CORRECTING WHERE, PRESERVING HOW: COMPOSITIONAL GENERALIZATION FOR VISION-LANGUAGE-ACTION MODELS VIA REFERENTIAL GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

While Vision-Language-Action (VLA) models enable flexible action generation, their generalization across diverse environmental elements, including manipulated objects, destinations, and backgrounds, is limited by the lack of diversity in robotic training data. Trained end-to-end on such data, VLAs tend to exploit visual shortcuts, associating actions with task-irrelevant visual features rather than the intended task semantics. These shortcuts block recomposition of elements already seen by the policy, that is, *compositional generalization*. Existing approaches mitigate such entanglement through task-relevant perception or targeted data diversification, but offer no explicit mechanism for unseen recomposition and require backbone-specific modifications with retraining. We observe that under such recomposition, VLAs often fail at global grounding while retaining local manipulation skills that recover near the correct target in familiar configurations. Therefore, we propose **Referential Guidance (ReGuide)**, a training-free wrapper that, given object poses from a grounding module, combines semantic and geometric rebinding to guide the end-effector into demonstration-supported configurations of the instructed referent, where the frozen policy can resume execution. Experiments in simulation across multiple VLA backbones as well as on a real robot show that ReGuide improves success rates under compositional shifts by up to 56.8 and 75.0 percentage points, respectively, while preserving standard-task performance.

1 INTRODUCTION

Vision-Language-Action (VLA) models draw on the broad semantic knowledge from pretrained vision-language backbones and map visual observations and language instructions directly to robot actions (Kim et al., 2025b; Zitkovich et al., 2023). However, due to the high cost of robotic data collection, objects, destinations, and backgrounds in robotic datasets tend to appear in fixed combinations (Xing et al., 2025; Saxena et al., 2025; Zhou et al., 2026). Trained end-to-end on such data with direct low-level output, VLAs receive supervision only on the final action, where the contributions of object, skill, and environment are collapsed into raw action values (Hancock et al., 2026; Black et al., 2025). Which part of an action belongs to the interaction with the *target object/destination*, which to the *skill* itself, and which to the constraints of the *environment* is hard to disentangle, since these factors rarely vary independently in training datasets (Gao et al., 2024). As a result, the policy often fails when a test episode breaks a training combination (*e.g.,* manipulating an object in a different scene, or pairing a target object with a new destination), even though it has seen every element before (Xing et al., 2025). It binds to the wrong referent, disregards the instruction, and blindly replays a training trajectory anchored on the dominant visual cues (de Haan et al., 2019; Peng et al., 2026b; Hou & Zhao, 2026). The compositional generalization that VLAs should in principle possess is thus limited by robotic data and end-to-end training (Chen et al., 2025).

Recent work on this limitation falls into three strands, summarized in Fig. 1. The first decorrelates the factors in the data, collecting or synthesizing demonstrations in which objects, goals, and scenes are re-paired (Vo et al., 2025; Gao et al., 2024; Xing et al., 2025; Peng et al., 2026a;b). Such methods buy composition with coverage, so the cost grows with the number of factors, and pairings outside the collected coverage still fail (Xie et al., 2024). The second changes how the policy grounds the referent, by injecting explicit target cues (Li et al., 2025), or by separating declarative from

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.

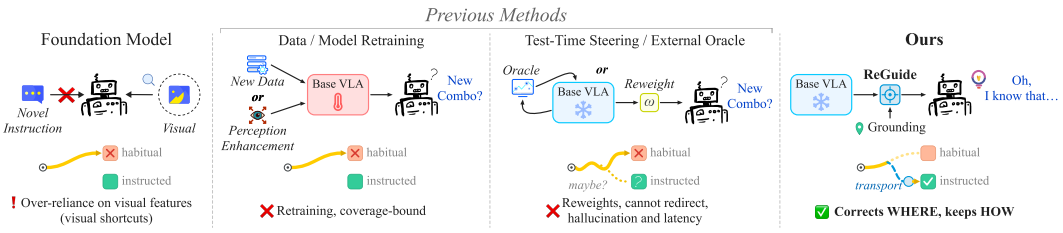

Figure 1: **Approaches to compositional failures.** Foundation VLAs over-rely on visual features. Previous methods require coverage-bound retraining or use test-time steering and external oracles, while ReGuide corrects the referent and preserves the frozen policy's manipulation competence. The second and third panels from the left correspond to strands 1–2 and strand 3 respectively.

procedural knowledge in a new architecture (Tsagkas et al., 2026; Lian et al., 2026; Huang et al., 2025b). These methods retrain the policy to incorporate grounding cues, often with backbone-specific modifications (Vo et al., 2025). Generalization to unseen pairings still depends on what the retrained mapping covers (Huang et al., 2025a). A third line of research keeps the policy frozen and steers it at inference. Some works amplify the language-dependent component of the policy's own action distribution, which only reweights what the policy already proposes and cannot carry the end-effector to a referent it never considered (Zhan et al., 2026; Li, 2026; Fang et al., 2026). Others keep an external oracle in the loop, such as learned action scorers or language-model planners staging the robot (Nakamoto et al., 2025; Zhang et al., 2026b; Du & Song, 2025). Every correction then depends on that stack, which adds latency and, for generative planners, the risk of infeasible subgoals (Ahn et al., 2022; Li et al., 2026). Across all three strands, the failure is treated as missing knowledge to be rebuilt or as a distribution to be reweighted or overridden, without taking a closer examination: *which part of the behavior actually fails, and which part still works? Can its competence and flexibility stay in charge with minimal agentic assistance?*

Our observation is that failures usually stem from incorrect global grounding rather than a loss of local manipulation skills. The policy commits to the wrong referent, yet once its end-effector reaches a trained configuration, it can resume local interaction with the object. The policy often errs in *where* it acts while retaining the skills needed for *how* it interacts. We therefore propose **Referential Guidance (ReGuide)**, a training-free wrapper which corrects a wrong referent commitment by rebinding rather than retraining or reweighting the policy. Specifically, the rebinding consists of two components. The semantic component determines which referent to act on, while the geometric component determines the local pose from which the policy can take over. On the *semantic side*, it resolves the instructed referent via a lightweight grounding module and uses the policy's action predictions as a signal for detecting referent mismatches without an external scorer. On the *geometric side*, it transports the end-effector along a bounded flow into a demonstration-supported pre-contact set of the correct referent. The frozen policy then resumes local interaction from the true observation, with subsequent actions constrained by the demonstrated motion envelope. ReGuide derives its reference configurations and motion bounds from the policy's training demonstrations. Our contributions are as follows:

- We find that VLAs retain local manipulation skills under compositional shifts, which can be reused by guiding the end-effector into demonstration-supported configurations of correct referents.
- We propose **Referential Guidance (ReGuide)**, a training-free, backbone-agnostic wrapper that reuses local interaction skills through *semantic* and *geometric rebinding*.
- ReGuide raises compositional success by up to 56.8 percentage points in simulation and 75.0 points on a real robot, while preserving standard-task performance.

2 Related Work

2.1 Vision-Language-Action Models

VLA models fine-tune a pretrained VLM to map images, proprioception, and an instruction directly to robot actions (O'Neill et al., 2024). Early designs emit actions as discrete tokens through autoregressive decoding (Kim et al., 2025b; Zitkovich et al., 2023; Pertsch et al., 2025), while recent generalist policies attach a diffusion or flow-matching action expert that outputs continuous action

chunks (Black et al., 2026; Zheng et al., 2026; NVIDIA et al., 2025; Shukor et al., 2025), and fine-tuning methods further improve inference speed and task success (Kim et al., 2025a). Their generalization is inherited from the VLM backbone (Zitkovich et al., 2023), yet fine-tuning on robot data that is far less diverse than the backbone's pretraining corpus weakens the grounding it started with (Huang et al., 2025b). This has motivated frozen encoders and knowledge insulation during training (Driess et al., 2025), and test-time sampling with verifiers that improve robustness without touching the weights (Kwok et al., 2025; Nakamoto et al., 2025). ReGuide leaves the model, its training, and its decoding untouched and wraps any of these policies as a frozen executor.

2.2 Compositional Generalization in Robot Manipulation

Compositional generalization in manipulation can mean composing skills into sequences (Chen et al., 2025; Yang et al., 2026) or, as studied here, recombining task elements (Gao et al., 2024; Li, 2026). We organize prior works by whether they act during training or at test time.

Training-time methods. Data-side work attributes shortcuts to limited factor combinations rather than insufficient data (Xing et al., 2025), and answers with protocols covering factor pairs (Gao et al., 2024) or synthesis pipelines that assemble recombined trajectories by LLM decomposition with wrist masking (Peng et al., 2026b) or by steered rollouts filtered in simulation (Peng et al., 2026a). Model-side work identifies the referent through zero-initialized cross-attention on object masks (Li et al., 2025), a lifted 3D point modulating the action head (Tsai et al., 2026), or object-centric tokens (Bendikas et al., 2025), or reshapes attention through relevance supervision (Jia et al., 2026; Sun et al., 2026), separate where and what streams (Tsagkas et al., 2026), and insulated or mutual-information objectives (Driess et al., 2025; Lian et al., 2026).

Inference-time steering. Inference-time methods differ in the signal they steer with. Language-side steering interpolates text latents between tasks (Li, 2026), contrasts the policy against a language-dropped branch (Fang et al., 2026), or redistributes attention to instruction tokens (Zhang et al., 2026a), each always on with hand-set scales. Oracle-side steering re-ranks samples with a value or verifier (Nakamoto et al., 2025; Kwok et al., 2025; Wu et al., 2026b), guides denoising with a dynamics or reward model (Du & Song, 2025; Liu et al., 2026), or stages the robot before contact, after a stall through an affordance field combining a planner model, a segmenter, depth, and a motion planner (Xu et al., 2026b), or through a memory-guided agent (Zhang et al., 2026b).

ReGuide instead needs no retraining, no external planner or scorer, and no guidance-weight tuning, directly reusing local manipulation skills across multiple VLA backbones.

3 Methodology

3.1 Problem Formulation

Setting. A frozen VLA policy π_θ maps an observation o_t and an instruction ℓ to an action chunk A_t, executing a prefix before replanning. Each action $a = (u, w, g)$ contains translation, rotation-vector and gripper commands for the end-effector pose $x_t = (p_t, R_t)$. The instruction specifies an object and a destination, inducing stages such as grasping and placing. At stage k, the instructed referent is $e_k^\star$, and the moving point q_t is the end-effector position before the object is held and the object position afterwards. A grounding module identifies the instructed referent and supplies entity poses and axis-aligned bounding boxes.

Restoring the conditions for local interaction. A successful stage requires reaching the instructed referent and executing a compatible interaction. For stage k, let B_k denote entry into the correct referent's approach region (Sec. 3.2) and S_k denote stage success. Since success requires entry, with $\Pr$ denoting probability,

$$\Pr(S_k) = \underbrace{\Pr(B_k)}_{\beta_k \text{ (binding)}} \underbrace{\Pr(S_k \mid B_k)}_{\gamma_k \text{ (competence)}}, \tag{1}$$

where binding β_k captures reaching the correct interaction and competence γ_k captures completing it. The design objective is to restore access to the interaction configurations in which the policy's local skills remain usable. This requires both identifying the correct referent and approaching it in a suitable position and posture. Our framework addresses these requirements through two components

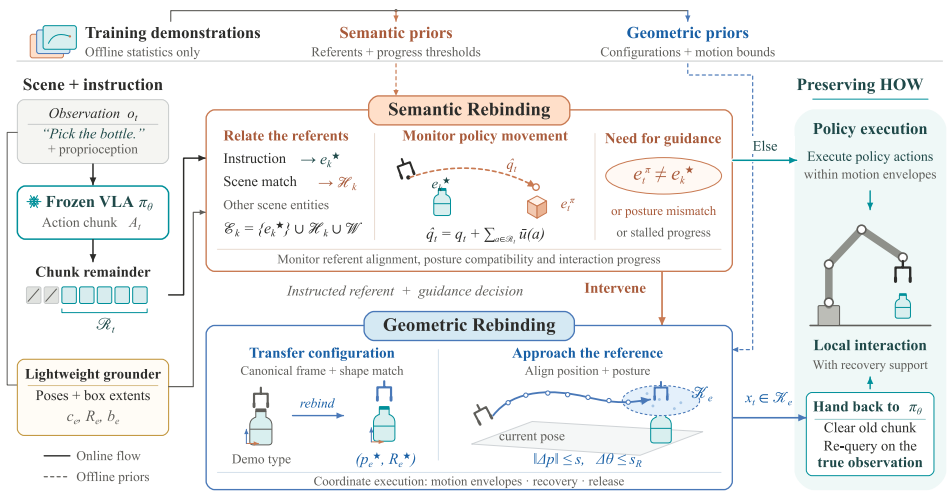

Figure 2: **Overview of ReGuide.** *Semantic rebinding* monitors referent alignment and interaction progress. *Geometric rebinding* transfers demonstration-derived configurations and coordinates guidance, frozen-policy execution and recovery.

(Fig. 2). *Semantic rebinding* relates the policy's movement to the instructed referent and determines when guidance is needed. *Geometric rebinding* transfers a local interaction configuration to that referent and coordinates transport, hand-back and subsequent execution. For example, destination reassignment changes where the held object should go, while the demonstrated release configuration specifies how it should arrive. Appendix A.3 gives the corresponding intervention model.

Demonstrations as references. The demonstrations $\mathcal{D}$ that trained π_θ supply target configurations, motion bounds and progress thresholds. For a scalar quantity ξ, $Q_\alpha[\xi]$ denotes its empirical quantile within a referent class. Targets use $Q_{.5}$, upper and lower bounds use $Q_{.95}$ and $Q_{.05}$, and commitment thresholds use $Q_{.25}$. These levels and the execution constants are shared across tasks (Appendix A.7).

3.2 Semantic Rebinding

Relating intended and competing referents. The instructed referent identifies the desired interaction. Alternatives reveal whether the policy follows a familiar but incorrect association. We retrieve the training scene nearest to the first frame under a frozen image encoder and collect the objects and destinations moved by its tasks into a habitual set $\mathcal{H}_k$. The candidates are $\mathcal{E}_k = \{e_k^\star\} \cup \mathcal{H}_k \cup \mathcal{W}$, where $\mathcal{W}$ contains the remaining movable entities. Each candidate inherits demonstration statistics from the training type matched by geometry rather than name.

