SEEK, GROUND, RECOVER: AN AGENTIC FRAMEWORK FOR AUTONOMOUS HUMANOID LOCO-MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autonomous long-horizon humanoid loco-manipulation in open scenes requires task decisions to adapt as locomotion and posture changes alter observations, body-relative target geometry, and action feasibility. Reliable execution requires acquiring sufficient perceptual evidence, converting semantic targets into accurate metric goals, and restoring physical preconditions after failures. We present an agentic framework that couples active evidence acquisition, explicit geometric perception, and execution feedback in a closed decision loop. A vision-language model (VLM) selects observation actions, identifies and verifies task targets, and provides initial target localization. Explicit geometric perception refines these predictions into metric 3D goals for pretrained whole-body policies, separating semantic verification from metric accuracy. During and after execution, the VLM interprets temporal feedback to assess progress and identify failures; the runtime combines these assessments with controller checks to coordinate further observation, recovery, and continuation. Persistent task state retains objectives, perceptual evidence, and execution progress, supporting target re-localization and resumption of interrupted tasks after recovery. A shared goal-and-outcome interface separates semantic coordination from physical control and supports extension of the motor-skill library. Experiments demonstrate the benefits of embodied evidence acquisition and explicit geometric perception for target localization, and characterize recovery under controlled physical failures.

1 INTRODUCTION

Autonomous long-horizon humanoid loco-manipulation requires grounding open-ended instructions while preserving task objectives as observations and physical conditions change. An instruction such as "put the red cup on the table beside the sofa" combines semantic search and instance verification with metric localization, locomotion, and manipulation. Targets may be unseen, occluded, or too small to identify, requiring further observation. Body motion changes visibility and body-relative geometry; failed grasps, drops, and falls can invalidate earlier decisions. The agent must therefore repeatedly establish the evidence and physical conditions needed for its next action.

Vision-language-action (VLA) policies map visual-language observations to robot actions (Zitkovich et al., 2023; Kim et al., 2024; Jiang et al., 2026), with recent extensions incorporating active perception through virtual view selection (Liu et al., 2026c) and executed camera actions (Liu et al., 2026a), and persistent object representations for execution feedback (Ren et al., 2026b). Modular systems connect foundation-model reasoning to executable skills and spatial representations (Ahn et al., 2022; Huang et al., 2023). Recent work strengthens learned spatial representations (Chen et al., 2025; Lu et al., 2026) and exposes spatial predictions through typed spatial grounding interfaces calibrated to the robot frame (Chen et al., 2026). Agentic frameworks additionally integrate persistent state, geometric verification, and recovery (Qin et al., 2026; Ren et al., 2026a). The remaining challenge for long-horizon humanoid execution is to coordinate these capabilities as body motion changes visibility, body-relative target geometry, and action feasibility. Verifying the intended object does not establish an accurate metric goal, and detecting a failure does not establish readiness to resume. The agent must therefore assess target identity, metric-goal validity, and physical readiness separately to decide when to observe, revise goals, or recover.

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

This motivates our central question: *How can a VLM-based agent coordinate geometric grounding and failure recovery for autonomous, long-horizon humanoid loco-manipulation?*

Our agentic framework couples VLM reasoning, explicit geometric perception, and pretrained whole-body experts through persistent task state (Figure 1). When evidence is insufficient or outdated, the VLM selects observation actions and proposes or selects target regions for verification against the instruction. Detector-supported refinement and calibrated geometry convert verified regions into metric goals, separating localization accuracy from semantic identification. During execution, VLM assessments of temporal visual evidence and physical-state summaries combine with controller checks to request observation, revise goals, or invoke recovery. Retained task state supports resumption once physical preconditions are restored.

Our main contributions are as follows:

- We introduce a modular and extensible agentic framework for autonomous long-horizon humanoid loco-manipulation, coordinating VLM reasoning, perception, and whole-body control while supporting the integration of new tools and skills.
- We develop humanoid agentic skills that organize multi-step reasoning and tool use into reusable procedures for embodied perception, target grounding, and recovery.
- We couple VLM-guided active perception with embodied action to acquire and verify task-relevant evidence. Explicit geometric perception translates verified targets into metric goals for whole-body control.
- We integrate VLM-guided execution monitoring and recovery to assess failures, restore physical conditions, and resume interrupted tasks.

2 Related Work

2.1 Foundation Models for Robot Decision Making

Vision-language-action (VLA) models learn to map visual observations and language instructions to robot actions from large-scale demonstrations (Kim et al., 2024), supporting generalist control across embodiments (NVIDIA et al., 2025) and whole-body humanoid loco-manipulation (Jiang et al., 2026). HumanVLA distills a privileged-state teacher into a vision-language policy for simulated humanoid rearrangement, with active-rendering supervision for gaze (Xu et al., 2024). Recent extensions couple action generation with evidence acquisition and assessment by selecting informative virtual views within learned 3D representations (Liu et al., 2026c), executing camera motions in the environment (Liu et al., 2026a), and maintaining persistent object representations for post-action verification (Ren et al., 2026b).

Agentic architectures complement learned policies by coordinating reasoning, tools, and execution through runtime interfaces. Skill affordances (Ahn et al., 2022) and semantic interfaces (Guo et al., 2026a) connect decisions to executable behavior, while persistent task state and observation-based verification support replanning and correction (Qin et al., 2026; Guo et al., 2026b). For humanoids, hierarchical vision-language planning combines pretrained skill sequencing with video-based completion monitoring (Schakkal et al., 2025). Our framework coordinates VLM reasoning, explicit geometric perception, and pretrained whole-body experts in long-horizon humanoid loco-manipulation, reassessing target identity, metric goals, and recovery readiness as observations and physical conditions change.

2.2 Active Perception and Visual Search

Active perception and visual search provide complementary evidence for target grounding. VLM-guided observation actions adjust camera viewpoint and orientation for semantic queries (Sripada et al., 2024). Within acquired images, guided search localizes fine details in high-resolution, cluttered scenes (Wu & Xie, 2024); zoom-in analysis attributes much of the benefit to reduced background distraction (Liu et al., 2026b). Our VLM coordinates image inspection with head and body motions that acquire new observations, retaining positive and negative evidence to guide search and verification.

2.3 Geometric Perception for Manipulation

Physics-based humanoid manipulation policies such as OmniGrasp rely on accurate object state and shape information (Luo et al., 2024a). To enable control from visual observations, existing work distills privileged-state teachers into vision-based students and trains policies for robustness to perceptual limitations (Xu et al., 2024; He et al., 2026b; Zhang et al., 2025).

Vision-language control improves spatial precision through spatially guided training and geometric features (Chen et al., 2025; Lu et al., 2026), or explicit guidance such as typed spatial targets calibrated to the robot frame (Chen et al., 2026) and coarse 2D paths for 3D-aware policies (Li et al., 2025). Geometric estimators provide 6D object poses (Wen et al., 2024; Liu et al., 2025), while RGB-D predicates support execution verification (Ren et al., 2026a). We refine verified regions into metric goals, assessing localization separately from semantic correctness.

2.4 Failure Monitoring and Physical Recovery

Closed-loop execution uses LLMs to interpret structured multisensory summaries for failure diagnosis and correction (Liu et al., 2023), and VLMs to monitor plan-derived constraints or generate programs for spatiotemporal failure detection (Guo et al., 2023; Zhou et al., 2025). Humanoid systems further integrate geometric verification, diagnostic feedback, re-grounding, and VLM-guided skill refinement (Ren et al., 2026a; He et al., 2026a).

Object loss and falls additionally require physical recovery before task continuation. Learned humanoid policies support autonomous recovery (Wu et al., 2026), while HumanCLAW factors out balance and motor-tracking failures to isolate VLM decision making (Li et al., 2026). Our VLM interprets temporal visual evidence and contact and posture summaries to assess failures and guide further observation and recovery. The agent combines these judgments with target re-localization, pretrained floor-grasp and get-up policies, and controller completion checks, retaining task state to resume interrupted stages after recovery is verified.

3 Method

3.1 Task and System Overview

We consider natural-language relocation instructions specifying a movable source, a placement destination, and optional attributes or spatial relations within the evaluated scenes. Execution requires identifying the requested targets, estimating their geometry relative to the humanoid, and maintaining the physical conditions needed to manipulate them. The runtime captures paired RGB-D observations and aggregates recent proprioceptive and hand-contact measurements. The VLM interprets the instruction, selected RGB evidence when needed, and textual state, sensor, and tool summaries; geometric tools estimate robot-relative goals, and pretrained motor experts execute movement. Each reasoning call returns a structured *action*: an invocation of a tool, agentic skill, or motor expert, or an assessment such as a verification outcome or failure hypothesis. A *tool* performs a basic operation, an *agentic skill* coordinates several reasoning and tool steps, and a *motor skill* executes motion through pretrained whole-body control.

At decision event k, the runtime offers admissible options $\mathcal{A}_k = \Gamma(m_k, o_k)$ from persistent task state m_k and observation o_k. The VLM returns

$$\widetilde{a}_k = \langle \alpha_k, \eta_k \rangle, \quad (\widetilde{a}_k, \kappa_k, r_k) = F_\theta(\ell, m_k, o_k, \mathcal{A}_k), \quad \alpha_k \in \mathcal{A}_k, \tag{1}$$

where α_k identifies the tool, skill, expert, or assessment type, η_k gives its arguments or content, $\kappa_k \in [0, 1]$ is confidence, and r_k is an evidence rationale. The runtime checks assessments and maps them to applicable tools, skills, or experts.

The closed loop in Figure 1 links *Seek* for acquiring and verifying target evidence, *Ground* for checked metric estimation, and *Recover* for interpreting physical feedback and restoring execution conditions. Localization failure can require renewed Seek; an object drop can require both Seek and Ground before regrasp.

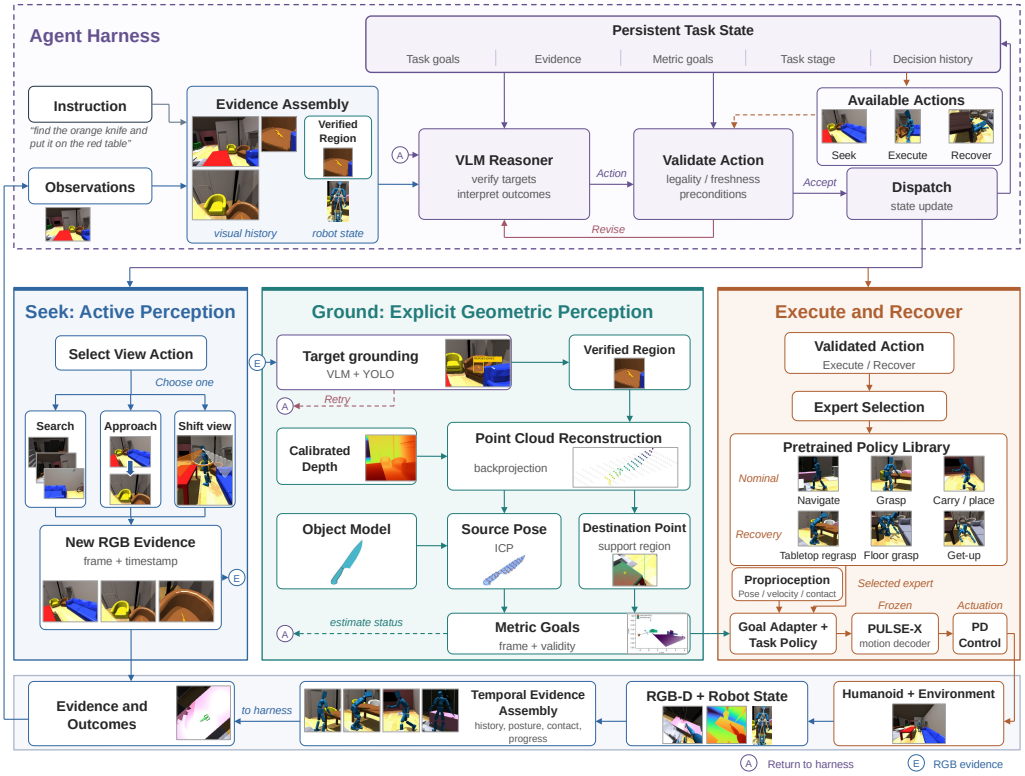


Figure 1: Closed-loop coordination of Seek, Ground, and Recover. Persistent task state links VLM reasoning, explicit geometric perception, and pretrained experts through validated decisions and execution feedback. Matching letters connect interfaces.

3.2 Agent Coordination

The runtime maintains the persistent task state

$$m_k = \langle \ell, E_k, G_k, s_k, D_k \rangle, \tag{2}$$

comprising instruction ℓ, retained observation, assessment, and tool evidence E_k, metric goals G_k, stage and progress s_k, and decision and failure records D_k. When an asynchronous reply arrives, the runtime checks that its evidence provenance matches the originating request and that it remains in the admissible set recomputed from live state. A valid proposal is applied as $a_k = \widetilde{a}_k$; otherwise it is discarded, and the active expert continues only when safe. The invoked tool, skill, or expert returns an outcome y_{k+1} with a status such as running, completed, or failed and its supporting evidence, updating the state as

$$m_{k+1} = \mathcal{U}(m_k, o_k, a_k, y_{k+1}), \tag{3}$$

retaining objectives and history while revising progress and goal validity. Appendices A.2 and A.3 give the grounding and execution instantiations.

Humanoid agentic skills. We develop humanoid agentic skills as bounded procedures for multi-step reasoning and tool use. Tiled visual search and embodied evidence acquisition address limited egocentric visibility; precision object localization supplies target geometry; dropped-object search and contact-and-posture assessment support recovery. Skills can invoke pretrained motor experts and return evidence and outcomes to the task state. Appendix A.1.3 details their contracts and implementations.

3.3 Seek: Active Evidence Acquisition

Seek acquires and checks evidence for target identity and subsequent geometric estimation. Further observation is needed when a target is outside the view, occupies too few pixels, is occluded, or remains ambiguous among similar instances. The VLM uses the instruction, selected head-camera

observations, retained hypotheses, and tool feedback to select tools such as *crop candidate*, *focus gaze*, *shift viewpoint*, or *approach candidate*, or to *commit* or *give up*. VLM-proposed candidates undergo stage-specific verification of appearance, support, and spatial relations.

Within existing images, *tiled visual search* isolates and enlarges relevant regions, reduces irrelevant context, and then maps proposals back to the parent image and reconciles candidates across tiles or views. Gaze, viewpoint, and approach tools acquire new observations: the runtime checks the maneuver's preconditions and budget before invoking head-control or navigation routines. Physical approaches use A* on an inflated occupancy grid derived from scene meshes (Hart et al., 1968). A fresh observation follows successful motion, with stabilization where configured; selected negative evidence discourages redundant inspection. Source and destination candidates are accepted independently, requiring both verifier acceptance and the role-specific confidence threshold. Seek stops when all required roles are accepted, the agent chooses to give up, or the evidence-acquisition budget is exhausted. Appendix A.2.4 gives the formal criteria. Appendix A.2 details the observation and verification interfaces.

3.4 GROUND: EXPLICIT GEOMETRIC LOCALIZATION

Semantic verification does not establish metric accuracy. For a verified source, the *precision object localization* skill estimates the canonical object origin, which generally differs from the visible-surface center. After the VLM locks the source, a fine-tuned YOLO detector (Jocher et al., 2023) refines its region; a proposal whose class label disagrees with the instruction is used only after VLM identity verification (Appendix B.3.1). The accepted region provides calibrated depth points; support-plane removal and foreground selection produce a partial cloud P, with optional appearance filtering and part recovery. Appearance filtering can suppress background but may also discard valid target points.

Camera intrinsics and the camera-to-humanoid transform, obtained from mounting calibration and current head/body kinematics, express these points in the robot frame without absolute world localization. When a canonical mesh cloud M is available, multi-start ICP (Besl & McKay, 1992) estimates its pose. At iteration r, nearest-neighbor associations $q_j^{(r)}$ and an adaptive distance threshold define inliers $\mathcal{I}_r$; rigid alignment updates

$$T_{bo}^{(r+1)} = \arg\min_{T \in \mathrm{SE}(3)} \sum_{j \in \mathcal{I}_r} \|p_j - T q_j^{(r)}\|_2^2, \tag{4}$$

where T_{bo} maps the canonical object frame into the humanoid frame. Initial translations align cloud centroids or bounds, and multiple orientations undergo local refinement; depth-visibility evaluation and fit checks select the estimate without guaranteeing a global optimum.

For a destination, the VLM verifies the intended support object. The metric backend uses calibrated depth and an available shape model to estimate its geometry, transforming a canonical model reference point by the fitted pose. This reference differs from a surface contact point: the execution adapter and controller construct the placement goal and check its feasibility. Destination identity and the accuracy of its geometric reference are therefore assessed separately.

The geometry service returns a goal record

$$g_\rho = \langle \hat{\mathbf{x}}_\rho,\ v_\rho,\ \epsilon_\rho,\ o_\rho \rangle, \tag{5}$$

with robot-relative pose or reference point $\hat{\mathbf{x}}_\rho$, fit-validity flag $v_\rho \in \{0, 1\}$, available registration-quality measures ϵ_ρ, and capture provenance o_ρ. Only valid records enter G_k; failed or invalidated estimates return to Seek or terminate unresolved. Appendix A.1.3 summarizes the localization procedure.

3.5 RECOVER: EXECUTION MONITORING AND TASK RESUMPTION

Once targets are grounded, pretrained experts perform navigation, grasping, transport, placement, and release. Their adapters define observations, reference construction, applicability, and outcomes. Execution uses proprioception, contact feedback, and perceptual goals from Ground, without simulator-provided object-state truth. Each expert i is a goal-conditioned policy

$$\boldsymbol{a}_t \sim \pi_i(\cdot \mid \boldsymbol{s}_t, \boldsymbol{g}_t^i), \tag{6}$$

where $\boldsymbol{s}_t$ is the robot state and $\boldsymbol{g}_t^i$ the skill-specific goal. Each expert is trained by reinforcement learning with skill-specific rewards. Low-level action $\boldsymbol{a}_t$ differs from task-level action $\widetilde{a}_k$. The shared adapter interface supports adding experts, such as the tabletop-grasp, floor-grasp, release, and get-up experts in Appendix Table S6; Appendix A.3.6.1 gives the training objective, latent-action formulation, and rewards.

