. As shown in Tab. 4, Gaussian guidance reduces the symmetric Chamfer distance to LiDAR ground truth from 11.689 to 5.517 cm, outperforming both unguided reconstruction and GS-Mesh. Evaluation details and qualitative comparisons appear in Appendix B.1.

Table 3. Reconstruction Cost. Time and allocated subscription cost by stage.

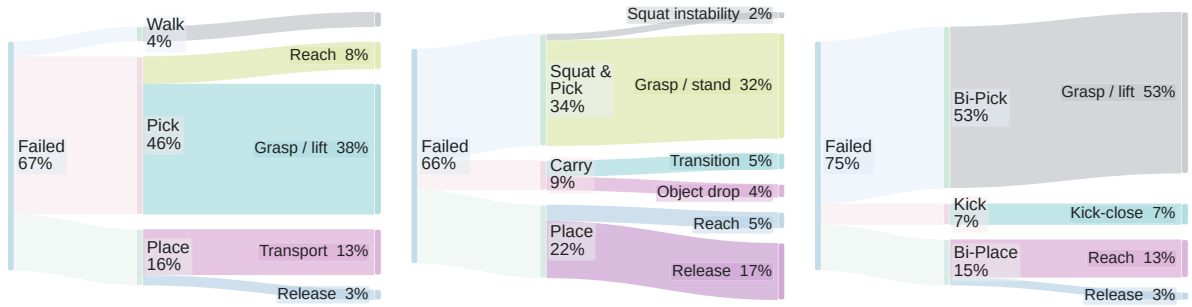
Stage	Time	Cost (USD)
Input → Blender	1h 47m	6.46
Blender → IsaacLab	1h 21m	5.08
Object reconstruction (7 assets)	4h 05m	11.28

Table 4. Geometry Accuracy. Symmetric CD to LiDAR ground truth.

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

4.2.2 Generation Success and Collection Time

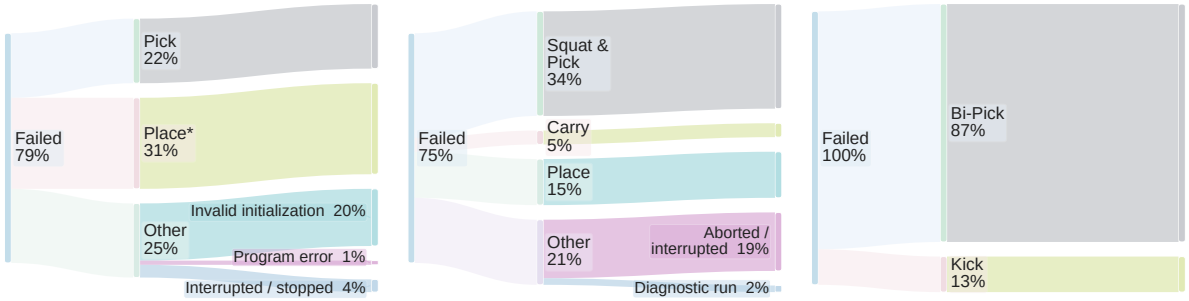
We analyze the generation success and failure modes of Sim-Ours and Sim-Astra under the same task-success, stability, and collision criteria. As shown in Figs. 5 and 6, Sim-Ours achieves success rates of 33%, 34%, and 25% on Tasks A–C, respectively, compared with 21.38% and 24.70% for Sim-Astra on Tasks A and B. Sim-Astra produces no successful Task C trajectories. Grasp-related failures dominate across tasks, while Sim-Astra additionally encounters invalid initialization, program errors, and interrupted runs. Each failed rollout is categorized by its earliest failed segment and a mutually exclusive failure type, with ribbon widths representing fractions of all attempts.



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

Figure 5. 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.

We also compare the time required to collect 50 successful demonstrations (Tab. 5). Sim-Ours achieves 2.4×, 1.6×, and 1.4× speedups over real-world teleoperation on Tasks A–C, respectively, averaging 1.8× across tasks. In contrast, Sim-Astra requires substantially longer collection times on Tasks A and B and produces no successful Task C demonstrations after more than eight hours.