Monitoring the policy's referent. The unexecuted chunk remainder $\mathcal{R}_t$ predicts where the policy is moving. Translation and rotation commands are converted to expected increments by $\bar{u} = \kappa_p \sigma_p u$ and $\bar{w} = \kappa_R \sigma_R w$, using controller scales σ and tracking gains κ fitted on demonstrations. The predicted endpoint and posture are

$$\hat{q}_t = q_t + \sum_{a \in \mathcal{R}_t} \bar{u}(a), \qquad \hat{R}_t = \exp\Big(\Big[\sum_{a \in \mathcal{R}_t} \bar{w}(a)\Big]_\times\Big) R_t. \tag{2}$$

Commitment is assessed relative to each referent's approach geometry. For referent position c_e and box half-extents b_e, the componentwise offset $\delta_e(q) = \max(|q - c_e| - b_e, 0)$ is zero along axes where q lies inside the box. The entry region has at most one nonzero component. Demonstration entry is the first frame before engagement in which the end-effector lies in this region. The commitment radius $\rho_e = Q_{.25}[\|\delta_e(p_t)\|]$ is measured at this event. A commitment requires the predicted endpoint to enter along two axes and approach along the remaining axis within this radius,

$$\chi_t(e) = \mathbb{1}\big[\|\delta_e(\hat{q}_t)\|_0 \le 1, \quad \delta_e^i(\hat{q}_t) < \rho_e, \quad \delta_e^i(\hat{q}_t) < \delta_e^i(q_t)\big], \tag{3}$$

where i indexes the largest endpoint offset and components below the position resolution ε count as zero. Among qualifying candidates, the box nearest to the predicted endpoint defines the policy's

referent e_t^π. A mismatch triggers guidance when it differs from $e_k^\star$, the moving point remains at least $\rho_{e_k^\star}$ from the instructed referent's box, and the gripper has not contacted it. This provides an anticipatory mismatch signal without an external scorer. During placement, the comparison uses horizontal object–destination distance and the demonstrated carry radius, requiring consecutive verdicts unless the object already dwells within a habitual radius. Grasp-stability and height-envelope gates coordinate this decision with the carrying state (Appendix A.5).

Monitoring progress toward interaction. A policy can select no competing referent yet fail to approach a usable configuration. ReGuide monitors posture compatibility and progress toward the demonstrated grasp configuration. Before descent beside the object, guidance is triggered if the predicted posture error exceeds its demonstrated tolerance and improves over the current posture by no more than one demonstrated rotation step. Failure to bind is declared after N steps without one demonstrated step of progress, with N exceeding the longest such demonstration interval by one. An object with no matching training type receives guidance at the first decision using geometric transfer.

3.3 Geometric Rebinding

Transferring an interaction configuration. Given the selected referent, geometric rebinding supplies the position and posture from which to interact with it. Demonstrations are grouped by object type and destination class. For an object of orientation R_e, poses at engagement are expressed in its gravity-aligned canonical frame R_eC_e, whose first axis follows the wider horizontal extent and whose third is vertical. Let t_e locate the top-face centre in this frame, $\bar{o}$ be the per-axis median grasp offset from it, and $\bar{R}$ the rotation medoid. Combining these with the median entry altitude $\bar{z}$ above the object top z_e^{top} gives

$$\begin{aligned} p^\star_{e,xy} &= c_{e,xy} + \big[R_eC_e(\bar{o} + t_e)\big]_{xy}, \qquad p^\star_{e,z} = z_e^{\mathrm{top}} + \bar{z}, \\ R_e^\star &= R_eC_e\bar{R}. \end{aligned} \tag{4}$$

An unseen object inherits statistics from the nearest training type under Euclidean distance between log descriptors of height, two widths and top fill. The canonical frame transfers the reference to its geometry (Appendix A.6).

Hand-back requires proximity in both position and posture. The pre-contact set is

$$\mathcal{K}_e = \Big\{(p, R) \;\Big|\; \|p - p_e^\star\| \le s,\;\; \|\Pi_\perp(p - p_e^\star)\| \le r_e,\;\; \vartheta(R_e^\star, R) \le \tau_R\Big\}. \tag{5}$$

Here s is the $Q_{.95}$ translation step. The lateral tolerance r_e is the smaller of the $Q_{.95}$ grasp scatter along the closing axis and the object half-extent along it. The posture tolerance τ_R is the $Q_{.95}$ rotation residual at entry. The projection $\Pi_\perp$ removes the approach axis, and ϑ measures geodesic error modulo the half-turn symmetry of a parallel gripper. These conditions specify how closely the transported configuration must match the reference before policy execution resumes.

For placement, we convert the median release offset in the destination frame to an end-effector target using the current grasp offset. The acceptance region Ω_d intersects the physical destination range with the $Q_{.95}$ release scatter. For destination class f, the median profile m_f and its $Q_{.05}$ lower envelope $\underline{m}_f$ describe the held object's bottom height against horizontal distance to the destination.

Approaching the reference. ReGuide transports the end-effector with translation and rotation bounded by demonstrated per-step motion. For position error $\Delta_t = p_e^\star - p_t$ and rotation error of unit axis ω_t and angle ϑ_t, it commands

$$u_t = \frac{\Delta_t}{\|\Delta_t\|} \cdot \frac{\min\{\|\Delta_t\|,\, s\}}{\kappa_p \sigma_p}, \qquad w_t = \omega_t \cdot \frac{\min\{\vartheta_t,\, s_R\}}{\kappa_R \sigma_R}, \tag{6}$$

where s_R is the $Q_{.95}$ rotation step and the denominators account for command scale and tracking gain. While carrying, the height target follows m_f and is floored by the intervening crest of $\underline{m}_f$, following the demonstrated order of lifting before translating. On entering $\mathcal{K}_e$, ReGuide discards the pending chunk and queries the frozen policy on the true observation.

Coordinating execution and recovery. The same references govern execution after hand-back. Policy actions are constrained by the lateral tolerance r_e during descent and by the height floor during carrying. Contact transitions also determine whether a transport remains applicable. It dissolves

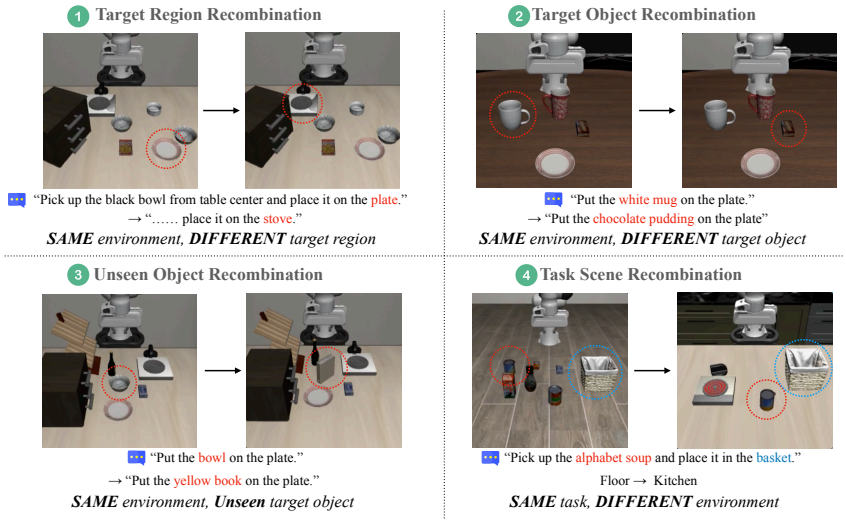

Figure 3: **Compositional Tasks on LIBERO**. Each setting changes exactly one factor in a training task while keeping the others fixed. Together, they isolate where compositional generalization fails and test whether the policy follows instructions beyond memorized task configurations.

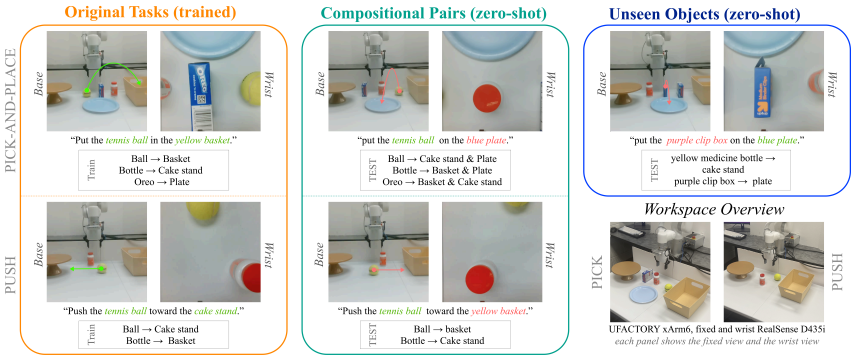

Figure 4: **Real-robot evaluation on the xArm6**. The setup includes five trained pairings, eight recombined pairings, and two unseen object cases across *pick and place* and *pushing*. Bare $\pi_{0.5}$ and ReGuide are evaluated on the same paired layouts per cell.

on a changed premise, such as a grasp or drop, or when its error stops decreasing over n replans. Each stage permits at most two transports. Completion certificates monitor loss of the destination, release outside Ω_d and insufficient progress during carrying or placement. They re-arm transport, or complete descent and gripper release at the demonstrated height when the held object remains inside Ω_d without descending for longer than the $Q_{.95}$ demonstrated dwell. Transport onto a supporting surface uses the same release procedure. Algorithm 1 and Appendix A specify the execution protocol.

4 Experiments

4.1 Experimental Setup

Simulation Setup. We build a benchmark on LIBERO that isolates compositional shifts by varying one factor at a time (Liu et al., 2023) (Fig. 3). We evaluate 4 settings of 4 tasks, yielding 16 composition cells. Each cell modifies a training task along 1 axis: the destination (*Target Region*), the target object (*Target Object*), an object never manipulated in the 4 training suites (*Unseen Object*), or the host scene (*Task Scene*). Another 10 unmodified training tasks form *Orig.* to assess whether the policy retains its original capabilities. All methods replay the same 50 initial states per cell, giving 1300 paired trials each. We evaluate 5 VLA backbones (Kim et al., 2025a; Black et al., 2026; 2025;

Table 1: **Success rate (%) on the single-factor composition evaluation of LIBERO.** Each axis recombines one factor of a training task and pools four cells × 50 paired trials. *Orig.* reports the ten unmodified anchor tasks under the same protocol; *All* pools composition and *Orig.* trials, while *Comp. Avg.* pools the 16 composition cells only. Subscripts are 95% Wilson confidence half-widths.

	Method	Year	Orig.	Recombination Axes				Average	
				Target Region	Target Object	Unseen Object	Task Scene	All	Comp. Avg.
Found.	OpenVLA-OFT	2025	**97.6**$_{\pm 1.4}$	13.5$_{\pm 4.7}$	1.5$_{\pm 1.9}$	34.5$_{\pm 6.5}$	6.5$_{\pm 3.5}$	46.2$_{\pm 2.7}$	14.0$_{\pm 2.4}$
	π_0	2025	86.0$_{\pm 3.0}$	5.0$_{\pm 3.1}$	1.5$_{\pm 1.9}$	27.0$_{\pm 6.1}$	0.5$_{\pm 1.3}$	38.3$_{\pm 2.6}$	8.5$_{\pm 1.9}$
	$\pi_{0.5}$	2025	94.0$_{\pm 2.1}$	40.5$_{\pm 6.7}$	25.0$_{\pm 6.0}$	49.5$_{\pm 6.9}$	41.0$_{\pm 6.8}$	60.2$_{\pm 2.7}$	39.0$_{\pm 3.4}$
	X-VLA	2026	**97.6**$_{\pm 1.4}$	39.0$_{\pm 6.7}$	9.5$_{\pm 4.1}$	34.0$_{\pm 6.5}$	0.0$_{\pm 0.9}$	50.2$_{\pm 2.7}$	20.6$_{\pm 2.8}$
	GR00T N1.7	2026	94.6$_{\pm 2.0}$	14.5$_{\pm 4.9}$	9.5$_{\pm 4.1}$	28.0$_{\pm 6.2}$	0.0$_{\pm 0.9}$	44.4$_{\pm 2.7}$	13.0$_{\pm 2.3}$
Comp.	OTTER	2025	82.2$_{\pm 3.3}$	0.0$_{\pm 0.9}$	0.0$_{\pm 0.9}$	13.0$_{\pm 4.7}$	0.5$_{\pm 1.3}$	33.7$_{\pm 2.6}$	3.4$_{\pm 1.3}$
	GuidedVLA	2026	94.8$_{\pm 2.0}$	2.0$_{\pm 2.1}$	6.5$_{\pm 3.5}$	26.5$_{\pm 6.1}$	0.5$_{\pm 1.3}$	41.9$_{\pm 2.7}$	8.9$_{\pm 2.0}$
	APT	2026	91.4$_{\pm 2.5}$	37.0$_{\pm 6.6}$	33.5$_{\pm 6.5}$	25.5$_{\pm 6.0}$	1.5$_{\pm 1.9}$	50.2$_{\pm 2.7}$	24.4$_{\pm 3.0}$
	CAG ($\pi_{0.5}$)	2026	81.4$_{\pm 3.4}$	53.0$_{\pm 6.9}$	35.0$_{\pm 6.6}$	44.0$_{\pm 6.8}$	33.5$_{\pm 6.5}$	56.8$_{\pm 2.7}$	41.4$_{\pm 3.4}$
Ours	ReGuide (OpenVLA-OFT)	–	96.8$_{\pm 1.6}$	87.0$_{\pm 4.7}$	49.0$_{\pm 6.9}$	72.5$_{\pm 6.1}$	74.5$_{\pm 6.0}$	80.8$_{\pm 2.1}$	70.8$_{\pm 3.1}$
	ReGuide ($\pi_{0.5}$)	–	94.2$_{\pm 2.1}$	**96.0**$_{\pm 2.8}$	**93.0**$_{\pm 3.6}$	**94.5**$_{\pm 3.2}$	94.5$_{\pm 3.2}$	**94.4**$_{\pm 1.3}$	**94.5**$_{\pm 1.6}$
	ReGuide (GR00T N1.7)	–	94.2$_{\pm 2.1}$	69.0$_{\pm 6.4}$	52.5$_{\pm 6.9}$	59.0$_{\pm 6.8}$	**96.0**$_{\pm 2.8}$	78.8$_{\pm 2.2}$	69.1$_{\pm 3.2}$

Table 2: **Real-robot evaluation on the xArm6.** Entries are successes/trials for trained (*Orig.*), recombined (*Comp.*), and unseen-object (*Unseen*) tasks. Average success rates (%) pool 50 original, 100 recombined and unseen-object, or all 150 trials per method. Subscripts give 95% Wilson confidence half-widths. Both methods use the same paired layouts.

Method	Pick-and-Place			Push		Average (%)		
	Orig.	Comp.	Unseen	Orig.	Comp.	Orig.	Comp.	All
Bare $\pi_{0.5}$	26/30	7/60	8/20	17/20	0/20	86.0$_{\pm 9.6}$	15.0$_{\pm 7.0}$	38.7$_{\pm 7.7}$
+ ReGuide	28/30	53/60	17/20	18/20	20/20	92.0$_{\pm 7.8}$	90.0$_{\pm 6.0}$	90.7$_{\pm 4.7}$

Zheng et al., 2026; NVIDIA et al., 2025; NVIDIA, 2026) and 4 methods that strengthen grounding or instruction following (Huang et al., 2025b; Jia et al., 2026; Xu et al., 2026a; Fang et al., 2026). We wrap 3 backbones with ReGuide using the same demonstration statistics and oracle simulator state to assess the performance ceiling (Appendix B). We also evaluate perception-based grounding.