Recover monitors grasp formation, object retention, placement, and body stability. The *contact-and-posture assessment* skill summarizes recent sensor measurements. For hand $h \in \{\mathrm{L}, \mathrm{R}\}$ with force norm $f_{h,t}$, and pelvis height z_t relative to standing reference z_{stand},

$$c_{h,t} = \mathbb{1}[f_{h,t} \geq \tau_f], \qquad \bar{c}_{h,t} = \frac{1}{W} \sum_{u=t-W+1}^{t} c_{h,u}, \qquad \zeta_t = \frac{z_t}{z_{\mathrm{stand}}}, \tag{7}$$

these features and $\Delta\zeta_t$ summarize current and recent contact and posture changes. The VLM interprets them with the current stage, selected decision records, and RGB evidence when needed to assess progress, uncertainty, and failures. Sensor cues and VLM assessments can trigger targeted rechecks or independent audits. Contact does not identify the touched object, and low posture alone does not establish a fall.

Where supported, the VLM proposes continuation, grasp/placement/release retries, renewed grounding, or termination. The runtime checks applicability, budget, and safety before dispatching accepted assessments or actions; deterministic guards can also trigger recovery. Each downstream recovery strategy allows four attempts per episode; needing a fifth terminates the task as a failure. Uncertain identity or stale geometry prompts Seek or Ground, respectively.

Following object loss, *dropped-object search* retains target identity, destination, and interrupted stage while acquiring new views through bounded head search and small locomotion adjustments, verifying the candidate and its support, and re-localizing it. The runtime selects tabletop or floor regrasp from validated target/support evidence, subject to approach and posture checks; finding the object alone does not establish regrasp. A detected fall can trigger get-up. A recovery skill or expert σ completes only when its checks eval_σ (contact and retention for regrasp; height and orientation for get-up) hold for L_σ consecutive control steps. Resumption additionally requires a valid next-stage target:

$$\mathrm{Done}_\sigma(t) = \prod_{u=t-L_\sigma+1}^{t} \mathrm{eval}_\sigma(u), \qquad \mathrm{Resume}(t) = \mathrm{Done}_\sigma(t) \wedge v_{\mathrm{next}} = 1, \tag{8}$$

where v_{next} is the next-stage goal's validity flag. Otherwise the runtime re-observes, retries within budget, or stops. Appendix A.3 details these interfaces.

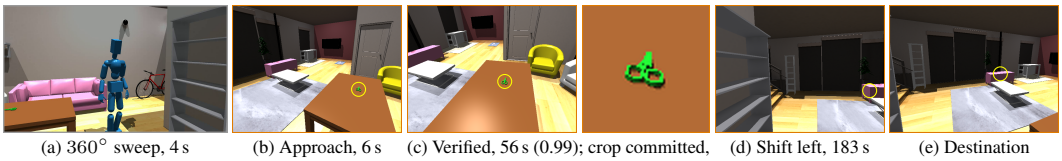
4 Experiments

4.1 Experimental Protocol

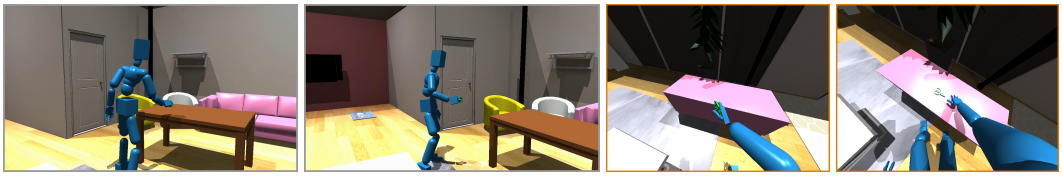
Experiments use physics-based humanoids in ReplicaCAD (Szot et al., 2021), with experts trained in Isaac Gym (Makoviychuk et al., 2021) and frozen before evaluation, using AMASS references for body/head tracking (Mahmood et al., 2019). We use 100 development and 100 held-out test relocation tasks in one furnished layout, with different object placements and instruction instances. We evaluate gpt-5.6-luna (*Luna*), adapted Qwen3.8-27B (*Qwen-FT*), and its untuned base for the adaptation comparison (Section 4.5). Main-table Luna conditions use three runs with fresh model calls and fixed non-model settings; Qwen-FT uses one deterministic run. Appendix B.1 details preprocessing, repeat accounting, and metrics.

4.2 Grounding and Agentic Evidence

Grounding episodes execute observation actions with frozen experts (Figure 2) and end when both roles are committed, at abstention, or after 1200 s. Grasping and placement are not scored. *Loc@2cm* requires the committed source position at capture time to lie within 2 cm of ground truth; *Destination* measures support identity, not placement accuracy. Abstentions and timeouts count as failures.



(a) 360° sweep, 4 s (b) Approach, 6 s (0.96) (c) Verified, 56 s (0.99); crop committed, 72 s (d) Shift left, 183 s (0.98) (e) Destination commit, 212 s (0.96)

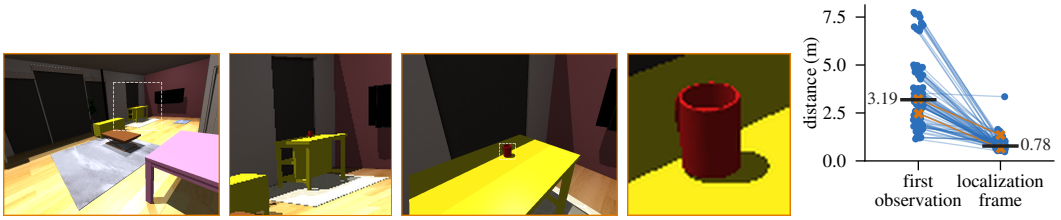


(f) Lift after 3 attempts, 39 s (g) Transport, 48 s (h) Release decision, 52 s (i) Completion verified, 60 s

Figure 2: Relocation in two recordings: (a–e) localization; (f–i) execution and verification. Times are recording-relative; parentheses show VLM confidence.

Table 1: Full-agent grounding on dev/test splits. Luna: mean ± SD over three runs; Qwen-FT: one deterministic run. Time: task median/P95.

VLM	Split	N	Loc@2cm (%) ↑	Destination (%) ↑	Abstain (%) ↓	Mean VLM calls ↓	Time (s) ↓
Luna	dev	100	84.3±3.2	82.3±4.2	11.7±3.2	34.6±1.5	635 / 1171
Luna	test	100	84.3±5.5	75.7±7.6	14.0±6.0	33.6±0.1	594 / 1033
Qwen-FT	dev	100	**88.0**	**88.0**	10.0	30.2	383 / 585
Qwen-FT	test	100	**91.0**	**88.0**	7.0	28.9	362 / 563



(a) Before approach, 7.58 m (b) Ambiguous crop (c) After approach, 0.94 m (d) Verified crop (e) Paired distances (83 tasks)

Figure 3: Approach improves evidence for a distant target (a–d). Panel (e) compares initial and localization distances for 83 tasks with estimates; 17 yielded none. Crosses indicate errors ≥ 2 cm. Orange borders denote VLM inputs; dashed boxes are crop overlays. Distances and errors are evaluation-only (Appendix B.2).

Table 1 evaluates both reasoners on both splits. Table 2 compares fixed-RGB Luna references with the Luna full agent and includes a Qwen-FT action ablation on 100 development tasks. The references vary both reasoning and available evidence, so they do not isolate embodiment. The ablation keeps reasoning, audit, and geometry fixed while restricting source actions: *sweep only* disables source body motion, *in-place gaze* replaces approach with gaze without translation, and the full agent can approach candidates.

The full agent achieves approximately 84% source localization with Luna and 88–91% with Qwen-FT across splits. Within the controlled action ablation, Loc@2cm rises from 32% without source motion to 67% with gaze and 88% with approach. Agent Destination scores follow the same order but depend on first committing the source and are for reference only. Fixed-evidence predictions rarely meet the 2 cm criterion, and adding verification mainly increases abstention rather than correct localization (Appendix B.2.3). Figure 3 shows how approach improves visual evidence. A 30-task Luna ablation and offline action-prefix diagnostics provide complementary evidence (Appendix B.2); the latter do not replace the live controlled ablation.

4.3 Explicit Geometry and Grasp Sensitivity

We separate region accuracy, metric localization, and controller sensitivity. On 50 archived views, Table 3 crosses single-image Luna or YOLO boxes (Jocher et al., 2023) with depth-median or mesh-registration estimation, using identical foreground points per box and measuring error at the canon-

Table 2: Luna baselines/full agent and Qwen-FT action ablations on 100 development tasks. Luna: mean ± SD, three runs; Qwen-FT: one deterministic run. Baseline Loc@2cm uses XY box/depth error (Appendix B.1). Agent Destination requires source commitment and is for reference only. Bold: block best.

Condition	N	Loc@2cm (%) ↑	Destination (%) ↑	Abstain (%) ↓	Mean VLM calls ↓
Luna: direct prediction on fixed evidence and full agent					
Initial-view one-shot	100	0.7±0.6	8.0±1.0	73.3±1.2	1.00
Cached-360 one-shot	100	3.0±0.0	62.7±4.2	9.7±1.2	1.00
Initial-view + verifier/appearance	100	0.3±0.6	5.7±0.6	85.3±0.6	1.60
Cached-360 + verifier/appearance	100	2.7±0.6	28.7±2.3	66.0±2.6	4.59
Full agent (Ours)	100	**84.3±3.2**	**82.3±4.2**	11.7±3.2	34.6±1.5
Qwen-FT: embodied-action ablation					
Sweep only (no source motion)	100	32.0	38.0	59.0	22.8
+ in-place gaze	100	67.0	65.0	25.0	29.1
Full agent (Ours)	100	**88.0**	**88.0**	10.0	30.2

Table 3: Offline localization on 50 views using single-image Luna or YOLO boxes. IoU is mean overlap with the visible target. Error quantiles exclude invalid estimates; threshold counts include all views.

Box source	IoU	Geometry	Valid	Median (cm)	P90 (cm)	Error ≤ 5 cm
Luna	0.870	Depth median	50/50	2.654	5.712	42/50
Luna	0.870	Mesh + ICP	49/50	0.113	2.971	47/50
YOLO	0.895	Depth median	50/50	2.502	5.719	42/50
YOLO	0.895	Mesh + ICP	50/50	0.128	2.717	49/50

ical object origin. Despite similar region overlap, registration reduces median origin error by over an order of magnitude for both box sources. This reflects shape information and alignment, with iterative refinement providing additional improvement (Appendix B.3.3).

With the original frozen OmniGrasp policy (Luo et al., 2024a), constant position bias reduces grasp success from 88–90% at zero bias to about 54% at 5 cm and 9% at 10 cm (five repetitions; Appendix B.3.4). This motivates precise grounding, without establishing a universal tolerance or an end-to-end geometry benefit. Table 4 instead uses non-privileged grasp inputs.

4.4 Failure Detection and Recovery

Table 4 reports 30 events per failure category. The first three use non-privileged grasp inputs and require a stable grasp after re-localization. Detection is reliable, but correction remains harder, especially for floor drops. Additional attempts help except for tabletop recovery. Figure 4 illustrates recovery; Appendix B.4 details the protocol.

We also report end-to-end success on 30 cases uniformly sampled from the task-level localization results of deterministic Qwen-FT, with Luna serving as the execution agent. Table 5 gives cumulative success after each stage over all 30 cases, including localization failures; 11/30 (36.7%) complete placement.

4.5 Adaptation of the Self-Hosted Reasoner

We adapt Qwen3.8-27B (Qwen Team, 2026) with QLoRA (Dettmers et al., 2023) on successful Luna grounding demonstrations selected by privileged localization scores; geometry, runtime checks, and motor experts remain fixed. Qwen-FT handles grounding, while Luna monitors execution and recovery. On all 100 held-out test tasks, adaptation raises Loc@2cm from 63% to 91% and destination identification from 50% to 88% (Table 1; Appendix B.6).

5 Conclusion and Limitations

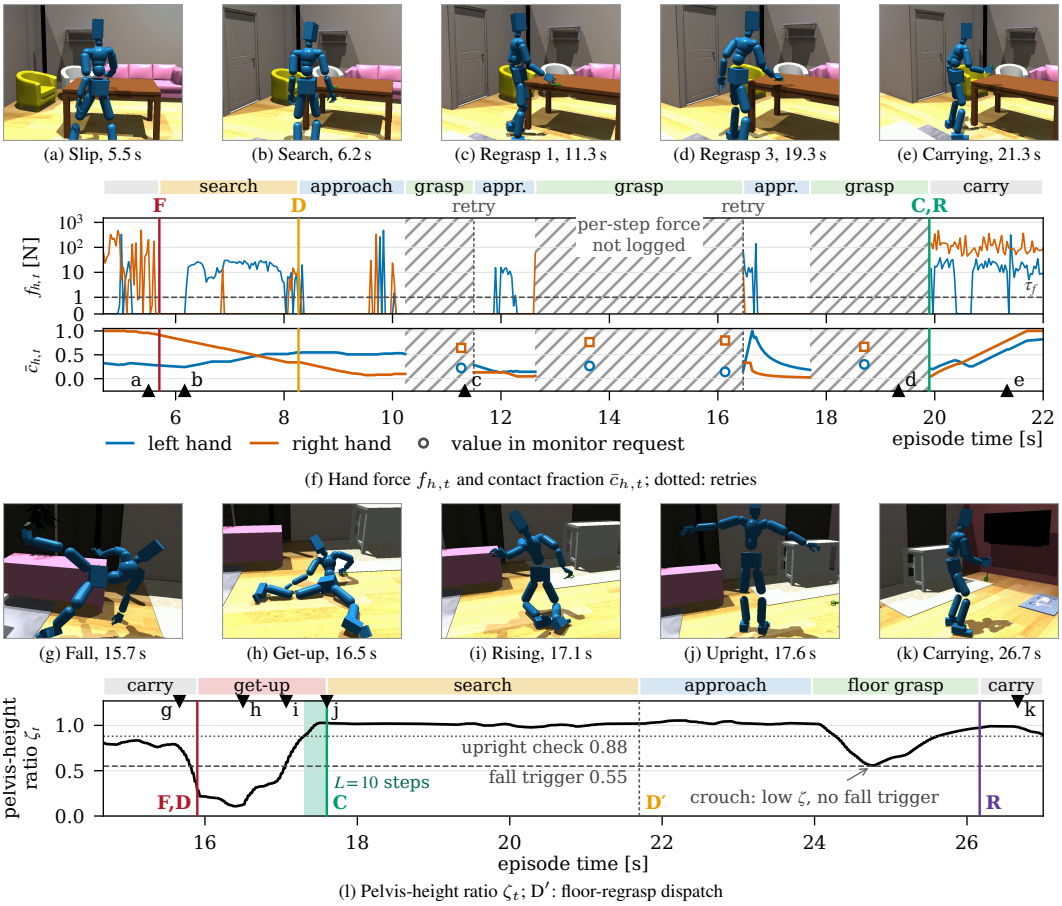
We presented an agentic framework that coordinates VLM reasoning, explicit geometric perception, and pretrained whole-body experts for autonomous humanoid loco-manipulation. Persistent task state connects Seek, Ground, and Recover to acquire evidence, form metric goals, and restore

Table 4: Recovery counts out of 30 events per failure category. Correct@k denotes physical correction within at most k recovery attempts, not stage resumption or final task completion. Grounding errors concern the source object; "–" means detection is not reported. Tabletop drops land on elevated supports (Appendix B.4).

Metric	Grounding error	Tabletop drop	Floor drop	Whole-body fall
Detect ↑	–	29/30	29/30	30/30
Correct@1 ↑	21/30	18/30	11/30	16/30
Correct@4 ↑	23/30	18/30	15/30	25/30

Table 5: Cumulative stage success on 30 end-to-end cases.

Stage	Localization	Grasping	Carrying	Placement
Success ↑	27/30	22/30	12/30	11/30



(a) Slip, 5.5 s (b) Search, 6.2 s (c) Regrasp 1, 11.3 s (d) Regrasp 3, 19.3 s (e) Carrying, 21.3 s

(f) Hand force $f_{h,t}$ and contact fraction $\bar{c}_{h,t}$; dotted: retries

(g) Fall, 15.7 s (h) Get-up, 16.5 s (i) Rising, 17.1 s (j) Upright, 17.6 s (k) Carrying, 26.7 s

(l) Pelvis-height ratio ζ_t; D′: floor-regrasp dispatch

Figure 4: Recovery from contact loss (a–f) and a fall (g–l). F/D/C/R denote detection, dispatch, completion, and resumption; triangles mark keyframes and hatching denotes unavailable per-step force logs. Traces are tactile/proprioceptive; third-person views are illustrative. Details: Appendix A.3.5.

conditions for task resumption. Experiments demonstrate improved localization through embodied observation and geometric refinement and characterize recovery under controlled failures.

Evaluation is limited to simulation: grounding splits share one furnished layout, and recovery uses small controlled event sets. Registration requires known shape models. Self-hosted adaptation covers visual grounding; end-to-end execution uses Luna. Real-robot deployment requires validating perception, calibration, contact sensing, and control transfer; full-pipeline fine-tuning, including execution monitoring and recovery, and onboard reasoning remain future work.

AI USE STATEMENT

Generative AI tools were used to aid and polish writing by improving clarity and wording, to support retrieval and discovery of related work, and to draft portions of the paper and supplementary material. The authors reviewed and revised all AI-assisted text and take responsibility for the final content, including its claims and references. Vision-language models used as components of the proposed system are described separately in the paper.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv preprint arXiv:2204.01691, 2022. URL https://arxiv.org/abs/2204.01691.

Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, Weiyang Jin, Hao Li, Yao Mu, Jiangmiao Pang, Yu Qiao, Yang Tian, Bin Wang, Bolun Wang, Fangjing Wang, Hanqing Wang, Tai Wang, Ziqin Wang, Xueyuan Wei, Chao Wu, Shuai Yang, Jinhui Ye, Junqiu Yu, Jia Zeng, Jingjing Zhang, Jinyu Zhang, Shi Zhang, Feng Zheng, Bowen Zhou, and Yangkun Zhu. InternVLA-M1: A Spatially Guided Vision-Language-Action Framework for Generalist Robot Policy. arXiv preprint arXiv:2510.13778, 2025. URL https://arxiv.org/abs/2510.13778.

Xiwen Chen, Zelin Li, Zhiruo Zhou, Huiming Chen, Chenwei Wang, and Xiaojun Zhu. Pointing-VLA: Typed Spatial Grounding Interfaces for Vision-Language-Action Manipulation. arXiv preprint arXiv:2608.23138, 2026. URL https://arxiv.org/abs/2608.23138.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Meng-Hao Guo, Zhe-Han Mo, Jia-Jun Wang, Yi Zhang, Kejin Wang, Yi-Xuan Deng, Jia-Peng Zhang, Yongming Rao, and Shi-Min Hu. Transferring the Intelligence of VLMs to Robotic Control. arXiv preprint arXiv:2609.22966, 2026a. URL https://arxiv.org/abs/2609.22966.

Pengyuan Guo, Zhonghao Mai, Zhengtong Xu, Kaidi Zhang, Quan Khanh Luu, Heng Zhang, Zichen Miao, Arash Ajoudani, Zachary Kingston, Qiang Qiu, and Yu She. PLanAR: Planning-Language-Grounded Agentic Reasoning for Robot Manipulation. arXiv preprint arXiv:2602.01662, 2026b. URL https://arxiv.org/abs/2602.01662.

Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. DoReMi: Grounding language model by detecting and recovering from plan-execution misalignment. arXiv preprint arXiv:2307.00329, 2023. URL https://arxiv.org/abs/2307.00329.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Shuliang He, Shuai Wang, Bo Yue, Junchi Teng, Changyu Wang, and Guiliang Liu. RoboReact: Agentic skill distillation from generated egocentric videos for generalizable whole-body manipulation. arXiv preprint arXiv:2608.03387, 2026a. URL https://arxiv.org/abs/2608.03387.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562, 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

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. In *International Conference on Learning Representations*, 2026.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. Software, 2023. URL https://github.com/ultralytics/ultralytics.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An Open-Source Vision-Language-Action Model. arXiv preprint arXiv:2406.09246, 2024. URL https://arxiv.org/abs/2406.09246.

Siyao Li, Jiawei Gu, Shuai Liu, Kairui Hu, Zekun Li, Linjie Li, Chengcheng Tang, Po-Chen Wu, Ivan Shugurov, Lingni Ma, Michael Zollhoefer, Sizhe An, Abhay Mittal, Amy Zhao, Ranjay Krishna, Manling Li, Ziwei Liu, and Chuan Guo. HumanCLAW: Can vision-language models act through a body? arXiv preprint arXiv:2607.27180, 2026. URL https://arxiv.org/abs/2607.27180.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Caelan Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. HAMSTER: Hierarchical Action Models for Open-World Robot Manipulation. In *International Conference on Learning Representations*, 2025.

Mengzhen Liu, Enshen Zhou, Cheng Chi, Yi Han, Shanyu Rong, Liming Chen, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. SaPaVe: Towards active perception and manipulation in vision-language-action models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37164–37174, 2026a.

Sheng Liu, Zhe Li, Weiheng Wang, Han Sun, Heng Zhang, Hongpeng Chen, Yusen Qin, Arash Ajoudani, and Yizhao Wang. ActivePose: Active 6d object pose estimation and tracking for robotic manipulation. arXiv preprint arXiv:2509.11364, 2025. URL https://arxiv.org/abs/2509.11364.

Xianjie Liu, Yiman Hu, Yixiong Zou, Liang Wu, Jian Xu, and Bo Zheng. HiDe: Rethinking the zoom-IN method in high resolution MLLMs via hierarchical decoupling. In *International Conference on Machine Learning*, 2026b.

Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3468–3484, 2023.

Zhenyang Liu, Yongchong Gu, Yikai Wang, Xiangyang Xue, and Yanwei Fu. ActiveVLA: Injecting active perception into vision-language-action models for precise 3d robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8141–8151, 2026c.

Yehao Lu, Jiarui Yang, Yuning Su, Yufeng Xie, Yu Zhong, Yazhou Zhang, Haiyu Lan, Kaixiang Lu, Peiwen Lin, Chuang Wang, Zequn Qin, Enyu Li, and Xi Li. V-Link: Recovering Lost Visual Representations in Action DiT for Vision-Language-Action Models. arXiv preprint arXiv:2608.25308, 2026. URL https://arxiv.org/abs/2608.25308.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10861–10870, 2023.

Zhengyi Luo, Jinkun Cao, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. OmniGrasp: Grasping Diverse Objects with Simulated Humanoids. In *Advances in Neural Information Processing Systems*, volume 37, pp. 2161–2184, 2024a. URL https://papers.nips.cc/paper_files/paper/2024/hash/040ace837dd270a87055bb10dd7c0392-Abstract-Conference.html.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal Humanoid Motion Representations for Physics-Based Control. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/fb19caf91f4966e426026685028cb73a-Paper-Conference.pdf.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5442–5451, 2019.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470, 2021. URL https://arxiv.org/abs/2108.10470.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734, 2025. URL https://arxiv.org/abs/2503.14734.

Xue Qin, Simin Luan, John See, Cong Yang, and Zhijun Li. AEROS: A Single-Agent Operating Architecture with Embodied Capability Modules. arXiv preprint arXiv:2604.07039, 2026. URL https://arxiv.org/abs/2604.07039.

Qwen Team. Qwen3.8-27B. Model card, 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Peng Ren, Haoyang Ge, Chuan Qi, Cong Huang, Hong Li, Jiang Zhao, Pei Chi, and Kai Chen. CyboWaiter: A Physical Agentic Framework for Humanoid Whole-Body Locomotion-Manipulation. arXiv preprint arXiv:2603.10675, 2026a. URL https://arxiv.org/abs/2603.10675.

Peng Ren, Haoyang Ge, Jiang Zhao, Cong Huang, Yukun Shi, Pei Chi, and Kai Chen. Closing the Loop in Humanoid VLA: Persistent 3D Object Tokens for Verifiable Loco-Manipulation. arXiv preprint arXiv:2607.18016, 2026b. URL https://arxiv.org/abs/2607.18016.

André Schakkal, Ben Zandonati, Zhutian Yang, and Navid Azizan. Hierarchical vision-language planning for multi-step humanoid manipulation. arXiv preprint arXiv:2506.22827, 2025. URL https://arxiv.org/abs/2506.22827.

Venkatesh Sripada, Samuel Carter, Frank Guerin, and Amir Ghalamzan. Scene Exploration by Vision-Language Models. arXiv preprint arXiv:2409.17641, 2024. URL https://arxiv.org/abs/2409.17641.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems*, volume 34, pp. 251–266, 2021.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879, 2024.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094, 2024.

Tianshu Wu, Xiangqi Kong, Yue Chen, Qize Yu, Hang Ye, Jia Li, Yizhou Wang, and Hao Dong. SUGAR: A Scalable Human-Video-Driven Generalizable Humanoid Loco-Manipulation Learning Framework. arXiv preprint arXiv:2605.20373, 2026. URL https://arxiv.org/abs/2605.20373.

Xinyu Xu, Yizheng Zhang, Yong-Lu Li, Lei Han, and Cewu Lu. HumanVLA: Towards vision-language directed object rearrangement by physical humanoid. In *Advances in Neural Information Processing Systems*, volume 37, pp. 18633–18659, 2024.

Hui Zhang, Zijian Wu, Linyi Huang, Sammy Christen, and Jie Song. RobustDexGrasp: Robust dexterous grasping of general objects. arXiv preprint arXiv:2504.05287, 2025. URL https://arxiv.org/abs/2504.05287.

Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. Code-as-Monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

SUPPLEMENTARY MATERIAL

This supplement provides the technical and evaluation details supporting the main paper. Appendix A describes the agentic system, persistent task state, grounding and execution loops, prompt and tool interfaces, expert policies and rewards, and implementation settings. Appendix B presents the experimental protocols and extended evaluations of failure recovery, local VLM reasoning, explicit geometry, and grasp sensitivity.

A METHOD AND IMPLEMENTATION DETAILS

This appendix expands the method of Sections 3.1–3.5. Appendix A.1 details the general agent model of Sections 3.1 and 3.2: the components of each decision step, the persistent task state, and the agentic skills. The system runs this model as two coupled loops that are nested in each other. Appendices A.2 and A.3 describe the grounding and execution agents, each through its state and evidence, admissible actions, decision prompts, and validation. Appendix A.3 closes with the whole-body experts that actions invoke.

A.1 AGENTIC SYSTEM COMPONENTS

A.1.1 AGENTIC SYSTEM MODEL

This subsection expands the compact formulation of the main paper (Equations 1–3). There, o_k abbreviates the assembled evidence $e_k = \Phi_q(o_k, m_k^q)$ below, and the validation step described in prose corresponds to Apply_q. Let $q \in \{\mathrm{G}, \mathrm{X}\}$ denote the grounding or execution loop. At task-relevant event k, an agent instantiation is characterized by

$$\mathfrak{A}_q = (\mathcal{M}_q, \Phi_q, \Gamma_q, F_\theta, \mathrm{Apply}_q, \mathcal{U}_q), \tag{S1}$$

where $\mathcal{M}_q$ is the persistent state space and the remaining components are the functions of one decision step. The runtime first assembles evidence from the current observation and the retained state,

$$e_k^q = \Phi_q(o_k, m_k^q),$$

then constructs the options admissible in the current state,

$$\mathcal{A}_k^q = \Gamma_q(m_k^q, e_k^q),$$

and serializes them with the instruction into a decision context:

$$\xi_k^q = (\ell, m_k^q, e_k^q, \mathcal{A}_k^q).$$

The reasoner returns a proposed action together with its confidence and rationale,

$$(\widetilde{a}_k, \kappa_k, r_k) = F_\theta(\xi_k^q).$$

The runtime validates the proposal and, if it is accepted, applies it through a registered tool or expert,

$$(a_k, y_{k+1}) = \mathrm{Apply}_q(\widetilde{a}_k, \mathcal{A}_k^q, m_k^q),$$

and finally writes the outcome back to the task state:

$$m_{k+1}^q = \mathcal{U}_q(m_k^q, e_k^q, a_k, y_{k+1}). \tag{S2}$$

Here ℓ is the retained instruction, o_k the current observation, e_k^q the assembled evidence, $\mathcal{A}_k^q$ the admissible options, and ξ_k^q the serialized decision context. As in the main paper, $\widetilde{a}_k = \langle \alpha_k, \eta_k \rangle$, where $\alpha_k \in \mathcal{A}_k^q$ identifies a tool, agentic skill, motor expert, or assessment type and η_k gives its arguments or content; κ_k is its confidence and r_k a concise evidence rationale. a_k is the applied action; a rejected proposal yields a null, hold, or protective action. The outcome y_{k+1} is either an immediate tool result or a later expert outcome.

A.1.2 PERSISTENT TASK STATE

The runtime owns the persistent task state m_k and serializes its relevant fields at each decision event; Table S1 lists its components, which may be stored in separate runtime structures. In the compact notation of the main paper, $m_k = \langle \ell, E_k, G_k, s_k, D_k \rangle$, ℓ is the instruction grounding, E_k the visual hypotheses and retained evidence, G_k the metric goals, s_k the execution state, and D_k the decision and failure records, which the execution loop splits into decision records D_k and recovery state Ω_k. Grounding updates hypotheses and metric goals; execution updates the stage, physical feedback, and recovery state.

Table S1: Logical components of the persistent task state.

Component	Principal fields	Purpose
Request identity	request and episode identifiers	Matches each reply to the state that produced the request and rejects stale replies.
Instruction grounding	source and destination nouns with their qualifiers	Retains the semantic roles throughout grounding, execution, and recovery.
Visual hypotheses	candidate boxes or points, verifier acceptance, rejected candidates	Keeps competing candidates without collapsing them into one target.
Metric goals	robot-relative pose and its localization method	Connects a verified region to a controller-consumable goal.
Execution state	stage, readiness gates, contact and item state	Describes the current physical task without exposing low-level actions.
Decision records	proposed and dispatched actions with their validation results	Supports recovery and detects repeatedly rejected actions.
Recovery state	failed stage, targets to re-localize, retry counts	Allows the original instruction to resume after an object drop or fall.

A.1.3 AGENTIC SKILLS

An *agentic skill* is a bounded procedure that addresses a specific difficulty through several reasoning, sensing, or geometric steps. An action is one validated decision, an agentic skill can span several such decisions, and a motor skill executes the physical motion. We write a skill as

$$\mathcal{S}_j = (\mathrm{Entry}_j, \mathrm{Obj}_j, \mathrm{Proc}_j, \mathrm{End}_j, \mathrm{Ret}_j), \tag{S3}$$

with entry conditions, local objective, conditional procedure, termination conditions, and returned evidence. Entry conditions and budgets remain runtime-controlled, and every step passes the same action checks. Table S2 lists the skills.

Table S2: Agentic skills. Each procedure can contain several actions and invoke existing motor policies.

Agentic skill	Entry condition and local objective	Returned evidence and termination
Precision object localization	A verified source needs a controller-usable pose; refine its region with a detector, backproject its depth into the humanoid frame, remove the support plane, and register the canonical mesh.	Robot-relative pose and fit validity; terminates on an accepted fit or a declared failure, which returns control to evidence acquisition.
Tiled visual search	No reliable candidate is visible in the full image; magnify local regions of a multi-view sweep and reconcile candidates.	A candidate in the original view with its verification outcome; terminates on acceptance or an exhausted budget.
Embodied evidence acquisition	Identity or extent of a candidate is uncertain; gaze, approach, shift, or search to obtain a better view.	A stabilized new observation and the maneuver outcome; terminates on the maneuver or budget limit.
Dropped-object search	A carried item is lost; search the floor and re-establish the item's identity and pose.	A verified and localized recovery target, or an unresolved status.
Contact-and-posture assessment	Contact continuity or balance needs interpretation; relate tactile and pelvis-height histories to recent RGB.	An evidence-supported state and applicable response, or explicit uncertainty.

A.2 GROUNDING AGENT

A.2.1 STATE AND EVIDENCE

The grounding state specializes the task state as

$$m_k^{\mathrm{G}} = (\ell, \rho_k, H_k^s, H_k^d, E_k, G_k, B_k), \tag{S4}$$

where $\rho_k \in \{\mathrm{src}, \mathrm{dst}\}$ is the active role, H_k^s and H_k^d are role-specific candidate sets, E_k is the retained evidence, G_k the verified metric goals, and B_k the rejected or exhausted attempts. Keeping the two roles separate lets the source be re-localized while a still-valid destination is retained. All evidence comes from the humanoid's head camera; a search sweep captures 16 egocentric RGB views at 22.5-degree yaw intervals, which remain available by view identifier.

A first text-only call parses the instruction into roles and qualifiers:

```
You are parsing a pick-and-place instruction into a structured JSON
object. Return only valid JSON.

Schema:
{"source":"<lowercase movable-object noun>",
 "source_qualifier":"<attributes or null>",
 "target":"<lowercase destination noun or null>",
 "target_qualifier":"<attributes or null>"}

Rules:
- Keep lexicalized furniture names such as "side table" together.
- Put color, size, instance, and spatial descriptions in qualifiers.
- Use a null target only when the instruction has no destination.

Instruction: <raw instruction>
```

A.2.2 GROUNDING ACTIONS

At each turn the runtime advertises only the actions whose inputs and controller preconditions are currently available (Table S3). Availability depends on whether a candidate is bound in the current view, whether a sweep is pending, which evidence has already been used, and the remaining motion and search budgets; localization is offered only for the source.

Table S3: Grounding actions. All inputs are first-person head-camera observations.

Action	Role	Returned evidence or effect
Candidate recognition over a sweep		
`candidate_list`	both	Ranked candidates with view, center, size, and confidence; an empty list means absent.
`approach_candidate`	both	Bounded walk toward a small or partly visible candidate, followed by a new view.
`shift_view`	both	Lateral move that reveals an occluded object or the rest of a surface.
Verification and localization		
`ground_candidate`	both	Binds a box drawn on the current view for verification.
`request_candidate_crop`	both	Tight or context crop around the bound candidate.
`request_object_reference`	both	Appearance-only reference images of the category, without scene location.
`request_embodied_evidence`	both	One maneuver (focus gaze, stabilize view, approach, shift, or search) and a stabilized new observation.
`request_search_view_detail`	both	Neighboring full frames of selected sweep views.
`select_search_view`	both	Turns back to a chosen sweep view.
`request_detection_icp_localization`	source	Robot-relative pose by precision object localization.
`commit`	both	Submits the role binding with its confidence for verification.
`give_up`	both	Returns a bounded failure with its reason.

