HERO: Learning Humanoid End-Effector Control for Visual Whole-Body Open-Vocabulary Object Grasping

Runpei Dong[†] **Ziyan Li**[†] **Arjun Gupta** **Xialin He** **Saurabh Gupta**
University of Illinois Urbana-Champaign

hero-humanoid.github.io



Figure 1: Open-vocabulary visual loco-manipulation with humanoids. We enable a humanoid to *autonomously* loco-manipulate *novel* objects in *novel* scenes using onboard sensors. A modular system combines large vision models for visual generalization with an accurate end-effector tracking policy. It achieves an 83.8% average success rate at reaching and picking up novel objects in challenging real-world scenes that require whole-body bending, squatting, and twisting.

Abstract: Visual loco-manipulation of arbitrary in-the-wild objects requires accurate end-effector (EE) control and a generalizable understanding of the scene from visual inputs (*e.g.*, RGB-D images). Existing imitation and sim2real methods jointly learn both these aspects via monolithic end-to-end learning and are thus hard to scale. In this work, we bring to bear the best tools for each of these problems – large vision models for generalizable scene understanding and simulated training for accurate EE control – leading to an overall modular loco-manipulation system that exhibits strong generalization. Our core technical innovation is HERO, an accurate residual-aware EE tracking policy made possible by combining classical robotics with machine learning. It uses a) inverse kinematics to convert residual end-effector targets into reference trajectories, b) a learned neural forward model for accurate forward kinematics, and c) goal adjustment and replanning. Together, these innovations reduce the end-effector tracking error to 2.44cm, outperforming the strongest prior method by 5.5×. Our overall system operates in diverse real-

[†]Equal contribution.

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

world environments, from offices to coffee shops, where the robot reliably grasps various everyday objects (*e.g.*, mugs, apples, toys) on surfaces ranging from 43cm to 92cm in height. Systematic modular and end-to-end tests demonstrate the effectiveness of our proposed design. We believe our advances open up new ways of training humanoids to interact with daily objects.

1 Introduction

Think about reaching to pick up the objects placed on the various table tops in Fig. 1. As humans we can reliably and robustly use our whole bodies to execute such pick ups. We can reach across a table with our back, rotate our torso for objects to the side, or squat for low coffee tables, all while balancing on two legs. We can pick up seen objects on seen tables, but also novel objects on novel tables in novel scenes. Once we have glanced at the object and scene, we can even do this with our eyes closed. In this paper, we develop a framework that equips humanoids with this fundamental capability: *autonomously reach over to pick up novel objects in novel everyday environments.*

Humanoids are doing backflips [1–5], so why would we be writing about such a mundane and seemingly unimpressive task? There are two key differences that make the problem of manipulating novel objects harder. First, in-the-wild operation means that neither reference motions nor privileged sensors (*e.g.* MOCAP) are available. But instead we need to process high-dimensional RGB-D image observations to infer object locations and plan trajectories. Second, object manipulation requires *precise* and *goal-directed* behavior: a robot needs to get its hand *where the object actually is*, different from a backflip where the focus is on landing safely rather than at a precise location. Operation in novel environments, sensing using on-board RGB-D sensors, precise end-effector (EE) control, and maintaining balance while reaching around, make this problem challenging.

The state-of-the-art for training humanoids for such tasks is end-to-end imitation learning in the real world [6–10] or end-to-end visual RL in simulation [11]. However, the difficulty of real-world data collection and of setting up photo- and physically-realistic scenes in simulation [11] limits the amount of data diversity and thus the generalization capabilities of learned policies. This causes them to fall short of the goal of manipulating *novel* objects in *novel* environments. In this paper, we pursue an alternate approach. We take inspiration from strong results with modular systems for tabletop object manipulation problems [12–14]. We use large vision models to translate high-level instructions (*e.g.* `grasp the red coke can`) into actionable plans by identifying the target objects in complex scenes and synthesizing a grasp on them; and a simulation-trained low-level control module then conveys the robot EE to the grasp location. Being able to use large pre-trained models enables broad generalization and even open-vocabulary reasoning. In many ways, this is the more direct, obvious, and performant way to build such a system. So why is such a modular method not already the go-to approach for building a humanoid object manipulation system?

While it is easy to get a Franka Emika robot to where you want, it turns out it is current methods aren't able to accurately control a humanoid hand. SONIC [15], the leading tracker achieves only a 13.38 cm tracking error in the real world, an error rate that is simply too large for object manipulation.[1] Our key technical contribution is to develop, HERO, an accurate EE tracking policy that cuts this error down by 5.5× to 2.44cm. This unlocks the possibility of developing modular humanoid systems for object manipulation that generalize without large-scale real-world imitation demos.

So what are the ingredients of building a highly accurate end-effector tracker? Our accurate end-effector control algorithm is based on multiple innovations. First, rather than directly trying to get the end-effector to the target location, we use a motion planner to generate an upper-body reference motion that gets the end-effector to the desired target. Second, the policy receives as input not just the current and target joint angles (output of the motion planner), but also the current and target end-effector position. Third, it is important to obtain a high-quality estimate of the current end-effector position, as we found that the default analytical forward kinematics and odometry on a low-cost

[1] Existing works do not focus on end-effector tracking accuracy. Imitation learning stacks built on top of inaccurate trackers still work because the IL policy learns to correct for the inaccurate tracking.

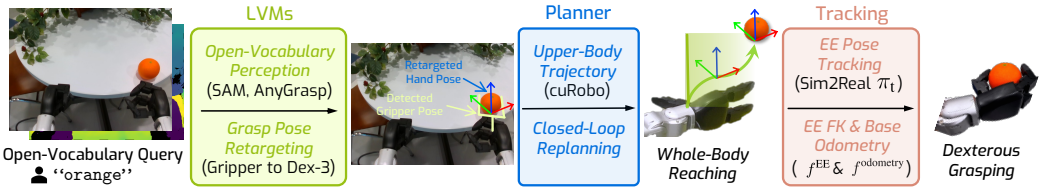


Figure 2: Overall architecture for our proposed modular system for open-vocabulary object grasping. Given a free-form natural language text query indicating which object needs to be picked, we use open-vocabulary large vision models (LVMs: Grounding DINO [16] and SAM [17]) to segment out the object of interest and predict parallel jaw grasps (using the AnyGrasp model [18]). We retarget the predicted grasp to a Dex-3 hand. We use our proposed whole-body end-effector tracker to convey the robot arm to the predicted grasp before picking up the object. By decomposing *action planning* (*i.e.* identifying which object to pick and using what grasp) from *action execution* (*i.e.* actual control of the robot), we inherit the strong visual generalization from pre-trained models as well as strong control capabilities for simulated training of the tracking policy.

humanoid robot like Unitree G1 are not accurate. We mitigate this issue by training neural forward models: an upper body model that provides accurate EE pose relative to the base, and a base odometry model that provides an accurate base pose relative to the stationary feet. Fourth, because whole-body balancing causes the planned reference to drift relative to the target, we periodically replan from the current state and odometry-corrected goal in a closed loop. Finally, a goal-adjustment loop iteratively nudges the commanded pose to remove the systematic sim2real gap.

Using this performant end-effector tracking policy, we develop a modular system for picking up open-vocabulary novel objects in novel everyday environments. This modular system leverages an open-vocabulary perception module to detect and segment the target object using large pre-trained vision models (Grounding DINO 1.5 [16] and SAM-3 [17]). We next use the AnyGrasp model [18] to produce parallel jaw grasps on the candidate object. We retarget them to the Dex-3 dexterous hand on the Unitree robot. Finally, we use our tracker as a low-level controller to achieve the grasp pose. In real-world testing for grasping open-vocabulary object queries in novel environments, our system achieves a success rate of 90% on 10 daily objects across standard and short table heights, 73.3% success rate on generalization to 10 daily scenes, and 80% success rate on cluttered scenes.

2 Related Works

Loco-manipulation via Motion Tracking. Motion tracking is a central tool for humanoid loco-manipulation: teleoperation or generated references provide whole-body motions, and RL policies learn to track them for sim-to-real transfer [6, 9, 19–22]. Recent work improves tracking accuracy, agility, robustness, reachability, and generalization [5, 15, 23–31], including force- and stability-aware end-effector control [32–34]. Teleoperation systems further enable imitation learning for rich manipulation [35–43]. In contrast, HERO does not require human teleoperation at test time; it converts visual grasp goals into end-effector targets and executes them with a learned tracker.

Visual Loco-Manipulation. Visual loco-manipulation often uses imitation data collected by teleoperation [6–9, 44] or reference-state policies that map depth observations to high-level commands or generated motions [45–48]. End-to-end visual RL has also shown strong results on selected object categories [11], and recent foundation-model approaches aim at universal humanoid policies [49, 50]. These systems remain limited by collected demonstrations, generated assets, or object categories. HERO instead pairs open-vocabulary perception with modular task-space control, enabling novel language queries and objects without task-specific teleoperation demonstrations.

System Identification. Real robots deviate from their nominal models because of hardware inaccuracies, joint elasticity, and dynamics [51–53]. System identification mitigates this sim-to-real gap either through online adaptation [54–59] or offline correction from collected data [5, 60, 61]. Our learned residual FK and base-odometry models follow the offline direction, using MOCAP data to correct the task-space signals required by the tracking policy.

3 HERO: Humanoid End-Effector ContROl

Given a desired end-effector pose in the robot frame, HERO outputs motor commands for all 29 DoFs of a Unitree G1 humanoid [62]. Although the feet remain planted, far-reaching motions require coordinated waist, torso, and leg motion. We therefore combine classical IK and motion planning with learned task-space feedback instead of learning a monolithic pose-to-action mapping. The overall tracker is shown in Fig. 3. HERO first converts the target $T^{\text{EE}} \in \text{SE}(3)$ into a base height h and upper-body joint goal $\mathbf{q}^* \in \mathbb{R}^{17}$ using IK, then uses a collision-free planner [63] to produce reference joint and end-effector trajectories. A learned whole-body policy π_t tracks these references with joint position commands at 50Hz. To make the residual task-space feedback accurate, π_t uses learned end-effector FK η and base odometry ξ. During execution, we also replan periodically and apply a small goal adjustment to compensate systematic sim-to-real bias.

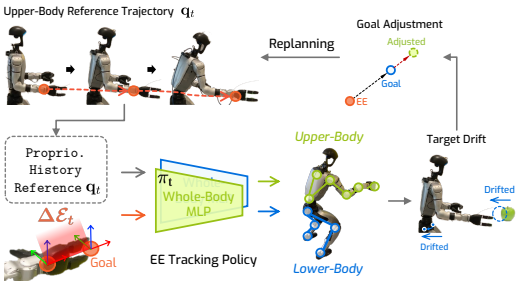


Figure 3: HERO end-effector control. HERO outputs IK and planned references from an EE goal, then tracks them with a learned whole-body policy, which leverages residual EE error computed from learned FK and odometry. Replanning and goal adjustment compensate drift and sim2real bias.

3.1 Whole-body End-Effector Tracking Policy, π_t

To track the target EE pose $T^{\text{EE}} \in \text{SE}(3)$ defined in the robot frame, our whole-body EE tracking policy π_t first obtains an upper-body reference trajectory $\{\mathbf{q}_t\}_{t=1}^{T}$ and the corresponding 6-DoF EE reference trajectory $\{ee_t\}_{t=1}^{T}$ from a motion planner. Given the trajectory, the current proprioceptive state $\mathbf{s}_t$, and other commands, π_t outputs the 29-DoF joint angles commands that are passed to per-joint PD controllers.