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

Figure 6. 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.

4.2.3 Data Generation Ablations

Motion Adaptation Choices. We ablate motion adaptation and correction on Task A under BASE, targeting 100 successful trajectories per variant. As shown in Tab. 6, removing robot-pose-guided $SE(2)$ correction causes the largest success drop, from 61% to 16%. Removing object-level $SE(2)$ correction, Z-axis compensation, or velocity-adaptive transitions reduces success to 49%, 51%, and 55%, respectively. These results highlight the importance of scene correction and motion adaptation for reliable interaction.

Table 5. Collection Time. Hours per 50 successes; Real excludes setup.

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 6. Generation Ablations. Task A, BASE; 100 successes 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

Table 7. Mesh vs. GS. Task A real-world success.

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

Visual Observation Choices. With 100 simulated demonstrations per setting, we compare Gaussian and mesh rendering for real-world transfer. As shown in Tab. 7, Gaussian observations improve Task A full-task success from 3/20 to 7/20. Picking improves substantially (3/20 to 10/20), while walking shows a smaller gain (18/20 to 20/20), suggesting that visual fidelity particularly benefits object interaction. Fig. 7 compares both renderings with real-world observations.

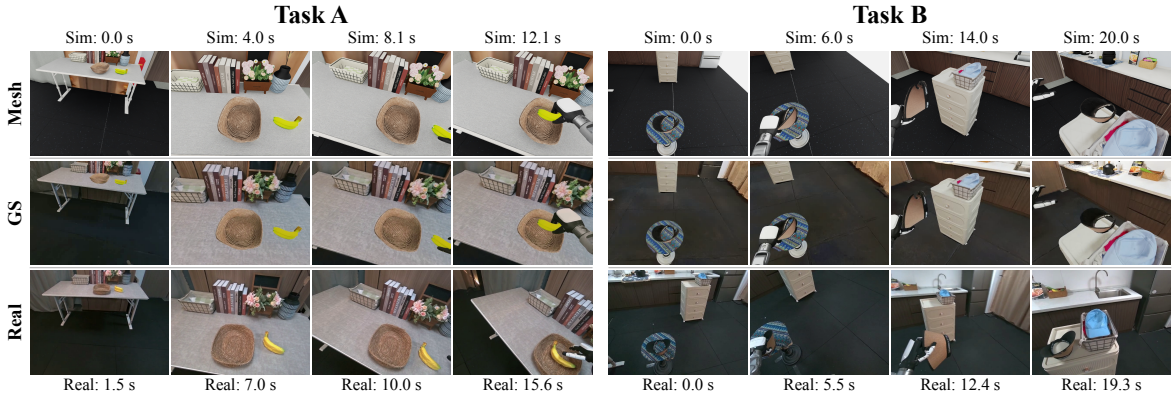


Figure 7. Mesh, Gaussian-Splatting, and Real-World Observations. Tasks A and B. Mesh and GS render the same simulation sequence; Real shows comparable stages from a separate physical rollout. Timestamps are independent across domains, not frame-aligned.

4.3 Direct Sim-to-Real Transfer and Generalization

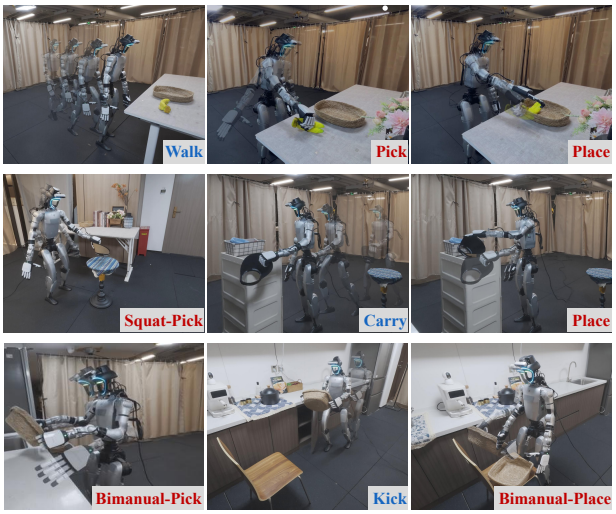
Direct Sim-to-Real Transfer. To answer **Q2**, we compare GR00T N1.7 fine-tuned on Sim-Ours(100) and Sim-Ours(800) with Real-Only(100) and Sim-Astra(100) under the BASE configuration. We evaluate all three tasks, except that Sim-Astra has no usable Task C data. As shown in Tab. 8, Sim-Ours(100) achieves 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. Increasing simulation data from 100 to 800 demonstrations per task raises mean success across three tasks from 43.3% to 66.7%, surpassing Real-Only(100) at 46.7%. Fig. 8 further details how real-world performance scales with the amount of our generated data on Task A.