Real-World Setup. We fine-tune $\pi_{0.5}$ on 250 scripted demonstrations of 3 pick-and-place and 2 push tasks on a UFACTORY xArm6 with base and wrist RGB-D cameras. The target objects and destinations vary in shape and size. The 15 evaluation cells cover the 5 training pairings, 8 recombined pairings and two with an unseen object, each under 10 layouts shared by the bare policy and ReGuide, that is 150 paired trials per method. Grounding uses an open-vocabulary detector with depth and measured object dimensions to assess real-world performance (Appendix D).

4.2 Experimental Results and In-Depth Analysis

ReGuide recovers most of the loss caused by recomposition. The comparison methods in Table 1 strengthen instruction conditioning but do not explicitly rebind execution to demonstration-derived interaction configurations. Their successes concentrate in the LIBERO-Goal scene, trained under 10 instructions (187/195 for APT), or in cells where training behavior already satisfies the new instruction (50/71 for GuidedVLA). CAG improves *Target Region* and *Target Object* by 12.5 and 10.0 points, but raises *Comp. Avg.* only from 39.0 to 41.4 while reducing *Orig.* from 94.0 to 81.4. Without ReGuide, *Task Scene* remains at or below 6.5% for all non-$\pi_{0.5}$ policies despite unchanged training instructions. ReGuide improves *Comp. Avg.* by 55.5–56.8 points across backbones, with *Orig.* changing by at most 0.8. Remaining failures on OpenVLA-OFT and GR00T N1.7 concentrate after hand-back, notably on the moka pot (2/50 each versus 49/50 for $\pi_{0.5}$), consistent with differences in local competence after reaching the interaction configuration (equation 1). With perception-based grounding, our strongest backbone, $\pi_{0.5}$, achieves 87.5% compositional success, 7.0 points below oracle grounding while retaining a 48.5-point gain over the bare policy (Appendix C.6). On the xArm6, ReGuide raises *Comp.* success from 15.0% to 90.0%, with 75 paired wins and no losses, while *Orig.* improves from 86.0% to 92.0% (Table 2). Gains span recombined pairings and unseen objects (Figs. 5 and 6).

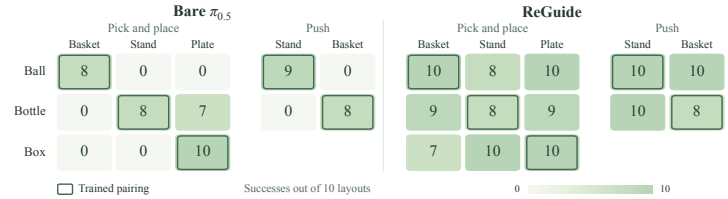

Figure 5: **Real-robot success per object and destination pairing.** Off the trained diagonal the bare policy succeeds on one pairing (7/10); ReGuide reaches at least 7/10 on every pairing.

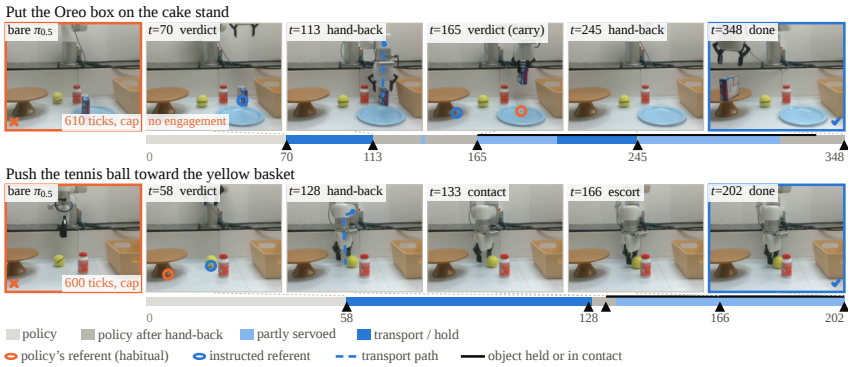

Figure 6: **Paired rollouts under recomposition.** The bare policy acts on the habitual referent, while ReGuide transfers the interaction configuration to the instructed referent and guides execution there.

Table 3: **Component-group ablations and hand-back assistance (success, %).** The first 4 removals test semantic and geometric rebinding; the final removal isolates post-hand-back assistance.

Variant	Recombination Axes				Comp. Avg.	Δ vs Full
	Target Region	Target Object	Unseen Object	Task Scene		
Bare $\pi_{0.5}$	$40.5_{\pm 6.7}$	$25.0_{\pm 6.0}$	$49.5_{\pm 6.9}$	$41.0_{\pm 6.8}$	$39.0_{\pm 3.4}$	-55.5^{***}
Naive transport	$43.5_{\pm 6.8}$	$70.5_{\pm 6.3}$	$75.5_{\pm 5.9}$	$84.5_{\pm 5.0}$	$68.5_{\pm 3.2}$	-26.0^{***}
w/o competing referents	$86.5_{\pm 4.7}$	$77.0_{\pm 5.8}$	$\underline{91.5}_{\pm 3.9}$	$88.5_{\pm 4.4}$	$85.9_{\pm 2.4}$	-8.6^{***}
w/o chunk commitment	$89.0_{\pm 4.4}$	$83.5_{\pm 5.1}$	$81.5_{\pm 5.4}$	$\underline{91.5}_{\pm 3.9}$	$86.4_{\pm 2.4}$	-8.1^{***}
w/o pre-contact sets	$87.0_{\pm 4.7}$	$70.5_{\pm 6.3}$	$75.5_{\pm 5.9}$	$70.5_{\pm 6.3}$	$75.9_{\pm 3.0}$	-18.6^{***}
w/o bounded transport	$84.0_{\pm 5.1}$	$\underline{87.5}_{\pm 4.6}$	$90.0_{\pm 4.2}$	$89.0_{\pm 4.4}$	$\underline{87.6}_{\pm 2.3}$	-6.9^{***}
w/o hand-back assistance	$\underline{94.0}_{\pm 3.4}$	$76.5_{\pm 5.8}$	$86.5_{\pm 4.7}$	$81.5_{\pm 5.4}$	$84.6_{\pm 2.5}$	-9.9^{***}
ReGuide (full)	$\mathbf{96.0}_{\pm 2.8}$	$\mathbf{93.0}_{\pm 3.6}$	$\mathbf{94.5}_{\pm 3.2}$	$\mathbf{94.5}_{\pm 3.2}$	$\mathbf{94.5}_{\pm 1.6}$	–

Δ: paired difference to Full in pp over the same 800 trials; McNemar exact test, $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Local interaction skills remain usable under recomposition. We transport the end-effector to the instructed object's demonstration-derived pre-contact target, then run frozen $\pi_{0.5}$ without further guidance. Over 50 paired trials per cell, grasp success rises from 64.5% to 95.8% on 8 recomposed cells, supporting access to local grasping through configuration transfer (Appendix C.1).

The policy's own actions matter after hand-back. To test execution from these configurations, we replace VLA actions after the first hand-back while retaining the wrapper. Over 800 paired composition trials, success falls from 94.5% to 38.1% with a default descent-and-close action and 28.0% with a hold action (exact McNemar, both $p < 10^{-100}$). Grasp success falls from 98.6% to 44.0% and 33.5%, respectively (Appendix C.7).

Guidance tracks referents and interaction progress. In simulation, transport occurs in 83.9–96.5% of composition versus 5.0–16.2% of original episodes (Fig. 7). Only 3/1500 original episodes trigger wrong-binding corrections. Other interventions establish compatible posture, recover stalled approaches, or support completion. Among jointly successful trials, median length changes are -5

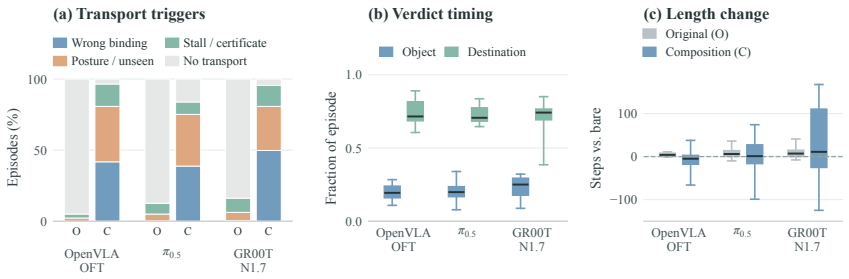

Figure 7: **When, how often and at what cost ReGuide intervenes.** Classes are the verdicts of ReGuide, excluding one-step transports cancelled at a clause switch.

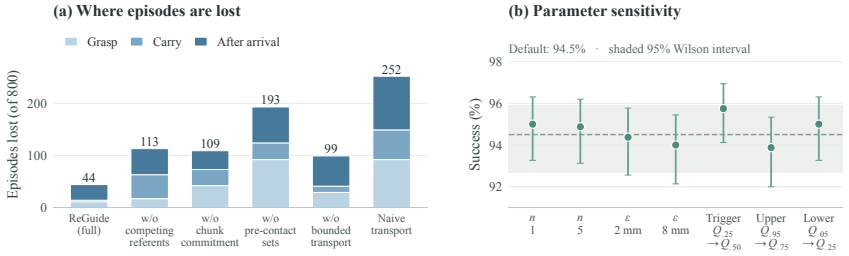

Figure 8: **Failure stages and parameter sensitivity.** (a) Failure stage by removed component. (b) No tested parameter variant differs significantly from the default under paired McNemar tests.

to $+11$ steps on composition tasks and $+4$ to $+7$ on original tasks. Transport occupies a median 14–15% of control steps in successful composition episodes with interventions (Appendix C.3).

Comparison with agent-based guidance and runtime. On one paired initial state per cell, ReGuide solves 23/26 tasks and Harness VLA (Zhang et al., 2026b) solves 18/26, an exploratory comparison reported separately from Table 1. ReGuide's median replanning latency is 129.2 ms versus 129.0 ms for the bare policy, while Harness incurs a median 6.1 s wait per planner response. Beyond latency, Harness incurs an additional GPT-5.5 API cost averaging $1.84 per episode (Appendix C.8).

4.3 Ablation Studies

Component groups and hand-back assistance. Table 3 tests referent monitoring, interaction configurations, and execution coordination. Naive transport recovers 29.5 of the 55.5-point gain by servoing to a fixed height above the grounded object and destination. Removing pre-contact sets or the chunk commitment group mainly impairs grasping, removing competing referents impairs destination selection, and removing the bounded transport group increases post-arrival failures (Fig. 8a). Without pre-contact sets, objects held fall from 789/800 to 708/800. Without competing referents, destination arrivals fall from 785/800 to 737/800, as habitual destinations are excluded. Disabling post-hand-back constraints and completion assistance while retaining subsequent transports yields 84.6% success and 97.1% grasp success, showing the contribution of execution support after configuration transfer (Appendix C.4). Stability across tested constants and quantile levels supports the framework's shared settings without precise tuning (Fig. 8b).

5 Conclusion

Under recomposition, a substantial fraction of VLA failures arise from incorrect referent binding, while local manipulation competence often remains recoverable from demonstration-derived configurations. ReGuide combines semantic and geometric rebinding to restore referent alignment and coordinate guidance, local policy execution, and recovery using demonstration-derived configurations and motion bounds. Without retraining the policy, ReGuide raises compositional success by 55.5 to 56.8 percentage points across three backbones in simulation and from 15.0% to 90.0% on a real robot, while maintaining comparable performance on original tasks.

REPRODUCIBILITY STATEMENT

ReGuide is training-free, so reproducing it requires the policy checkpoints, the demonstrations that trained them, and the procedure of Sec. 3. Algorithm 1, the reference construction, execution protocol and demonstration statistics with their quantile levels are given in Appendix A. Appendix B defines the 16 composition cells and the ten original tasks, the paired initial states, the step budget, the success criterion and the statistical tests, and lists the checkpoint and execution setting of every backbone and comparison method, including the two models that we trained ourselves. Appendix C reports per-cell results for all methods, the intervention statistics behind Fig. 7, the perception-based grounding evaluation, hand-back action replacements, targeted ablations, and the Harness VLA comparison with runtime measurements. Appendix D describes the robot, the fine-tuning recipe, the scripted demonstrations, the grounding module and the evaluation protocol of the real-world study.

AI USE STATEMENT

In this work, we used generative AI tools (ChatGPT and Claude) to polish the writing, including grammar correction and wording refinement. We also used these tools to assist with the implementation of parts of the method and with scripts for data analysis and aggregation. The method itself, including its design and algorithmic decisions, was developed by the authors. We did not use generative AI tools for formulating research hypotheses, designing experiments, developing theoretical or mathematical claims and their proofs, or interpreting results. Other tasks requiring disclosure under the ICLR generative AI policy were not applicable to this work. We have reviewed all AI-assisted work, and all AI-assisted code was verified and tested by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

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, 2022. URL https://arxiv.org/abs/2204.01691.

Rokas Bendikas, Daniel Dijkman, Markus Peschl, Sanjay Haresh, and Pietro Mazzaglia. Focusing on what matters: Object-agent-centric tokenization for vision language action models, 2025. URL https://arxiv.org/abs/2509.23655.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/black25a.html.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164v4.

Yangtao Chen, Zixuan Chen, Nga Teng Chan, Junting Chen, Junhui Yin, Jieqi Shi, Yang Gao, Yong-Lu Li, and Jing Huo. RoboHiMan: A hierarchical evaluation paradigm for compositional generalization in long-horizon manipulation, 2025. URL https://arxiv.org/abs/2510.13149.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/947018640bf36a2bb609d3557a285329-Abstract.html.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. In *Advances in Neural Information Processing Systems*, volume 38, pp. 102867–102888, 2025. URL https://papers.neurips.cc/paper_files/paper/2025/hash/94e936034d12bcd04834ec2773f02aff-Abstract-Conference.html.

Maximilian Du and Shuran Song. DynaGuide: Steering diffusion polices with active dynamic guidance. In *Advances in Neural Information Processing Systems*, volume 38, pp. 44192–44221, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3ec6c6fc9065aa57785eb05dffe7c3db-Abstract-Conference.html.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs, 2026. URL https://arxiv.org/abs/2602.17659.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models, 2025. URL https://arxiv.org/abs/2510.13626.

Jensen Gao, Annie Xie, Ted Xiao, Chelsea Finn, and Dorsa Sadigh. Efficient data collection for robotic manipulation via compositional generalization. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.013. URL https://www.roboticsproceedings.org/rss20/p013.html.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning VLMs into VLAs without catastrophic forgetting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7a0f8055c838df8e62329a76c7c6403d-Abstract-Conference.html.

Yuchen Hou and Lin Zhao. LangGap: Diagnosing and closing the language gap in vision-language-action models, 2026. URL https://arxiv.org/abs/2603.00592.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. RoboGround: Robotic manipulation with grounded vision-language priors. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22540–22550, 2025a. doi: 10.1109/CVPR52734.2025.02099. URL https://openaccess.thecvf.com/content/CVPR2025/html/Huang_RoboGround_Robotic_Manipulation_with_Grounded_Vision-Language_Priors_CVPR_2025_paper.html.

Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. OTTER: A vision-language-action model with text-aware visual feature extraction. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 25566–25580. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/huang25u.html.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, Qingwen Bu, Haidong Cao, Chao Wu, Qifeng Li, Zhenjie Yang, Chenhe Zhang, Hongyang Li, Zuxuan Wu, Junchi Yan, and Yu-Gang Jiang. GuidedVLA: Specifying task-relevant factors via plug-and-play action attention specialization, 2026. URL https://arxiv.org/abs/2605.12369.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025a. URL https://arxiv.org/abs/2502.19645.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025b. URL https://proceedings.mlr.press/v270/kim25c.html.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. RoboMonkey: Scaling test-time sampling and verification for vision-language-action models, 2025. URL https://arxiv.org/abs/2506.17811.

Peihan Li, Zijian An, Shams Abrar, and Lifeng Zhou. Large language models for multi-robot systems: A survey. *Autonomous Robots*, 50(3):30, 2026. doi: 10.1007/s10514-026-10257-4. URL https://link.springer.com/article/10.1007/s10514-026-10257-4.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, and Siyuan Huang. ControlVLA: Few-shot object-centric adaptation for pre-trained vision-language-action models. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1898–1913. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/li25c.html.

Quanyi Li. VLAs are confined yet capable of generalizing to novel instructions, 2026. URL https://arxiv.org/abs/2505.03500v5.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. LangForce: Bayesian decomposition of vision language action models via latent action queries. In *Forty-third International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65457.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Shuo Liu, Ishneet Sukhvinder Singh, Yiqing Xu, Jiafei Duan, and Ranjay Krishna. VLS: Steering pretrained robot policies via vision-language models, 2026. URL https://arxiv.org/abs/2602.03973.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. Steering your generalists: Improving robotic foundation models via value guidance. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 4996–5013. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/nakamoto25a.html.

NVIDIA. GR00T-N1.7-3B. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/GR00T-N1.7-3B.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" 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, 2025. URL https://arxiv.org/abs/2503.14734.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Kai Peng, Jie Lu, and Xiaojiang Peng. Unleashing more actions via action compositional training for VLA models, 2026a. URL https://arxiv.org/abs/2607.00351.

Xiaojiang Peng, Kai Peng, Jie Lu, Zheng Lian, Zitong YU, and Xiaobo Wang. AC-VLA: Robust out-of-distribution action execution via compositional learning, 2026b. URL https://arxiv.

org/abs/2607.15714.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Vaibhav Saxena, Matthew Bronars, Nadun Ranawaka Arachchige, Kuancheng Wang, Woo Chul Shin, Soroush Nasiriany, Ajay Mandlekar, and Danfei Xu. What matters in learning from large-scale datasets for robot manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/16466b6c95c5924784486ac5a3feeb65-Abstract-Conference.html.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics, 2025. URL https://arxiv.org/abs/2506.01844.

Xiatao Sun, Yuan Zhuang, Mateo Sanchez Lopez Negrete, Matei-Victor Coldea, Chen Liang, Haoyang Zhang, Che Liu, Ziyao Zeng, Shawn Li, Qian Wang, Fei Miao, and Daniel Rakita. Artificial foveated perception for mitigating shortcut learning in robotic foundation models, 2026. URL https://arxiv.org/abs/2607.10655.

Nikolaos Tsagkas, Andreas Sochopoulos, Chris Xiaoxuan Lu, Oisin Mac Aodha, and Alexandros Kouris. Decoupling the declarative from the procedural in vision-language-action models, 2026. URL https://arxiv.org/abs/2606.21496.

Shiang-Feng Tsai, Jin-Cheng Jhang, Yen-Ling Tai, Jia-Hong Lai, Shih-Yun Wong, KangTung-Hsu, and Yi-Ting Chen. Direct action-head injection of a grounded 3D point unlocks spatial and task generalization, 2026. URL https://arxiv.org/abs/2606.27663.

Khoa Vo, Taisei Hanyu, Yuki Ikebe, Trong Thang Pham, Nhat Chung, Minh Nhat Vu, Duy Nguyen Ho Minh, Anh Nguyen, Anthony Gunderman, Chase Rainwater, and Ngan Le. Clutter-robust vision-language-action models through object-centric and geometry grounding, 2025. URL https://arxiv.org/abs/2512.22519.

Yilin Wu, Anqi Li, Tucker Hermans, Fabio Ramos, Andrea Bajcsy, and Claudia Pérez-D'Arpino. Do what you say: Steering vision-language-action models via runtime reasoning-action alignment verification, 2026a. URL https://arxiv.org/abs/2510.16281v2.

Yilin Wu, Zilin Si, Zeynep Temel, Oliver Kroemer, and Andrea Bajcsy. Inference-time policy steering via vision and touch, 2026b. URL https://arxiv.org/abs/2606.14981.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160. IEEE, 2024. doi: 10.1109/ICRA57147.2024.10611331. URL https://arxiv.org/abs/2307.03659.

Youguang Xing, Xu Luo, Junlin Xie, Lianli Gao, Hengtao Shen, and Jingkuan Song. Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3239–3266. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/xing25a.html.

Kechun Xu, Zhenjie Zhu, Anzhe Chen, Rong Xiong, and Yue Wang. APT: Action expert pretraining improves instruction generalization of vision-language-action policies, 2026a. URL https://arxiv.org/abs/2606.12366.

Siyu Xu, Zijian Wang, Yunke Wang, Chenghao Xia, Tao Huang, and Chang Xu. Affordance field intervention: Enabling VLAs to escape memory traps in robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37206–37215, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xu_Affordance_Field_Intervention_Enabling_VLAs_to_Escape_Memory_Traps_in_CVPR_2026_paper.html.

Yue Yang, Shuo Cheng, Yu Fang, Homanga Bharadhwaj, Mingyu Ding, Gedas Bertasius, and Daniel Szafir. LiLo-VLA: Compositional long-horizon manipulation via linked object-centric policies, 2026. URL https://arxiv.org/abs/2602.21531.

Zhihao Zhan, Yuhao Chen, Jiaying Zhou, Qinhan Lyu, Hao Liu, Keze Wang, Liang Lin, and Guangrun Wang. Stable language guidance for vision–language–action models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4137–4159, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.190. URL https://aclanthology.org/2026.acl-long.190/.

Ninghao Zhang, Bin Zhu, Shijie Zhou, and Jingjing Chen. Restoring linguistic grounding in VLA models via train-free attention recalibration, 2026a. URL https://arxiv.org/abs/2603.06001.

Yixian Zhang, Huanming Zhang, Feng Gao, Xiao Li, Zhihao Liu, Yi Nie, Chunyang Zhu, Jiaxing Qiu, Yuchen Yan, Jiyuan Liu, Wenhao Tang, Jiaji Rao, Zhengru Fang, Changxu Wei, Yu Wang, Wenbo Ding, and Chao Yu. Harness VLA: Steering frozen VLAs into reliable manipulation primitives via memory-guided agents, 2026b. URL https://arxiv.org/abs/2607.08448.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007740.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization, 2026. URL https://arxiv.org/abs/2510.03827v2.

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 Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Method Details

A.1 Notation

An action chunk is $A_t = (a_t^1, \ldots, a_t^H)$, with $h \leq H$ actions executed before replanning. The end-effector pose is $x_t = (p_t, R_t) \in SE(3)$, with translation and rotation-vector commands in $\mathbb{R}^3$. Tracking gains satisfy $\kappa_p, \kappa_R \in (0, 1]$.

A.2 Algorithm

Algorithm 1 links referent monitoring to configuration transfer. It checks predicted commitment and approach progress, transports toward the selected reference when needed, and resumes policy execution from the true observation. Motion envelopes and completion conditions coordinate the subsequent interaction.

Table 4: Notation used in the method and its implementation details.

Symbol	Meaning
π_θ, A_t, H, h	frozen policy, action chunk, chunk length, executed actions per replan
$a = (u, w, g)$, $x_t = (p_t, R_t)$	translation, rotation and gripper command, end-effector pose
σ, κ	command scales of the controller, tracking gains fitted on demonstrations
e, c_e, R_e, b_e	scene entity, its position, orientation and bounding-box half-extents
$e_k^\star$, q_t	instructed referent of stage k, moving point (end-effector or held object)
$\mathcal{R}_t$, $\hat{q}_t$, $\hat{R}_t$	unexecuted chunk remainder, predicted endpoint and posture
δ_e, ρ_e, $\chi_t(e)$, e_t^π	offset from the box of e, commitment radius, commitment indicator, policy's referent
$\mathcal{H}_k$, $\mathcal{W}$, $\mathcal{E}_k$	habitual referents, remaining movable entities, candidate set
C_e, t_e, $(\bar{o}, \bar{R})$, $\bar{z}$	canonical alignment, top-face centre, grasp offset and rotation medoid, entry altitude
$p_e^\star$, $R_e^\star$, $\mathcal{K}_e$	pre-contact target and hand-back tolerance set
s, s_R, τ_R, r_e	step bounds, posture tolerance, lateral tolerance of the descent
m_f, $\underline{m}_f$, Ω_d	median height profile of class f, its lower envelope, acceptance region of d
$\phi(e)$	shape descriptor (height, two widths, top fill)
Q_α, n, N, ε	empirical quantile, consecutive count, failure-to-bind window, position resolution
β_k, γ_k, $\tilde{\gamma}_k$, ν_k, η_k	binding, competence, competence after a transport, false and missed interventions

Algorithm 1 ReGuide control loop for one stage k

Require: frozen policy π_θ, instruction ℓ, grounding module, demonstration statistics
budget $\leftarrow 2$, transport $\leftarrow$ none, $\mathcal{R}_t \leftarrow \emptyset$
for each control step t **do**
  read x_t and the entities of the scene, update the interaction state and the moving point q_t
  if a transport is active **then**
    if its premise changed **or** its error did not decrease over n replans **then**
      dissolve the transport
    else if $x_t \in \mathcal{K}_e$ (or the held object is one step above the release height over d) **then**
      end the transport, discard $\mathcal{R}_t$ ▷ hand-back on the true observation
    else
      execute the bounded command of equation 6, with the height following m_f while an object is held
      continue
    end if
  end if
  if $\mathcal{R}_t = \emptyset$ **then** $A_t \leftarrow \pi_\theta(o_t, \ell)$, $\mathcal{R}_t \leftarrow A_t$
  end if
  compute $\hat{q}_t$, $\hat{R}_t$ (Eq. 2) and the policy's referent e_t^π (Eq. 3)
  if budget > 0 **and** (wrong binding **or** posture mismatch **or** failure to bind **or** unseen object at the first decision) **then**
    set the target of equation 4 or the destination target, start a transport, budget $\leftarrow$ budget -1
  else
    pop the next action of $\mathcal{R}_t$, clamp it to the demonstrated envelope, execute it
  end if
  if a completion certificate holds **then** re-arm the transport or complete the release
  end if
end for

A.3 Binding and Competence Under Intervention

A wrapper intervenes on a correct binding with probability ν_k, misses a wrong binding with probability η_k, and on every intervention delivers a state with competence $\tilde{\gamma}_k$. If intervention and outcome are independent given binding, then

$$\Pr_{\mathrm{RG}}(S_k) = \beta_k\big[(1-\nu_k)\,\gamma_k + \nu_k\,\tilde{\gamma}_k\big] + (1-\beta_k)(1-\eta_k)\,\tilde{\gamma}_k. \tag{7}$$

Derivation of equation 7. Let I_k be the event that the wrapper intervenes in stage k. Partitioning on B_k and I_k gives $\Pr(S_k) = \Pr(B_k)\big[\Pr(\neg I_k \mid B_k)\Pr(S_k \mid B_k, \neg I_k) + \Pr(I_k \mid B_k)\Pr(S_k \mid B_k, I_k)\big] + \Pr(\neg B_k)\Pr(I_k \mid \neg B_k)\Pr(S_k \mid \neg B_k, I_k)$, where the missing term vanishes because $S_k \subseteq B_k$ under the bare policy, so a wrong binding that is not corrected cannot succeed. With $\nu_k = \Pr(I_k \mid B_k)$, $\eta_k = \Pr(\neg I_k \mid \neg B_k)$, $\Pr(S_k \mid B_k, \neg I_k) = \gamma_k$ by the independence assumption, and $\Pr(S_k \mid \cdot, I_k) = \tilde{\gamma}_k$, equation 7 follows. Setting $\nu_k = \eta_k = 0$ and $\tilde{\gamma}_k = \gamma_k$ gives $\Pr_{\mathrm{RG}}(S_k) = \gamma_k$, the nominal competence level motivating restoration of the interaction conditions.

A.4 Stages, Events and Interaction States

Each relocation clause specifies an object and a destination, with an approach stage followed by a carry stage (Sec. 3.1). Contact events determine which referent, configuration and motion statistics apply.

Demonstration events. Engagement is the first frame from which the entity moves with the end-effector over a window of eight frames. The effect starts when the entity rises more than 3 cm above its rest height, and termination is the end of co-motion. These events use relative motion without reading gripper width. Approach statistics precede engagement, while carry statistics span effect onset to termination.

Execution state. The same detector tracks five states, free, near, attached, effecting and detaching. The held predicate corresponds to effecting. Posture compatibility is checked only while free, and the failure-to-bind counter runs only without physical interaction with the instructed object. In multi-clause tasks, the active instance switches when the policy commits to or touches an object of another unsatisfied clause. Such an object is a legitimate referent. After a completed clause, the next instance is armed once the end-effector leaves the previous object.

A.5 Details of Semantic Rebinding

Competing referents. The first frame after the scene has settled is embedded with an ImageNet-pretrained ResNet-50 and matched to the first frames of the training demonstrations. The goal predicates of the training tasks of the retrieved scene provide the habitual objects and destinations, and every relocation clause of a habitual task contributes. A habitual object is grounded in the current scene by name when an entity of that name exists and otherwise by geometric identity (Appendix A.6). Objects of the instruction itself are never habitual referents. The watch set $\mathcal{W}$ contains every other body with a free, sliding or hinge joint.

Commitment radii. The entry event of a demonstration is the first frame at which the end-effector lies inside the bounding box of the entity along two of the three world axes, which covers top grasps and side grasps alike. The commitment radius ρ_e is the $Q_{.25}$ of the distance to the box surface at that frame for the training type of e. Typical values in LIBERO are 3.3 cm for the bowl, 7.8 cm for the moka pot, 7.2 cm for the mug, 1.1 cm for the wine bottle and 1.8 cm for the milk carton.

Referent monitoring during carrying. Equation 3 uses the held object's predicted horizontal position relative to each destination. A mismatch requires commitment to a habitual destination, by heading or dwelling, without commitment to the instructed destination. Heading must persist for $n = 3$ consecutive steps, while dwelling counts immediately. Transport also requires a stable grasp and compatibility with the carry envelope. The gripper width must remain at rest for three steps within the demonstrated band, and the held object must lie inside the demonstrated height envelope. Otherwise, the height floor first lifts it into that envelope.