Residual-Aware End-Effector Tracking. π_t output actions $\mathbf{a}_t$ at time t as follows:

$$\mathbf{a}_t = \pi_t\left(\mathbf{s}_t, h_t, \mathbf{q}_t, \Delta\mathcal{E}_t, \mathbf{v}_t, \mathbf{s}_{t-5:t-1}, \mathbf{a}_{t-5:t-1}\right), \tag{1}$$

where $\mathbf{s}_t$ is the current proprioception, h_t is the reference base height, $\mathbf{q}_t$ are the reference upper-body joint angles, $\mathbf{v}_t$ are the linear and angular velocity locomotion commands, and $\mathbf{s}_{t-5:t-1}, \mathbf{a}_{t-5:t-1}$ are five time steps of proprioception and action history. The proprioception include the robot's joint angles, joint velocities, angular velocity, projected gravity, and roll and pitch encoded from the IMU. We don't use the IMU yaw as it is inaccurate [9]. $\Delta\mathcal{E}_t$ represents the residual pose error between the current and target end-effector pose in the robot frame, *i.e.*,

$$\Delta\mathcal{E}_t = f^{\text{EE}}(\mathbf{x}_t) \ominus ee_t, \tag{2}$$

where $f^{\text{EE}}(\mathbf{x}_t)$ maps the arm states $\mathbf{x}_t \in \mathbb{R}^{17}$ to the end-effector pose $T_t^{\text{EE}} \in \text{SE}(3)$, and $\ominus$ is the inverse pose composition operator.[2]

Architecture and Training. π_t uses two three-hidden-layer MLPs that share the same observation but separately predict upper- and lower-body actions; their outputs are combined into 29-DoF joint commands. We train π_t in simulation with PPO [64], using AMASS [65] ($\sim$8K motion sequences) and a curated set of everyday reaching targets ($\sim$8K). Targets are sampled in the robot frame from $[0.1\text{m}, -0.5\text{m}, 0.65\text{m}]$ to $[0.5\text{m}, 0.5\text{m}, 1.15\text{m}]$, with yaw in $[-60°, 60°]$. A motion planner converts each target into upper-body and end-effector reference trajectories for policy training.

3.2 Learned Residual Neural Forward Models

Residual Neural FK. Our residual neural forward kinematics function, η, learns a correction to the output of the analytical forward kinematic function, FK, to output accurate end-effector poses.

[2] We use $\oplus$ to denote pose composition: $\mathbf{T}_1 \oplus \mathbf{T}_2 = \mathbf{T}_1 \cdot \mathbf{T}_2$ and $\ominus$ for inverse composition: $\mathbf{T}_1 \ominus \mathbf{T}_2 = \mathbf{T}_2^{-1} \cdot \mathbf{T}_1$.

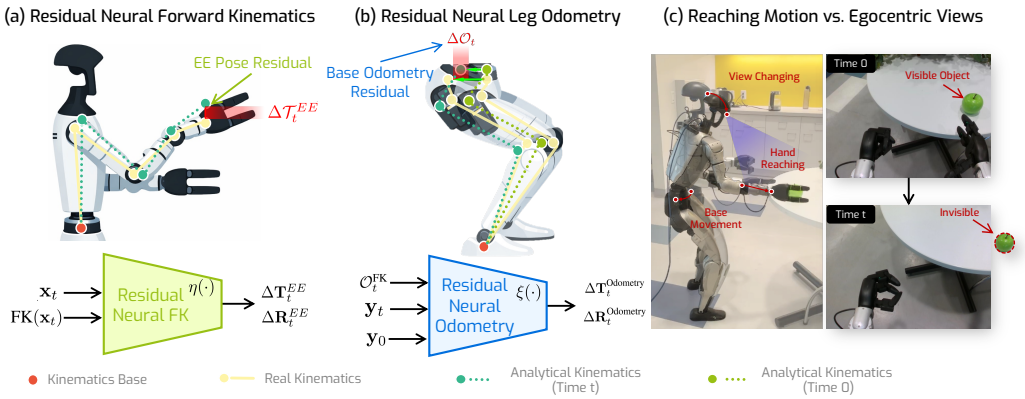


Figure 4: Learned neural forward kinematics and odometry models. (a) A residual FK model η corrects analytical FK by predicting EE translation and rotation residuals. (b) A residual odometry model ξ estimates base motion from lower-body joints. (c) Whole-body reaching can move the object out of the egocentric view, making purely visual correction unreliable.

Specifically, given the current proprioceptive state of one operating arm and waist $\mathbf{x}_t \in \mathbb{R}^{10}$ at timestep t, and output from analytical FK, $\text{FK}(\mathbf{x}_t)$, the final end-effector pose $f^{\text{EE}}(\mathbf{x}_t)$ is obtained as:

$$f^{\text{EE}}(\mathbf{x}_t) = \text{FK}(\mathbf{x}_t) \oplus \eta\big(\mathbf{x}_t, \text{FK}(\mathbf{x}_t)\big). \tag{3}$$

Note that the analytical forward kinematics function $\text{FK}(\mathbf{x}_t)$ uses the robot geometry and coordinate transformations to compute the 6-DoF end-effector pose in the robot base frame. For precise robots, FK is itself quite accurate, however as our experiments will show, FK is inaccurate for our humanoid [51, 52], necessitating the need for learning a correction.

Residual Neural Leg Odometry. Different from fixed-base object manipulation tasks, where the robot's base is fixed and stable, humanoid robots' base needs to be dynamically balanced during whole-body reaching. This movement makes the original reaching target inaccurate, as the reaching target defined in the robot frame is no longer the same place where the object lies. One might consider using the egocentric visual information for replanning or motion adjustment. However, as shown in Fig. 4, the egocentric view of the robot is too narrow for the robot to see the object when the robot's arm and waist movements are large. As a result, using the robot's odometry to adjust the reaching goal is critical. We assume the feet to be static on the ground and use the lower body joint angles to predict the base pose. By assuming the robot ankle joint as the root joint and the robot base as the end-effector, we can compute the base pose using forward kinematics.

However, similar to the error in analytical FK for EE, analytical FK to compute the base pose is also inaccurate (see analysis in Sec. 5). Similar to our solution for EE, we adopt a residual model to mitigate these inaccuracies. We estimate the base pose transformation relative to time step 0, rather than the absolute base pose, which avoids depending on a global frame and directly tells the controller how much the base has shifted since planning.

Concretely, let $\mathbf{y}_t \in \mathbb{R}^6$ be the 6DOF state of the left (or right) leg motors. We can get analytical base odometry, *i.e.* base pose relative to base pose at time step 0, $\mathcal{O}^{\text{FK}}(\mathbf{y}_t, \mathbf{y}_0) \in \text{SE}(3)$, using analytical forward kinematics and SE(3) difference: $\mathcal{O}^{\text{FK}}(\mathbf{y}_t, \mathbf{y}_0) = \text{FK}(\mathbf{y}_0) \ominus \text{FK}(\mathbf{y}_t)$. Our residual neural leg odometry model ξ learns the residual:

$$f^{\text{odometry}}(\mathbf{y_t}, \mathbf{y_0}) = \mathcal{O}^{\text{FK}}(\mathbf{y}_t, \mathbf{y}_0) \oplus \xi(\mathbf{y}_t, \mathbf{y}_0, \mathcal{O}^{\text{FK}}(\mathbf{y}_t, \mathbf{y}_0)). \tag{4}$$

Architecture and Training. Both residual models are lightweight 3-layer MLPs. η predicts a residual SE(3) correction for the end-effector, while ξ predicts the residual between MOCAP odometry and analytical leg-FK odometry. Each model uses separate heads for translation and rotation, outputting a residual translation in $\mathbb{R}^3$ and a residual rotation via the 6D rotation representation [66].

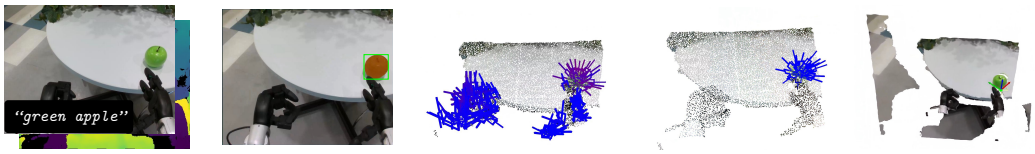


Figure 5: Visual perception for language-conditioned grasping. An example of target segmentation, grasp proposals, filtering, and retargeted EE poses from onboard RGB-D observations.

We train both models on MOCAP data [67]. For η, a tracking policy sweeps the EE through the workspace while we record motor encoders and MOCAP EE/base poses; marker poses are converted with the Kabsch-Umeyama algorithm [68, 69] (< 1.5mm RMSE). We collect 3 hours of data and use the first 2 hours for training and the last 1 hour for validation. ξ uses the same split, with supervision formed by sampling temporal pairs from trajectories. Both models are optimized with MSE losses on residual translations and rotations, using MOCAP-derived ΔT_t^{EE} and $\Delta \mathcal{O}_{n-m}$ as ground truth.

3.3 Replanning and Goal Adjustment

During execution, the robot drifts from the planned reference because of base-pose changes from whole-body balancing and residual sim-to-real bias, leaving the policy with an out-of-distribution input or a target that is no longer reachable. We therefore replan the remaining reference $\{\mathbf{q}_t, ee_t\}$ every $k = 300$ steps (6 sec) with cuRobo [63] ($\sim$20 ms), using the current state and the odometry-corrected goal so each fresh trajectory accounts for the latest base displacement.

To further compensate the steady-state EE tracking offset, we maintain a commanded grasp g initialized to T^{EE} and apply the damped translational update $g^p \leftarrow g^p - \beta, e_t^p$, where $g^p, e_t^p \in \mathbb{R}^3$ denote the translational components of g and the SE(3) residual $e_t = f^{\mathrm{EE}}(\mathbf{x}_t) \ominus T^{\mathrm{EE}}$, $\beta = 0.6$, only when $|e_t^p| \leq 15\mathrm{cm}$, clipped to $5\mathrm{mm}$/step, and stopped once $|e_t^p| < 1.5\mathrm{cm}$. We adjust translation only because scaling the rotation residual the same way does not empirically improve tracking.

4 A Modular System for Open-Vocabulary Humanoid Object Grasping

Our task is to pick up novel objects in novel environments from free-form language queries, using only onboard sensing. Our modular pipeline builds on HERO: Grounding DINO [16] detects the queried object, SAM-3 [17] segments it, and AnyGrasp [18] proposes parallel-jaw grasps. We filter and retarget the selected grasp to the Dex-3 hand before HERO executes the reach. Fig. 5 visualizes the pipeline; detailed filtering and retargeting steps are given in Sec. A.5. We use MOCAP to accurately calibrate the egocentric camera, Fig. 6 shows the MOCAP setup and camera calibration.

For Dex3 retargeting, we rotate the AnyGrasp pose by 45° around the z-axis so the thumb opposes the other two fingers, improving contact area and force closure. We also clip end-effector rotation within 70° to avoid twisted IK postures that degrade tracking.



Figure 6: MOCAP and camera calibration. (a) Using measured marker-to-link offsets, Kabsch–Umeyama [68, 69] recovers 6-DoF EE/base poses from MOCAP coordinates (< 1.5 mm RMSE). (b–c) The D435i camera is mounted on the G1 head with a passive neck pitch joint. (d–e) We manually move an ArUco board [70] with MOCAP markers through 60–70 poses, recording board poses in the camera frame and board/base poses in the MOCAP frame. Tsai–Lenz [71] then solves the camera-to-base transform. More details can be found in Sec. B.1 and Sec. B.2.

Table 1: Open-vocabulary grasping success. Compared with direct PD control and automated SONIC [15], HERO achieves 90% success across objects at two table heights.

Language Query	PD Controller		SONIC [15]		Ours	
	Gen. 0.74 m	Short 0.56 m	Gen. 0.74 m	Short 0.56 m	Gen. 0.74 m	Short 0.56 m
red coke can	2/3	0/3	0/3	2/3	3/3	3/3
e-stop button	0/3	0/3	2/3	0/3	3/3	3/3
red piranha plant	0/3	0/3	0/3	0/3	3/3	3/3
orange cube	0/3	0/3	0/3	0/3	3/3	3/3
olive oil bottle	0/3	1/3	0/3	0/3	2/3	2/3
game cartridge	1/3	0/3	0/3	0/3	2/3	3/3
chip can	1/3	1/3	0/3	0/3	2/3	3/3
hand soap bottle	0/3	0/3	2/3	3/3	3/3	3/3
robot hand	0/3	0/3	0/3	0/3	3/3	2/3
red apple	0/3	0/3	0/3	1/3	3/3	2/3
Total	4/30	2/30	4/30	6/30	**27/30**	**27/30**

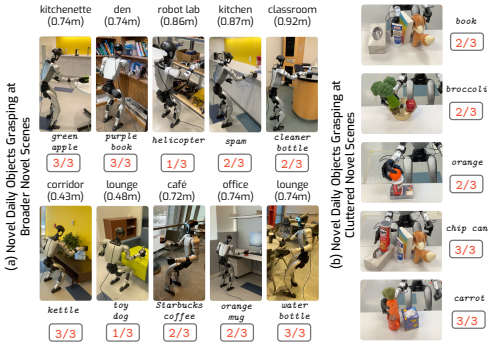


Figure 8: Open-vocabulary grasping in real-world scenes. HERO achieves 22/30 (73.3%) success across broader novel scenes and 12/15 (80%) success in cluttered layouts.