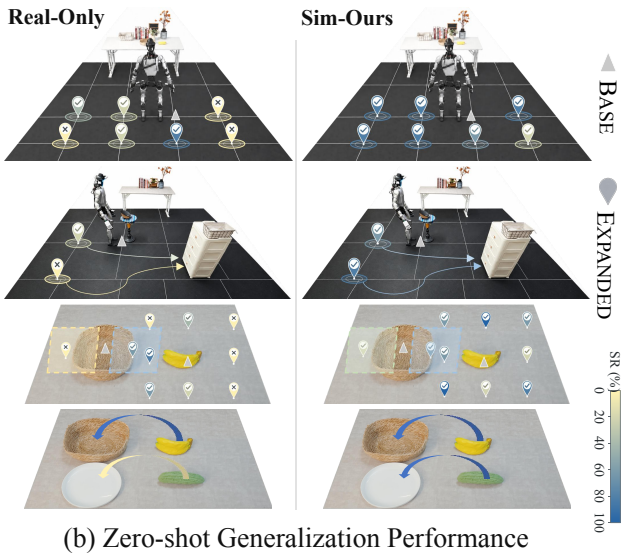
Figure 8. Sim-Data Scaling on Task A in Real-World Evaluation.

Table 8. 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 9. Tasks and Generalization Evaluation. (a) Task A–C stage sequences. (b) Robot starts for Tasks A and B, picking and placement locations, and an alternative object–goal pair (top to bottom). Triangles/pins mark BASE/EXPANDED; colors show real-world successes per five trials, from yellow (0/5) to blue (5/5).

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. 9, 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 scalable simulation data can substantially broaden real-world policy coverage without requiring additional physical demonstrations. Detailed per-configuration results are provided in Appendix B.3.

Table 9. Zero-Shot Transfer across Configurations. Real-world success rates (%) across robot-start, tabletop-location, and object–goal shifts, excluding Base. Avg. pools successful trials across all five settings (100 total trials).

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**

Table 10. Sim–Real Consistency on Task A. Full-task successes under matched BASE; denominators give trial counts.

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

4.4 Understanding Sim-to-Real Transfer

Can Simulation Reflect Real-World Performance? We examine sim–real consistency using Zero-HLM-Bench (Section 4.5). Specifically, we evaluate Task A policies trained on Real-Only(100) and Sim-Ours(100) in both simulation and the real world under matched BASE configurations. As shown in Tab. 10, Real-Only(100) achieves 40% real-world success versus 31% in simulation, while Sim-Ours(100) achieves 35% and 39%, respectively. The corresponding sim–real gaps are 9 and 4 percentage points.

These results suggest that the benchmark captures a comparable performance range across both training sources without consistently overestimating real-world success. However, the reversed policy ranking indicates that agreement in absolute success rates does not guarantee consistent relative performance. Thus, Zero-HLM-Bench provides an approximate measure of task difficulty under matched configurations, rather than an exact predictor of policy rankings. Further benchmark evaluations, including model comparisons and data scaling, are presented in Section 4.5.

What Gaps Remain in Actions and Observations? To examine the transferability of simulated actions and visual observations, we evaluate the banana pick-and-place subtask in Task A, varying its picking location to construct four training sets (Fig. 10). R-25 contains 25 real demonstrations with the banana placed in the lower region of the table, while S-25 contains 25 simulation demonstrations with the banana in the upper region. Mix-50 combines both datasets to cover both picking regions. For Mix-50*, we successfully replay the S-25 trajectories on the physical robot by manually adjusting object placements to match the simulated interactions. We then replace the simulated observations in S-25 with recorded real images while retaining the original simulation action labels. Combining these trajectories with R-25 yields Mix-50*, which preserves the same action supervision as Mix-50.