Posture compatibility. Before descent beside the object and while the interaction state is free, guidance is triggered when $\vartheta(R_e^\star, \hat{R}_t) > \tau_R$ and $\vartheta(R_e^\star, \hat{R}_t) \geq \vartheta(R_e^\star, R_t) - s_R$. The predicted error

must therefore exceed the demonstrated tolerance without improving over the current error by more than one rotation step.

Progress toward the grasp configuration. Progress is measured on three channels, the position residual to the demonstrated grasp point in units of s, the posture residual in units of s_R, and the closing of the gripper. The window is $N = 40$ steps, one more than the longest run without progress in the approach segments of the single-clause demonstrations, and $N = 75$ for an instance issued after a completed clause.

A.6 Details of Geometric Rebinding

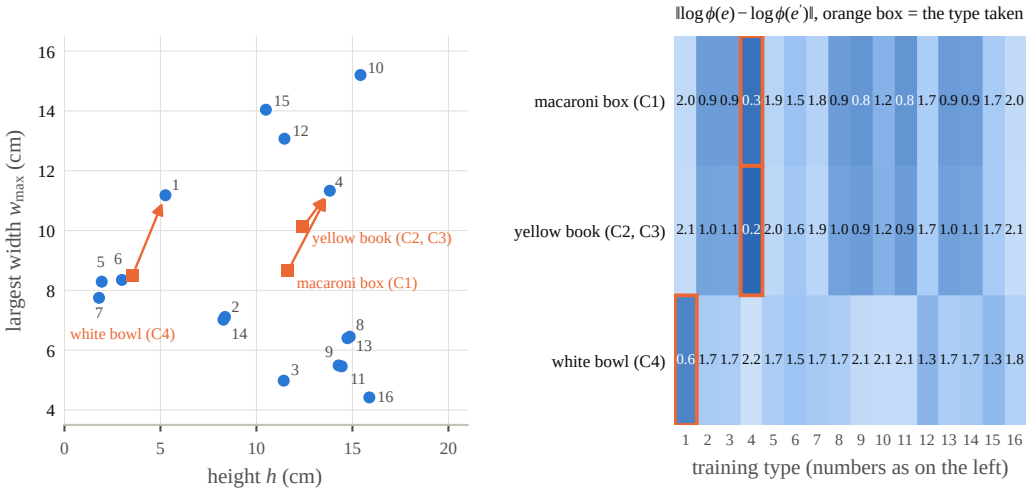

Figure 9: Geometric identity. Left, height and largest width of the 16 training types (1 black bowl, 2 alphabet soup, 3 bbq sauce, 4 black book, 5 butter, 6 chocolate pudding, 7 cream cheese, 8 ketchup, 9 milk, 10 moka pot, 11 orange juice, 12 porcelain mug, 13 salad dressing, 14 tomato sauce, 15 white and yellow mug, 16 wine bottle) and of the three unseen objects of the benchmark, with an arrow to the type whose statistics they take. Right, the distance $\|\log\phi(e) - \log\phi(e')\|$ over the four components of the descriptor between each unseen object and every type. The macaroni and cheese box and the yellow book take the black book, at 0.33 and 0.19, and the white bowl takes the black bowl, at 0.58.

Canonical frame and geometric identity. The alignment C_e defines the gravity-aligned frame R_eC_e of equation 4, ordering horizontal axes by extent at rest. The descriptor $\phi(e)$ holds the height, the two widths and the top fill, which is the fraction of a 24×24 grid of downward rays whose first hit lies in the top quarter of the object and separates hollow from solid objects. An object is of a training type when its three sorted extents agree with those of the type within ε, and it is unseen otherwise, in which case it takes the statistics of the type with the smallest $\|\log\phi(e) - \log\phi(e')\|$. Names do not enter. A renamed copy of a training object is seen, and a new geometry under a familiar name is unseen. When the footprint of an object is rotationally symmetric within ε, its rotation about the vertical axis is not observable, and the reference configuration uses only the up axis and the top face. Fig. 9 places the 16 training types and the three unseen objects of the benchmark in the descriptor space.

Approach configuration. The target of equation 4 hovers above the demonstrated grasp point at the $Q_{.5}$ entry altitude of the type. While the yaw residual exceeds its $Q_{.95}$ tolerance, the hover height is raised to the demonstrated altitude at which the alignment is completed, since the demonstrations align before the final descent. The grasp offset, the rotation medoid and the tolerances are taken from the slice of demonstrations with the same object type and, in order of preference, the same destination, the same destination class, or the class with the closest release tilt.

Placement references and carry envelope. Statistics are pooled by destination class, the declared type of an entity or its interaction region. The median release offset is expressed in the fixture frame

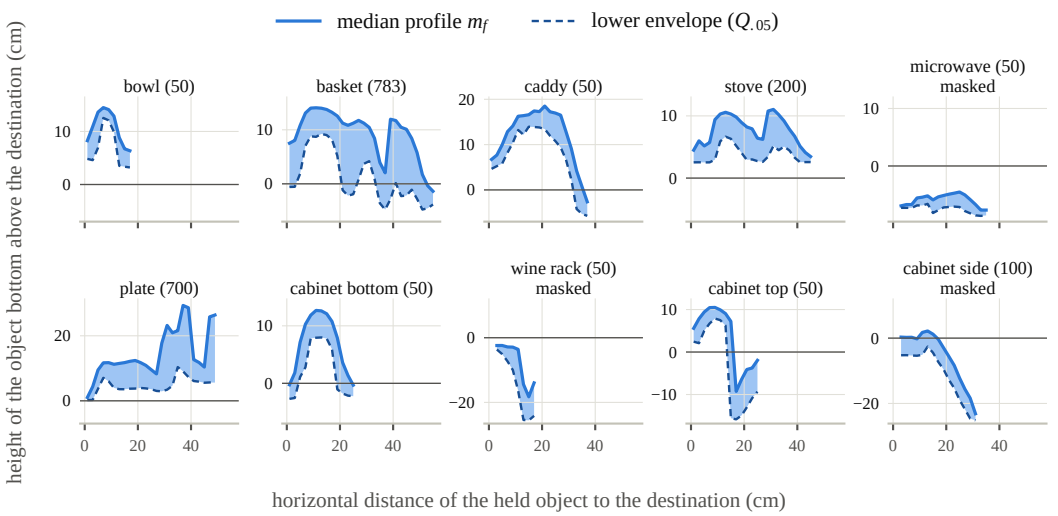

Figure 10: Height profiles of the ten destination classes. The height of the bottom of the held object above the destination plane is plotted against the horizontal distance to the destination, as the median m_f (solid) and the lower envelope $\underline{m}_f$, the $Q_{.05}$ (dashed), in bins of 2 cm with at least 20 samples. Numbers in parentheses are the contributing demonstrations. In the three masked classes the envelope stays below the destination plane over the final approach, and a transport onto them keeps the current height.

and rotated by its yaw. Adding the grasp offset measured at transport onset converts this object reference to an end-effector target. The height profile uses 2 cm bins with at least 20 samples per bin and 10 demonstrations per class. Its floor is the maximum of $\underline{m}_f$ between the destination and the nearer of the object and the first peak of m_f, coordinating lifting with translation. Three of the ten destination classes are approached from the side in LIBERO demonstrations, the microwave, wine rack and cabinet-top side region. Their lower envelope stays below the destination plane, so height-profile transport is masked for these classes. Transport keeps the current height and the policy performs placement. Fig. 10 shows the profiles and envelopes.

Transport applicability. A reference is tied to the interaction state at transport onset. The transport dissolves if the held predicate changes, if the object's up axis leaves the rest-frame class used by the demonstrations, or if error fails to decrease by more than ε over n replans. Each stage permits at most two transports, with the second requiring a completion certificate.

Recovery and release. Completion certificates distinguish loss of the destination from lack of progress toward completion. A transport is re-armed if the held object leaves Ω_d for n steps after transport, or if it rests outside Ω_d after release and the policy has disengaged. Recovery also applies when the held object makes no progress toward the destination for N steps, or dwells outside Ω_d within the carry radius longer than the $Q_{.95}$ demonstrated dwell. If it instead remains inside Ω_d without descending for longer than the $Q_{.95}$ demonstrated dwell, the wrapper completes descent to the $Q_{.5}$ release height while holding horizontal position at $p_d^\star$, then opens the gripper within the demonstrated release band. Every transport onto a supporting surface uses the same release procedure.

A.7 Demonstration Statistics and Declared Constants

Table 5 lists the reference statistics used for commitment, configuration transfer and execution. These statistics are computed once from the demonstrations that trained the policy. In LIBERO these are the demonstrations of the four training suites, which give 16 object types and 10 destination classes, the tracking gains are $\kappa_p = 0.237$ and $\kappa_R = 0.211$, and the step bounds are $s = 1.24$ cm in the approach, 1.34 cm in the carry and $s_R = 0.027$ rad. The constants that are declared rather than mined are the consecutive count $n = 3$, the position resolution $\varepsilon = 4$ mm, the bin width of 2 cm, the evidence floors of 20 samples per bin and 10 demonstrations per class, the event thresholds of Sec. A.4, and the

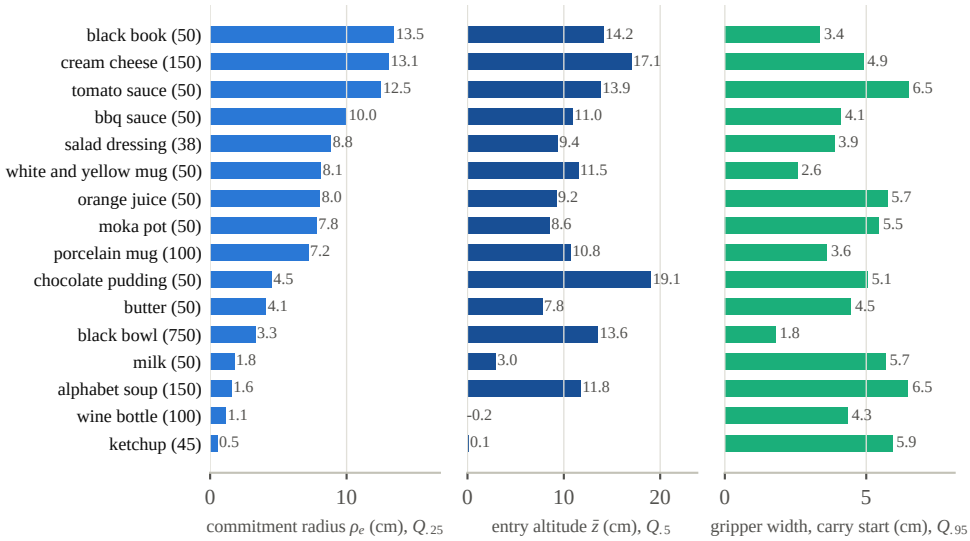

Figure 11: Per-type demonstration statistics of the LIBERO training suites. The commitment radius ρ_e is the $Q_{.25}$ of the distance to the box surface at the entry event, the entry altitude $\bar{z}$ the $Q_{.5}$ of the height of the end-effector above the object top at the entry event, and the gripper band the $Q_{.95}$ of the gripper width at the start of a carry. Numbers in parentheses are the demonstrations of the type. The entry altitude is close to zero for the wine bottle and the ketchup.

Table 5: Demonstration statistics read by ReGuide. A slice is a pair of object type and task.

Statistic	**Measured on**	**Level and role**
tracking gains κ_p, κ_R	realized against commanded motion, per demonstration	$Q_{.5}$, unit conversion
step bounds s, s_R	per-step displacement and rotation, approach and carry	$Q_{.95}$, upper bound
commitment radius ρ_e	distance to the box at the entry event, per object type	$Q_{.25}$, trigger
carry radius	distance to the destination during demonstrated carries, per class	$Q_{.25}$, trigger
entry altitude $\bar{z}$	height above the object top at the entry event, per type	$Q_{.5}$, target
grasp offset $\bar{o}$, rotation $\bar{R}$	end-effector pose at engagement in the canonical frame, per slice	median and medoid, target
posture tolerance τ_R, yaw tolerance	rotation residual at the entry event, per slice	$Q_{.95}$, upper bound
lateral tolerance r_e	scatter of the grasp along the closing axis, per slice	$Q_{.95}$, upper bound
release offset $\bar{o}_f$, tolerance	object position at release in the fixture frame, per class	$Q_{.5}$ target, $Q_{.95}$ bound
height profile m_f, envelope $\underline{m}_f$	bottom of the held object against distance to the destination	$Q_{.5}$ target, $Q_{.05}$ lower bound
dwell and release durations	frames near the destination, frames from release to rest	$Q_{.95}$, upper bound
gripper band at the start of a carry	gripper width and its change, per object type	$Q_{.95}$, upper bound
shape descriptor ϕ, frame C_e	geometry of the training object types	identity and transfer

budget of two transports per stage. Sec. C.4 varies n, ε and the quantile levels. Fig. 11 shows three of the per-type statistics for the 16 object types.

B Simulation Benchmark and Protocol

Benchmark design rationale. Existing LIBERO extensions perturb tasks along appearance, layout, wording, viewpoint or lighting (LIBERO-PRO (Zhou et al., 2026), LIBERO-Plus (Fei et al., 2025)), or extrapolate a new task from two trained ones (libero-ood (Li, 2026), the behaviour-composition tasks of LIBERO-10-R (Wu et al., 2026a)). Our cells instead change exactly one factor of one training task: the destination, the object, an object never manipulated in training, or the host scene. A failure in a cell is therefore attributable to that factor, object and destination bindings are probed separately, and recombination is separated from object novelty.

B.1 Cells

Table 6: The 16 composition cells. Axes A to D are Target Region, Target Object, Unseen Object and Task Scene.

Cell	Host scene and anchor task	Change	Instruction
A1	Spatial, bowl at table centre to plate	destination is the stove	pick up the black bowl from table center and place it on the stove
A2	Goal, cream cheese to bowl	destination is the open top drawer	put the cream cheese in the top drawer of the wooden cabinet
A3	Goal, bowl to plate	destination is the open top drawer	put the black bowl in the top drawer of the wooden cabinet
A4	Goal, wine bottle to rack	destination is the stove	put the wine bottle on the stove
B1	Living room 6, white mug to plate	the mug is replaced in place by a moka pot	put the moka pot on the plate
B2	Living room 6, white mug to plate	the target is the chocolate pudding of the scene	put the chocolate pudding on the plate
B3	Goal, bowl to plate	the target is the cream cheese of the scene	put the cream cheese on the plate
B4	Goal, bowl to plate	the target is the wine bottle of the scene	put the wine bottle on the plate
C1	Object, alphabet soup to basket	object is a macaroni and cheese box	pick up the macaroni and cheese box and place it in the basket
C2	Goal, bowl to plate	object is a yellow book	put the yellow book on the plate
C3	Object, ketchup to basket	object is a yellow book	pick up the yellow book and place it in the basket
C4	Goal, bowl to plate	object is a white bowl	put the white bowl on the plate
D1	Object, alphabet soup to basket	moved to the Spatial kitchen table	pick up the alphabet soup and place it in the basket
D2	Object, salad dressing to basket	moved to the table of Living room 6	pick up the salad dressing and place it in the basket
D3	Object, alphabet soup to basket	moved to the counter of Kitchen 8	pick up the alphabet soup and place it in the basket
D4	Object, milk to basket	moved to the counter of Kitchen 8	pick up the milk and place it in the basket

Every cell is anchored on one training task of LIBERO and changes one factor. The object and the destination of a cell are placed at positions that the arm manipulated in the demonstrations of the host scene, native movable objects that conflict with the new task are removed, and the official region names are reused. We validate scene stability and confirm that the goal predicate is false at initialization and true when the object is placed at the specified goal. Table 6 lists the 16 cells, and Figs. 12 and 13 show the initial state of every cell next to its anchor task. The unseen objects are the macaroni and cheese box, which occurs in no LIBERO suite, and the yellow book and the white bowl, which occur in LIBERO-90 but in none of the four suites on which the evaluated checkpoints and the statistics of ReGuide are based. C4 is kept as a reference cell in which the trained behaviour already satisfies the new instruction. The ten original tasks are the anchors of the cells.