5 Experiments

We evaluate HERO on open-vocabulary grasping, learned FK, and tracking gains from residual feedback, replanning, and goal adjustment.

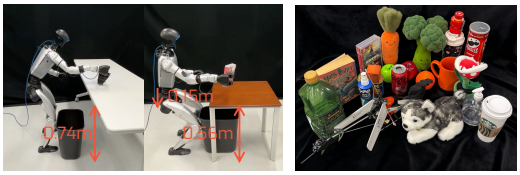


Figure 7: Novel test setups and objects. (a-b) Standard (0.74m) and short (0.56m) table setups. (c) Diverse daily objects used for the test.

Experiments use an *unmodified* Unitree G1 humanoid with Dex-3 dexterous hands (3-finger hands), a head-mounted Intel D435i RGB-D camera, proprioception, and a base IMU. Visual trials use novel objects and environments; FK and tracking evaluations use a 13-camera Optitrack MOCAP room (see Sec. B.1). Selected design choices are validated in Isaac Gym [72] and MuJoCo [73]. In addition, Sec. A reports language sensitivity, moving-object grasping, door opening, walking FoV analysis, tracking-error CDFs, trajectory curves, and workspace coverage.

5.1 End-to-end System Testing

Experimental Setup. Each trial starts 10-20 cm from a table (43-92cm high). The robot must grasp the queried object using onboard sensing and lift it for more than 2 seconds. We run 3 trials per object per table height; Fig. 7 shows the objects and Tab. 1 lists the queries.

Baselines. We compare HERO to two alternate controllers. To maximally isolate the impact of the tracking quality, we plug in these alternate controllers into the same overall modular pipeline as HERO. *PD Controller* tracks IK reference joints without a learned policy. For *SONIC* [15], we use the three-point VR teleoperation interface. We use the predicted grasp as the right EE target and place the head and the left EE targets via FK from upper-body joint positions obtained from right EE IK.

Results: ❶ 10 Daily Objects. Tab. 1 shows that HERO succeeds in 90% of trials across objects, queries, and two table heights, *vs.* 10% for PD control and 16.7% for automated SONIC. Accurate whole-body end-effector tracking is essential for turning visual grasp goals into physical grasps. **❷ 10 Daily Scenes.** Fig. 8(a) evaluates broader scenes such as robot labs and classrooms, where HERO reaches 73.3% success. **❸ 5 Cluttered Layouts.** In cluttered layouts (Fig. 8(b)), HERO achieves 80% success without teleoperation demonstrations, remaining language-sensitive under distractors. On 1,450 queries constructed from OCID-Ref images [74], our perception module detects targets in 73.8% of canonical-name queries and 89.7% of queries with attributes and spatial relations (Tab. 7).

5.2 End-effector Tracking Accuracy Evaluation and Ablation Study

We evaluate tracking on 180 reaching goals in simulation and real MOCAP: 60 poses across three table heights, sampled 5-15cm above the table. We report EE translation/orientation error against

Table 2: Simulation tracking comparison. End-effector translation, orientation, and joint-tracking errors averaged over three table heights in MuJoCo. HERO attains the lowest error; SONIC lacks in EE tracking feedback and joint position only baselines fall behind.

Method	Trans. (cm)	Orient. (deg)	Joint (rad)
FALCON [33]	13.57 ± 4.41	19.33 ± 5.30	$\mathbf{0.02 \pm 0.00}$
AMO [25]	8.29 ± 3.82	13.85 ± 5.91	$\mathbf{0.02 \pm 0.00}$
SONIC [15]	4.10 ± 2.12	16.65 ± 9.27	–
HERO (ours)	$\mathbf{2.48 \pm 1.15}$	$\mathbf{11.23 \pm 3.97}$	0.17 ± 0.03

Table 3: Real-world tracking comparison. MoCap-measured EE errors on hardware. HERO retains sub-3 cm accuracy in the real world, while SONIC baseline degrade by 4–5× compared to simulation. Lacking dynamic gravity compensation, the PD controller leaves the EE persistently low and misoriented.

Method	Trans. (cm)	Orient. (deg)	Joint (rad)
PD Controller	12.09 ± 3.14	40.82 ± 36.27	0.24 ± 0.03
SONIC [15]	13.38 ± 1.43	16.75 ± 10.06	–
HERO (full)	$\mathbf{2.44 \pm 0.86}$	$\mathbf{8.22 \pm 3.52}$	$\mathbf{0.21 \pm 0.05}$

Table 4: Forward-model ablation. Replacing analytic FK with our learned models reduces sim2real bias and approaches the MoCap oracle.

EE Pose	Base Pose	Trans. (cm)	Orient. (deg)	Joint (rad)
FK	FK	4.67 ± 1.30	14.59 ± 3.99	0.20 ± 0.03
Ours	FK	3.35 ± 0.70	14.07 ± 3.93	0.19 ± 0.03
FK	Ours	3.89 ± 1.06	14.28 ± 4.75	0.20 ± 0.04
Ours	Ours	$\mathbf{2.56 \pm 1.23}$	$\mathbf{12.06 \pm 4.38}$	$\mathbf{0.18 \pm 0.03}$
MoCap	*MoCap*	$\mathit{2.44 \pm 0.86}$	$\mathit{8.22 \pm 3.52}$	$\mathit{0.21 \pm 0.05}$

Table 5: Closed-loop feedback ablation. Real-world MoCap tracking with replanning or goal adjustment removed. Replanning significantly improves tracking accuracy, while goal adjustment further reduces errors.

Method	Trans. (cm)	Orient. (deg)	Joint (rad)
HERO (full)	$\mathbf{2.44 \pm 0.86}$	$\mathbf{8.22 \pm 3.52}$	0.21 ± 0.05
w/o Replan	5.17 ± 2.21	16.13 ± 4.66	0.21 ± 0.03
w/o Goal Adj.	2.71 ± 0.87	9.38 ± 2.72	$\mathbf{0.20 \pm 0.03}$

MOCAP and upper-body joint error from motors. Baselines include the PD controller, AMO [25], FALCON [33], and automated SONIC [15]. We also conduct ablation study on forward model accuracy (Sec. 3.2), closed-loop adapation strategies (Sec. 3.3).

Comparisons against state-of-the-art. Tab. 2 reports simulation results averaged over three table heights. HERO achieves the lowest errors, with 2.48cm translation error versus 4.10cm for SONIC, 8.29cm for AMO, and 13.57cm for FALCON. SONIC is the closest translation baseline since it also reasons about EE targets, but its lack of end-effector error feedback leads to higher tracking error. The gap between joint and task-space metrics also shows that joint accuracy alone does not ensure end-effector accuracy. Tab. 3 shows the same trend on real-world MOCAP: full HERO achieves 2.44cm translation and 8.22° orientation error, outperforming both PD control (12.09cm, 40.82°) and automated SONIC (13.38cm, 16.75°). In simulation, analytical FK and odometry are exact, so HERO uses them directly; replanning and goal adjustment remain enabled as in real-world deployment without further specifications.

Forward models and closed-loop adaptation. Tab. 4 shows that learned EE/base estimates reduce analytical-FK bias and give the best non-oracle tracking accuracy. Using learned estimates for both EE and base poses reduces translation error from 4.67cm to 2.56cm, close to the MOCAP oracle at 2.44cm. Tab. 5 ablates the feedback terms on real hardware. Removing replanning roughly doubles translation error from 2.44cm to 5.17cm, while removing goal adjustment gives a smaller degradation. These results indicate that learned pose estimation and closed-loop replanning improve accuracy.

Vanilla FK *vs.* Learned FK. We measure FK error by marking the robot base and EE and using their MOCAP relative transform as ground truth. We record poses and joint angles, then compare our method to a) analytical FK and b) kinematic calibration, where we learn URDF geometric model parameters [75–77].

Table 6: FK and odometry accuracy. EE/base translation and rotation errors from MOCAP; kinematic calibration and residuals reduce bias.

Method	a) End-effector Pose		b) Base Odometry	
	Translation Error (cm)	Rotation Error (deg)	Translation Error (cm)	Rotation Error (deg)
Analytical FK	1.30	6.03	1.00	0.49
Kinematic Calibration	0.58	**2.83**	**0.51**	0.51
Learned FK, no residual (ours)	4.44	7.06	0.53	0.75
Learned FK (ours)	**0.50**	3.18	0.52	**0.46**

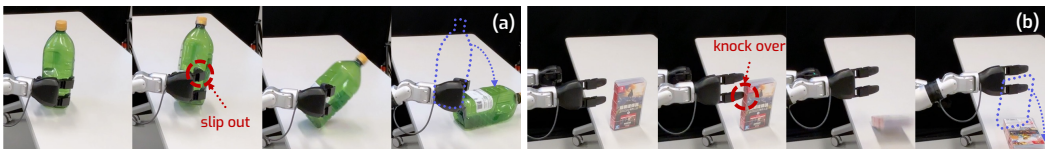


Figure 9: Failure mode examples. (a) Object slips from the hand. (b) Object knocked over.

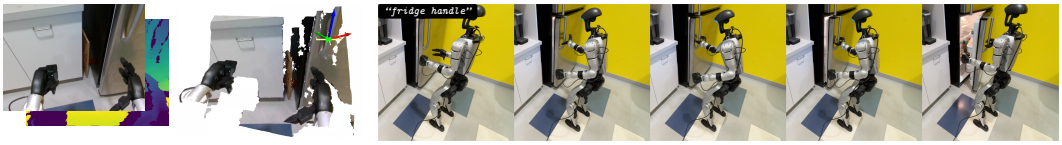


Figure 10: Door opening with HERO. (a) The egocentric RGB-D visual inputs. (b) Given a language query for the door handle (*e.g.*, ``fridge handle'' prompted here), our modular system obtains the grasping pose, same as the pipeline for picking up objects. (c) The robot executes the reaching trajectory and closes the hand when reaching the target poses closely, and then returns to the default pose with the door opened. Note that while the door is heavy, HERO successfully manages to open the door with a smooth and stable door motion.

Tab. 6 shows systematic bias in analytical FK for both EE pose and base odometry. FK corrections, through kinematic calibration or our residual model, reduce this bias. Ours achieves lower errors on 2 of 4 metrics, possibly because the neural model may be able to capture non-geometric effects (*e.g.*, elasticity [51, 53]). While we use the neural model for HERO, this experiment suggests that the calibrated model would also offer improvements over the default calibration.

5.3 Failure Mode Analysis

Fig. 9 shows two common failures: *object slipping* on large or irregular objects, and *object knock-over* when grasp orientation or hand size contacts unstable objects such as standing books.

5.4 Extending HERO to Other Tasks Like Door Opening

As HERO constructed a modular system that coordinates high-level planning and low-level end-effector control, it reveals a possibility of extending HERO to broader tasks like door opening, which is a challenging loco-manipulation task [78–80]. In Fig. 10, we directly employ HERO to identify the target grasping pose for `fridge door handle`, followed by the same pipeline as object grasping, enabling the humanoid to grasp the fridge handle and finally open it when returning to the default position. Note that the door requires a large force to open because of magnetic attraction; we leave the door unlatched beforehand. This result shows *HERO*'s modular potential, and it is also possible to extend our system to broader loco-manipulation tasks by incorporating off-the-shelf trajectory generation frameworks [13, 81–83].

6 Discussions and Limitations

We present a modular humanoid system that separates open-vocabulary *action planning* from simulator-trained *action controller*, achieving 90% success on daily-object grasping without large-scale real-world imitation. This decomposition lets large vision models handle visual generalization while a learned motion tracker executes whole-body reaching. The experiments show two lessons: task-space state must be accurate (analytical FK has 1.30cm EE error while ours reaches 0.50cm), and explicit error feedback during closed-loop execution effectively reduces real-world drifts (our tracker reaches 2.5cm error). Remaining limitations include restricted egocentric FoV for far or high targets, motivating active neck control and visual servoing [41, 84, 85]; inefficient motions, motivating learned motion priors; and modular failures [14, 83] plus dexterity limits, motivating better hands and retargeting.