As shown in Tab. 11, R-25 and S-25 fail to generalize to their respective unseen picking regions, achieving 0/5 success. Although Mix-50 covers both regions, it achieves only 2/5 and 4/5 success in the upper and lower regions, respectively. Notably, upper-region performance drops from 4/5 for S-25 to 2/5 after mixing, suggesting that broader spatial coverage alone does not ensure consistent improvements. In contrast, Mix-50* achieves 5/5 success in both regions without changing the demonstration count or action supervision. These results support the transferability of simulation-generated action labels while identifying visual mismatch as an important limitation in sim–real co-training.

Figure 10. Spatial Coverage of Data. S-25 covers the upper table region; R-25 covers the lower region. Mix combines both.

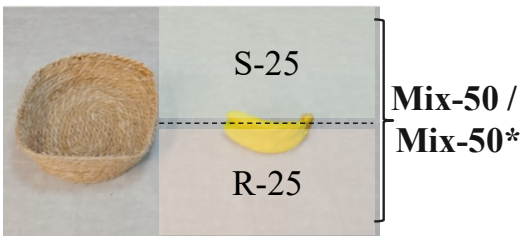


Table 11. Different Data Compositions. R denotes real-world data; S denotes simulation data.

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**

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, with 20 real-world evaluation trials per task.

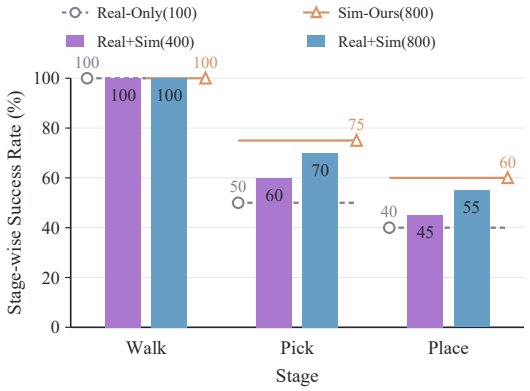


Figure 11. Co-Training Performance on Task A.

As shown in Fig. 11, 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). Tab. 12 reports the full stage-wise results for Tasks A–C. Increasing 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. Intermediate-stage performance does not improve uniformly: Task B Squat-Pick and Carry and Task C Kick decrease slightly. Thus, simulation data benefits co-training, yet combining real and simulation data does not necessarily outperform simulation-only training. Together with Tab. 11, these results suggest that the remaining visual-domain gap may limit effective sim–real co-training.

Table 12. 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; parentheses give the added simulation count.

Training Data	Real:Sim	Task A			Task B			Task C			Avg. (%)
		Walk	Pick	Place	Sq-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**

4.5 Zero-HLM-Bench: Humanoid Loco-Manipulation Evaluation

We introduce **Zero-HLM-Bench**, a humanoid loco-manipulation benchmark comprising the three task environments and nine skill stages under shared execution and evaluation protocols. We compare task-specific imitation learning, pretrained vision–language–action policies, and video–action modeling using the same generated training data.

4.5.1 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, never as policy inputs.

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. 12 visualizes the task layouts and sampled configurations used for benchmark evaluation.

4.5.2 Model Training and Execution

All model comparisons use Sim-Ours(800)* for Tasks A and B, and Sim-Ours(800) for Task C, with 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 adopt the LeRobot implementation of Diffusion Policy [43] for task-specific imitation learning using RGB observations, robot states, and action sequences. Images are resized to 224×224, with robot states and actions normalized during training. The policy executes 16 actions per update.