A.2.3 DECISION PROMPTS

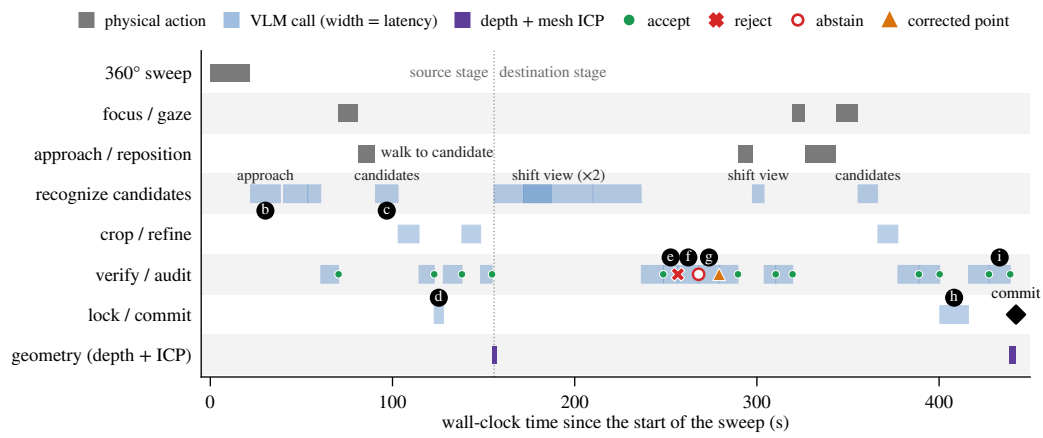
Grounding runs in two stages: candidate recognition over the sweep, followed by the tool loop for the active role. Their prompts are:

```
You are the <role> candidate recognizer.
<Batch of a fixed-position 360-degree RGB sweep, or a fresh view
 captured after a bounded action, with the completed-action lineage.>

Instruction: <raw instruction>
Current role: source | destination
Target appearance: <role label>
Frames: [{attachment, view_id, yaw_deg, size}, ...]

Search every attached full frame. Return plausible candidates even when
small, but do not guess a bounding box. If the same physical candidate is
visible in several frames, return one observation per frame; the
controller tracks it across views. Give the center of the visible target
pixels, not of its supporting furniture, and the size level of a square
crop that contains the whole target (tiny, small, medium, or large). The
controller creates all crops. Scene colors are literal; do not use
reference renders as scene-location evidence. If a plausible source is
tiny or only partly visible, request approach_candidate or shift_view
toward its visible pixels instead of listing it.

Reply with ONLY one of:
{"action":"candidate_list","role":"<role>",
 "candidates":[{"view_id":"<view>","center":[x,y],
   "size_level":"tiny|small|medium|large","confidence":<0..1>,
   "evidence":"<visible attributes and uncertainty>"}],
 "reason":"<summary>"}
{"action":"approach_candidate|shift_view","role":"<role>",
 "view_id":"<view>","center":[x,y],"size_level":"<level>",
 "confidence":<0..1>,"evidence":"<unresolved detail>"}
```

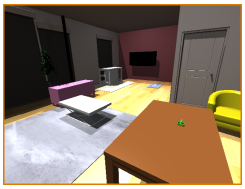
```
You are an embodied perception agent. Verify only the active role:
pickup_source or placement_destination. Inspect all accumulated RGB
and tool evidence, then choose exactly one currently allowed tool.

Rules:
- A pickup source must match the requested category and qualifier and
  be a visibly discrete, relocatable instance. A resting support is
  valid; an installed, fastened, or integrated object is not.
- For a destination, verify the requested furniture and qualifier.
  The source need not be visible in the destination image.
- Detector labels and boxes are evidence, not synonyms or ground
  truth. Request semantic association when a label conflicts with the
  instruction.
- If confidence is >=0.80 and commit is available, commit. Otherwise
  name the missing discriminating feature and request the one best
  crop, reference, focus, stabilization, shift, approach, or search.
- After visual identification, request exact localization before the
  final source commitment.
- Do not repeat an unchanged action. Respect controller budgets.

Return only:
{"action":"<allowed tool>","args":{...},"confidence":<0..1>,
 "thought":"<short visible evidence or unresolved feature>"}
```

A maneuver request names no velocity, path, or arbitrary destination; the controller executes a bounded, collision-aware motion, stabilizes the head camera, and returns a new observation.

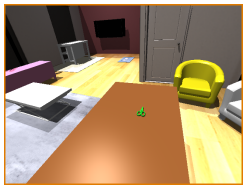
Recorded decision trace. Figure S1 illustrates how observation actions, verification outcomes, and physical motions interact in a recorded grounding episode. The trace is a worked example of the decision loop, not a fixed action script.



(a) Timeline of physical actions, VLM calls, and registration



(b) Step 89, sweep view: approach candidate (0.96)



(c) Step 839, after approach: candidate list (0.99)



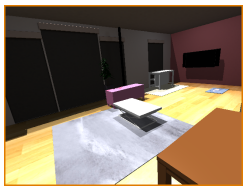
(d) Candidate crop of the step-89 view: lock (0.99)



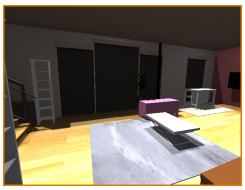
(e) Target-point marker crop (dot controller-drawn): reject (0.99)



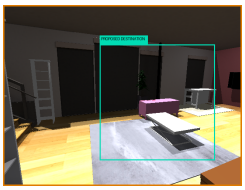
(f) Step 121, full frame: abstain (0.98)



(g) Step 81, full frame: target point corrected (0.99)



(h) Step 3174, full frame: destination lock (0.96)



(i) Verifier context (box controller-drawn): accept (0.98)

Figure S1: Decision trace of one full grounding episode (Luna). (a) Each row is an action type; grey bars are physical actions executed by the whole-body experts, blue bars are VLM calls (width is latency) with their verification outcome, and purple bars are depth and mesh registration. Lettered markers link calls to panels (b)–(i), which show their exact egocentric RGB inputs (orange borders). The agent finds a tiny source in the sweep (b), approaches and relists it (c) and locks it (d); for the destination it rejects a target point on the wrong furniture (e), abstains on that view (f), corrects the point in another view (g), and after two viewpoint shifts locks the stand (h) and passes the final verification (i).

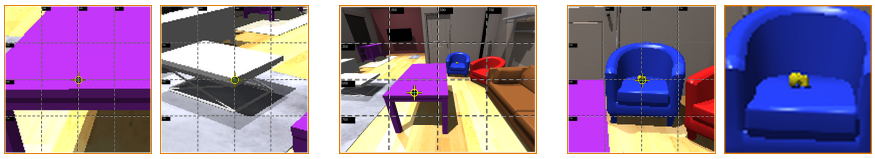
A.2.4 Commit Validation

A commit must be parseable, lie inside the declared image, keep source and destination distinct from each other and from the source's support, and pass a separate VLM verification call that returns `verified`, `revise`, or `reject`. A revision triggers a local box refinement; a rejection is kept as negative evidence, and the loop requests different evidence instead of silently accepting another candidate. Figure S2 shows such rejections and revisions on the RGB inputs actually received by the agent: identity rejections of wrong instances and same-color distractors (a, b), a lock crop that cuts the target at its boundary and is replaced by a complete crop (c), and a corrected destination (d).

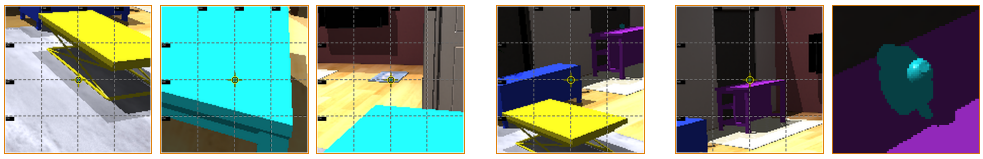
For each role $\rho \in \{\mathrm{src}, \mathrm{dst}\}$, a hypothesis $h_\rho = \langle b_\rho, \omega_\rho, \kappa_\rho \rangle$ records a view-specific region b_ρ, verification outcome $\omega_\rho \in \{\mathrm{accept}, \mathrm{reject}, \mathrm{revise}, \mathrm{uncertain}\}$, and confidence κ_ρ. Roles are accepted independently, and Seek stops when

$$\mathrm{Acc}(h_\rho) = \mathbb{1}[\omega_\rho = \mathrm{accept} \wedge \kappa_\rho \geq \tau_\rho]\,,$$
$$\mathrm{Stop}_k = \bigwedge_{\rho \in \mathcal{Q}_k} \mathrm{Acc}(h_\rho) \;\vee\; \alpha_k = \textit{give up} \;\vee\; n_k \geq N, \tag{S5}$$

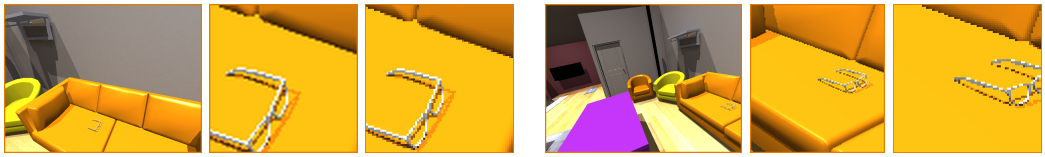
where $\mathcal{Q}_k$ contains the roles still required, τ_ρ is the role-specific commit threshold, and n_k counts evidence-acquiring actions against budget N.



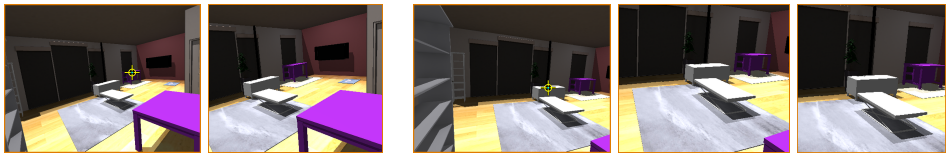
(a) “find the yellow doorknob and put it on the white tv stand”. Rejected: proposals at steps 119 (purple table), 111 (table base); revised: step 119 re-audit (knob on chair); accepted: step 119 corrected point, lock crop. Source error 2.4 mm (eval. only).



(b) “find the cyan alarm clock and put it on the yellow chair”. Rejected: proposals at steps 98 (table base), 114 (cyan table), 114 (blue mat); revised: step 106 (clock); accepted: step 106 corrected point, step 3355 close-view detail. Source error 0.4 mm (eval. only).



(c) “find the white eyeglasses and put it on the red tv stand”. Rejected: step 1638 close view locked, then its lock crop and native detail (cut off); accepted: step 105 earlier full view re-locked, with its lock crop and native detail (audit: accept). Source error 1.0 mm (eval. only).



(d) Same instruction as (a), separate destination-only run. Rejected: step 3124 locked center and support focus (purple, open-leg stand); accepted: step 73 re-locked center and support focus (white cabinet), then final verifier crop (accept, commit). Destination not scored.

Figure S2: **Rejections and revisions before commitment.** Each subfigure shows, left to right, VLM inputs of one episode that were rejected, revised, or accepted (a wider gap separates the groups); subcaptions give the policy step and the returned decision. (a) VLM-proposed target points landed on a purple table and on a coffee-table base; a full-frame re-audit redirected the agent to the doorknob on the chair. (b) Same-color distractors were refused until a compact cyan clock was found and confirmed from a close view. (c) A close-range lock crop cut the eyeglasses at the left edge; a complete crop from an earlier view was re-locked and accepted. (d) The locked destination was a small purple stand rather than the requested white TV stand; the identity audit rejected it and the agent re-locked the white cabinet. All images are complete VLM inputs (orange border); marker and grid overlays are controller-drawn attachments that the VLM received. Source errors are evaluation-only.

A.3 Execution Agent

A.3.1 State and Request

The execution state specializes the task state as

$$m_k^{\mathrm{X}} = (\ell, G_k, s_k, p_k, E_k, D_k, \Omega_k), \tag{S6}$$

where G_k holds the retained roles and metric goals, s_k is the task stage, p_k contains progress and readiness, E_k is the temporal evidence window, D_k the decision records, and Ω_k the retry and recovery state. At a safe decision event, the runtime sends a request with the chronological head-camera frames, the root self-sensing and hand tactile summaries, the stage and its elapsed frames, the admissible actions, and, when available, the grounding context and recent decision records.

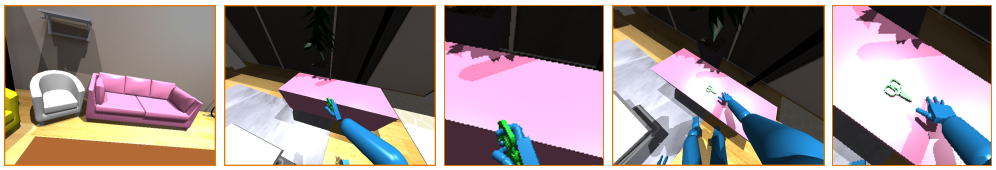
A.3.2 Evidence

The execution evidence consists of a bounded window of chronological egocentric RGB and the compact physical summaries of Table S4, computed only from first-person proprioceptive and tactile signals. Transient outcomes, such as an item that is still settling, do not immediately count as failures; a confirmed failure requires temporally consistent evidence.

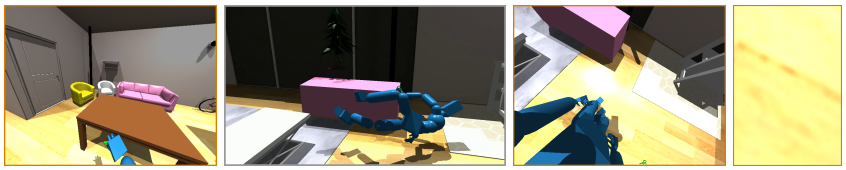
Table S4: Compact temporal feedback supplied to semantic reasoning.

Group	Example fields	Interpretation
Progress	stage frames, bounded progress, transition readiness	Distinguishes a running expert from a completed or stalled one.
Posture	pelvis-height ratio and trend, coarse upright/low/fallen state, stability duration	Separates controlled low reaching from balance loss and a whole-body fall.
Hand contact	left/right force norms, recent contact fractions, current bilateral/one-hand/no-contact pattern	Distinguishes contact establishment, retained contact, and recent contact loss.
Body contact	coarse support/contact regions and recent impact category	Provides evidence of collision, fall, or support without exposing full simulator state.
Item outcome	retained, separated-settling, separated-stable, floor-visible, or uncertain	Supports wait, regrasp, placement retry, or additional evidence.
Route outcome	progressing, blocked, no-safe-route, or replanning-failed	Supports continuation, route replanning, rollback, or termination.
Perception status	verified, rejected, not verified, or unavailable	Separates positive evidence, contradiction, and absence of a verifier result.

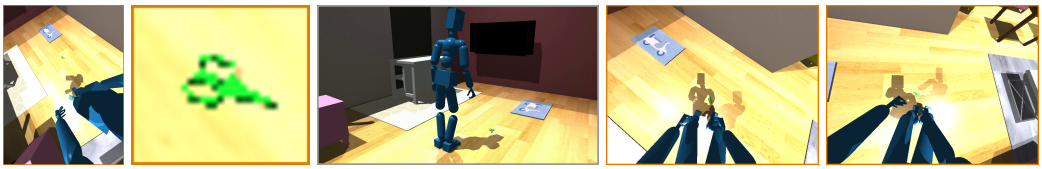
In the recorded runs, drop confirmation during grasping relies on the tactile and posture summaries without attached images; the two retried grasps in Figure S12 (g–l) follow such text-only confirmations. RGB frames enter the monitor when the item's location matters: Figure S3 shows the recorded inputs and decisions during nominal release and post-fall re-localization.



(a) Episode 1, carry the green scissors to the pink tv stand. GRASP check, step 30 (undetermined → none); tick 778 frame, goal crop (ready → release); tick 898 frame, goal crop (released → complete). Object on the stand (eval. only); run ended on its monitor-call budget.



(b) Episode 2, same task. Tick 217 frame (held, confidence 0.72 → inspect); third-person frame of the fall near tick 477; tick 604 frame after get-up (on floor → localize); tick 604 candidate crop (identity: reject).



(c) Episode 2, continued. Tick 651 frame (on floor → localize), candidate crop (identity: accept); third-person frame of the re-grasp approach; ticks 886, 933 (held → continue). Item lost again near tick 1202; run ended in an indoor-safety stop, not a placement.

Figure S3: **Execution-stage monitoring from head-camera RGB.** (a) In a nominal episode, the GRASP-stage check is inconclusive because the item is not in view; during carry, the monitor judges the scissors ready to release at the stand and then released at the destination. (b) In a second episode of the same task, the robot falls while carrying. After get-up, the monitor reports the item on the floor, but the first floor candidate is rejected by the identity check because its crop shows only floor. (c) A later frame yields a confirmed floor candidate; the agent re-approaches and re-grasps, and subsequent monitor calls report the item held. Orange borders mark monitor inputs; grey borders mark third-person frames shown for presentation only, which the agent never received. Subcaptions give, left to right, the monitor tick, the input, and the returned state and action.

A.3.3 EXECUTION ACTIONS

Each monitor judgment returns a verdict (nominal, degraded, failure, or unsafe) and one action from the stage-specific admissible set (Table S5). A retry must name its cause, and `retry_grounding` is offered only when a grounding context exists.

Table S5: Execution actions and the stages in which they are admissible.

Action	Effect	Admissible stages
`none`	Continue the active expert.	all
`retry_grasp`	Re-approach the source and regrasp.	grasp, place, release
`retry_place`	Repeat the placement.	place, release
`retry_release`	Repeat the release.	release
`retry_grounding`	Return a role to the grounding loop.	navigation (both roles), grasp (source), place (destination)
`terminate`	Stop the episode with a reason.	all

A.3.4 DECISION PROMPT AND VALIDATION

```
You are the humanoid's execution-monitoring VLM agent. Inspect the
chronological first-person RGB frames and the humanoid's own
proprioceptive and tactile readings.
Current phase: <stage>; elapsed frames in this phase: <n>.
<decision records; grounding context; floor-search note when active>
Root self-sensing: height, standing baseline, ratio, trend, balance state.
Hand tactile: force norms, recent contact fractions, temporal contact
state, contact pattern and continuity.

Rules:
- critical_height_loss has safety priority, even when an item is visible
  or hand forces are high; the hands may be bracing on the floor.
- One noisy frame is insufficient; combine chronological RGB with tactile
  persistence. High hand force is not proof of retention.
- GRASP: controlled crouching and early contact transitions are normal in
  the first 45 frames. Afterwards, lost contact, or an item still on its
  support at 90 frames, is a failed grasp: retry_grasp.
- PLACE: an item resting on the floor is not retained: retry_grasp.
  Confirm over at least three observations. Use retry_place only when
  transport cannot continue.
- RELEASE: an item still held after 75 frames: retry_release.
- Visibility gate: the item and a hand must be visible in the newest
  frame; otherwise retention and lift are inconclusive.
- A non-none action cannot be nominal; name the cause matching it.

Action meanings: none continues the current expert; retry_grasp,
retry_place, and retry_release reinitialize that stage's expert;
retry_grounding returns to the active grounding role; terminate ends
the episode.

Allowed actions: <stage-specific list>. Reply ONLY:
{"verdict":"nominal|degraded|failure|unsafe",
 "action":"<allowed action>","confidence":<0..1>,
 "intervention_cause":"none|body_collision|object_or_grasp_failure|
   place_failure|balance_failure|release_failure|grounding_failure",
 "reason_code":"<code>","summary":"<visible change between frames>"}
```

Raw model text is never executed. The runtime extracts one JSON proposal and applies it only if its action belongs to the admissible set recomputed from live state, its cause matches the action, and its confidence is at least 0.65; otherwise the current expert keeps running. Inference is asynchronous: the active expert continues at control rate while a request is pending.