Acknowledgement

We are grateful to the Coordinated Science Laboratory and the Center for Autonomy for access to experimental space and the G1 humanoid robot used in this work. We thank John Hart for his invaluable help with the humanoid robots. This material is based upon work supported by an NSF CAREER Award (IIS-2143873).

References

[1] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11691–11698. IEEE, 2026.

[2] Q. Liao, T. E. Truong, X. Huang, Y. Gao, G. Tevet, K. Sreenath, and C. K. Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11 (117):eadx8924, 2026.

[3] X. He, R. Dong, Z. Chen, and S. Gupta. Learning Getting-Up Policies for Real-World Humanoid Robots. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[4] T. Huang, J. Ren, H. Wang, Z. Wang, Q. Ben, M. Wen, X. Chen, J. Li, and J. Pang. Learning Humanoid Standing-up Control across Diverse Postures. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[5] T. He, J. Gao, W. Xiao, Y. Zhang, Z. Wang, J. Wang, Z. Luo, G. He, N. Sobanbabu, C. Pan, Z. Yi, G. Qu, K. Kitani, J. K. Hodgins, L. Fan, Y. Zhu, C. Liu, and G. Shi. ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[6] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. In *8th Annual Conference on Robot Learning*, 2024.

[7] Helix: A vision-language-action model for generalist humanoid control, 2025. URL https://www.figure.ai/news/helix.

[8] Tesla. Artificial intelligence & autopilot, 2021. URL https://www.tesla.com/AI.

[9] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. M. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024.

[10] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint arXiv:2507.05331*, 2025.

[11] T. He, Z. Wang, H. Xue, Q. Ben, Z. Luo, W. Xiao, Y. Yuan, X. Da, F. Castañeda, S. Sastry, et al. Viral: Visual sim-to-real at scale for humanoid loco-manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13430–13441, 2026.

[12] M. Dalal, M. Liu, W. Talbott, C. Chen, D. Pathak, J. Zhang, and R. Salakhutdinov. Local policies enable zero-shot long-horizon manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13875–13882. IEEE, 2025.

[13] Z. Qi, W. Zhang, Y. Ding, R. Dong, X. Yu, J. Li, L. Xu, B. Li, X. He, G. Fan, J. Zhang, J. He, J. Gu, X. Jin, K. Ma, Z. Zhang, H. Wang, and L. Yi. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[14] P. Liu, Y. Orru, J. Vakil, C. Paxton, N. M. M. Shafiullah, and L. Pinto. Demonstrating ok-robot: What really matters in integrating open-knowledge models for robotics. In D. Kulic, G. Venture, K. E. Bekris, and E. Coronado, editors, *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024.

[15] Z. Luo, Y. Yuan, T. Wang, C. Li, S. Chen, F. Castañeda, Z.-A. Cao, J. Li, D. Minor, Q. Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *arXiv preprint arXiv:2511.07820*, 2025.

[16] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[17] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. S. Coll-Vinent, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. HAZRA, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollar, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. SAM 3: Segment anything with concepts. In *The Fourteenth International Conference on Learning Representations*, 2026.

[18] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 39(5):3929–3945, 2023.

[19] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143, 2018.

[20] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pages 8944–8951. IEEE, 2024.

[21] X. Cheng, Y. Ji, J. Chen, R. Yang, G. Yang, and X. Wang. Expressive Whole-Body Control for Humanoid Robots. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[22] Z. Qi, X. Chen, J. Wang, C. Lin, Y. Lian, W. Zhang, X. Yu, H. Wang, and L. Yi. Humanoid generative pre-training for zero-shot motion tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20834–20844, June 2026.

[23] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control. In *RSS 2025 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2025.

[24] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9989–9996. IEEE, 2025.

[25] J. Li, X. Cheng, T. Huang, S. Yang, R.-Z. Qiu, and X. Wang. AMO: Adaptive Motion Optimization for Hyper-Dexterous Humanoid Whole-Body Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[26] T. Portela, A. Cramariuc, M. Mittal, and M. Hutter. Whole-body end-effector pose tracking. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11205–11211. IEEE, 2025.

[27] Z. Chen, M. Ji, X. Cheng, X. Peng, X. B. Peng, and X. Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025.

[28] Z. Zhang, C. Chen, H. Xue, J. Wang, S. Liang, Y. Liu, Z. Zhang, H. Wang, and L. Yi. Unleashing humanoid reaching potential via real-world-ready skill space. *IEEE Robotics and Automation Letters*, 11(2):2082–2089, 2025.

[29] Z. Cao, L. Yan, Y. Zhang, S. Chen, J. Ma, T. Zhan, S. Fu, Y. Jia, C. Lu, and Y. Gao. Hiwet: Hierarchical world-frame end-effector tracking for long-horizon humanoid loco-manipulation. *arXiv preprint arXiv:2602.06341*, 2026.

[30] X. Luo, X. Chen, X. Yin, H. Wu, B. Xia, Z. Chen, J. Li, B. Chen, and X. Cheng. Ceer: Compliant end-effector and root control as a unified interface for hierarchical humanoid loco-manipulation. *arXiv preprint arXiv:2605.19981*, 2026.

[31] J. Ren, Y. Li, K. Zhang, P. Fu, H. Jiang, Y. Pan, G. Zeng, T. Huang, W. Guo, P. Lu, et al. Smash: Mastering scalable whole-body skills for humanoid ping-pong with egocentric vision. *arXiv preprint arXiv:2604.01158*, 2026.

[32] Y. Li, Y. Zhang, W. Xiao, C. Pan, H. Weng, G. He, T. He, and G. Shi. Hold my beer: Learning gentle humanoid locomotion and end-effector stabilization control. In *Conference on Robot Learning*, pages 4506–4523. PMLR, 2025.

[33] Y. Zhang, Y. Yuan, P. Gurunath, I. Gupta, S. Omidshafiei, A.-a. Agha-mohammadi, M. Vazquez-Chanlatte, L. Pedersen, T. He, and G. Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. In *8th Learning for Dynamics & Control Conference*, 2026.

[34] J. Jang, Z. Wang, Z. Zhou, F. Wu, and Y. Zhao. Seec: Stable end-effector control with model-enhanced residual learning for humanoid loco-manipulation. *arXiv preprint arXiv:2509.21231*, 2025.

[35] C. Stanton, A. Bogdanovych, and E. Ratanasena. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In *Proc. Australasian Conference on Robotics and Automation*, volume 8, page 51, 2012.

[36] O. Porges, M. Connan, B. Henze, A. Gigli, C. Castellini, and M. A. Roa Garzon. A wearable, ultralight interface for bimanual teleoperation of a compliant, whole-body-controlled humanoid robot. In *2019 International Conference on Robotics and Automation, ICRA 2019*. IEEE, 2019.

[37] Q. Ben, F. Jia, J. Zeng, J. Dong, D. Lin, and J. Pang. HOMIE: Humanoid Loco-Manipulation with Isomorphic Exoskeleton Cockpit. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[38] Y. Ze, Z. Chen, J. P. Araujo, Z.-a. Cao, X. B. Peng, J. Wu, and K. Liu. Twist: Teleoperated whole-body imitation system. In J. Lim, S. Song, and H.-W. Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 2143–2154. PMLR, 27–30 Sep 2025.

[39] F. Liu, Z. Gu, Y. Cai, Z. Zhou, H. Jung, J. Jang, S. Zhao, S. Ha, Y. Chen, D. Xu, et al. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *IEEE Robotics and Automation Letters*, 2025.

[40] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5364–5371. IEEE, 2025.

[41] L. Wei, X. Peng, R.-Z. Qiu, X. Cheng, and X. Wang. Hmc: Learning heterogeneous meta-control for contact-rich loco-manipulation. In *RSS 2025 Workshop on Whole-body Control and Bimanual Manipulation: Applications in Humanoids and Beyond*, 2025.

[42] T. Wu, X. Kong, Y. Chen, Q. Yu, H. Ye, J. Li, Y. Wang, and H. Dong. Sugar: A scalable human-video-driven generalizable humanoid loco-manipulation learning framework. *arXiv preprint arXiv:2605.20373*, 2026.

[43] K. Lin, A. Mandlekar, C. R. Garrett, N. Chernyadev, Y. Fang, R. Ding, Y. Xie, J. Tran, L. Fan, and Y. Zhu. Humanoidmimicgen: Data generation for loco-manipulation via whole-body planning, 2026.

[44] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song. Umi-on-legs: Making manipulation policies mobile with manipulation-centric whole-body controllers. In *Conference on Robot Learning*, pages 5254–5270. PMLR, 2025.

[45] M. Liu, Z. Chen, X. Cheng, Y. Ji, R.-Z. Qiu, R. Yang, and X. Wang. Visual whole-body control for legged loco-manipulation. In *Conference on Robot Learning*, pages 234–257. PMLR, 2025.

[46] S. Yin, Y. Ze, H.-X. Yu, C. K. Liu, and J. Wu. Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation. *arXiv preprint arXiv:2509.20322*, 2025.

[47] Z. Liang, Y.-L. Wei, X. Chen, M. Lin, Y.-X. He, Z. Luo, J.-H. Liu, K.-Y. Lin, and W.-S. Zheng. Humanoid whole-body manipulation via active spatial brain and generalizable action cerebellum. *arXiv preprint arXiv:2605.21133*, 2026.

[48] Z. Hu, Z. Xu, D. Chang, H. Yin, L. Tran, R. Martín-Martín, P. Stone, J. Qiao, and J. Biswas. Vofa: Visual object goal pushing with force-adaptive control for humanoids. *arXiv preprint arXiv:2605.01518*, 2026.

[49] S. Wei, H. Jing, B. Li, Z. Zhao, J. Mao, Z. Ni, S. He, J. Liu, X. Liu, K. Kang, et al. ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026.

[50] B. Chen, Y. Chen, L. Qiu, J. Bai, Y. Ge, and Y. Ge. Unit: Toward a unified physical language for human-to-humanoid policy learning and world modeling. *arXiv preprint arXiv:2604.19734*, 2026.

[51] M. Filipović, V. Potkonjak, and M. Vukobratović. Elasticity in humanoid robotics. *Scientific Technical Review, Military Technical Institute, Belgrade*, 1:24–33, 2007.

[52] J. Tenhumberg and B. Bäuml. Calibration of an elastic humanoid upper body and efficient compensation for motion planning. In *20th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2021, Munich, Germany, July 19-21, 2021*, pages 98–103. IEEE, 2021.

[53] J. Tenhumberg, D. Winkelbauer, D. Burschka, and B. Bäuml. Self-contained calibration of an elastic humanoid upper body using only a head-mounted RGB camera. In *21st IEEE-RAS International Conference on Humanoid Robots, Humanoids 2022, Ginowan, Japan, November 28-30, 2022*, pages 702–707. IEEE, 2022.

[54] W. Yu, C. K. Liu, and G. Turk. Policy transfer with strategy optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[55] W. Yu, J. Tan, Y. Bai, E. Coumans, and S. Ha. Learning fast adaptation with meta strategy optimization. *IEEE Robotics Autom. Lett.*, 5(2):2950–2957, 2020.

[56] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: rapid motor adaptation for legged robots. In *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021.

[57] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In K. Liu and D. K. andD Jeffrey Ichnowski, editors, *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, volume 205 of *Proceedings of Machine Learning Research*, pages 138–149. PMLR, 2022.

[58] N. Fey, G. B. Margolis, M. Peticco, and P. Agrawal. Bridging the Sim-to-Real Gap for Athletic Loco-Manipulation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[59] H. Wang, H. Luo, W. Zhang, and H. Chen. CTS: concurrent teacher-student reinforcement learning for legged locomotion. *IEEE Robotics Autom. Lett.*, 9(11):9191–9198, 2024.

[60] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018.

[61] J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Sci. Robotics*, 4(26), 2019.

[62] Unitree. Unitree G1: Humanoid Agent AI Avatar. 2024. URL https://www.unitree.com/g1.

[63] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023.

[64] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[65] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019.

[66] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[67] Optitrack. URL https://optitrack.com/.

[68] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

[69] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 2002.

[70] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[71] R. Y. Tsai, R. K. Lenz, et al. A new technique for fully autonomous and efficient 3 d robotics hand/eye calibration. *IEEE Transactions on robotics and automation*, 5(3):345–358, 1989.

[72] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac gym: High performance GPU based physics simulation for robot learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.