B.2 Protocol

For every cell, 50 initial simulator states are generated once and stored, and every method replays the same states, so trial i of any two methods starts from the same state. An episode starts with 10

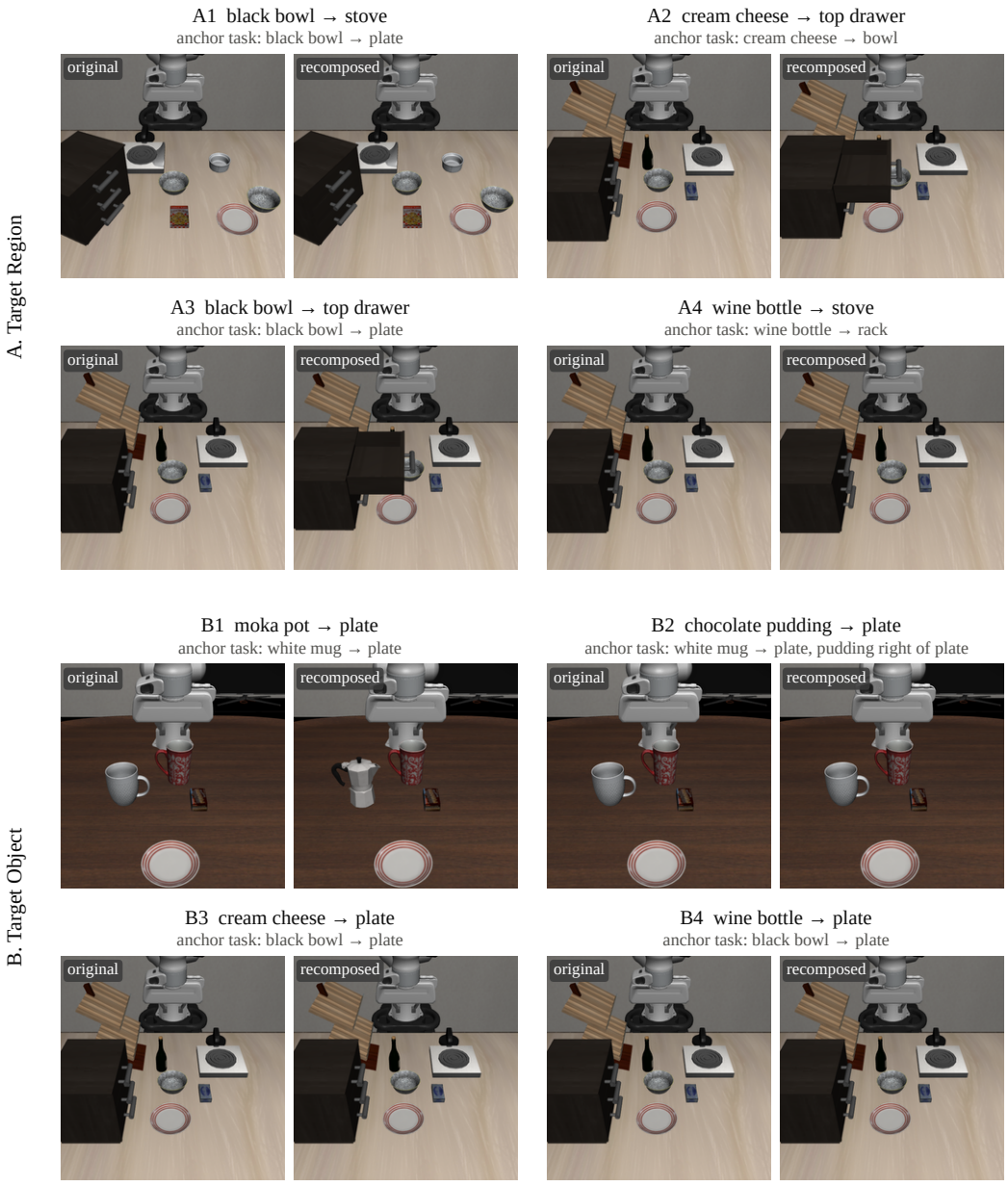

Figure 12: Composition cells of the Target Region and Target Object axes. For every cell the left image is the initial state of the unmodified anchor task and the right image the initial state of the cell, rendered from the first initial state of its evaluation bank. The changed factor of each cell is listed in Table 6.

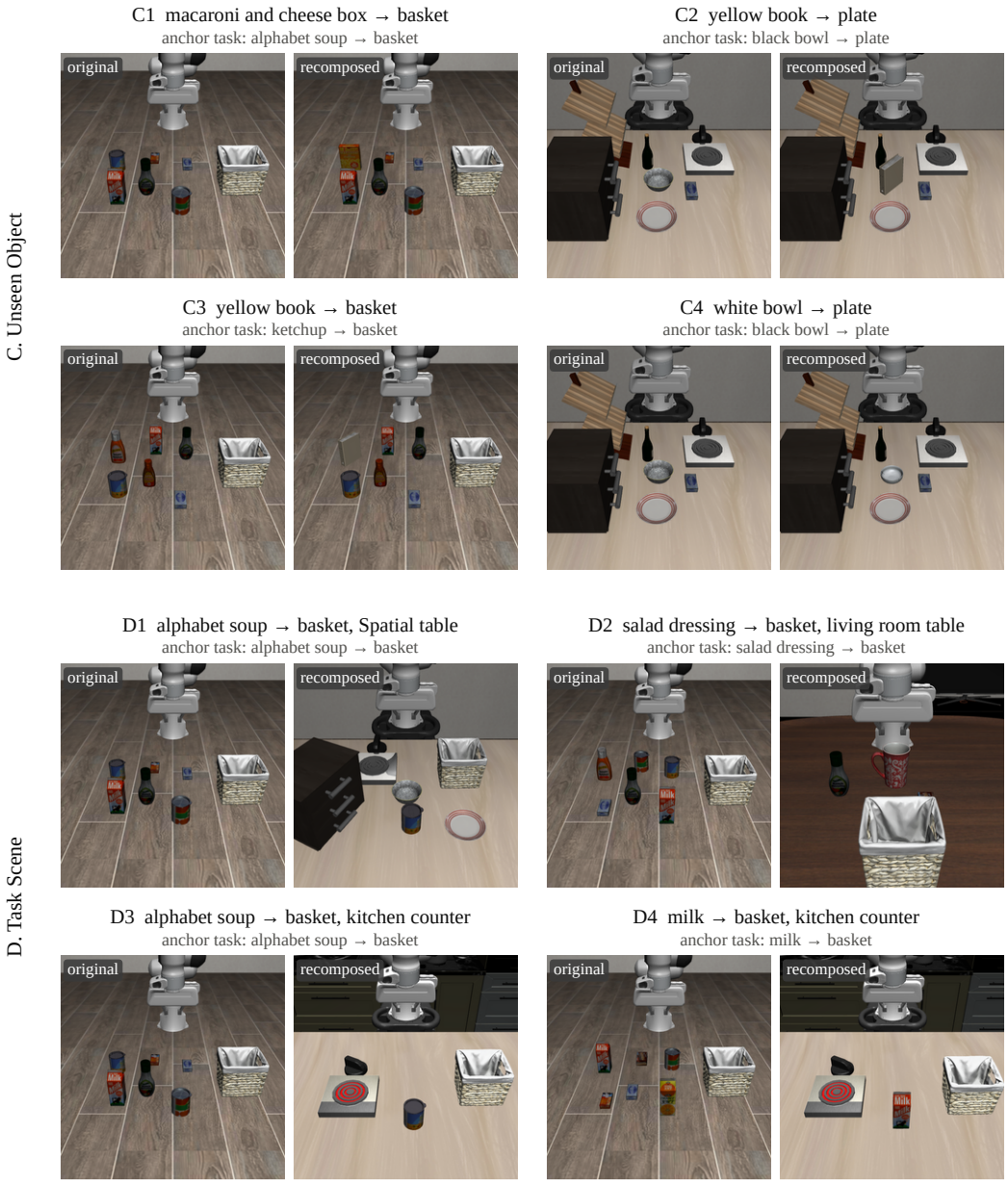

Figure 13: Composition cells of the Unseen Object and Task Scene axes, in the layout of Fig. 12. For the Task Scene cells the left image shows the source task in its own scene and the right image the same task on the host surface.

idle steps and ends at success according to the goal predicate of LIBERO or after 520 steps, the largest official budget, for all cells and methods. Table subscripts are 95% Wilson half-widths. Paired comparisons use the exact two-sided McNemar test on the trials of the two methods with the same initial state.

B.3 Policies and Comparison Methods

Table 7: Checkpoints and execution settings.

Method	Checkpoint	Execution
π_0, $\pi_{0.5}$	official LIBERO checkpoints of openpi	official client, 5 actions executed per replan, chunks of 50 actions for π_0 and 10 for $\pi_{0.5}$
OpenVLA-OFT	official checkpoint trained on the four suites	chunk of 8 executed open loop, two images and proprioception
X-VLA	official LIBERO checkpoint	official client with absolute end-effector targets, the full chunk of 30 actions executed open loop
GR00T N1.7	fine-tuned by us on the four suites, see below	official horizon, 16 actions predicted and 8 executed
OTTER	trained by us with the official code, as no checkpoint is released, see below	official evaluation loop, one action per step with a context of 12 frames
GuidedVLA	released checkpoint, built on π_0, see below	official server with corrected depth-adapter checkpoint loading
APT	official LIBERO checkpoint	official inference service with an ensemble of 4
CAG	training-free variant on the official $\pi_{0.5}$ checkpoint	authors' server script, guidance weight $\omega = 1.5$ as set in their paper for $\pi_{0.5}$

Checkpoints without an official four-suite release. GR00T N1.7 has no released checkpoint that covers the four suites. We fine-tune the released base model with the official fine-tuning script on the four official LIBERO datasets with no-op actions removed, which hold 1,693 demonstrations and 273,465 frames. The language model and the vision encoder stay frozen, and the projector, the diffusion transformer and the vision-language layer norm are trained. Training runs for 80,000 steps on two GPUs with a global batch size of 640, AdamW, a peak learning rate of 10^{-4} with a cosine schedule and a warm-up ratio of 0.05, a weight decay of 10^{-5}, a state dropout of 0.2 and colour jitter. OTTER is trained with the authors' code on the same four datasets, sampled in proportion to their number of frames. It uses a frozen CLIP ViT-L/14, a chunk of 12 actions, a batch size of 64, a peak learning rate of 3×10^{-4} that decays to zero after 2,000 warm-up steps, a weight decay of 0.01 and 80,000 steps on one RTX A6000. The GuidedVLA checkpoint is the one released by its authors for their full configuration with object, depth and skill heads. According to the released configuration it starts from the π_0 base model and is trained for 30,000 steps with a batch size of 64 on the LIBERO dataset released by the authors, which holds the 40 tasks of the four suites in 1,722 demonstrations and 277,947 frames with object masks and skill labels.

ReGuide in simulation. The three wrapped backbones share demonstration references, intervention criteria and the execution protocol. Only the chunk length and the number of executed actions differ, which are 10 and 5 for $\pi_{0.5}$, 16 and 8 for GR00T N1.7, and 8 and 8 for OpenVLA-OFT. The grounding module is replaced by the simulator. The identity of the instructed object and destination is read from the specification of the cell, and the poses and extents of the entities are read from the simulator state. The instruction text is used only by the policy and by the retrieval of habitual referents. Appendix C.6 evaluates perception-based grounding in simulation, and Appendix D describes the detector used on the real robot.

C Additional Simulation Results

C.1 Pre-contact Initialization Diagnostic

Protocol. We isolate configuration transfer by testing whether a demonstration-derived starting pose restores local grasping under recomposition. Using $\pi_{0.5}$, we compare the default start with a single transport to the instructed object's demonstration-derived pre-contact target at the first decision. After arrival, the pending action chunk is discarded and the policy is queried on the true observation, with all subsequent ReGuide interventions and action constraints disabled. Both conditions use the same 50 initial states per cell and the same episode budget.

The diagnostic covers 10 original tasks and 8 recomposed cells from the Target Object and Task Scene axes, totaling 1800 episodes. The pooled original-task analysis uses the 8 single-clause tasks.

O5 and O6 are reported separately because the bare-condition instrumentation tracks only the first clause's object, making their grasp measurements asymmetric across conditions. This adjustment was recorded before the affected cells were run. All 900 initialization transports arrived, with no gripper–object contact or object displacement before arrival.

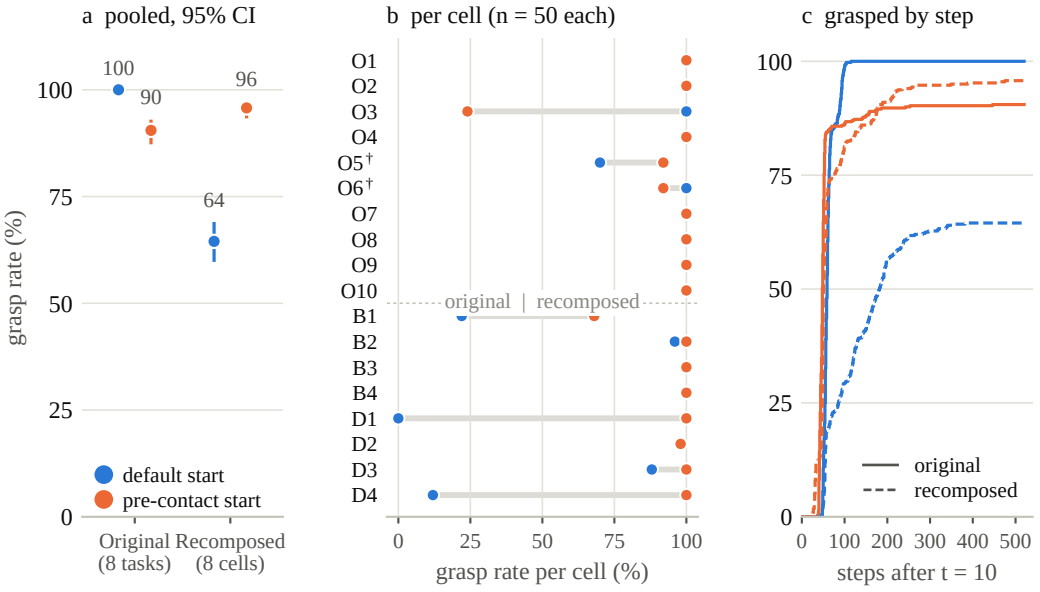

Figure 14: **Pre-contact initialization diagnostic.** Default and pre-contact starts for frozen $\pi_{0.5}$. (a) Pooled grasp success with 95% confidence intervals. (b) Per-cell grasp success; O5 and O6 are multi-clause tasks excluded from the pooled original-task results. (c) Cumulative grasp success from the first decision, including initialization time.