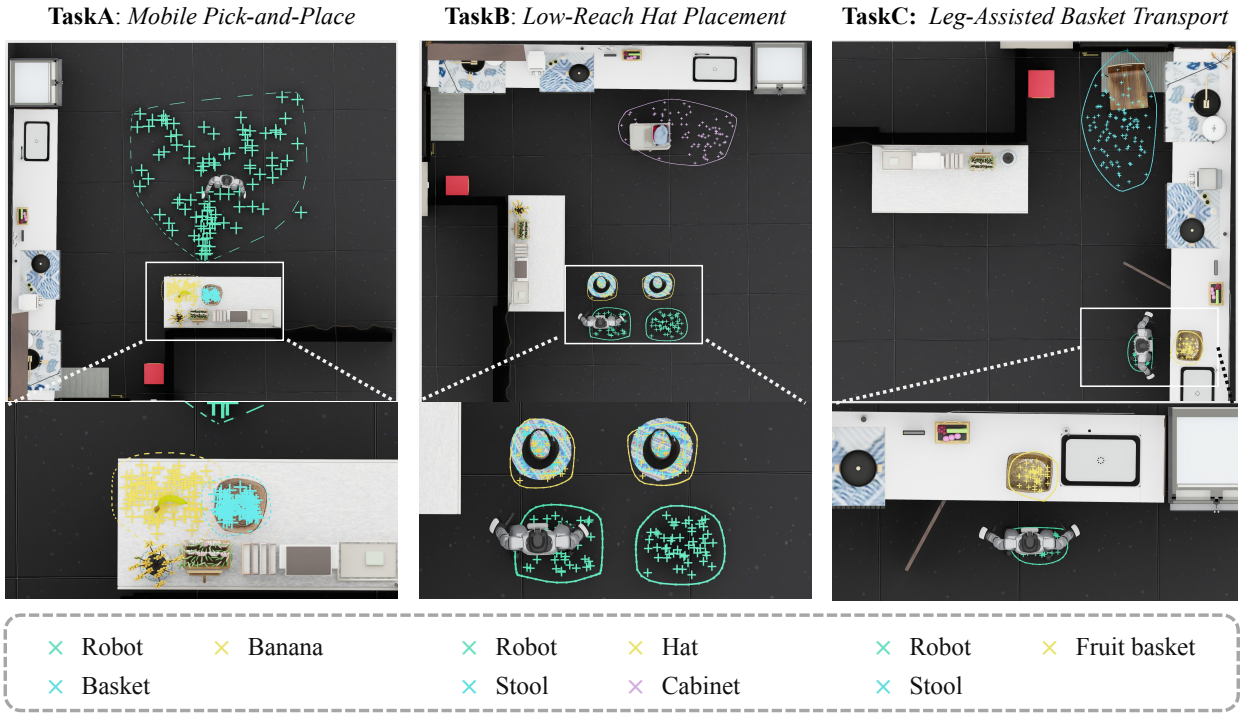


Figure 12. Benchmark Setup. Top-down layouts and sampled robot and object configurations for Tasks A–C, with enlarged views of the interaction regions.

$\pi_{0.5}$. We fine-tune the pretrained OpenPI $\pi_{0.5}$ checkpoint [34] using RGB observations, robot states, and a fixed task instruction. The model predicts a 50-step horizon of 66-dimensional actions.

GR00T N1.7. We adapt pretrained GR00T N1.7 [3] to the G1–SONIC embodiment. Training updates the action-diffusion module, projection layers, and selected vision–language normalization layers, while keeping both backbones frozen. The model predicts a 40-step action horizon.

DiT4DiT. DiT4DiT [36] combines a pretrained Cosmos video backbone with an action-diffusion transformer. We follow its native video–action training recipe with a 40-step horizon of 66-dimensional actions.

We distinguish prediction horizons from execution chunks. Diffusion Policy executes 16 actions per update across all tasks, while the VLA and video–action models execute 16 on Task A and 40 on Tasks B and C.

4.5.3 Evaluation Protocol and Metrics

Each model–task pair is evaluated over 100 simulation trials. We report full-task success rates alongside independently evaluated stage-wise performance to assess individual skill capabilities. Unlike full-task success, stage-wise evaluations measure each skill independently rather than requiring successful completion of all preceding stages.