[73] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE, 2012.

[74] K.-J. Wang, Y.-H. Liu, H.-T. Su, J.-W. Wang, Y.-S. Wang, W. Hsu, and W.-C. Chen. OCID-ref: A 3D robotic dataset with embodied language for clutter scene grounding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5333–5338. Association for Computational Linguistics, 2021. doi:10.18653/v1/2021.naacl-main.419.

[75] Z. Roth, B. Mooring, and B. Ravani. An overview of robot calibration. *IEEE Journal on Robotics and Automation*, 3(5):377–385, 1987.

[76] D. Huczala, T. Kot, J. Mlotek, J. Suder, and M. Pfurner. An automated conversion between selected robot kinematic representations. In *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 47–52. IEEE, 2022.

[77] G. Franzese, M. Spahn, J. Kober, J. Alonso-Mora, and C. Della Santina. Accurate and affordable cobot calibration without external measurement devices. *Communications Engineering*, 2026.

[78] A. J. Schmid, N. Gorges, D. Goger, and H. Worn. Opening a door with a humanoid robot using multi-sensory tactile feedback. In *2008 IEEE international conference on robotics and automation*, pages 285–291. IEEE, 2008.

[79] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orlowski. The darpa robotics challenge finals: Results and perspectives. In *The DARPA robotics challenge finals: Humanoid robots to the rescue*, pages 1–26. Springer, 2018.

[80] H. Xue, T. He, Z. Wang, Q. Ben, W. Xiao, Z. Luo, X. Da, F. Castañeda, G. Shi, S. Sastry, et al. Opening the sim-to-real door for humanoid pixel-to-action policy transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6642–6652, 2026.

[81] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9488–9495. IEEE, 2024.

[82] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *Conference on Robot Learning*, pages 4573–4602. PMLR, 2025.

[83] A. Gupta, M. Zhang, R. Sathua, and S. Gupta. Demonstrating MOSART: Opening Articulated Structures in the Real World. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[84] H. Qi, Y.-J. Wang, T. Lin, B. Yi, Y. Ma, K. Sreenath, and J. Malik. Coordinated humanoid manipulation with choice policies. *arXiv preprint arXiv:2512.25072*, 2025.

[85] Y. Ze, S. Zhao, W. Wang, A. Kanazawa, R. Duan, P. Abbeel, G. Shi, J. Wu, and C. K. Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[86] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[87] Y. Chen, Z. Qi, W. Zhang, X. Jin, L. Zhang, and P. Liu. Reasoning in space via grounding in the world. In *The Fourteenth International Conference on Learning Representations*, 2026.

[88] T. Ren, Q. Jiang, S. Liu, Z. Zeng, W. Liu, H. Gao, H. Huang, Z. Ma, X. Jiang, Y. Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

[89] H.-S. Fang, H. Yan, Z. Tang, H. Fang, C. Wang, and C. Lu. Anydexgrasp: General dexterous grasping for different hands with human-level learning efficiency. *arXiv preprint arXiv:2502.16420*, 2025.

[90] A. Tabb and K. M. Ahmad Yousef. Solving the robot-world hand-eye (s) calibration problem with iterative methods. *Machine Vision and Applications*, 28(5):569–590, 2017.

[91] M. Shah, R. D. Eastman, and T. Hong. An overview of robot-sensor calibration methods for evaluation of perception systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 15–20, 2012.

[92] G. Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[93] R. Y. Tsai. Efficient and accurate camera calibration technique for 3d machine vision. In *IEEE conference on computer vision and pattern recognition*, 1985.

[94] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin, et al. Isaac lab: A GPU-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

[95] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[96] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

APPENDIX

A Additional Experimental Analysis **17**

A.1 Language Sensitivity 17

A.2 Tracking Error Distribution Analysis 18

A.3 Moving Object Grasping with Visual Replanning 18

A.4 Field of View Analysis 18

A.5 Visual Perception Details 20

A.6 Analytical FK Error Visualization 21

A.7 Tracking Error Curves with and without Replanning 21

A.8 Whole-Body Reaching Workspace Analysis 22

B Additional Implementation Details **23**

B.1 MOCAP Setup 23

B.2 Onboard Egocentric RGB-D Camera 24

B.3 Hyper-parameters 24

B.4 Rewards 24

B.5 Policy Training 26

B.6 Deployment Hardware 26

B.7 Testing Assets Details 27

B.8 Testing Scenes Details 27

A Additional Experimental Analysis

A.1 Language Sensitivity

We evaluate target detection on OCID-Ref images [74] using 1,450 language queries; Tab. 7 reports accuracy for selected query types.

Table 7: OCID-Ref query sensitivity. Detection accuracy on our 1,450-query evaluation (58 classes, five views, five query types). Selected types are shown; overall includes all five.

Query type	Detection accuracy (%)
Canonical name	73.8
Canonical name + attribute	86.6
Canonical name + attribute + spatial relation	89.7
Overall (all five query types)	82.6

Fig. 11 illustrates red- and green-apple selection among distractors in robot grasping.

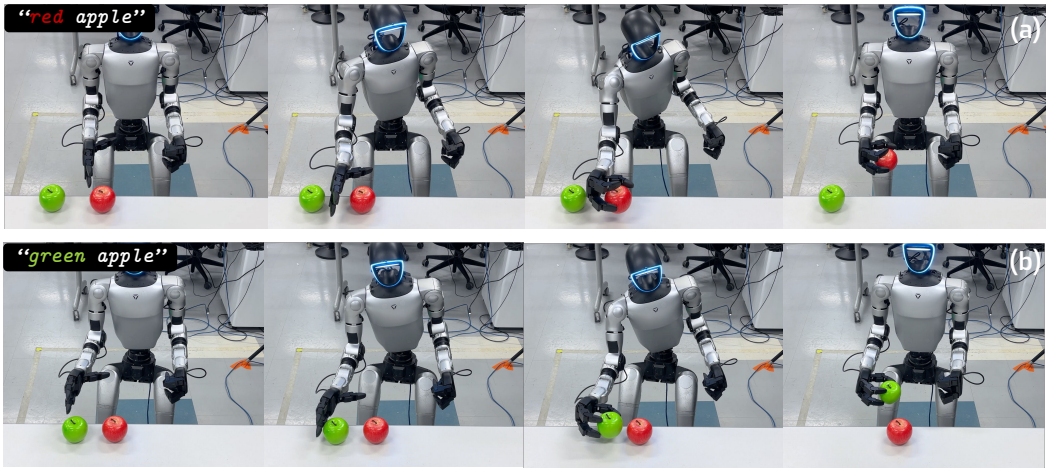


Figure 11: HERO is able to distinguish the target object via language queries. (a) Picking up a `red apple`; (b) Picking up a `green apple` instead of `red apple`.

A.2 Tracking Error Distribution Analysis

We visualize the CDFs of EE tracking errors in Fig. 12.

Top row. HERO dominates all baselines: at 80%, HERO achieves 3.9 cm / 6.7° (pos/rot), versus 20.9 cm / 25.3° for FALCON and 9.8 cm / 19.5° for AMO. At 90%, HERO remains below 4.6 cm and 8.2°, indicating strong tail robustness.

Bottom row. Replanning (Sec. 3.3) is critical: at 80%, HERO achieves 3.1 cm vs. 6.3 cm without replanning (2.1× worse), and at 90% 3.4 cm vs. 7.1 cm. Rotation gains are smaller but consistent (median 13.6° vs. 16.1°).

A.3 Moving Object Grasping with Visual Replanning

HERO derives target grasping poses from vision and language queries, enabling the robot to capture moving objects through closed-loop replanning. Fig. 13(a-b) illustrates this dynamic adaptation: while the system initially generates an EE trajectory based on the first vision perception, it re-estimates the pose as the object moves. This visual feedback triggers the update of the target grasp, allowing the robot to seamlessly adjust its trajectory and successfully secure the moving object. Note that, in these two trials, the robot successfully sees the object after moving, but the object can only be seen at the corner due to the rather limited field of view.

A.4 Field of View Analysis

Our system uses the onboard camera for visual perception, which, however, has a limited field of view. As shown in Fig. 14, the robot first stands randomly at a distance of 1.28m from the object, while the target object (`stapler`) is not visible at this distance; after walking forward under a consistent velocity command to a distance of 0.6m and continuously detecting the object, the egocentric view successfully captures the target object, which makes the robot stops at about 0.5m from the object. Then the robot coordinates the whole-body reaching motion and successfully grasps the object. This visualization indicates that the robot's onboard egocentric view is limited, and the robot can only see the object within a short distance (< 0.6m), which makes the availability of the 3D spatial understanding beforehand [14, 86, 87] critical for searching objects, which could be a future exploration.

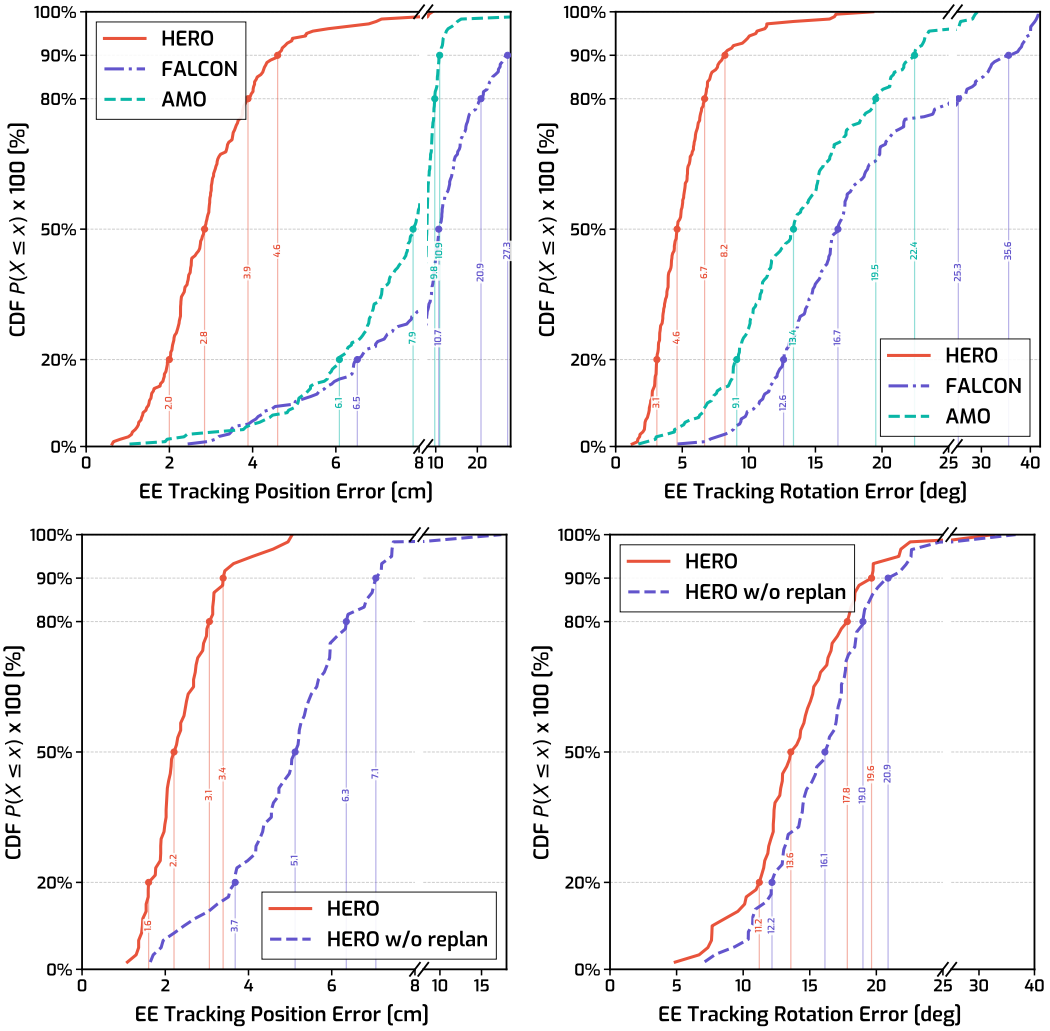


Figure 12: CDF analysis of end-effector tracking errors. Top row: Comparison of translation (left) and rotation (right) error distributions across HERO, FALCON [33], and AMO [25] for all table heights. The steeper curves of HERO indicate consistently lower errors and tighter distributions. Bottom row: Ablation study showing translation (left) and rotation (right) error distributions with and without replanning (Sec. 3.3). The steeper CDF curves with replanning demonstrate their significant contribution to tracking accuracy.