Table 8: **Diagnostic outcomes.** Grasp requires lifting the instructed object more than 3 cm with attachment maintained. Task success uses the original benchmark criterion. All trials in the pooled cells are retained.

Metric	Task type	Default start	Pre-contact start
Grasp	Original	400/400	362/400
	Recomposed	258/400	383/400
Task success	Original	397/400	343/400
	Recomposed	136/400	266/400

Recovery of local grasping. On recomposed cells, grasp success increases from 64.5% to 95.8%. Of the paired trials, 131 succeed only with pre-contact initialization and 6 only with the default start (exact McNemar $p < 10^{-31}$). The median time from policy takeover to grasp decreases from 116 to 35 steps. Recovery is especially pronounced on D1 and D4, where grasp success rises from 0% and 12%, respectively, to 100%. On D1, task success remains 0/50 despite recovered grasping, locating the remaining difficulty in the carry stage.

C.2 Per-Cell Results

Tables 9 and 10 give the successes out of 50 for every cell and every original task (Fig. 15), and Table 11 the paired tests of ReGuide against its bare backbone. Without ReGuide the successes concentrate on the seven cells of the Goal table scene, A2 to A4, B3, B4, C2 and C4, which hold 187 of the 195 composition successes of APT and 92 of the 104 of GR00T N1.7, and within them on C4, where the trained behaviour already satisfies the instruction.



Figure 15: The ten original tasks O1 to O10 of Table 10, each shown at the first initial state of its evaluation bank.

Table 9: Successes out of 50 per composition cell.

Method	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4	Σ
OpenVLA-OFT	0	0	27	0	3	0	0	0	3	0	31	35	0	13	0	0	112
π_0	0	0	10	0	0	0	3	0	0	7	0	47	0	0	1	0	68
$\pi_{0.5}$	1	5	42	33	4	13	9	24	3	19	27	50	0	42	37	3	312
X-VLA	0	47	31	0	1	6	12	0	0	2	16	50	0	0	0	0	165
GR00T N1.7	0	0	8	21	2	1	10	6	0	1	9	46	0	0	0	0	104
OTTER	0	0	0	0	0	0	0	0	0	0	15	11	0	0	1	0	27
GuidedVLA	0	0	4	0	0	0	13	0	0	1	2	50	0	0	1	0	71
APT	0	31	5	38	3	1	49	14	0	0	1	50	0	3	0	0	195
CAG ($\pi_{0.5}$)	2	32	39	33	8	13	25	24	5	6	29	48	0	37	30	0	331
ReGuide (OpenVLA-OFT)	50	30	47	47	2	49	24	23	30	21	50	44	49	34	40	26	566
ReGuide ($\pi_{0.5}$)	50	48	46	48	49	49	49	39	48	46	46	49	49	42	48	50	756
ReGuide (GR00T N1.7)	50	27	25	36	2	23	48	32	32	2	48	36	49	43	50	50	553

Table 10: Successes out of 50 per original task. O1 black bowl from table centre to plate, O2 cream cheese to bowl, O3 wine bottle to rack, O4 bowl to plate, O5 both moka pots to stove, O6 white mug to plate and chocolate pudding to the right of the plate, O7 alphabet soup, O8 ketchup, O9 salad dressing, O10 milk to basket.

Method	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	Σ
OpenVLA-OFT	49	49	50	50	46	46	50	50	48	50	488
π_0	49	49	40	50	5	40	49	48	50	50	430
$\pi_{0.5}$	50	50	47	50	31	44	50	50	48	50	470
X-VLA	49	49	46	50	48	47	50	50	50	49	488
GR00T N1.7	50	49	49	49	34	44	50	50	48	50	473
OTTER	46	42	31	50	21	30	41	50	50	50	411
GuidedVLA	49	49	43	49	39	45	50	50	50	50	474
APT	50	50	48	50	14	45	50	50	50	50	457
CAG ($\pi_{0.5}$)	49	46	44	50	10	17	48	47	46	50	407
ReGuide (OpenVLA-OFT)	49	47	50	50	47	42	50	50	49	50	484
ReGuide ($\pi_{0.5}$)	49	50	50	49	34	40	49	50	50	50	471
ReGuide (GR00T N1.7)	49	49	50	49	33	46	49	47	49	50	471

Table 11: ReGuide against its bare backbone on paired trials. Wins are trials solved only by ReGuide, losses trials solved only by the bare policy, and p is the exact McNemar test.

Backbone	**Split**	**Bare**	**ReGuide**	**Δ** (pp)	**Wins**	**Losses**	p
OpenVLA-OFT	Target Region	27/200	174/200	+73.5	148	1	<0.001
	Target Object	3/200	98/200	+47.5	97	2	<0.001
	Unseen Object	69/200	145/200	+38.0	81	5	<0.001
	Task Scene	13/200	149/200	+68.0	136	0	<0.001
	Comp.	112/800	566/800	+56.8	462	8	<0.001
	Orig.	488/500	484/500	-0.8	6	10	0.454
$\pi_{0.5}$	Target Region	81/200	192/200	+55.5	116	5	<0.001
	Target Object	50/200	186/200	+68.0	139	3	<0.001
	Unseen Object	99/200	189/200	+45.0	95	5	<0.001
	Task Scene	82/200	189/200	+53.5	115	8	<0.001
	Comp.	312/800	756/800	+55.5	465	21	<0.001
	Orig.	470/500	471/500	+0.2	22	21	1.000
GR00T N1.7	Target Region	29/200	138/200	+54.5	114	5	<0.001
	Target Object	19/200	105/200	+43.0	92	6	<0.001
	Unseen Object	56/200	118/200	+31.0	76	14	<0.001
	Task Scene	0/200	192/200	+96.0	192	0	<0.001
	Comp.	104/800	553/800	+56.1	474	25	<0.001
	Orig.	473/500	471/500	-0.4	18	20	0.871

Table 12: Failures of ReGuide on the 800 composition trials, by the stage at which the episode was lost.

ReGuide on	**Failures**	**Grasp**	**Carry**	**After arrival**	**Cells with most failures**
$\pi_{0.5}$	44	11	3	30	B4 (11), D2 (8)
GR00T N1.7	247	106	36	105	B1 (48), C2 (48)
OpenVLA-OFT	234	113	32	89	B1 (48), C2 (29)

Remaining failures. Table 12 splits the failures of ReGuide by the last stage that was reached. For GR00T N1.7 and OpenVLA-OFT the failures concentrate in cells where the instructed object is reached and the resumed policy does not complete the grasp, as in B1, where the object is held in 8 and 7 of 50 episodes against 50 for $\pi_{0.5}$.

C.3 Intervention Statistics

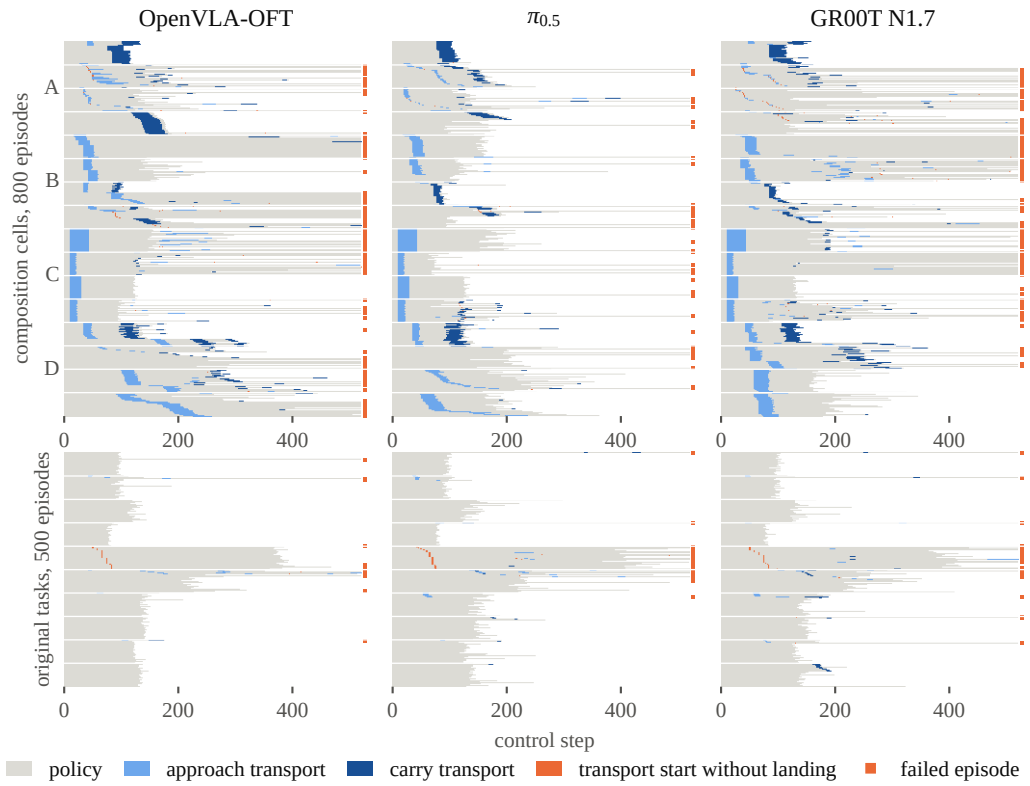

Figure 16: Every ReGuide episode in simulation as one row, composition cells above and original tasks below, sorted within each cell by the start of the first transport. Light blue marks the steps of an approach transport and dark blue those of a carry transport. Orange marks a transport start without a landing, that is a transport that was dissolved, hijacked or cancelled, and the squares on the right mark failed episodes.

An episode is assigned to one class with the priority wrong binding, posture, reactive, none. The posture class contains the unseen-object rule, which by construction fires at the first decision of every episode of the Unseen Object axis. The reactive class contains stalled-approach recovery and completion certificates. These classes distinguish referent mismatch, configuration compatibility and interaction progress. The classes are the verdicts of the wrapper and not ground-truth labels of the binding. On the original task with two clauses in Kitchen 8, a transport that is issued at the switch of the clause is cancelled one step later in every episode, and these one-step transports are not counted. With them the transport rates on original tasks are 20.4, 24.2 and 14.4%. The length change compares trials that both methods solve from the same initial state, which are two independent rollouts.

Wrong-binding corrections occur in 1/500, 2/500 and 0/500 original episodes for $\pi_{0.5}$, GR00T N1.7 and OpenVLA-OFT, respectively. Among successful composition episodes with at least one transport, the median fractions of control steps spent in transport are 14.3%, 14.6% and 15.4%, respectively.

Figs. 16–18 detail transport timing, causes, and paired episode lengths. On the 800 composition episodes $\pi_{0.5}$ issues 559 approach and 304 carry transports, GR00T N1.7 729 and 408, and OpenVLA-OFT 751 and 391. The approach transports mostly start in the first hundred steps, with a median first transport at step 35, 42 and 34 for $\pi_{0.5}$, GR00T N1.7 and OpenVLA-OFT, and the carry transports

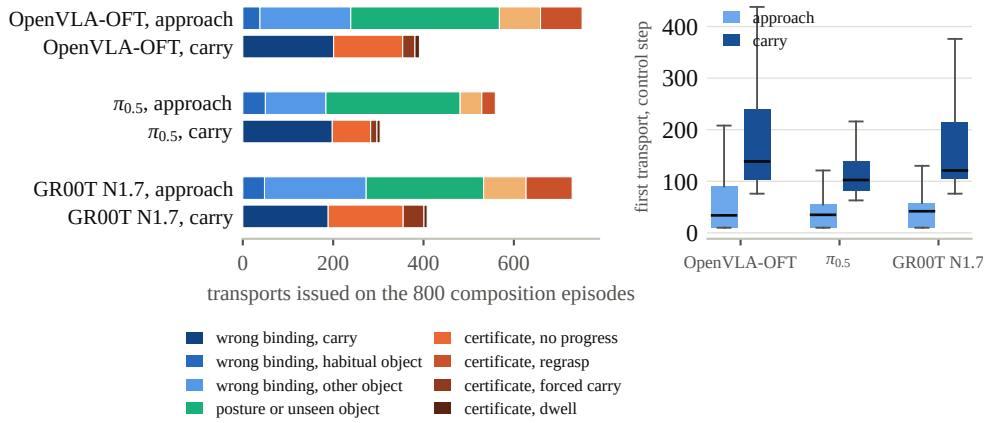

Figure 17: Causes of the transports issued on the 800 composition episodes (left) and the step of the first transport of an episode (right, boxes are quartiles and whiskers the range without outliers). Wrong binding denotes a predicted commitment to an incorrect object during approach or a habitual destination during carrying. Posture covers the posture mismatch and the unseen-object rule, and the remaining causes are the failure to bind and the completion certificates.

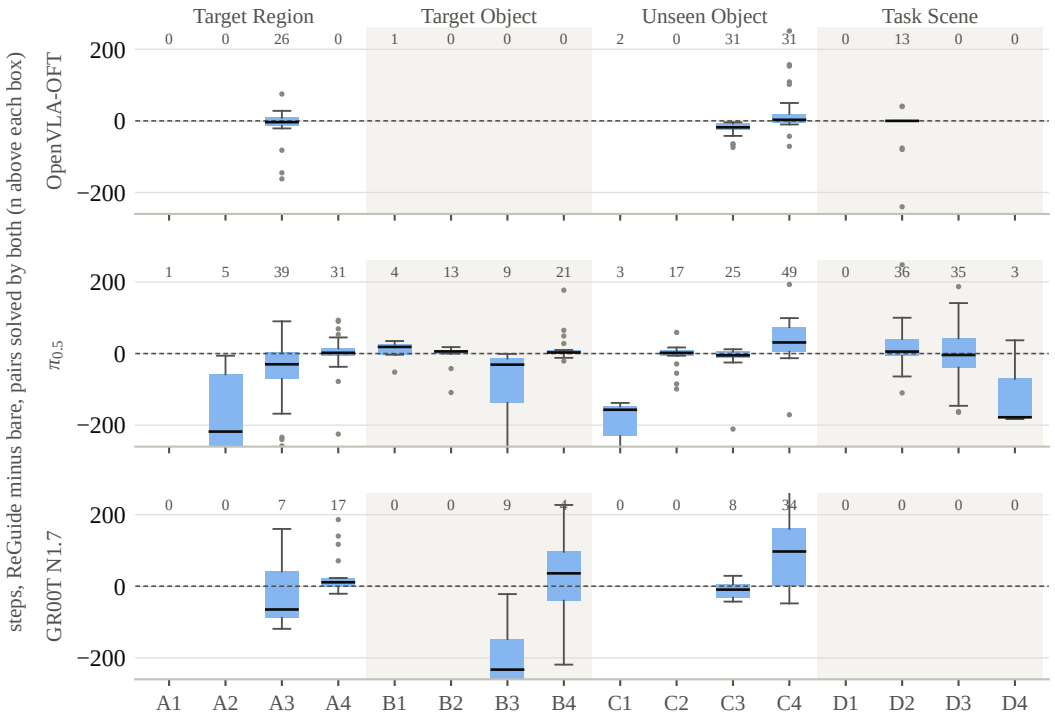

Figure 18: Change of the episode length under ReGuide per composition cell, in steps, over the pairs of trials solved by both the bare backbone and ReGuide. The number of pairs is given above each box, and cells with fewer than three pairs are left empty.

follow the grasp. On the original tasks transports are rare, and the short marks without a landing lie almost entirely on the two-clause tasks O5 and O6, 56 of 56 for $\pi_{0.5}$, 56 of 62 for GR00T N1.7 and 55 of 55 for OpenVLA-OFT.