A.3.5 FAILURE DETECTION AND RECOVERY

Detection. Failures are detected from first-person signals only. Two tactile checks cover grasping and transport. The grasp stage hands over to transport only after both hands have risen at least 3 cm since first contact and the net external force on the hands is at least 0.05 N; two hands clasped without an object cancel each other and do not pass. If this confirmation is not reached within 10 s, the grasp is declared unsecured. During transport and placement, a net hand force below 0.05 N for 1 s declares the item lost. A fall is detected from the pelvis-height ratio and body tilt. The VLM

execution monitor can additionally report an item or balance failure that these checks miss, but it cannot declare success on its own. The net-force check measures contact with anything outside the humanoid, not object load: a hand pressed against a support can pass confirmation, and a loss during transport can go unnoticed while one hand still touches the environment.

A confirmed failure retains the instruction, the role bindings, and the interrupted stage, and recovery restores the physical preconditions of that stage before the task resumes.

Drop onto the support. When a grasp or transport failure leaves the item on its elevated support, for example on the tabletop, `retry_grasp` starts a support re-approach. The humanoid first fixes its gaze on the item, then shifts its viewpoint by a small lateral offset so that the next attempt starts from a different view, and finally repeats the grasp with the nominal grasp expert on a refreshed estimate. If the item was displaced beyond the retained estimate, `retry_grounding` returns the source to the grounding loop instead.

Drop onto the floor. When the item falls to the floor, the runtime first holds the current posture until the item has settled. It then searches through downward head motions and small locomotion adjustments to acquire new views, and the reasoner judges whether the manipulated item is visibly resting on the floor; unrelated floor objects count as negative. The item is re-localized, and a floor re-approach executes a floor search, a viewpoint shift, and a floor grasp with a dedicated floor-grasp expert that is allowed a lower root height. After a stable lift, the placement stage resumes with the retained destination.

Whole-body fall. The runtime dispatches the get-up policy while retaining the instruction, target records, and interrupted stage (Appendix A.3.6.1). The same fall check runs when a re-approach begins, so a humanoid that is still low or tilted after a failed grasp first stands up. After the get-up passes its height, orientation, and persistence checks, current evidence determines whether the interrupted stage resumes, the target is re-localized, or the episode terminates.

Termination. An episode ends as a failure when a recovery strategy exhausts its retry budget, a get-up does not complete within its time limit, the controller's indoor safety stop triggers, or an interrupted stage cannot be restarted. Section B.5 reports how often each detection fires and how each failed episode ends.

Illustrated recovery thresholds. For the main-paper sensor trace, contact-loss checks use a 120-step contact window. Get-up starts when normalized pelvis height falls below 0.55; completion requires height at least 0.88 and tilt at most 45° for ten consecutive steps. The low posture during floor grasp does not trigger get-up.

A.3.6 Physical Expert Policies

Actions are executed by the physical experts in Table S6. For each expert, the *preconditions* are the state conditions the runtime checks before dispatching it, and the *completion and failure evidence* are the first-person signals from which the runtime and the execution monitor decide that it has succeeded or failed; these signals form the outcome y_{k+1} returned to the task state.

Table S6: Physical experts, the conditions under which they are dispatched, and the first-person evidence of their completion or failure.

Expert	Goal	Preconditions	Completion/failure evidence
Navigate/ approach (head tracking)	head trajectory along a collision-free route to the interaction pose	metric goal available; humanoid upright	arrival at the interaction pose; blocked or missing route; loss of pelvis height

Expert	Goal	Preconditions	Completion/failure evidence
Gaze adjustment (head tracking)	head reference toward a candidate or sweep view: focus gaze, stabilize view, lateral view shift, in-place search sweep, downward floor search	humanoid upright; a bound candidate or cached sweep view	head reaches the requested view and the camera is stable for capture; otherwise the maneuver is reported as failed
Crouched view (head tracking)	lowered head reference facing the candidate region	source candidate in view; humanoid upright	head reaches the lowered height with a stable view; tracking lag is reported
Stance refinement and posture hold (full-body tracking)	full-body reference pose	a grasp is about to start, or a dropped item is still moving	pose, or dropped item, has settled
Tabletop grasp	object pose on an elevated support	source localized on the support; humanoid at the interaction pose	success: both hands rise at least 3 cm with net external hand force of at least 0.05 N; failure: no such confirmation within 10 s
Floor grasp	object pose on the floor	item localized on the floor; controlled low reach	stable lift at the lower admissible root height; contact loss or balance degradation
Carry/place	route to the destination and placement pose on the verified support	item held; destination localized	success: item resting on the destination; failure: net hand force below 0.05 N for 1 s, or transport blocked
Release	withdrawal of the hands from the placed item	item placed and stable	success: item supported independently while hand contact ends; failure: item still held after 75 frames
Get up	upright posture from the current fallen state	fall detected from pelvis height and tilt	height, orientation, and persistence checks pass; failure: timeout

Controller adapters define the expert identity, input construction, applicability checks, termination conditions, and summarized outcome. Adding an expert requires a compatible policy and adapter, including reference construction and recovery handling; sharing a task-level contract does not imply identical policy architectures or observation formats.

A.3.6.1 SKILL-SPECIFIC REWARD DETAILS

The PULSE-X-based motion and manipulation experts use the latent-action hierarchy of OmniGrasp (Luo et al., 2024a), which extends PULSE (Luo et al., 2024b) to dexterous humanoid control. Full-body and head-tracking policies use AMASS motion references (Mahmood et al., 2019). A task policy predicts a residual latent action, and the pretrained prior and decoder provide the motor representation. The get-up policy uses the same latent-action hierarchy without an object or reference-motion goal; its reward is given separately below.

OmniGrasp latent-action formulation. Following Equation (3) of OmniGrasp, adapted only to index our PULSE-X-based experts,

$$\boldsymbol{a}_t^i = D^{\text{PULSE-X}}\big(\pi_i(\boldsymbol{z}_t^i \mid \boldsymbol{s}_t, \boldsymbol{g}_t^i) + \boldsymbol{\mu}_t^p\big)\,, \tag{S7}$$

where the policy output is a residual latent code, $\boldsymbol{\mu}_t^p$ is the conditional prior mean, and the decoder produces a PD target. Task-specific goal features can include future references and perceptual object estimates; recurrent policy state is implicit in this notation. The decoder also conditions on

proprioception, suppressed in this compact notation. This formulation does not impose an additional latent-norm projection. Our execution interface supplies perceptual object estimates when constructing object-related goal features; it does not require simulator object-state truth at execution time. Privileged quantities used to train or score a policy are distinct from its permitted execution inputs.

High-level training objective. Each expert π_i is trained to maximize the skill-level discounted return

$$J(\pi_i) = \mathbb{E}_{\tau\sim\pi_i}\left[\sum_{t=0}^{T-1}\gamma^t r_t^i\right] \tag{S8}$$

In the OmniGrasp formulation, the high-level task policy optimizes discounted return while the pretrained prior and decoder remain fixed:

$$\phi_i^\star = \arg\max_{\phi_i}\mathbb{E}_{\tau\sim\pi_{\phi_i}}\left[\sum_{t=0}^{T-1}\gamma^t r_t^i\right]. \tag{S9}$$

The rollout τ is induced by the policy, its recurrent state when present, the fixed decoder, and environment dynamics; r_t^i is the skill reward and γ the discount. The same objective applies to the get-up policy with the reward in Equation S15. The PD target $\boldsymbol{a}_t^i$ in Equation S7 is distinct from a task-level action $\widetilde{a}_k$.

Expert reward vocabulary. For tracked entities $\mathcal{E}$ and $y \in \{p, R, v, \omega\}$, let

$$\begin{aligned} e_y(\mathcal{E}) &= \frac{1}{|\mathcal{E}|}\sum_{j\in\mathcal{E}} d_y(y_{j,t}, y_{j,t}^*)^2, \\ \mathcal{T}(\mathcal{E}) &= \sum_y w_y \exp[-k_y e_y(\mathcal{E})]. \end{aligned} \tag{S10}$$

where d_R is geodesic rotation error and the other d_y are Euclidean. A skill combines the relevant tracking kernels and event terms, with optional power $\mathcal{P}_t = \sum_n |\tau_{n,t}\dot{q}_{n,t}|$ and contact-slippage $\mathcal{S}_t$ penalties:

$$r_t^i = \sum_m \alpha_{i,m}\mathcal{T}(\mathcal{E}_{i,m}) + b_i(\chi_t) - \lambda_P\mathcal{P}_t - \lambda_S\mathcal{S}_t. \tag{S11}$$

We use $\mathcal{H}$, $\mathcal{C}$, $\{O\}$, and $\mathcal{B}$ for the head, active hands, object, and full-body tracking sets. Navigation uses $r_t^{\text{nav}} = \mathcal{T}(\mathcal{H}) - \lambda_P\mathcal{P}_t$ on the head reference induced by the collision-aware route. Tabletop and floor grasp use the same two-phase manipulation objective. The grasp policy is additionally trained with objects resting on support surfaces at multiple heights. Before contact, active hands receive distance-progress reward when far from the demonstrated contact poses and position–orientation tracking when near them. After contact, grasp, carry, and place track the reference object state:

$$\begin{aligned} r_t^{\text{man}} = c_t &\left[\sum_{y\in\{p,R,v,\omega\}} w_y e^{-k_y e_y(\{O\})} + w_c\right] \\ &- \lambda_P\mathcal{P}_t - \lambda_S\mathcal{S}_t, \end{aligned} \tag{S12}$$

where c_t requires a consistent hand–object contact/lift proxy. Tabletop and floor grasp use elevated and ground-level references, respectively; the latter is drawn from ground-level initial states and low-reaching motions. Carry and place instantiate r_t^{man} with transport-route and support-settling object references. Release instead tracks a static withdrawal reference and adds a terminal bonus after the withdrawal remains stable for a fixed interval.

The remaining pre-contact and release rewards are defined as follows:

Let $\mathcal{C}$ denote the active hands, $d_t^{\mathcal{C}}$ their mean distance to the demonstrated contact poses, and $\delta_{\mathcal{C}}$ the near-contact threshold. The pre-contact grasp reward switches from bounded distance progress to position–orientation tracking:

$$r_t^{\text{pre}} = \begin{cases} 0.9e^{-k_p e_p(\mathcal{C})} + 0.1e^{-k_R e_R(\mathcal{C})}, & d_t^{\mathcal{C}} \le \delta_{\mathcal{C}}, \\ \text{clip}(d_{t-1}^{\mathcal{C}} - d_t^{\mathcal{C}}, 0, \bar{\Delta}), & d_t^{\mathcal{C}} > \delta_{\mathcal{C}}. \end{cases} \tag{S13}$$

Release tracks a static full-body withdrawal reference $\mathcal{B}^*$ and adds a terminal hold bonus after both the tracking and duration thresholds are satisfied:

$$r_t^{\text{rel}} = \mathcal{T}(\mathcal{B}) + b_{\text{rel}}\mathbf{1}[T_t^{\text{rel}} \geq \bar{T} \wedge \mathcal{T} \geq \bar{r}] - \lambda_P \mathcal{P}_t. \tag{S14}$$

Get-up policy. The get-up expert builds on the fail-state recovery setting of PHC (Luo et al., 2023). PHC creates fallen states by randomly dropping the humanoid and trains a recovery primitive with $r_t^{\text{g-recover}} = 0.5\, r_t^{\text{g-point}} + 0.5\, r_t^{\text{amp}} + 0.1\, r_t^{\text{energy}}$, where $r_t^{\text{g-point}} = d_{t-1} - d_t$ rewards reducing the distance d_t between the simulated and reference roots, so that the humanoid gets up and returns to motion imitation. Our get-up expert keeps the fallen-state training setting but has no reference motion to approach and acts in the PULSE-X latent space. Each episode starts from a random root orientation and random joint configuration that settles under random joint targets for 3 s before control begins. The policy input is normalized proprioception together with a fixed condition embedding, and a recurrent actor (an MLP followed by a GRU) outputs the residual latent action decoded by Equation S7.

The reward couples head height with foot support. Let $c_t^{\text{L}}, c_t^{\text{R}} \in \{0, 1\}$ indicate that the left or right sole has contact force above τ_f, let $\nu_t = c_t^{\text{L}} + c_t^{\text{R}}$, and let z_t^{head} denote head height. Then

$$r_t^{\text{up}} = w_{\text{ch}}\, \text{sig}(\lambda_{\text{c}}\,(\nu_t - \nu_0))\, \beta\, z_t^{\text{head}} + C\, c_t^{\text{L}}\, c_t^{\text{R}}, \tag{S15}$$

where sig is the logistic function, $\lambda_{\text{c}} = 4$, $\nu_0 = 1.5$, $\beta = 1$, $C = 1$, $w_{\text{ch}} = 1$, and $\tau_f = 1\,\text{N}$. The sigmoid gate withholds most of the head-height reward until both soles support the body, which discourages raising the head while lying or kneeling. The discriminator reward, a reference-pose term, and a power penalty are disabled. Episodes last 300 control steps and do not terminate on low posture, because lying on the ground is the initial state. These values are the settings recorded in the training workspace when the deployed checkpoint was saved; that workspace is not version-controlled, so earlier training segments may have used different settings. At deployment, completion is decided by the runtime's upright checks in the get-up recovery protocol, not by this reward.

The reward quantities above concern training the PULSE-X-based experts. The agent observes semantic progress and completion/failure summaries rather than reward values.

B Experimental Protocols and Extended Evaluations

This section provides the experimental protocol and additional evaluations of controlled recovery, actionable-grounding sensitivity, and local VLM reasoning.

B.1 Experimental Protocol and Reproducibility

B.1.1 Data Sources, Task Manifest, and Preprocessing

The experiments use AMASS for full-body and head-tracking policy training and ReplicaCAD for furnished indoor scenes, as cited and motivated in the main paper. We introduce no new benchmark dataset. The paper-specific grounding data consist of a 100-entry experimental manifest that binds existing scene and object assets to source and destination roles, colors, world initializations, and generated relocation instructions. This manifest, the prompt-development records, the split metadata, and the required source code will be released under a license permitting free research use; raw AMASS, ReplicaCAD, GRAB, SMPL, and SMPL-X assets are not redistributed and must be obtained from their official sources under their respective licenses.

The preprocessing scripts extract the required SMPL-H fields, apply the documented joint ordering and upright coordinate convention, resample to 30 Hz where required, and serialize the motion records used by the loaders. ReplicaCAD scene instances retain their official mesh and scene descriptions. Runtime utilities construct collision geometry, occupancy grids, and sampled camera or head trajectories from those descriptions. The perception-data generator renders the task manifest in the same ReplicaCAD/GRAB placement and camera code path used by evaluation; records used for prompt or detector development are identified separately and do not enter the final test-set denominators.

B.1.2 GROUNDING DATA SPLITS

The localization tables in the main paper report the complete 100-task development split and the complete 100-task held-out test split in every reported run; the two splits share one scene and differ in object placement and instruction instances. Prompt development and failure analysis use the development split. No task of one split enters the metrics of the other. The main grounding comparisons use the full splits; the 30-task Luna action ablation is described in Appendix B.2. The separate 30-case end-to-end evaluation uniformly samples task-level outcomes from deterministic Qwen-FT localization results and uses Luna for execution (Table 5). Its stage-success counts are cumulative over all sampled cases, including those that fail localization.

Closed-loop grounding starts the humanoid at its annotated pose in the reconstructed Isaac Gym scene. Instructions identify source objects and destination supports by category, color, or spatial relation. The agent can request a 360-degree egocentric sweep, crops, candidate-centred gaze, bounded physical approach, and viewpoint shifts; movement uses frozen whole-body experts shared across conditions. Episodes end when both roles are committed, at abstention, or after 1200 s; downstream grasping and placement are not scored in this evaluation.

The four direct-prediction rows in the main table use three Luna runs over the same frozen RGB evidence: the 360-degree sweep recorded at the start pose of each development task. Each run issues fresh model calls over these immutable inputs rather than re-executing the simulator, which preserves identical visual evidence across stochastic runs.

The initial-view and cached-360 conditions use the initial RGB observation or cached sweep, respectively. Each is evaluated as one-shot prediction or with rendered appearance references of the source and verifier re-grounding. No physical actions are executed. These references differ from the full agent in both reasoning procedure and evidence acquisition; the source-action ablation instead holds reasoning, audit, and geometry fixed while restricting the permitted physical actions.

B.1.3 RANDOMNESS AND EVALUATION RECORDS

The physics and learning entry point seeds Python `random`, NumPy, PyTorch, and all CUDA devices; the base training configuration uses seed 0 and the fixed-RGB grounding comparison uses case-order seed 20260729. Seed sufficiency is partial: the released configuration sets `torch_deterministic=False`, and the Luna endpoint exposes no reproducible sampling seed; the fine-tuned Qwen3.8-27B reasoner is decoded deterministically.