Task A. Walk requires the robot to reach the designated target region 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 computed as the arithmetic mean of full-task success rates across Tasks A–C, excluding intermediate-stage evaluations.

Sim–Real Consistency. For the separate comparison in Tab. 10, we evaluate the same Task A policies in simulation and the real world under the BASE configuration. Each policy is tested in 100 simulation and 20 real-world trials.

4.5.4 Stage-Wise Benchmark Results

Tab. 13 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 13. Stage-Wise Simulation Benchmark. Stage-wise successes are reported as counts out of 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 rate (%) across the three tasks.

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

Policy Comparison and Data Scaling. Tab. 13 compares four policies under the same evaluation protocol. GR00T N1.7 achieves the highest mean full-task success (69.7%), while DiT4DiT performs best on Task B (80%). Fig. 13 shows consistent gains with increasing simulation data for GR00T N1.7. The gains extend across all three tasks, indicating that additional generated demonstrations benefit distinct whole-body interaction sequences. Improvements are larger at smaller data scales and become more gradual at higher counts; thus, the benefit is shared across tasks but is not linear in dataset size. The policy comparison also shows that high success on an individual stage does not guarantee high full-task success, motivating evaluation of both stage-wise behavior and complete execution. For example, Diffusion Policy reaches 90% walking success on Task A but only 3% full-task success.

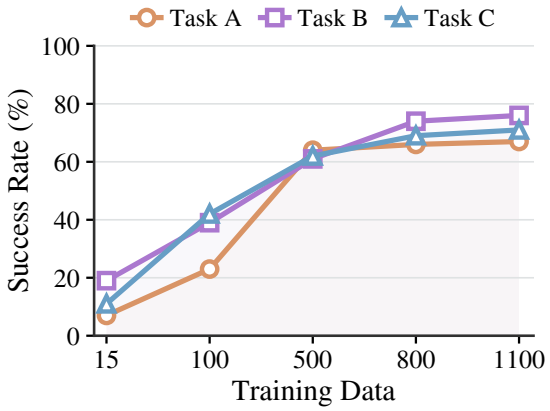


Figure 13. Sim-Data Scaling in the Benchmark. Full-task success improves with more generated demonstrations.

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.

References

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

[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.

[4] 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.

[5] 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*, 2026. URL https://arxiv.org/abs/2608.06375.

[6] 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.

[7] 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.

[8] 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/.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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*, pages 547–577. PMLR, 2025. URL https://proceedings.mlr.press/v305/yu25a.html.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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*, 11(4):4393–4400, 2026. doi: 10.1109/LRA.2026.3666395.

[24] 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.

[25] 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.

[26] 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.

[27] 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)*, pages 1667–1679. IEEE, 2025.

[28] 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)*, pages 13430–13441, June 2026.

[29] 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.

[30] 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.

[31] 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.

[32] 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*, 2026.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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*, pages 1820–1864. PMLR, 2023.

[40] 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.

[41] 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*, pages 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.

[42] 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.

[43] 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.

Appendix Contents

- **A Method Implementation Details**
  - A.1 Agentic Reconstruction: Implementation Details
- **B Experimental Details**
  - B.1 Agentic Reconstruction Evaluation
  - B.2 GPT-6 Astra Baseline
  - B.3 Generalization Evaluation Details
- **C Additional Qualitative Results**

A Method Implementation Details

We provide implementation details that complement the method described in Sec. 3. Section 3.2 details source annotation, source selection, and reference-motion construction; this appendix illustrates reconstruction.

A.1 Agentic Reconstruction: Implementation Details

This section provides implementation details for the reconstruction pipeline in Sec. 3.1. Fig. 14 illustrates the workflow from captured inputs to a physics-ready simulation environment. See Section 4.2.1 for reconstruction time and cost.