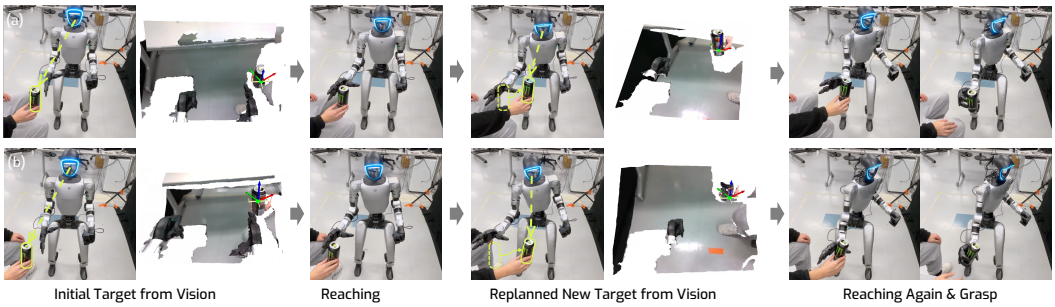


Figure 13: HERO enables a humanoid to grasp a moving object via visual closed-loop replanning. (a-b) Two examples of visual closed-loop replanning. The goal is to grasp `black can`.

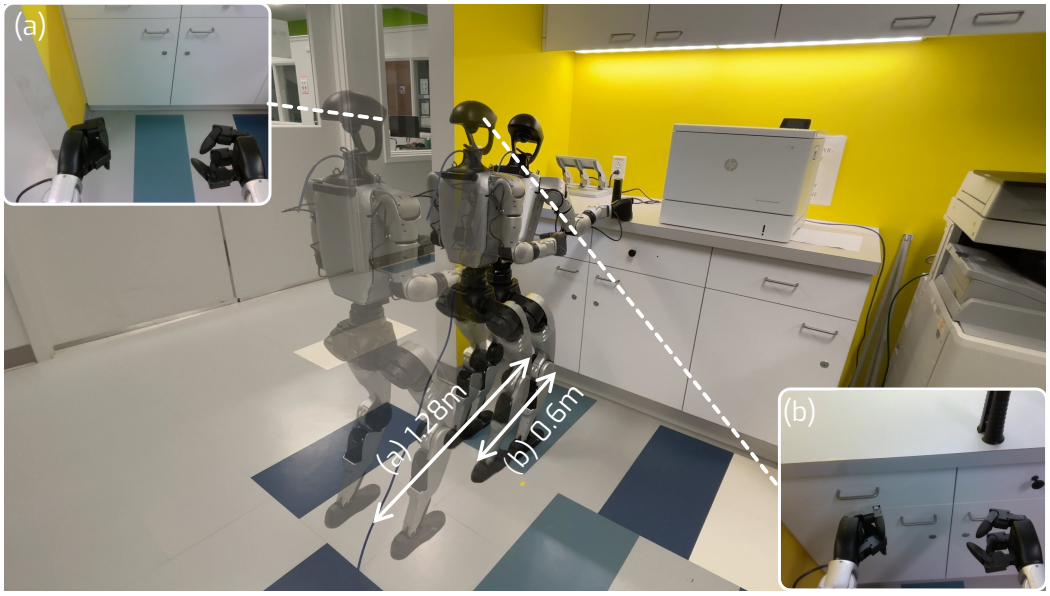


Figure 14: Field of view (FoV) visualization using HERO. We let the robot stand at a random distance from the object (*e.g.*, 1.28m), and the robot keeps walking forward until the target object (`stapler`) is detected. After successfully detecting the target object, the robot stops walking and grasps the object via whole-body coordination. The robot can only see the object within 0.6m, which makes it hard for the robot to search for the object in a random room.

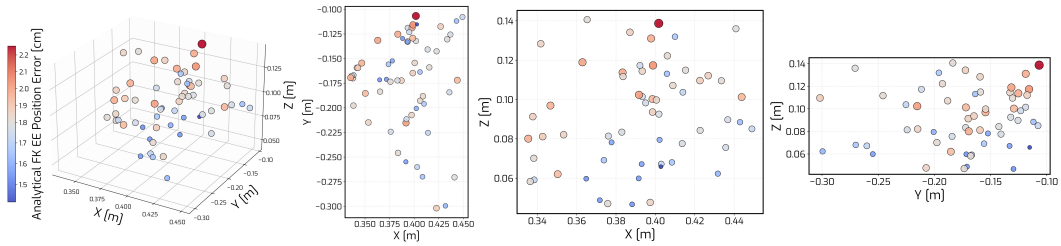


Figure 15: Visualization of analytical forward kinematics error. We plot the 60 data points collected in the MOCAP room. The error is indicated in the color bar on the left side of the figure, and the size of the scatter also increases with the error.

A.5 Visual Perception Details

Fig. 16 illustrates how HERO leverages LVMs to obtain the targeted EE grasping pose, following a modular perception-to-action design that is similar to prior systems [13, 14].

Object of Interest from Language. Given an ego-centric RGB-D observation and a natural-language query specifying the target object, HERO first applies GroundingDINO to produce a language-conditioned detection box [88]. The detected box is then used to prompt SAM-3 for segmentation of the object of interest [17].

Grasp Proposals. This mask serves as a spatial constraint for grasp proposal generation: HERO runs AnyGrasp [18] to produce a set of candidate grasps, and then filters out proposals outside the segmented object region. Note that AnyDexGrasp [89] can also be used here, but we find that the Dex-3 hand lacks dexterity, and the difference between these two methods is limited.

Grasp Selection. To select the best jaw grasp, we first filter out the grasp poses that lie on the opposite side of the object relative to the robot's hand (*e.g.*, for an object to the right side of the hand,

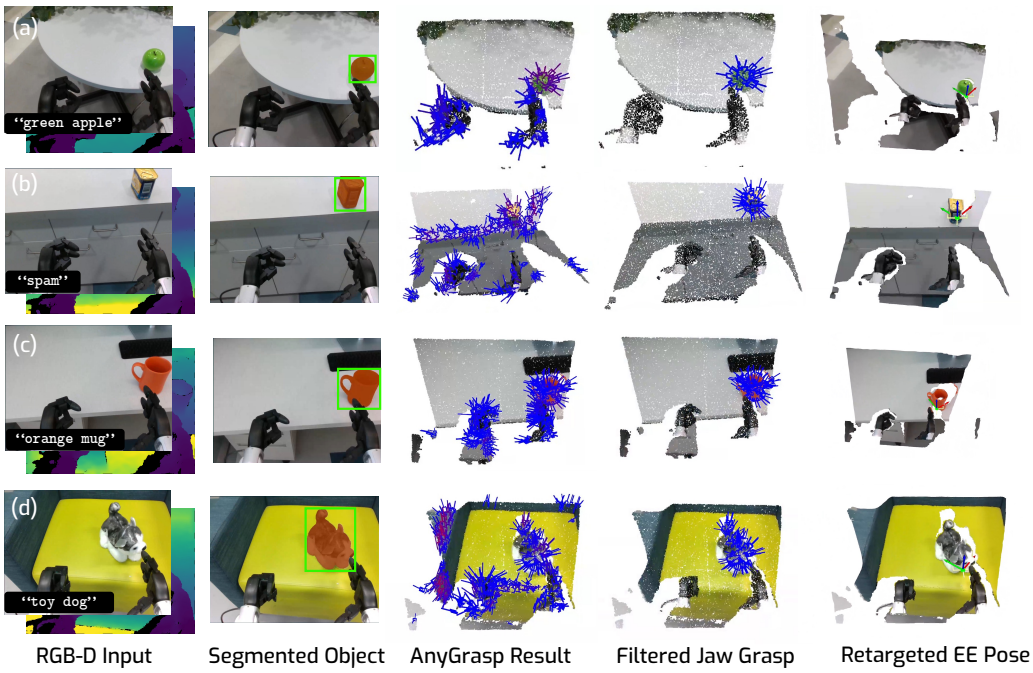


Figure 16: Visual perception for language-conditioned grasping. Examples of target segmentation, grasp proposals, filtering, and retargeted EE poses from onboard RGB-D observations.

the left approaching grasps are abandoned). Then we filter out grasps that are too high or too low based on a gravity-aligned height estimation of objects using depth. Afterward, we select the grasp that lies most parallel to the ground with the highest confidence as the final grasp.

Grasp Retargeting. The selected grasp is retargeted to a 6-DoF end-effector pose for dexterous grasping with the Dex-3 hand. We first rotate the gripper pose by 45 degrees around the z-axis to improve the grasp robustness and pose error tolerance. After that, we clip the yaw angle within 70 degrees to ensure the orientation is not too large.

Approach-angle sweep. In a preliminary $0°$–$90°$ sweep, wrist distortion increased from $22°$ to $104°$ as the angle increased, while the observed object knock-over rate fell from 45.1% at $0°$ to 7.4% at $45°$. We use the $45°$ offset as a compromise between these effects.

A.6 Analytical FK Error Visualization

In Fig. 15, we visualize the translation error of analytical forward kinematic results. We plot the error via the collected 60 samples in the MOCAP room, where the error is recorded when time is 1 minute. From the figure, we can observe that the error generally increases when the EE location becomes larger along the Y and Z axes, which may form a pattern that can be learned from a neural model.

A.7 Tracking Error Curves with and without Replanning

Fig. 17 shows how end-effector tracking error evolves over the course of a real-world reach, complementing the aggregate ablations in Tab. 5. Without replanning, the error plateaus at a higher level because the policy commits to a stale reference; with replanning, the error continues to decrease as fresh references compensate for accumulated drift. The fact that HERO with learned η, ξ closely tracks the MOCAP-oracle curve also confirms that our learned forward models suffice for accurate real-world deployment.

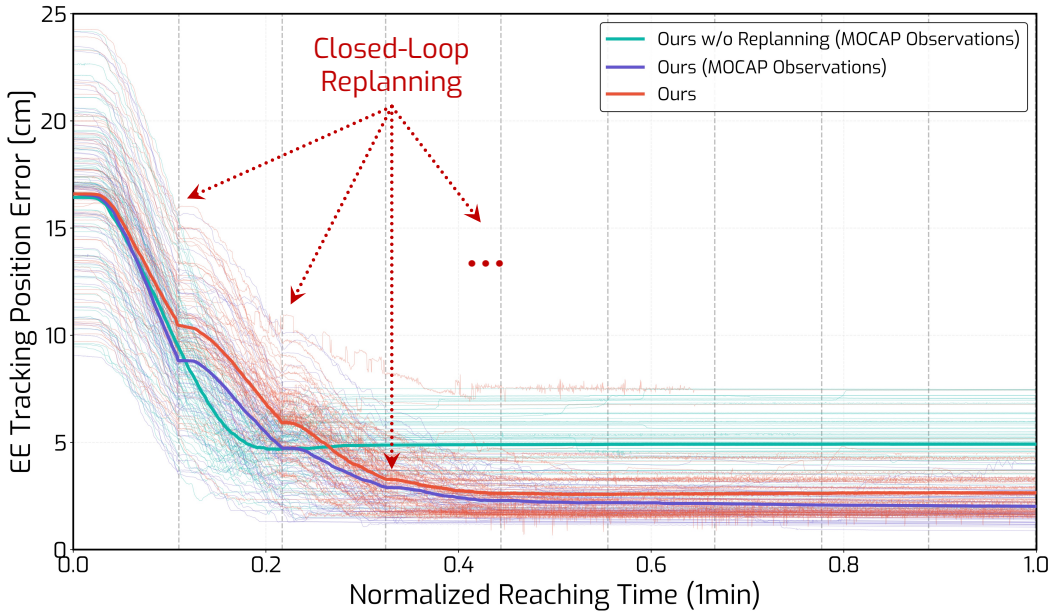


Figure 17: Impact of replanning on end-effector tracking error in the real world. We plot the end-effector translation error as a function of execution time steps. The plot shows 1 minute of execution at 50Hz. The transparent lines are individual 60 real-world rollouts, and the corresponding solid line indicates the average value. The gray vertical dashlines indicate the replanning every 6 seconds (0.15Hz). Cyan line shows HERO without replanning and while purple line shows HERO with re-planning. Re-planning leads to more accurate tracking. Orange line uses end-effector estimates from our neural model which leads to tracking performance very close to the oracle purple line that uses end-effector estimates from MOCAP.