Some attempts fail for reasons unrelated to the system under test, such as a lost connection to the model endpoint or a failure to write the result files. To exclude such failures without permitting selective reruns, an attempt is counted as infrastructure-invalid only when it ends because of such a transport or packaging failure before producing its result record. Invalid attempts are logged, excluded from both numerator and denominator, and rerun on the same task. All other failures, including abstentions, verifier rejections, localization failures, timeouts within a valid trial, and physical failures, count as task failures and are never rerun.

B.1.4 METRICS AND RESULT SUMMARIES

Loc@2cm, the primary grounding metric, requires the committed source position, measured at capture time, to lie within 2 cm of the true object position. Direct prediction commits a box without mesh registration, so its source position is recovered from the committed box and depth and scored on horizontal (XY) error. *Destination* scores semantic identification of the intended support, not metric placement accuracy. An abstained or timed-out task counts as wrong for both metrics; abstention is still reported separately, because refusing an unsupported commitment is behaviorally different from committing the wrong entity. The 2 cm threshold is motivated by the grasp expert's sensitivity to position error (Table S10), but meeting it does not guarantee grasp success.

Repeated runs of the same task are not independent samples. Each Luna condition is therefore summarized by the mean and sample standard deviation of its per-run rates rather than pooled across runs, and Qwen-FT, which decodes deterministically, is run once. Ablation conditions keep the same tasks and initial scene states. The recovery experiments have small event counts; their proportions are reported with numerators and denominators and without significance claims.

B.1.5 COMPUTING INFRASTRUCTURE

The historical mobile-manipulation evaluations use an NVIDIA GeForce RTX 4080 SUPER. The Qwen3.8-27B adaptation uses three NVIDIA A100-SXM4 80GB GPUs; serving uses vLLM with a BF16 base and the selected LoRA adapter. The recorded training environment uses Python 3.11, PyTorch 2.5.1 with CUDA 12.1, Transformers 5.6.0.dev0, PEFT 0.19.0, and bitsandbytes 0.49.2. These settings are distinct from the historical control experiments: the grasp policy is trained on an NVIDIA A800 and the remaining policies on an NVIDIA GeForce RTX 3090. The simulator is NVIDIA Isaac Gym.

B.2 ADDITIONAL GROUNDING ABLATIONS

B.2.1 ACTION ABLATIONS AND OFFLINE REPLAY

The ablation in the main text uses Qwen-FT. To check that the gain from embodied actions is not specific to the fine-tuned model, we repeat it with Luna on 30 development tasks selected in advance, stratified by the full-agent outcome (seed 20260924), with one run per condition. Table S7 (top) shows the same ordering: *sweep only* localizes 23.3% of sources, + *in-place gaze* 60.0%, and the full agent 83.3%.

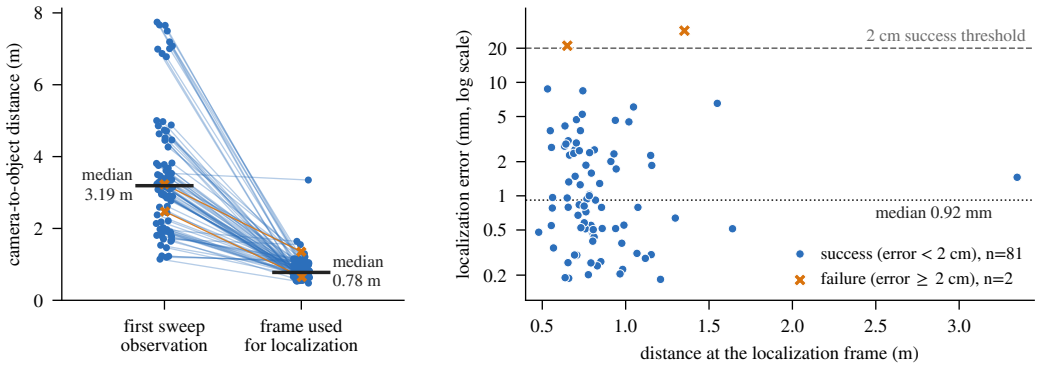
The live ablation shows that source motion matters, but not which motions contribute. An offline action-prefix replay of the recorded Luna run on the development split addresses this: it commits the source candidate held after the first k source actions and scores it with the same geometry back end, without new model calls (Table S7, bottom). The first approach raises Loc@2cm from 53% to 73%; the second action raises it to 74%, and the remaining actions bring the full-run rate to 83%. On the 77 tasks with a first approach, that approach reduces the camera-to-source distance from 3.19 m to 1.26 m, enlarges the median source box from 576 to 2304 pixels, and increases the ICP points observed on the source from 121 to 500. The closer view thus supplies the evidence that verification and metric geometry require.

Table S7: Embodied evidence ablation for source localization with Luna. *Top:* live runs on 30 development tasks. *Bottom:* offline replay of the recorded full-agent run on the development split, committing the source candidate held after k source actions. †The k=0 commit bypasses the lock audit and is an optimistic bound: on the 30 live tasks it gives 18, matching + *in-place gaze*, whereas *sweep only* reaches 7; it is not a no-action estimate.

Condition	Source translation	N	Loc@2cm (%) ↑
Live tasks			
Sweep only	–	30	23.3
+ in-place gaze	–	30	60.0
Full agent (Ours)	approach / shift	30	**83.3**
Action-prefix replay of the recorded run			
k=0 source actions†	–	100	53.0
k=1	approach	100	73.0
k=2	approach / shift	100	74.0
Full agent (recorded)	approach / shift	100	**83.0**

B.2.2 APPROACH DISTANCE AND VISUAL EVIDENCE

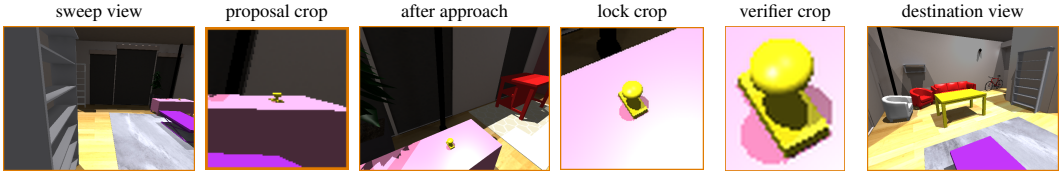
Figure S4 describes observation distances for the 83 tasks with an estimate; the remaining 17 produced none. These conditional statistics complement the controlled action ablation and offline replay, rather than independently establishing the benefit of approach. Figure S5 shows successful test sequences using the RGB inputs actually received by the agent. Distances and localization errors are evaluation-only quantities.



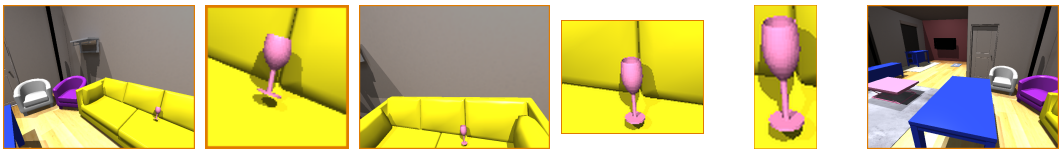
(a) Distance at the first sweep observation and at the localization frame

(b) Localization error against the distance at the localization frame

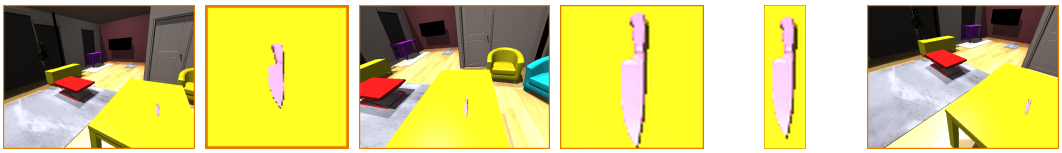
Figure S4: Camera-to-object distance and localization error across the development tasks (one run of the full agent with Luna), for the 83 tasks that produced a strict estimate; the remaining 17 tasks failed without an estimate. Distances are horizontal, between the head camera and the source; the localization frame is the frame used for mesh registration. Blue circles are successes and orange crosses failures; black bars and the dotted line mark medians, and the dashed line is the 2 cm success threshold. Distances and errors use the true object position and are for evaluation only.



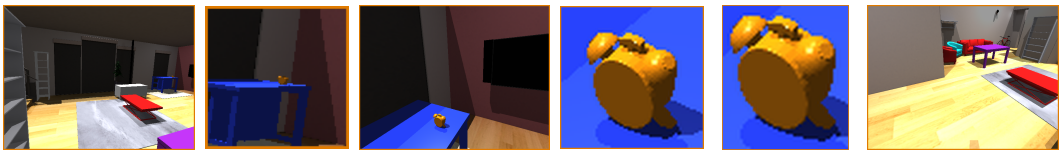
(a) "find the yellow doorknob and put it on the red sofa". Step 84: select proposal → step 1033: lock source; lock crop audit: accept → final verifier: accept, commit → step 2672: lock destination. Source error 0.9 mm (evaluation only).



(b) "find the pink wine glass and put it on the blue table". Step 138: select proposal → step 1036: lock source; lock crop audit: accept → final verifier: accept, commit → step 3221: lock destination. Source error 2.4 mm (evaluation only).



(c) "find the pink knife and put it on the yellow tv stand". Step 96: select proposal → step 644: lock source; lock crop audit: accept → final verifier: accept, commit → step 2372: lock destination. Source error 0.2 mm (evaluation only).

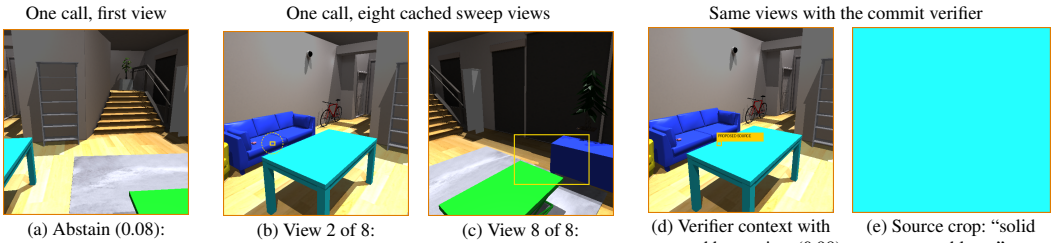


(d) "find the orange alarm clock and put it on the cyan chair" (fine-tuned student VLM). Step 136: select proposal → step 1164: lock source; lock crop audit: accept → final verifier: accept, commit → step 2998: lock destination. Source error 0.4 mm (evaluation only).

Figure S5: **What the VLM actually sees during Seek.** Each subfigure is one successful test episode, and every image is a complete RGB input attached to a recorded VLM call (orange border). From a passive sweep view in which the target occupies only a few pixels (left), the agent selects a VLM-proposed candidate, walks toward it, locks the source from a close-range view, audits the lock crop, passes the shared verifier, and then locks the destination; column headings give the input role, and subcaptions give the instruction, the policy steps, and the decision returned by each call. Crops are separate VLM attachments, not zooms added for the figure; lock crops share the step of their full frame. Rows (a)–(c) use the frozen VLM and row (d) the fine-tuned student. Source errors are evaluation-only scores, never shown to the agent.

B.2.3 VERIFICATION, FAILURES, AND ABSTENTION

Figure S6 illustrates verification converting a wrong commitment into abstention. Figure S7 shows failures and budget-limited abstentions; these cases complement the successful sequences above and the pre-commit revisions in Appendix A.2.



(a) Abstain (0.08): "missing RGB evidence"

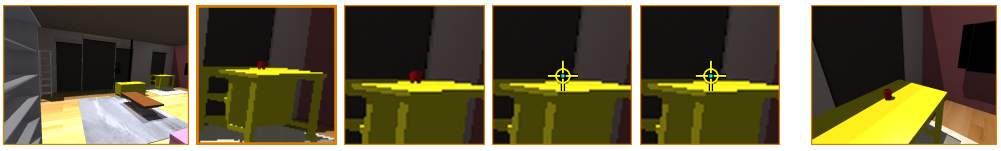
(b) View 2 of 8: committed source box (0.53), 0.35 m from the object

(c) View 8 of 8: committed destination box

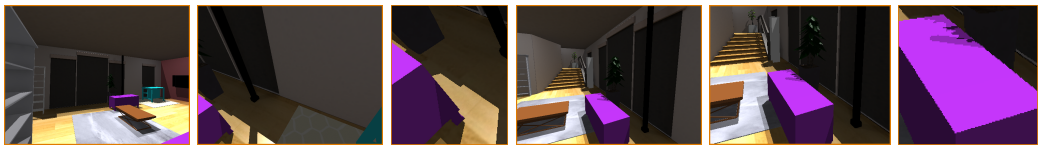
(d) Verifier context with proposed box: reject (0.98)

(e) Source crop: "solid cyan tabletop"

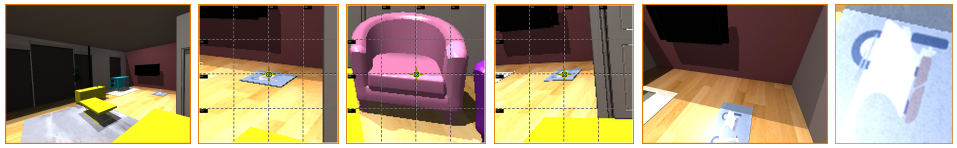
Figure S6: A commit verifier converts a wrong commitment into an abstention. (a) From the first view alone the VLM abstains because neither object is visible. (b, c) From eight cached sweep views, a single call commits to a pink blob on the sofa as the camera; scoring shows the source box contains no target pixels. (d, e) With the verifier, each proposed box is cropped and checked; all three proposals are rejected (the first check is shown), and the system abstains. Orange borders mark the exact VLM inputs; yellow marks in (b, c) are the committed boxes, drawn for the figure (the dashed circle locates the small source box), whereas the box in (d) is part of the verifier input.



(a) "find the red mug and put it on the brown table", approach disabled. Step 73: sweep, proposal crop (select) → lock crop (lock) → two marker checks (reject: point on table) → abstain. Right: separate run with approach, step 1037: lock, commit; error 0.09 mm (eval. only).



(b) "find the purple frying pan and put it on the orange chair". Step 73 sweep view 3 of 16 (inspect far table) → step 1255 after approach (lock) → verifier crop (reject: cut off) → step 3852 re-approach (re-lock) → lock crop, native detail (reject: furniture) → abstain.



(c) "find the blue stamp and put it on the yellow table". Step 81 sweep view (select proposal) → proposals at steps 81 (rug) and 89 (pink chair) rejected → step 89 proposal accepted (blue floor item) → step 1822 after approach (lock source) → detector crop rejected → abstain.

Figure S7: **Grounding failures and abstentions.** Every image is a complete VLM input (orange border); subcaptions give the policy step, the input, and the returned decision, left to right. (a) With physical approach disabled, the red mug remains a few pixels tall; the agent locks it from the sweep view, but two marker checks judge the selected point to lie on the table rather than the mug, and the episode ends in abstention. In a separate run with approach enabled, the same target is locked from close range (right, separated by a wider gap). (b) After inspecting a distant table, the agent approaches and locks a purple piece of furniture; the verifier rejects its crop as cut off and, after re-approach, the lock audit rejects it as furniture rather than a frying pan, exhausting the budget. (c) Proposals on a rug and a pink chair are refused, but a later check accepts a flat blue floor item; after approaching, the detector-identity check rejects it as a flat card rather than a stamp, and the agent abstains instead of committing a wrong object. No source pose was exported in any of the three failed episodes (evaluation-only).

B.3 Explicit Geometry and Grasp Sensitivity

Explicit geometric perception links perception to execution by improving the spatial accuracy of visual grounding and providing more precise goals for physical control. Section B.3.2 separates 2D region accuracy from 3D localization and evaluates the benefit of shape-based registration. Section B.3.3 then fixes the visual evidence to attribute this benefit to shape alignment, iterative refinement, and geometric processing. Finally, Section B.3.4 tests how position error affects grasp success through controlled bias injection.

B.3.1 Region Detector

Region detection uses YOLOv8n (Jocher et al., 2023) with coverage of 51 GRAB object categories. The 51-category model is initialized from COCO-pretrained weights and fine-tuned for 30 epochs with its first ten layers frozen. Its data comprise 7,260 training and 2,420 validation head-camera frames rendered during walking, object approach, gaze changes, and dropped-object search in ReplicaCAD, with original or randomized furniture appearance. Boxes are derived from simulator segmentation masks used as training labels.

After the VLM locks a source region, detector proposals are associated spatially with that region and filtered by safety checks, with matching class labels preferred. A mismatched label does not automatically reject a proposal: the VLM examines the full RGB image, the proposed crop, and appearance references to verify the requested object's identity before the region is used for depth extraction and mesh registration. For example, in selected archived runs, Luna accepted a mug region labelled "cup" and Qwen-FT accepted an apple region labelled "piggybank"; their final source-position errors were 4.42 mm and 0.76 mm, respectively. These examples demonstrate tolerance to detector misclassification when a usable region is available.

The fixed-view geometry comparison provides complementary evidence: Luna regions with mesh ICP yield a median error of 0.113 cm, versus 0.128 cm for the recorded YOLO regions (Table 3). This supports the usefulness of VLM-proposed regions for metric estimation. When the detector supplies no usable proposal, the system falls back to the VLM-proposed region for geometric localization. Thus, YOLO primarily improves geometric precision and provides complementary evidence for mutual verification with the VLM, while preserving the VLM's ability to interpret open-ended instructions.