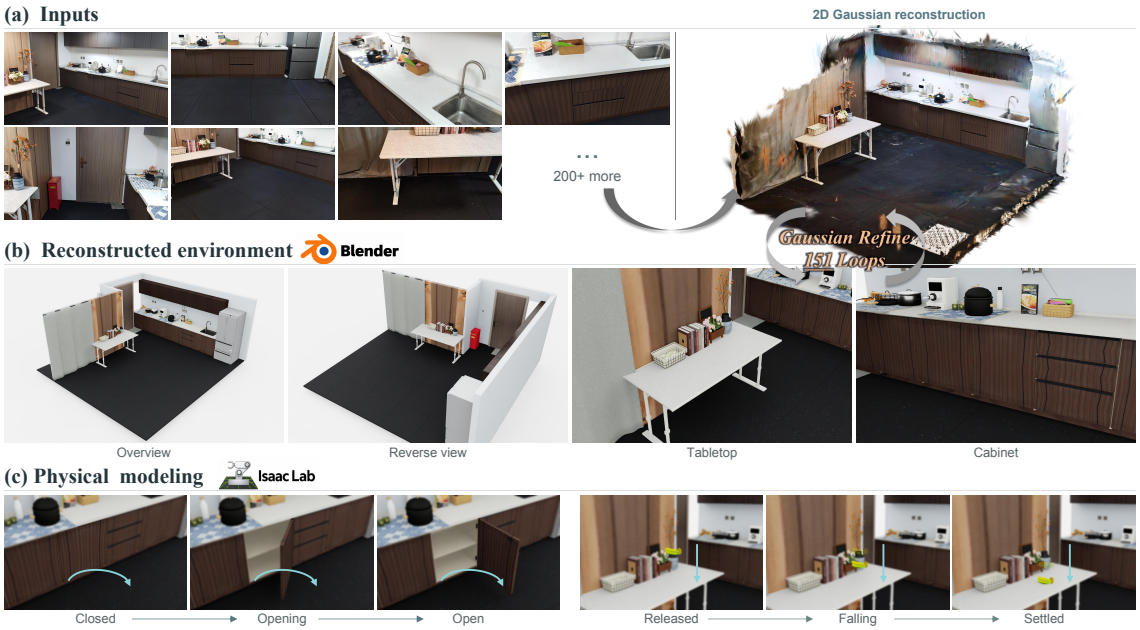


Figure 14. Agentic Reconstruction and Verification. (a) Captured RGB views, estimated camera poses, SfM geometry, and Gaussian reconstruction supply the inputs. (b) Overview and close-up views illustrate mesh-scene refinement in Blender. (c) Cabinet opening and object-release tests check articulation and supporting contacts in IsaacLab.

Visual Refinement and Physical Checks. The agent compares mesh and Gaussian renderings from matched viewpoints to refine furniture geometry and local object placement. After import into IsaacLab, cabinet-opening tests check articulated motion, while object-release tests check gravity-driven motion and contact with supporting surfaces. These checks complement the geometric comparison: visual agreement alone does not verify that a reconstructed scene supports the intended interactions.

B Experimental Details

This section supplements the experiments in Sec. 4 with reconstruction diagnostics, baseline details, and additional results. The system setup and evaluation configurations are presented in Section 4.1.

B.1 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. 4, 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. 15 further show that Gaussian guidance improves furniture dimensions, support-surface geometry, and local object placement.



Figure 15. 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 mesh–Gaussian detail views.

B.2 GPT-6 Astra Baseline

This section details the implementation of 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, with additional access to the full simulator state. The agent autonomously designs its data-generation pipeline, either by synthesizing complete reference motions or issuing incremental closed-loop commands. It can use the SONIC planner, inverse kinematics, and whole-body control interfaces. Fig. 16 presents the complete Task B prompt, specifying the available inputs, execution constraints, success criteria, and recording requirements.

We report results from the stabilized pipeline following human-assisted debugging and task alignment. Sim-Astra generates 100 successful trajectories per task for Tasks A and B, but none for Task C. Simulator-side planning and most agent reasoning latency are excluded from the recorded demonstrations, which contain only the executed robot trajectories used for policy training.

The generated trajectories frequently exhibit pauses and repeated attempts, often exceeding one minute compared with approximately 30 seconds for real-world demonstrations. Policies trained on these data show similar hesitation during real-world deployment, suggesting that such temporal irregularities may carry over into learned behavior.