A.8 Whole-Body Reaching Workspace Analysis

We quantify how enabling torso motion via the waist DoFs affects end-effector reachability. We compare two kinematic settings: *i) arms-only*, where IK optimizes the 14-DoF arm joints, and *ii) arms+waist*, where IK additionally optimizes the 3-DoF waist (17 DoFs total).

Workspace estimation. We define an axis-aligned 3D candidate region in the robot base frame:

$$x \in [0, 1.0]\,\mathrm{m}, \quad y \in [-1.0, 1.0]\,\mathrm{m}, \quad z \in [-0.5, 1.0]\,\mathrm{m}, \tag{5}$$

discretized at $0.02\,\mathrm{m}$ resolution. For each grid point $\mathbf{p}$ we attempt IK with cuRobo [63] under joint-limit constraints; a point is *reachable* if the solver converges within a fixed iteration budget and the EE position residual lies below a preset tolerance. The workspace volume is approximated by voxel counting,

$$V \approx N_{\mathrm{reach}} \cdot (0.02)^3, \tag{6}$$

where N_{reach} is the number of reachable points.

Effect of waist DoFs. Tab. 8 reports the resulting volumes. Enabling the waist substantially enlarges reachability: the combined two-arm workspace grows from $0.248\,\mathrm{m}^3$ to $0.523\,\mathrm{m}^3$ ($\sim 2.1\times$), and the single-arm workspace grows from $0.166\,\mathrm{m}^3$ to $0.426\,\mathrm{m}^3$. Bending and twisting the torso effectively repositions the shoulder frame, allowing the EE to cover farther-forward and lower-height targets that are infeasible with a non-actuated waist.

Workspace showcase. Fig. 18 illustrates HERO retrieving objects distributed across an expansive tabletop workspace. Every object is positioned beyond $0.4\,\mathrm{m}$ from the robot base, so the task requires whole-body coordination to reach precisely while remaining stable. HERO composes expressive whole-body motion with the precision needed for successful grasping.

Table 8: Reachable workspace volume across configurations.

Configuration	Single Arm (m^3)	Both Arms (m^3)
Arms-only (14 DoFs)	0.166	0.248
Arms+Waist (17 DoFs)	**0.426**	**0.523**

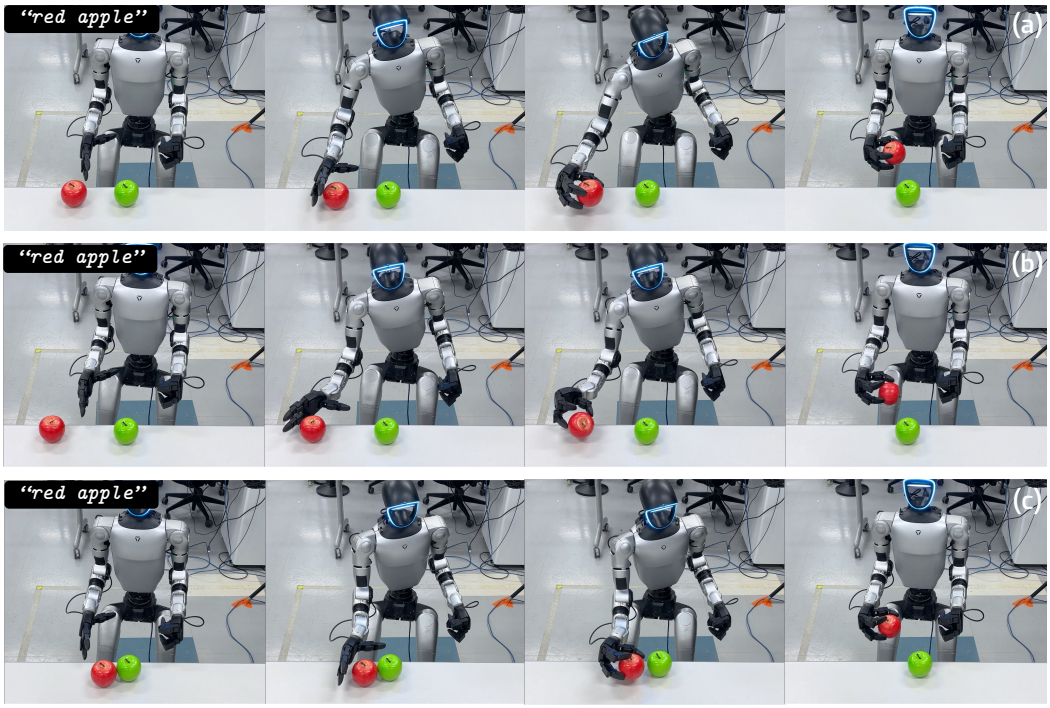


Figure 18: HERO enables a humanoid picking up objects from a standard table (0.74m) across a large workspace with open-vocabulary queries. (a-c) The robot can reach and pick up a `red apple` placed at different heights, poses, and locations.

B Additional Implementation Details

B.1 MOCAP Setup

MOCAP System. We use the modern MOCAP system Optitrack [67] with 13 cameras which provides $\leq$ 0.2mm measure accuracy.

Robot Link Pose. To obtain the end-effector pose in the robot frame, we put several markers onto both links, and we show markers on the hand in Fig. 6(a). Although the MOCAP system provides constructed asset poses via selected marker groups, there exists a misalignment between the MOCAP asset frame and the robot link frame. To address this, we carefully measure each marker's relative offset to the link's origin, followed by the Kabsch-Umeyama (KU) algorithm [68, 69] that transforms individual marker coordinates into 6-DoF link pose in the MOCAP frame within $<$ 1.5mm RMSE error. The relative transformation of EE and the robot base is thus obtained as they are all in the MOCAP frame. This approach ensures an accurate measurement of both the end-effector and the robot base, setting a solid ground for our evaluation and camera calibration, introduced next.

Training-data quantity. In a separate subsampling study, retraining the learned FK model with 100%, 50%, 25%, and 10% of a two-hour MOCAP dataset yielded mean EE translation errors of 0.51, 0.61, 0.68, and 0.70 cm, respectively. Base odometry remained usable with 25% of the data (about 30 minutes).

B.2 Onboard Egocentric RGB-D Camera

Setup. We use the onboard RGB-D camera D435i mounted on the humanoid's head, as shown in Fig. 6(b). The humanoid's neck features a `pitch` degree of freedom enabling head rotation within a limited range, necessitating precise camera calibration for accurate 3D perception.

Calibration with MOCAP. Standard hand-eye calibration [90, 91] typically relies on analytical forward kinematics to obtain end-effector poses. However, as shown in Tab. 6, analytical forward kinematics has a mean EE translation error of 1.30 cm due to hardware inaccuracies, limiting its suitability for precise camera calibration.

We instead leverage the MOCAP system for ground-truth pose measurement. Following the marker-based approach described previously, we attach reflective markers to an ArUco calibration board [70] and apply the KU algorithm [68, 69] for 6-DoF pose estimation. During data collection, we manually move the board through 60-70 diverse poses in front of the camera. For each pose i, we record: 1) the robot base pose in MOCAP frame $\mathcal{T}^{\text{base}}_{\text{MOCAP}}$, 2) the board pose in MOCAP frame $\mathcal{T}^{\text{board},i}_{\text{MOCAP}}$, and 3) the board pose in camera frame $\mathcal{T}^{\text{board},i}_{\text{camera}}$ via ArUco detection using the OpenCV library [92].

To compute the camera-to-base transformation $\mathcal{T}^{\text{camera}}_{\text{base}}$, we solve the eye-to-hand calibration problem:

$$\mathcal{T}^{\text{camera}}_{\text{base}} \oplus \mathcal{T}^{\text{board},i}_{\text{camera}} = \mathcal{T}^{\text{base}}_{\text{MOCAP}} \ominus \mathcal{T}^{\text{board},i}_{\text{MOCAP}} \tag{7}$$

using the Tsai-Lenz method [71, 93]. This MOCAP-assisted calibration achieves a reprojection error within 2.5mm, ensuring accurate egocentric 3D perception.

Image Resolution & FPS. We use the RGB-D images with a resolution of 640×480 in a 60Hz FPS.

B.3 Hyper-parameters

Motion planning For motion planning, we use cuRobo and set the planning dt to 7.25e-6.

Grasping Threshold When the robot approaches the object, it autonomously close the hand when the hand distance to the target grasp $\boldsymbol{\Delta\mathcal{E}}_t \leq \delta$ where $\delta > 0$ is a threshold. At the moment when this threshold is reached, we pass the same local waypoint of the planned motion trajectory to the policy to ensure stability, and the hand is immediately closed for grasping. In this paper, we utilize a threshold of $\delta = 1.5$cm, which we find most effective across tested objects.

B.4 Rewards

Tab. 9 summarizes reward components and weights used for RL training of π_t, which is structured into four categories: *tracking task*, *penalties*, *regularization*, and *locomotion task*. To ensure precise manipulation, the tracking rewards weigh the alignment of the end-effector based on our newly proposed residual $\boldsymbol{\Delta\mathcal{E}}_t$. Note that EE orientation is represented with the continuous 6D parameterization (first two columns of the rotation matrix) [66]. To encourage the planned upper-body posture (*e.g.*, waist bending or torso twisting), we also add a joint-space tracking term. Penalties strictly enforce safety constraints (*e.g.*, joint limits, termination), while regularization terms—such as costs on torque, acceleration, and stance symmetry—are essential for generating smooth, stable motions capable of robust and natural Sim2Real transfer. To train the robot to follow locomotion commands, we also use a flag variable to control the standing and waking mode switching.

Table 9: Reward terms and nominal weights. $m = 1$ denotes walking ($m = 0$ standing); h_c is commanded base height, $\mathbf{g}_{xy}$ projected gravity, and e_p, e_R EE position and mean axis-angle errors. Each weight multiplies the control-step duration; † terms also use a penalty scale $s \in [0, 1]$ initialized at 0.1. Here $[x]_+ = \max(x, 0)$, and soft position bounds cover the central 95% of each joint range.

TERM	EXPRESSION	WEIGHT
Tracking task		
EE position (stance)	$(1-m)\exp(-\lVert e_p\rVert_2^2/0.05)$	2.0
EE orientation (stance)	$(1-m)\exp(-e_R/0.10)$	2.0

Continued on next page

Table 9 (continued)