B.3.2 Separating Region Precision from Metric Localization

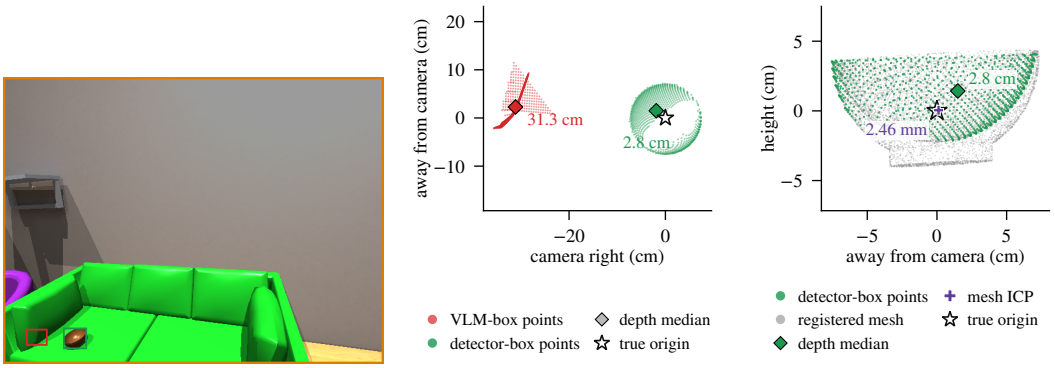
On 50 fixed target views from one scene, we compare untuned and adapted Qwen3.8-27B and Luna with the system's recorded YOLO regions (Jocher et al., 2023) (Table S8). Each VLM receives the same single RGB image and target instruction, predicts a box in normalized coordinates, and receives neither depth nor ground truth. For the complementary 3D analysis, Table 3 uses the archived pixel-coordinate Luna predictions on these same views, crossing region source with depth-median estimation or canonical-mesh ICP (Besl & McKay, 1992). Error is measured at the canonical object origin; missing estimates count as failures. This analysis uses one prediction per task and configuration, with identical foreground points for the two estimators within each region.

Table S8: Region accuracy on 50 fixed target views. Threshold columns report percentages over all 50 views; missing boxes receive zero IoU. VLM coordinates are normalized to $[0, 1000]$ and converted to pixels for scoring. Luna reports mean $\pm$ sample SD over three runs with medium reasoning; other rows use one run, with Qwen in non-thinking mode. YOLO includes the system's recorded target association. Bold marks the best mean in each column.

Box source	Mean IoU	IoU ≥ 0.5 (%)	IoU ≥ 0.75 (%)	IoU ≥ 0.9 (%)
Qwen3.8-27B	0.812	98	82	14
Qwen3.8-27B (fine-tuned)	0.777	98	72	8
Luna	0.849 ± 0.005	95.3 ± 1.2	88.0 ± 2.0	53.3 ± 6.4
YOLO (recorded selection)	**0.895**	**100**	**96**	**58**

On these fixed views, direct box predictions from the evaluated VLMs are less precise than YOLO regions. Together with the gains from mesh registration (Section B.3.3), this supports separating VLM-based semantic reasoning from region refinement and explicit metric geometry.

Figure S8 illustrates the distinction between a visible-surface estimate and the canonical object origin.



(a) VLM box (red), detector box (green)

(b) Depth median (top view)

(c) Mesh registration (side view)

Figure S8: From image regions to metric goals. (a) VLM input with overlaid predictions; (b) depth medians estimate visible surfaces, whereas (c) mesh registration estimates the object origin. Stars mark ground-truth origins used only for evaluation.

B.3.3 Attributing Metric Precision

To identify where metric precision comes from, we hold the RGB-D observation, detector region, calibration, and canonical object mesh fixed and vary only the geometric estimator. We replay 246 semantically committed source records from three runs on the same development tasks, without new VLM calls or physical rollouts. Each estimator uses the same source records; repeated records are not independent test tasks. We measure 3D error at the canonical object origin, using thresholds of 2 and 10 cm.

The estimator ladder progresses from center-pixel depth and depth median, through an upright support-height prior and non-iterative mesh alignment, to full registration. The final step adds iterative refinement and fit acceptance on the same cleaned point cloud. Separate ablations remove support-plane estimation, RGB filtering, or depth-visibility refinement; a fallback relaxes RGB filtering when too few points remain.

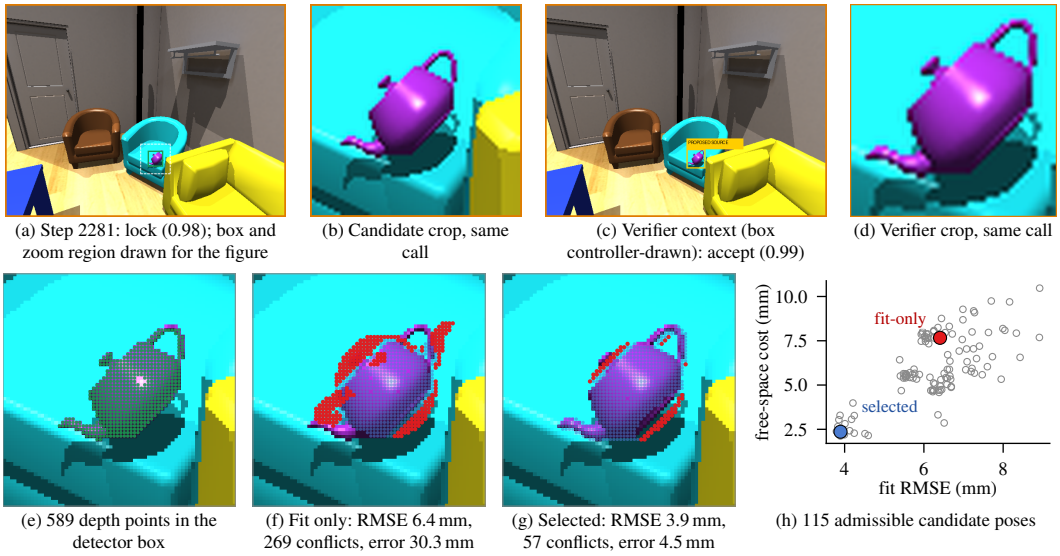
Table S9: Metric localization on fixed evidence, reported as mean ± SD across three runs (83, 76, and 87 records). Missing estimates count as failures; error quantiles use available estimates. Archived outputs are separated from replay with an updated backend. Oracle rows select hypotheses using ground truth and cover only records with available hypotheses (83, 76, and 85); they are diagnostics, not deployable methods. Bold indicates the best non-oracle mean in each column, including ties.

Estimator / processing variant	Available (%)	< 2 cm (%)	< 10 cm (%)	Median (mm)	P90 (mm)
Archived full-backend output	99.2±1.3	97.7±2.3	99.2±1.3	**0.77±0.13**	4.03±0.40
Center-pixel depth	**100.0±0.0**	31.0±1.7	99.6±0.8	27.39±1.43	60.45±0.44
Depth median	**100.0±0.0**	35.0±2.1	**100.0±0.0**	23.75±0.55	46.48±2.09
Depth median + support-height prior	**100.0±0.0**	47.1±1.3	99.6±0.7	20.74±0.88	40.99±3.78
Mesh alignment without iteration	99.2±1.3	94.4±1.5	99.2±1.3	4.75±0.45	14.96±1.90
Full registration (recomputed)	99.2±1.3	98.1±2.4	99.2±1.3	**0.77±0.13**	3.91±0.19
without support-plane removal	95.2±1.8	92.8±2.0	95.2±1.8	0.81±0.15	4.64±0.78
without RGB chromaticity filtering	**100.0±0.0**	96.0±1.6	**100.0±0.0**	1.27±0.27	8.65±3.31
with low-retention fallback	**100.0±0.0**	**98.4±1.8**	**100.0±0.0**	**0.77±0.13**	**3.72±0.55**
without depth-visibility refinement	99.2±1.3	93.6±1.4	99.2±1.3	**0.77±0.13**	11.26±2.95
Oracle initial-pose selection	100.0±0.0	97.6±1.2	100.0±0.0	3.71±0.22	10.93±1.24
Oracle visibility-hypothesis selection	100.0±0.0	100.0±0.0	100.0±0.0	0.40±0.01	0.95±0.05

Table S9 shows that shape alignment provides the largest improvement over depth-only estimates, while full registration further improves precision. The gain therefore cannot be attributed to ICP alone. Support-plane removal and visibility refinement are useful, whereas RGB filtering has mixed effects. The coarse 10 cm threshold obscures differences that the stricter threshold reveals.

Ground-truth-assisted hypothesis selection indicates remaining room to improve candidate ranking, but is diagnostic only. These results explain localization precision on fixed observations, not the effect of backend changes on closed-loop task success.

Depth-visibility example. Figure S9 illustrates why matching the observed surface alone is insufficient: depth-visibility evaluation penalizes candidate poses that place model surface in measured free space.



(a) Step 2281: lock (0.98); box and zoom region drawn for the figure
(b) Candidate crop, same call
(c) Verifier context (box controller-drawn): accept (0.99)
(d) Verifier crop, same call
(e) 589 depth points in the detector box
(f) Fit only: RMSE 6.4 mm, 269 conflicts, error 30.3 mm
(g) Selected: RMSE 3.9 mm, 57 conflicts, error 4.5 mm
(h) 115 admissible candidate poses

Figure S9: Depth-consistent mesh registration. (a–d) Exact VLM inputs at the lock step (orange borders): the full frame and candidate crop the VLM locked, and the annotated context and crop the joint verifier accepted. (e–g) Zoom of (a) with overlays drawn for the figure: green dots are observed depth points, blue dots model surface visible to the camera, and red dots free-space conflict pixels, where the model lies more than 1 cm in front of the measured depth. The pose chosen by surface fit alone (f) matches the points but places model surface where the sensor sees free space; the pose selected among the candidates in (h) by combined fit and free-space cost (g) reduces both conflicts and error. Errors use the true object position and serve only for evaluation; no ground truth enters the selection.

B.3.4 Physical Sensitivity to Position Error

We evaluate the frozen OmniGrasp policy (Luo et al., 2024a) on 51 objects in its original tabletop setting, without detector or VLM inference. A constant positional bias is applied to the controller's object-position observations and references while the physical object placement remains unchanged. We test horizontal (XY) and spatial (XYZ) biases of eight magnitudes from 0 to 10 cm over five repetitions, totaling 4,080 trials. Success requires a stable lift exceeding 3 cm with sustained hand contact; all trials, including zero-bias failures, remain in the denominator.

Table S10: Stable grasp success on 51 objects under perceived-position bias. Values are mean ± sample SD over five repetitions.

Bias norm (cm)	XY (%)	XYZ (%)
0	89.8 ± 1.6	88.2 ± 3.7
0.5	87.1 ± 1.8	87.8 ± 4.5
1	89.0 ± 6.3	88.2 ± 3.4
2	80.0 ± 2.9	83.9 ± 4.3
3	74.1 ± 7.0	79.2 ± 6.4
5	54.1 ± 1.1	53.7 ± 7.5
7	22.7 ± 2.6	21.6 ± 4.2
10	9.4 ± 2.6	9.4 ± 1.6

Grasp success degrades as positional bias grows, indicating that coarse localization accuracy is insufficient for reliable grasping. This controlled sensitivity study motivates precise metric grounding but does not directly measure the end-to-end benefit of the localization module.

Figure S10 plots the controlled bias results.

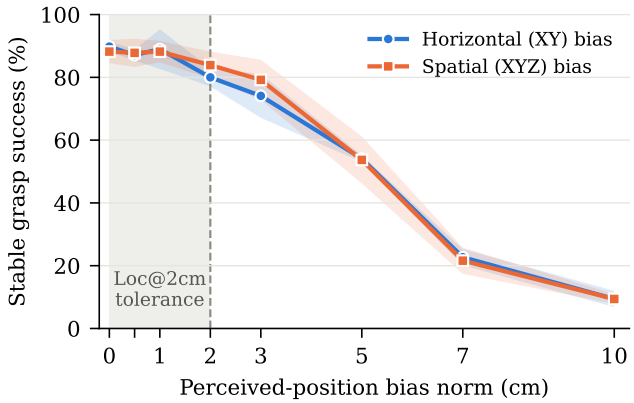
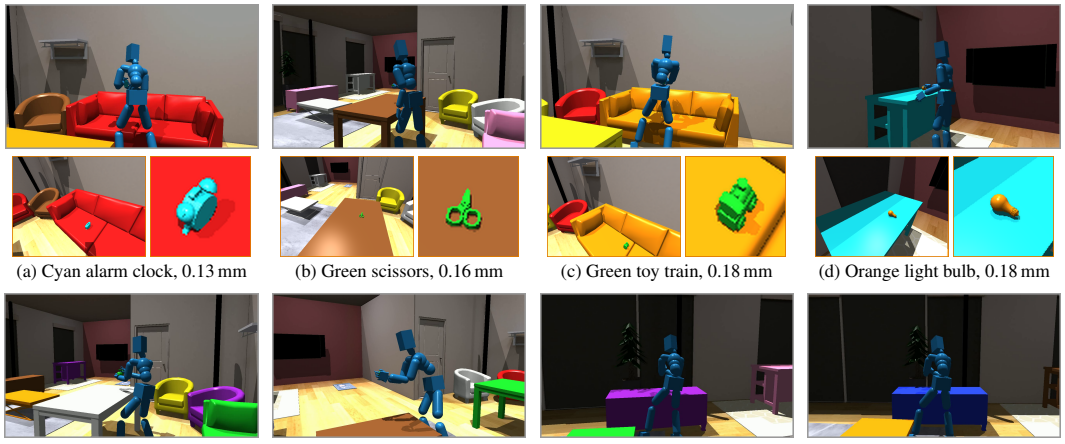


Figure S10: Stable grasp success of the frozen grasp policy on 51 objects under constant perceived-position bias (mean ± sample SD over five repetitions). The shaded region marks the 2 cm tolerance used by Loc@2cm.

B.3.5 Selected Downstream Grasp Examples

Figure S11 shows eight downstream stable lifts selected for the lowest source localization errors. This selected gallery is not a representative sample and does not estimate the causal grasp benefit of the localization module.



(a) Cyan alarm clock, 0.13 mm (b) Green scissors, 0.16 mm (c) Green toy train, 0.18 mm (d) Orange light bulb, 0.18 mm

(e) Green bunny figurine, 0.19 mm (f) Green light bulb, 0.19 mm (g) Orange toothbrush, 0.20 mm (h) Pink duck, 0.22 mm

Figure S11: Eight successful executions with the lowest source localization error (0.13–0.22 mm), selected cases rather than a representative sample; subcaptions give the object and its source localization error. In each panel, top: stable lift (grey border; third-person, presentation only); bottom: the egocentric image and crop the VLM received when committing the source (orange border; every commit had confidence 0.99). In this protocol, success is a stable grasp handed off to transport; placement is not recorded.

B.4 Controlled Failure Recovery

B.4.1 Failure-Injection Protocol and Evaluation Scope

The source-grounding-error, tabletop-drop, and floor-drop conditions in Table 4 all use grasp policies with non-privileged object-state inputs: perception supplies the target estimate and contact-based state estimation supports manipulation, without supplying simulator object-state truth to the grasp policy. Correction in these three conditions requires a stable retained lift. Thus, the source-grounding-error result includes grasp execution after re-localization and cannot be interpreted as localization-only success; a correct location alone does not count as recovery. The whole-body-fall column measures get-up correction separately.

The failure-injection protocol specifies the case, initial physical state, injection time, and perturbation parameters. Injection parameters and simulator-private state are recorded for scoring but are never attached to a VLM request. An injection must produce its declared physical event through the simulator; directly teleporting the system to a recovered state is not allowed.

Table S11: Controlled failure sets, injection definitions, and physical correction criteria. A successful expert dispatch is not itself a physical correction.

Failure set	Intended injection	Physical correction criterion
Source grounding error	Wrong source role, or an over-threshold source-pose offset	Correct source role and metric goal are re-established, and a non-privileged grasp restores a stable retained lift
Tabletop drop	Release the object and allow it to settle on an elevated support	Tabletop regrasp restores a stable retained lift
Floor drop	Apply one downward-velocity impulse and allow PhysX settling	Floor search, localization, and floor grasp restore a stable retained lift
Whole-body fall	Fixed tilt and linear/angular-velocity perturbation	The complete upright guard passes, restoring the physical precondition for stage resumption

Table 4 reports 30 events in each of four categories: source grounding errors, tabletop drops, floor drops, and whole-body falls. Detection is reported for the two drop categories and falls: 29/30 (96.7%), 29/30 (96.7%), and 30/30 (100%), respectively. Detection is not reported for source grounding errors; the dash does not denote a zero detection rate.

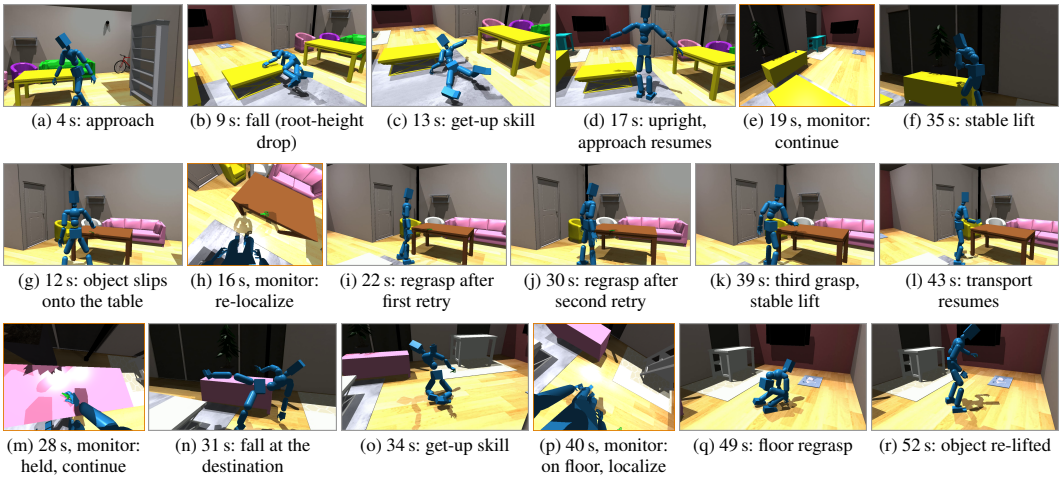
Correct@k is the fraction of all 30 events in a category that meet the physical correction criterion within at most k recovery attempts. Both detection and correction use the full failure-set size as the denominator; correction is not conditioned on successful detection. For source grounding errors, tabletop drops, floor drops, and falls, respectively, Correct@1 corresponds to 21/30 (70.0%), 18/30 (60.0%), 11/30 (36.7%), and 16/30 (53.3%). Correct@4 corresponds to 23/30 (76.7%), 18/30 (60.0%), 15/30 (50.0%), and 25/30 (83.3%). The larger attempt budget yields two additional grounding corrections, no additional tabletop correction, four additional floor-drop corrections, and nine additional fall corrections. Floor drops remain the least frequently corrected category within four attempts, although 29 of the 30 events are detected.