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 16. 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.3 Generalization Evaluation Details

This section details the five settings summarized in Tab. 9 (Sec. 4.3), using the locations in Fig. 4. 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 14. 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. 4(a).

Table 15. 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. 4(b).

Table 16. Task A: Picking Locations.

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

The banana varies across eight EXPANDED locations while the basket remains fixed (Fig. 4(c)).

Table 17. 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. 4(c)).

Table 18. 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. 9.

C 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.

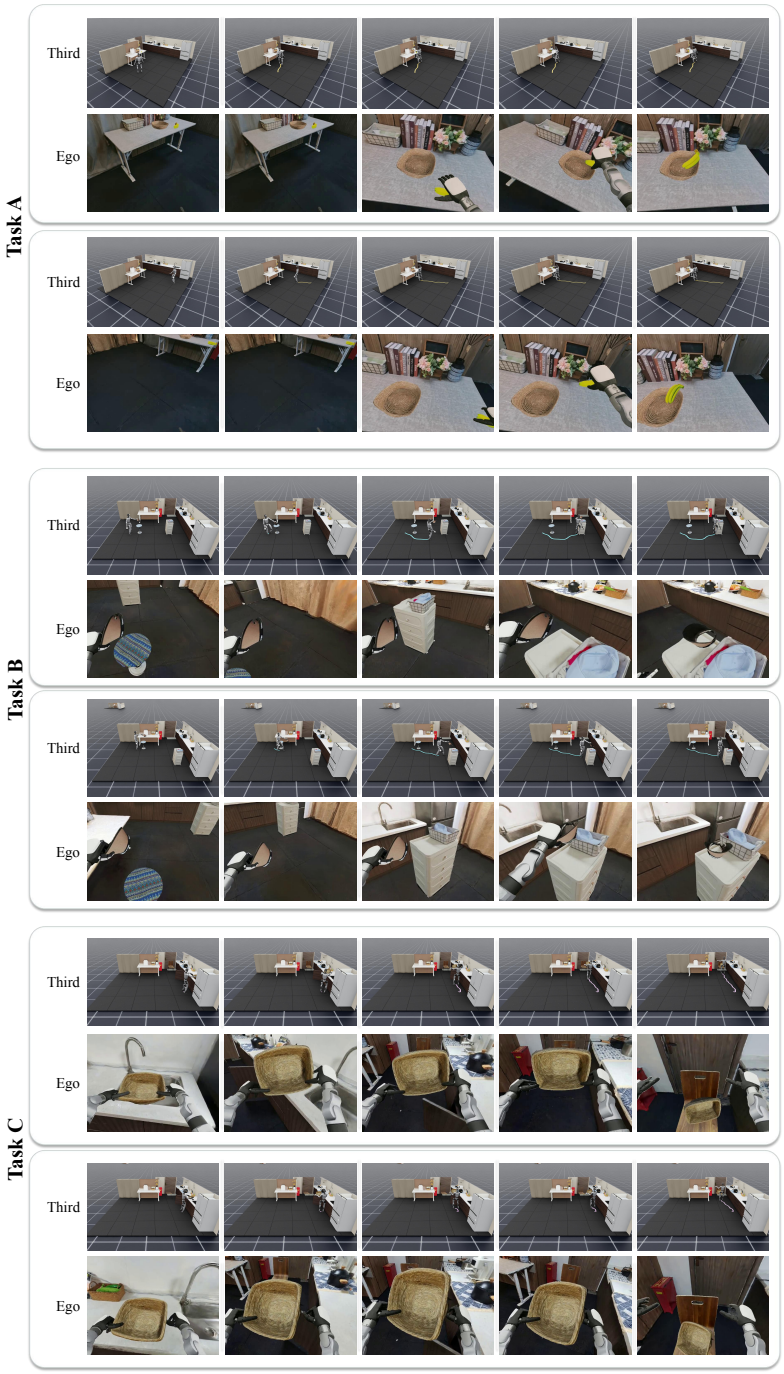


Figure 17. Generated Simulation Demonstrations. Third-person simulation views and corresponding egocentric renderings illustrate locomotion, manipulation, and leg-assisted interactions across Tasks A–C.