TERM	EXPRESSION	WEIGHT
Upper-body DoF	$\exp(-\|\mathbf{q}_{\text{arm}} - \mathbf{q}^{\text{ref}}_{\text{arm}}\|_2^2/0.1)$	4.0
Base height (stance)	$(1-m)\exp(-\|h - h_c\|/0.05)$	4.0
Base height (walk)	$m\exp(-\|h - h_c\|/0.05)$	1.0
Waist DoF (stance)	$(1-m)\exp(-\|\mathbf{q}_{\text{waist}} - \mathbf{q}^c_{\text{waist}}\|_2^2/0.05)$	3.0
Waist DoF (walk)	$m\exp(-\|\mathbf{q}_{\text{waist}} - \mathbf{q}^c_{\text{waist}}\|_2^2/0.05)$	0.5
Locomotion task		
Base velocity x	$\exp(-(v_x - v_x^c)^2/0.25)$	2.0
Base velocity y	$\exp(-(v_y - v_y^c)^2/0.25)$	1.5
Base yaw rate	$\exp(-(\omega_z - \omega_z^c)^2/0.25)$	4.0
Feet air time	$\mathbb{1}_{\|\mathbf{v}^c_{xy}\|_2>0.1}\sum_i \mathbb{1}_{t_i^{\text{air}}>0}$ $\mathbb{1}_{\max(F_{i,z,t},F_{i,z,t-1})>1}(t_i^{\text{air}} + \Delta t - 0.5)$	4.0
Foot-heading alignment	$\sum_{i\in\{\text{L,R}\}} \lvert \text{wrap}(\psi_i - \psi_{\text{base}})\rvert$	-0.25
Regularization		
Lower-body torque†	$\|\boldsymbol{\tau}_{\text{lower}}\|_2^2$	-10^{-5}
Upper-body torque†	$\|\boldsymbol{\tau}_{\text{upper}}\|_2^2$	-10^{-5}
Lower-body DoF velocity†	$\|\dot{\mathbf{q}}_{\text{lower}}\|_2^2$	-10^{-3}
Upper-body DoF velocity†	$\|\dot{\mathbf{q}}_{\text{upper}}\|_2^2$	-10^{-3}
Lower-body DoF acceleration†	$\|(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_{t-1})_{\text{lower}}/\Delta t\|_2^2$	-2.5×10^{-7}
Upper-body DoF acceleration†	$\|(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_{t-1})_{\text{upper}}/\Delta t\|_2^2$	-2.5×10^{-7}
Lower-body action rate†	$\|\mathbf{a}_{\text{lower},t} - \mathbf{a}_{\text{lower},t-1}\|_2^2$	-0.1
Upper-body action rate†	$\|\mathbf{a}_{\text{upper},t} - \mathbf{a}_{\text{upper},t-1}\|_2^2$	-0.1
EE linear-velocity change	$\sum_{i\in\{\text{L,R}\}} \|\mathbf{v}^{\text{EE}}_{i,t} - \mathbf{v}^{\text{EE}}_{\text{L},t-1}\|_2$	-0.2
EE angular-velocity change	$\sum_{i\in\{\text{L,R}\}} \|\boldsymbol{\omega}^{\text{EE}}_{i,t} - \boldsymbol{\omega}^{\text{EE}}_{\text{R},t-1}\|_2$	-0.02
Base vertical velocity	$m(v_z^{\text{base}})^2$	-2.0
Base angular velocity	$\|\boldsymbol{\omega}^{\text{base}}_{xy}\|_2^2$	-0.05
Torso angular velocity	$\|\boldsymbol{\omega}^{\text{torso}}_{xy}\|_2^2$	-1.0
Base orientation	$\|\mathbf{g}^{\text{base}}_{xy}\|_2^2$	-1.5
Torso orientation	$\|\mathbf{g}^{\text{torso}}_{xy}\|_1$	-1.0
Base-height deviation	$(5-4m)(h-h_c)^2$	-10.0
Hip roll/yaw position	$[m + (1-m)h_c]\,\|\mathbf{q}_{\text{hip,roll/yaw}}\|_2^2$	-2.5
Stance DoF velocity	$(1-m)\|\dot{\mathbf{q}}_{\text{lower}}\|_2^2$	-10^{-3}
Stance symmetry	$(1-m)(\|\mathbf{q}^s_{\text{L}} - \mathbf{q}^s_{\text{R}}\|_1 + \|\mathbf{q}^a_{\text{L}} + \mathbf{q}^a_{\text{R}}\|_1)$	-0.5
Ankle roll	$\lvert q_{\text{ankle,L}}\rvert + \lvert q_{\text{ankle,R}}\rvert$	-2.0
Stance root offset	$(1-m)\lvert[R_{\text{pel}}^\top(\mathbf{p}_{\text{pel}} - \frac{1}{2}(\mathbf{p}_{\text{L}} + \mathbf{p}_{\text{R}}))]_y\rvert$	-5.0
Stance foot offset	$(1 - m\mathbb{1}_{\lvert v_x^c\rvert>0})\lvert[R_{\text{pel}}^\top \text{abs}(\mathbf{p}_{\text{L}} - \mathbf{p}_{\text{R}})]_x\rvert$	-5.0
Zero-command translation	$m\,\mathbb{1}_{\|\mathbf{v}^c_{xy}\|<0.2}\|\mathbf{v}^{\text{root,world}}_{xy}\|_2$	-1.5
Zero-command rotation	$m\,\mathbb{1}_{\lvert\omega_z^c\rvert<0.1}\lvert\omega_z^{\text{root,world}}\rvert$	-1.5
Contact slip	$(10-9m)\sum_i \mathbb{1}_{\|\mathbf{F}_i\|_2>1}\|\mathbf{v}_i^{\text{foot}}\|_2^2$	-0.2
Foot-contact pattern	$(1-m)\mathbb{1}_{n_f<2} + m\mathbb{1}_{n_f\neq1}$, $n_f = \sum_i \mathbb{1}_{F_{i,z}>1}$	-4.0
Standing foot contact	$(1-m)\mathbb{1}_{\min_i F_{i,z}<0.1}$	-0.15
Feet orientation	$\sum_{i\in\{\text{L,R}\}} \|\mathbf{g}^i_{xy}\|_2$	-2.0
Foot target height	$m[\min_i \lvert z_i - 0.11\rvert - 0.02]_+$	-5.0
Swing-foot height	$\sum_{i:\|\mathbf{F}_i\|_2\leq1}(z_i - 0.11m - 0.025(1-m))^2$	-20.0

Continued on next page

Table 9 (continued)

TERM	EXPRESSION	WEIGHT
Feet too close	$\mathbb{1}_{\|\mathbf{p}_{\mathrm{L}}^{xy}-\mathbf{p}_{\mathrm{R}}^{xy}\|_2<0.17}$	-10.0
Knee below threshold	$\sum_{i\in\{\mathrm{L},\mathrm{R}\}}\mathbb{1}_{q_{\mathrm{knee},i}<0.2}$	-1.0
Feet air-time asymmetry	$\lvert t_{\mathrm{L}}^{\mathrm{last}}-t_{\mathrm{R}}^{\mathrm{last}}\rvert$	-5.0
Penalties		
Lower-body DoF position limit†	$\sum_{j\in\mathrm{lower}}\mathrm{dist}(q_j,[q_{j,\min}^{\mathrm{soft}},q_{j,\max}^{\mathrm{soft}}])$	-5.0
Upper-body DoF position limit†	$\sum_{j\in\mathrm{upper}}\mathrm{dist}(q_j,[q_{j,\min}^{\mathrm{soft}},q_{j,\max}^{\mathrm{soft}}])$	-5.0
Upper-body DoF velocity limit†	$\sum_{j\in\mathrm{upper}}\min\{1,[\lvert\dot{q}_j\rvert-\dot{q}_{j,\max}]_+\}$	-5.0
Upper-body torque limit†	$\sum_{j\in\mathrm{upper}}[\lvert\tau_j\rvert-0.9\tau_{\max,j}]_+$	-0.1
Termination	$b_{\mathrm{reset}}(1-b_{\mathrm{timeout}})$	-250

B.5 Policy Training

Simulation & Training Setup. We train our end-effector tracking policy π_{t} in Isaac Gym [72] or Isaac Sim via Isaac Lab [94], and transfer this policy to the MuJoco simulator [73] for Sim2Sim evaluations before deploying it in the real world. We train our policy with 4,096 environments for overall 20K iterations in parallel, with a learning rate of 1e-4 for both the actor and critic models. AdamW optimizer [95] is used with a weight decay of 1e-2. We use a high simulation frequency of 500Hz, with the low-level PD controller running at 50Hz. All the policy training is conducted on a single NVIDIA RTX 4090 or an L40S GPU.

Sim2Real Domain Randomization. Following previous works [3, 5, 9], we employ standard domain and dynamics randomization to facilitate Sim2Real transfer [96], including variations in link center of mass (CoM) and control delay. Notably, we identify that randomizing the *end-effector mass* is essential; without this specific randomization, the policy exhibits end-effector instability, leading to high-frequency hand oscillations that compromise tracking accuracy.

B.6 Deployment Hardware

We run all modules (*e.g.*, π_{t} and SAM-3 [17]) off-the-shelf on a 32-GB RAM laptop equipped with NVIDIA RTX 5070Ti GPU and Intel Core Ultra 9 275HX CPU processor (24 CPU cores / 24 threads). We run cuRobo with CUDA graph acceleration, which largely improve the efficiency on the edge [63]. For the detection module, we have tested both Grounding DINO base [16] and Grounding DINO 1.5 [88], where the base version can be deployed on the laptop, and Grounding DINO 1.5 only provides access through online APIs. However, we find that Grounding DINO base is sufficient for most scenes and objects. Tab. 10 reports deployment runtime; conservative robot motion accounts for most of the roughly 55-second execution.

Table 10: Deployment runtime. Measured on the local laptop with Grounding DINO base. GroundingSAM and AnyGrasp are included in perception; replanning is a per-update latency and is not an additional phase in the total.

Phase or component	Time
Perception (query to grasp)	1.33 s
GroundingSAM	950 ms
AnyGrasp	377 ms
Reaching	20.39 s
Closure	1.20 s
Return	32.14 s
Total (four phases)	$\approx$ 55 s
Replanning (per update)	$\approx$ 20 ms

B.7 Testing Assets Details

In the paper, we have tested HERO with 20 daily objects; these objects have different sizes and weights, while being made with different materials, making it challenging to grasp with a Dex-3 hand. We list the detailed sizes, weights, materials, and language queries of all objects tested in Tab. 11. Note that the size is roughly measured as the shape is irregular and cannot be easily described.

B.8 Testing Scenes Details

Tab. 12 lists the details of the novel scenes tested in this paper, which are mainly chosen from Coordinated Science Laboratory Studio (CSL Studio) and the Thomas M. Siebel Center for Computer Science at the University of Illinois Urbana-Champaign, Urbana, IL. The snapshot of these testing scenes can be found in Fig. 8.

Table 11: Testing objects, sizes, weights, materials, and language queries. Sizes are roughly measured due to irregular shapes. Weights are measured with an accurate food scale.

Object	Size	Weight	Material	Language Query
10 daily object evaluation.				
	4.9×4.9×4.9 cm	58.06 g	Wood	`orange cube`
	12×6×6 cm	14.97 g	Aluminum	`coke can`
	8.5×7.5×7.5 cm	15.88 g	Plastic	`red apple`
	9×5×5 cm	137.89 g	Plastic & Metal	`emergency stop button`
	15×6×5 cm	239.95 g	Plastic & Metal	`robot hand`
	16.5×10.5×3.3 cm	185.07 g	Plastic	`game cartridge`
	27.5×9.5×9.5 cm	79.83 g	Plastic	`olive oil bottle`
	16×8.5×8.5 cm	392.81 g	Plastic & Liquid	`hand soap`
	23×7.9×7.9 cm	43.09 g	Paperboard & Plastic	`chip can`
	17×11×11 cm	73.94 g	Plush	`red piranha plant`
10 daily objects used in 10 daily scenes evaluation.				
	21×12×43 cm	215.91 g	Plastic	`kettle`
	16.5×11×21 cm	213.19 g	Plush	`toy dog`
	17.2×9×9 cm	24.95 g	Paperboard & Plastic	`Starbucks coffee`
	11.5×9.4×9.4 cm	526.17 g	Ceramic	`orange mug`
	18.5×6.3×6.3 cm	286.22 g	Plastic & Metal	`water bottle`
	7×7.7×7.7 cm	14.97 g	Plastic	`green apple`
	18.8×2.4×13.3 cm	301.19 g	Paper	`purple book`
	14×8×39 cm	234.96 g	Plastic & Metal	`helicopter`
	8×5.6×10 cm	367.41 g	Metal & Spam	`spam`
	19×5×5 cm	86.18 g	Plastic & Liquid	`cleaner bottle`

Continued on next page

Object	Size	Weight	Material	Language Query
Additional objects.				
	24.5×8×6.5 cm	135.17 g	Plush	`carrot`
	23×8×8 cm	307.08 g	Plush	`broccoli`
	7.5×7.2×7.5 cm	19.05 g	Plastic	`orange`
	21×2.3×13.8 cm	376.03 g	Paper	`book`
	15.3×6.5×6.5 cm	18.14 g	Aluminum	`black can`

Table 12: Testing scenes, table heights, language queries. Here we list the novel scenes chosen in this paper for evaluation, and the corresponding table height. The snapshot of these scenes can be found in Fig. 8.

Scene	Location	Table Height	Language Query
corridor	CSL Studio UIUC	0.43m	`kettle`
office lounge	CSL Studio UIUC	0.48m	`toy dog`
building café	Siebel CS Building UIUC	0.72m	`Starbucks coffee`
office	CSL Studio UIUC	0.74m	`orange mug`
building lounge	Siebel CS Building UIUC	0.74m	`water bottle`
office kitchenette	CSL Studio UIUC	0.74m	`green apple`
building den	Siebel CS Building (RM 3333) UIUC	0.74m	`purple book`
robotics lab	CSL Studio UIUC	0.86m	`helicopter`
office kitchen	CSL Studio UIUC	0.87m	`spam`
classroom	Siebel CS Building (RM 1302) UIUC	0.92m	`cleaner bottle`