Table S11 specifies the failure definitions and physical correction criteria. Detection, corrective dispatch, physical correction, stage resumption, and final task completion are distinct outcomes. Dispatch alone does not count as correction; a fall is physically corrected when the complete upright guard passes, while resumption additionally requires a valid next-stage target. The reported correction rates do not measure stage resumption or final task completion, nor do they establish the end-to-end benefit of enabling recovery relative to disabling it.

B.4.2 Recovery Examples

Figure S12 illustrates how recovery unfolds during execution, alongside the monitor's recorded egocentric inputs. During approach, a fall triggers get-up before the humanoid resumes approaching and lifts the object. After a tabletop drop, the monitor identifies the object as unheld, prompts re-localization, and repeated grasp attempts eventually restore a stable lift and transport. Following a fall at the destination, the humanoid stands up, locates the object on the floor, and re-acquires it

with the floor-grasp expert. These examples show the sequence of observation, recovery action, and physical outcome; aggregate correction results are reported separately in Table 4.



(a) 4 s: approach (b) 9 s: fall (root-height drop) (c) 13 s: get-up skill (d) 17 s: upright, approach resumes (e) 19 s, monitor: continue (f) 35 s: stable lift

(g) 12 s: object slips onto the table (h) 16 s, monitor: re-localize (i) 22 s: regrasp after first retry (j) 30 s: regrasp after second retry (k) 39 s: third grasp, stable lift (l) 43 s: transport resumes

(m) 28 s, monitor: held, continue (n) 31 s: fall at the destination (o) 34 s: get-up skill (p) 40 s, monitor: on floor, localize (q) 49 s: floor regrasp (r) 52 s: object re-lifted

Figure S12: Recovery during execution. Orange-bordered images are the monitor's actual egocentric inputs, with its decision in the subcaption; grey-bordered third-person frames are for presentation only. Times are measured from the start of each recording. (a–f) A fall while approaching triggers the get-up skill; the monitor then sees no failure and the approach resumes to a stable lift. (g–l) The object slips back onto the table; the monitor sees it unheld, which triggers re-localization; after two confirmed failures and retries, the third grasp succeeds. (m–r) After a fall at the destination and a get-up, the monitor finds the object on the floor and a floor grasp re-acquires it.

B.5 Validation of the Non-Privileged Grasp Policy

The system is intended to run without privileged simulator information, but the grasp controller inherited from prior work observes the simulator object state throughout grasping and transport. We therefore fine-tune a non-privileged grasp policy that replaces this state with a hand-carried estimate after contact (Section B.5.2). To test whether it performs close enough to be used in the current system, we compare it with two privileged conditions that give the same policy the simulator object state, either after lift or throughout.

B.5.1 Object-Dependent Observation

Following OmniGrasp, the grasp policy $\pi_{\text{grasp}}(\boldsymbol{z}_t \mid \boldsymbol{s}_t, \boldsymbol{g}_t)$ of Equation S7 receives the proprioceptive state $\boldsymbol{s}_t$, including fingertip contact forces, and the goal state

$$\boldsymbol{g}_t = \big(\hat{\boldsymbol{p}}_{t+1:t+\varphi} - \tilde{\boldsymbol{p}}_t,\ \hat{\boldsymbol{R}}_{t+1:t+\varphi} \ominus \tilde{\boldsymbol{R}}_t,\ \hat{\boldsymbol{v}}_{t+1:t+\varphi} - \tilde{\boldsymbol{v}}_t,\ \hat{\boldsymbol{\omega}}_{t+1:t+\varphi} - \tilde{\boldsymbol{\omega}}_t,\ \tilde{\boldsymbol{p}}_t,\ \tilde{\boldsymbol{R}}_t,\ \{\boldsymbol{p}_t^{f_j} - \tilde{\boldsymbol{p}}_t\}_{j=1}^{10},\ \boldsymbol{\sigma}\big), \tag{S16}$$

expressed in the humanoid's root heading frame. Hatted quantities are the reference object trajectory over $\varphi = 20$ future samples, $\boldsymbol{p}_t^{f_j}$ is the position of fingertip j, $\boldsymbol{\sigma}$ is the object shape code, and $\ominus$ denotes the rotation difference. The tilde marks the object state $\tilde{\boldsymbol{x}}_t = (\tilde{\boldsymbol{p}}_t, \tilde{\boldsymbol{R}}_t, \tilde{\boldsymbol{v}}_t, \tilde{\boldsymbol{\omega}}_t)$ supplied to the controller. The three conditions differ only in how it is formed:

$$\tilde{\boldsymbol{x}}_t = \begin{cases} \boldsymbol{x}_t, & \text{fully privileged; post-lift privileged after the hand-over,} \\ \bar{\boldsymbol{x}}_t, & \text{before contact (non-privileged) or before the hand-over (post-lift privileged),} \\ \boldsymbol{T}_t^w \boldsymbol{x}^{\text{rel}}, & \text{non-privileged after contact,} \end{cases} \tag{S17}$$

where $\boldsymbol{x}_t$ is the simulator object state, $\bar{\boldsymbol{x}}_t$ is the object estimate, and $\boldsymbol{T}_t^w \boldsymbol{x}^{\text{rel}}$ carries the object pose $\boldsymbol{x}^{\text{rel}}$, frozen relative to the contacting wrist w at first contact, with the wrist pose $\boldsymbol{T}_t^w$ (Section B.5.2). The hand-over is the tactile transition from grasping to transport (Section A.3.5). All three conditions run the fine-tuned checkpoint; the privileged conditions only replace its object input with the simulator state.

B.5.2 NON-PRIVILEGED POLICY

After contact, the head camera can no longer observe the object reliably, so the non-privileged controller replaces the object state with a proxy carried by the hand. Because the pretrained policy never observed such a proxy, we fine-tune the complete policy with the training algorithm of OmniGrasp (Equation S9).

Training proxy. Let $\boldsymbol{p}_t^c$ be the fingertip centre of hand h, n_t^h its number of fingertips with contact force above 1 N, t_0^h its first contact step, and $\boldsymbol{p}_0$ the initial object position. The grasp is latched at

$$\begin{aligned} t_\ell &= \min\{t:\ \ell_{t-2} = \ell_{t-1} = \ell_t = 1\}, \\ \ell_t &= \mathbf{1}\Big[n_t^h \geq 2 \ \wedge\ \|\boldsymbol{p}_t^c - \boldsymbol{p}_0\| \leq 0.25\,\mathrm{m} \ \wedge\ z_t^c - z_{t_0^h}^c \geq 0.03\,\mathrm{m}\Big], \end{aligned} \tag{S18}$$

and the object state supplied to the policy is

$$\tilde{\boldsymbol{x}}_t = \begin{cases} \boldsymbol{x}_t, & t < t_\ell, \\ \boldsymbol{T}_t^c \big(\boldsymbol{T}_{t_\ell}^c\big)^{-1} \boldsymbol{x}_{t_\ell}, & t \geq t_\ell, \end{cases} \tag{S19}$$

where $\boldsymbol{T}_t^c$ is the frame at $\boldsymbol{p}_t^c$ with the wrist orientation of hand h. After the latch, the reference object trajectory lifts the object from its proxy pose at t_ℓ:

$$\hat{\boldsymbol{p}}_t = \tilde{\boldsymbol{p}}_{t_\ell} + 0.2\,\mathrm{m} \cdot S\left(\frac{t - t_\ell}{1\,\mathrm{s}}\right) \boldsymbol{e}_z, \qquad \hat{\boldsymbol{R}}_t = \tilde{\boldsymbol{R}}_{t_\ell}, \tag{S20}$$

with the smoothstep $S(u) = 3u^2 - 2u^3$ for $u \in [0, 1]$ and $S(u) = 1$ for $u > 1$.

Reward. The reward is the OmniGrasp-style grasp reward of the pretrained policy, unchanged and computed from the simulator state $\boldsymbol{x}_t$, whereas the policy observes $\boldsymbol{g}_t(\tilde{\boldsymbol{x}}_t)$ of Equation S16:

$$r_t = \begin{cases} r_t^{\mathrm{pre}}, & \text{before contact}, \\ r_t^{\mathrm{man}}(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t), & \text{after contact}, \end{cases} \qquad \max_{\pi_{\mathrm{grasp}}} \mathbb{E}\left[\sum_t \gamma^t r_t\right], \ \ \boldsymbol{z}_t \sim \pi_{\mathrm{grasp}}\big(\cdot \mid \boldsymbol{s}_t, \boldsymbol{g}_t(\tilde{\boldsymbol{x}}_t)\big), \tag{S21}$$

with r_t^{pre} and r_t^{man} from Equations S13 and S12. Terminations and the lift-success criterion also use $\boldsymbol{x}_t$.

Deployment. Before contact, $\tilde{\boldsymbol{x}}_t = \bar{\boldsymbol{x}}_t$. At the first step t_c with a contact force of at least 1 N, the proxy attaches to the wrist w with the larger force, $\boldsymbol{x}^{\mathrm{rel}} = (\boldsymbol{T}_{t_c}^w)^{-1} \bar{\boldsymbol{x}}_{t_c}$, and $\tilde{\boldsymbol{x}}_t = \boldsymbol{T}_t^w \boldsymbol{x}^{\mathrm{rel}}$ thereafter, through transport and placement. If contact is lost before the hand-over to transport, $\tilde{\boldsymbol{x}}_t$ reverts to $\bar{\boldsymbol{x}}_t$. A privacy check overwrites $\boldsymbol{x}_t$ with a constant and requires the transport observation to remain unchanged. Deployment thus attaches the proxy earlier than in training, and to the wrist rather than the fingertip centre.

B.5.3 CONTROLLED SETTING

To isolate the controller from localization error, all three conditions share the same initial estimate: the simulator pose read once the object has settled after scene initialization, rather than the detection, depth, and mesh-registration estimate used by the full system. The re-approach target of every grasp retry is likewise the current simulator pose. All conditions use the development split and the same detection checks, retry budget, and VLM monitor. Episodes rejected by the object-initialization guard are rerun and do not count as scientific outcomes. This controller-input comparison is distinct from the non-privileged recovery evaluation in Table 4; its controlled targets remove localization error, so its success rates are not directly comparable to that table.

B.5.4 METRICS AND RESULTS

The metrics are defined as follows:

- *First-attempt success* counts successes without any recovery or get-up event.

- *Failure detection* counts, among episodes with a failed grasp attempt, those in which the grasp failure was detected.
- *Detection precision* counts, among grasp-retry decisions, those made when the item was in fact not held. Held is scored from the simulator for evaluation only, as a lift of at least 5 cm with a hand–object distance of at most 10 cm.
- *Recovery* counts, among episodes with a detected grasp failure, those that end in success.

Table S12: Grasp-controller privilege comparison on the development split. Success and detection rows are counts over their denominators, as defined above; tasks that failed before reaching the grasp stage are excluded. Failure counts are terminal outcomes and sum to the number of failed episodes.

Metric	Non-privileged	Post-lift privileged	Fully privileged
Overall success	76/99	78/99	79/98
First-attempt success	64/99	64/99	67/98
Failure detection	30/30	30/30	25/25
Detection precision	81/87	78/82	51/66
Recovery	10/30	13/30	8/25
Episodes with a detected fall	20	17	16
Terminal failures			
Retry budget exhausted	11	11	12
Get-up not completed	8	7	6
Indoor safety stop	2	1	1
Stage restart failed	2	2	0

Without privileged object information, the policy succeeds about as often as with either privileged condition (Table S12), which supports using it in the current system. Detection is not the bottleneck: almost every failed grasp is detected, and almost every grasp retry is triggered while the item is indeed not held; the few false alarms are unsecured-grasp timeouts on an already lifted item. Most detected failures instead end by exhausting the retry budget or by an incomplete get-up.

B.6 Supervised Adaptation of the Local Reasoner

B.6.1 Demonstration Data

We fine-tune the publicly released Qwen3.8-27B (Qwen Team, 2026) on the grounding stages of the agent. Privileged localization scores select 83 successful Luna trajectories on development tasks: 81 satisfy Loc@2cm and two have near-threshold errors (Table S13). Every recorded grounding call in these trajectories becomes one example, with its prompt and egocentric RGB observations as input and Luna's recorded reply as the target. Privileged information is used only for this selection; it never enters inputs or targets.

Table S13: Grounding demonstration corpus.

Property	Value
Trajectories	83 (81 with Loc@2cm, 2 near-threshold)
Tasks (train / validation / test)	67 / 8 / 8
Calls (train / validation / test)	2,193 / 265 / 267
Distinct images	5,839

B.6.2 Objective and Configuration

We adapt the model with LoRA (Hu et al., 2022): each adapted linear layer keeps its pretrained weight $\boldsymbol{W}_0 \in \mathbb{R}^{d\times k}$ frozen and learns a low-rank update,

$$\boldsymbol{W} = \boldsymbol{W}_0 + \frac{\alpha}{r}\boldsymbol{B}\boldsymbol{A}, \qquad \boldsymbol{B} \in \mathbb{R}^{d\times r},\ \boldsymbol{A} \in \mathbb{R}^{r\times k},\ r \ll \min(d,k), \tag{S22}$$

with rank r and scaling α. Let θ_0 denote the frozen base parameters and $\Delta\theta = \{\boldsymbol{A}_l, \boldsymbol{B}_l\}_l$ the trainable parameters over the adapted layers l. QLoRA (Dettmers et al., 2023) further stores $\boldsymbol{W}_0$ in 4-bit NormalFloat (NF4) with doubly quantized scaling constants, dequantizes it to BF16 for each forward and backward pass, and backpropagates through it only into $\boldsymbol{A}$ and $\boldsymbol{B}$. This reduces training memory without changing the training objective.

Let x_i contain the recorded images and prompt, y_i the recorded response, and $\mathcal{T}_i$ its assistant-response token positions. We minimize

$$\mathcal{L}_{\mathrm{SFT}}(\Delta\theta) = -\frac{1}{\sum_i |\mathcal{T}_i|} \sum_i \sum_{t \in \mathcal{T}_i} \log p_{\theta_0+\Delta\theta}(y_{i,t} \mid x_i, y_{i,<t}), \tag{S23}$$

where image and user-prompt tokens are masked from the loss and the full assistant response, including its stage-specific decision, coordinates, rationale, and evidence fields, is supervised. Table S14 lists the configuration; the adapter is selected by validation loss.

Table S14: QLoRA configuration of the adapted Qwen3.8-27B.

Configuration	Setting
Base / adaptation	Qwen3.8-27B / QLoRA, NF4 double quantization, BF16 compute
Adapter rank / scaling / dropout	16 / 32 / 0.05; linear layers
Training hardware	Three NVIDIA A100-SXM4 80 GB GPUs
Batching	1 example per device; 2 accumulation steps; effective batch 6
Learning rate / seed	3×10^{-5} / 20260919
Context / loss	32,768 tokens, no truncation; full assistant-response supervision
Inference	BF16 base with LoRA through vLLM; additional thinking disabled

B.6.3 Base-versus-Adapted Evaluation

Token-level losses on held-out calls are computed with reference prefixes and do not reflect free-running decisions, so we compare Qwen3.8-27B with and without its adapter under identical base weights, inputs, decoding settings, and runtime. *Recorded-input replay* issues the same archived calls to both models and scores their replies without executing actions. *Closed-loop grounding* resets each case for each model and lets their observations diverge with their decisions. Table S15 reports the complete 100-task held-out test split. The untuned base localizes 63/100 sources within 2 cm and identifies the correct destination support in 50/100 tasks; the adapted model reaches 91/100 and 88/100, respectively. These are gains of 28 and 38 percentage points. Destination measures support identity, not placement accuracy, and uses all 100 tasks as the denominator.

Table S15: Closed-loop grounding with untuned and adapted Qwen3.8-27B on all 100 held-out test tasks. Entries are successful tasks over the full test-set denominator; destination scores support identity. Adapted results match Table 1.

Measure	Base	Adapted
Loc@2cm ↑	63/100	**91/100**
Destination ↑	50/100	**88/100**

Table S16 reports replay on 100 held-out calls from the demonstration corpus's eight-task test partition, distinct from the 100-task closed-loop test split. Decisions are scored against archived visible-target masks rather than against the demonstration replies. The adapted model almost always returns well-formed structured replies, whereas the base often does not, and it places point proposals on the target far more often. Both bind the correct candidate, while the adapted model occasionally accepts a wrong anchor. Because the runtime extractor recovers a structured object from every reply of both models, the format gain alone does not change runtime parsing. Request latency is similar.

Table S16: Untuned base versus adapted Qwen3.8-27B on 100 held-out recorded grounding calls (eight tasks). Decision measures use independent mask-based scoring on the scorable subset. Bold marks the better model; ties are unmarked.

Measure	Base	Adapted
Raw reply is a JSON object ↑	43/100	**97/100**
Raw reply contains all expected top-level keys ↑	33/100	**93/100**
Runtime extractor recovers an object ↑	100/100	100/100
Point proposal on target pixel ↑	14/27	**25/30**
on matched scorable calls ↑	14/26	**23/26**
Correct candidate binding (matched) ↑	9/9	9/9
Wrong anchor accepted ↓	**0/27**	2/27
Median request latency (s) ↓	6.04	**5.65**