C.4 Ablation Details

Table 13: Ablation on $\pi_{0.5}$, successes out of 50 per cell.

Variant	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4	Σ
Naive transport	35	2	0	50	7	50	50	34	16	38	48	49	39	34	46	50	548
w/o competing referents	34	49	45	45	49	49	14	42	44	48	44	47	38	43	48	48	687
w/o chunk commitment	49	41	43	45	45	49	50	23	25	47	49	42	44	44	49	46	691
w/o pre-contact sets	50	48	43	33	28	49	49	15	23	29	50	49	47	40	29	25	607
w/o bounded transport	50	31	41	46	45	41	49	40	49	50	38	43	41	44	47	46	701
ReGuide (full)	50	48	46	48	49	49	49	39	48	46	46	49	49	42	48	50	756

Table 14: Stage counts of the ablation over 800 trials. Object and Destination count episodes in which the end-effector reached the instructed object and the held object reached the instructed destination. Successful trials without a logged stage event are excluded from failure counts, so these need not equal differences between adjacent stage counts.

	Episodes reaching the stage				**Episodes lost**		
Variant	Object	Held	Destination	Success	Grasp	Carry	After arrival
Naive transport	795	708	651	548	92	57	103
w/o competing referents	796	783	737	687	17	46	50
w/o chunk commitment	789	758	726	691	42	31	36
w/o pre-contact sets	790	708	674	607	92	32	69
w/o bounded transport	787	771	759	701	29	12	58
ReGuide (full)	798	789	785	756	11	3	30

Table 15: Sensitivity to the declared constants and to two design forms, on the 800 composition trials with $\pi_{0.5}$. The last columns count trials solved only by the default or only by the variant.

Quantity	**Default**	**Variant**	$k/800$	**Success** (%)	**Default only**	**Variant only**	p
Default	–	–	756	$94.5_{\pm 1.6}$	–	–	–
n	3	1	760	$95.0_{\pm 1.5}$	27	31	0.694
n	3	5	759	$94.9_{\pm 1.5}$	25	28	0.784
ε (mm)	4	2	755	$94.4_{\pm 1.6}$	34	33	1.000
ε (mm)	4	8	752	$94.0_{\pm 1.7}$	36	32	0.716
trigger level	$Q_{.25}$	$Q_{.50}$	766	$95.8_{\pm 1.4}$	25	35	0.245
upper level	$Q_{.95}$	$Q_{.75}$	751	$93.9_{\pm 1.7}$	36	31	0.625
lower level	$Q_{.05}$	$Q_{.25}$	760	$95.0_{\pm 1.5}$	32	36	0.716
descent tolerance r_e	min of scatter and half-extent	max of the two	754	$94.2_{\pm 1.6}$	39	37	0.909
acceptance region Ω_d	range ∩ scatter	scatter only	750	$93.8_{\pm 1.7}$	38	32	0.550

Component groups. Table 13 retains the variant names used in Table 3. The first two removals concern referent monitoring in Sec. 3.2, and the last two concern configuration transfer and transport in Sec. 3.3.

Naive transport uses the same grounding information, servoing once to 0.25 m above the instructed object at the start and once to 0.25 m above the destination after the object is held, then handing back. Without *competing referents*, no habitual or watch set is compiled and objects are identified by name. Without *chunk commitment*, proximity replaces the endpoint test, and its accompanying mechanisms are removed, including suppression on the instructed object, carry gates, first-decision guidance for unseen objects, transport dissolution, failure-to-bind monitoring and completion certificates. Without *pre-contact sets*, a fixed 0.25 m target replaces the interaction configuration, with no posture target, descent tolerance or acceptance region. Without *bounded transport*, commands use clipped residuals without step bounds or tracking gains, and omit the height profile and floor. Table 14 locates the resulting losses within the interaction stages shown in Fig. 8.

Constants and design forms. Table 15 changes one declared constant or quantile level at a time, and two design forms. No variant differs significantly from the default under the paired McNemar test.

Targeted controls. Targeted controls separate local policy execution and post-hand-back coordination. Each control uses the frozen $\pi_{0.5}$ checkpoint, the same demonstration statistics, and the 800 paired composition initial states under the protocol of Appendix B. Full ReGuide results are from the main evaluation. Table 16 gives per-cell results. Action replacement is described in Appendix C.7.

Without hand-back assistance. After the first transport lands, we disable the descent corridor's lateral correction, the carry-height floor, completion-certificate recovery (rearming, regrasping, and hover-stall escort), immediate escort after landing, and escort descent and release. Failure-to-bind triggering, subsequent approach and carry transports, their targets, step laws, and budgets remain active. The switch occurs in 656/800 episodes. Success falls from 756/800 to 677/800, a paired difference of -9.9 percentage points with 95% bootstrap interval $[-12.8, -7.0]$. Full alone succeeds on 113 pairs and this variant alone on 34 ($p = 4.2 \times 10^{-11}$). Grasp success remains 97.1% versus 98.6% for full, while destination arrival falls from 98.1% to 93.9%. Most additional failures occur after grasping (Table 20).

Table 16: **Post-hand-back assistance by composition cell.** Successes out of 50. *No assistance* disables post-hand-back constraints and completion assistance.

Cell	Full	No assistance
A1	50	48
A2	48	49
A3	46	43
A4	48	48
B1	49	35
B2	49	48
B3	49	50
B4	39	20
C1	48	45
C2	46	43
C3	46	49
C4	49	36
D1	49	36
D2	42	46
D3	48	44
D4	50	37
Total	756	677

C.5 Guidance Weight of CAG

Table 17: CAG on $\pi_{0.5}$ under two guidance weights (success rate, %).

Method	Orig.	Target Region	Target Object	Unseen Object	Task Scene	Comp. Avg.
Bare $\pi_{0.5}$	94.0	40.5	25.0	49.5	41.0	39.0
CAG, $\omega = 1.5$	81.4	53.0	35.0	44.0	33.5	41.4
CAG, $\omega = 2.0$	64.6	59.0	27.0	38.0	20.0	36.0

Table 17 compares the guidance weight 1.5 used in Table 1 with 2.0, the example value in the authors' repository.

C.6 Perception-Based Grounding

Setup. We replace oracle grounding with an RGB-D perception module for ReGuide on $\pi_{0.5}$, keeping referent monitoring, geometric execution, demonstration statistics, and the execution budget unchanged. Inputs are the instruction, a 512-pixel agentview RGB-D image, and declared category

geometry from the object assets, analogous to the measured dimensions used on the real robot. GroundingDINO tiny uses box and text thresholds of 0.25 with the declared scene category vocabulary. Detected point-cloud segments are assigned to declared instances by maximum-weight one-to-one matching, with geometric gates on resting top height and footprint dimensions. Spatial phrases in the instruction distinguish instances of the same category. Depth back-projection provides positions, using the center of the declared top-surface slice for objects and the point-cloud bounding-box center for fixtures.

Registration occurs once when the instruction is issued, without subsequent visual tracking. The grounding module reads neither simulator entity identities nor ground-truth poses. Episodes in which the instructed object or destination cannot be registered count as failures, with no oracle fallback.

Table 18: **ReGuide with oracle and perception-based grounding on $\pi_{0.5}$ (compositional success rate, %).** Each composition axis contains 200 trials. Subscripts are 95% Wilson confidence half-widths. Registration failures are included.

Grounding	Target Region	Target Object	Unseen Object	Task Scene	Comp. Avg.
Oracle	$96.0_{\pm 2.8}$	$93.0_{\pm 3.6}$	$94.5_{\pm 3.2}$	$94.5_{\pm 3.2}$	$94.5_{\pm 1.6}$
Perception	$86.0_{\pm 4.8}$	$86.5_{\pm 4.7}$	$91.5_{\pm 3.9}$	$86.0_{\pm 4.8}$	$87.5_{\pm 2.3}$

Results. Compositional success remains 87.5% (700/800), compared with 94.5% (756/800) under oracle grounding, retaining a 48.5 percentage-point gain over the bare policy. The paired difference to oracle is -7.0 percentage points, with a 95% paired-bootstrap interval of $[-9.8, -4.4]$. Perception alone succeeds on 32 pairs and oracle alone on 88.

Error analysis. Identity confusion affects referent matching, while fragmented fixture point clouds and stale estimates after object motion affect the transferred configurations.

C.7 Hand-Back Action Replacement

Protocol. We test local policy execution while retaining ReGuide's referent monitoring, configuration transfer and execution support. From the first transport landing until episode termination, the action selected from the policy chunk is replaced before applying the wrapper's constraints. The VLA is still queried on the true observation at the standard cadence, and its predicted chunks continue to feed the commitment detector. Subsequent transports, motion envelopes and completion certificates remain active. Episodes without a hand-back follow the original execution path. Each condition uses the same 16 composition cells and 50 initial states per cell as full ReGuide.

Replacement actions. The *default action* holds the hand-back horizontal position and orientation, descends with the demonstrated step size to the demonstrated median grasp height, then closes and holds the gripper. Following a post-grasp hand-back, it instead descends to the demonstrated release height while keeping the gripper closed. In the 44 episodes without an available demonstration target, it closes and holds at the landing height. The *hold* condition sets translation and rotation increments to zero and retains the gripper command at the switch. Both replacements remain subject to the unchanged wrapper. Replacement covers 73.7% and 77.6% of execution steps for default and hold, respectively.

Results. Full ReGuide solves 756/800 trials (94.5%, 95% Wilson interval $[92.7, 95.9]$), compared with 305/800 for default (38.1%, $[34.8, 41.5]$) and 224/800 for hold (28.0%, $[25.0, 31.2]$). Full exceeds default by 56.4 percentage points with paired-bootstrap interval $[52.6, 60.1]$ (470 full-only and 19 default-only successes, $p = 9.4 \times 10^{-114}$), and hold by 66.5 points with interval $[63.0, 70.0]$ (547 full-only and 15 hold-only successes, $p = 1.5 \times 10^{-140}$). Grasp success is 98.6%, 44.0%, and 33.5%, and destination arrival is 98.1%, 42.1%, and 32.2%, respectively. Table 21 gives the per-cell counts.

Role of local execution. The largest loss occurs when replacement begins before grasping (Table 19). Default and hold incur 440 and 521 post-hand-back grasp failures, versus 6 for full (Table 20). Replacement after grasping is less disruptive, consistent with continued transport and release assistance during placement. Completion escort performs the final release in 86 full episodes (83 successful), 128 default episodes (119 successful), and 63 hold episodes (all successful). Together with the

Table 19: **Outcomes by first hand-back stage.** Successes/trials with percentages in parentheses. Groups are defined within each rollout, so their membership can differ across conditions.

Action	No hand-back	Before grasp	After grasp
VLA (full)	120/139 (86.3)	510/531 (96.0)	126/130 (96.9)
Default	126/147 (85.7)	52/513 (10.1)	127/140 (90.7)
Hold	121/141 (85.8)	0/529 (0.0)	103/130 (79.2)

initialization diagnostic, these controls link access to an interaction configuration with the value of executing the local policy from it.

Table 20: **Failure stages after hand-back.** Counts over 800 trials per condition. Failure categories after *No hand-back* apply only to episodes with a hand-back. Each row sums to 800. *No assistance* is the targeted control in Appendix C.4.

Condition	Success	No hand-back failure	No grasp	No destination arrival	No release	Failure after release
Full	756	19	6	1	1	17
Default	305	21	440	11	2	21
Hold	224	20	521	7	5	23
No assistance	677	17	16	23	17	50

Table 21: **Hand-back action replacement by cell.** Successes out of 50.

Cell	Full	Default	Hold	Cell	Full	Default	Hold
A1	50	50	49	C1	48	0	0
A2	48	2	1	C2	46	1	0
A3	46	9	9	C3	46	0	0
A4	48	47	42	C4	49	0	0
B1	49	0	0	D1	49	34	0
B2	49	15	13	D2	42	42	37
B3	49	37	30	D3	48	13	6
B4	39	39	34	D4	50	16	3

C.8 Agent-Based Guidance and Runtime

Harness VLA protocol. We evaluate the official Harness VLA/RPent implementation (Zhang et al., 2026b), using its released guides and memory (`RLinf/RPent-memory`) and the documented Codex planner with GPT-5.5 and at most 100 rounds. Its tools include VLA contact skills, waypoint motion, gripper commands, SAM3 segmentation, and depth back-projection. We use the same `pi05_libero` checkpoint, image preprocessing, 8-dimensional state, and 5 executed actions per query as in the main evaluation. We retain the 10 settling steps, 520-step execution budget, 20 Hz control, and per-step LIBERO success predicate.

To bound API cost, we run the first paired initial state (seed 0) in each of the 26 cells and compare against the matching bare-policy and ReGuide trials from the main evaluation. Harness uses its SAM3 and depth pipeline, while these ReGuide rollouts use oracle grounding. The released Harness memory remains unchanged throughout evaluation. It contains LIBERO-PRO seed-0 recipes and reference trajectories and is consulted in 16/26 episodes. In 2 original tasks, soup-to-basket and milk-to-basket, the retrieved reference has the same object and destination.

Outcomes and execution. ReGuide alone succeeds on 6 pairs and Harness alone on 1 ($p = 0.125$). Harness alone succeeds on 7 pairs and the bare policy alone on 3 ($p = 0.34$). With one rollout per cell, these are exploratory paired outcomes rather than precise success-rate estimates. In the 17 jointly successful episodes, median steps to success are 279 for Harness and 141 for ReGuide.

Table 22: **Exploratory comparison with Harness VLA.** Successes/trials on one paired initial state per cell. A–D denote the four composition axes. These counts are separate from the 50-trial-per-cell evaluation.

Method	A	B	C	D	Comp.	Orig.	All
Bare $\pi_{0.5}$	1/4	0/4	2/4	1/4	4/16	10/10	14/26
Harness VLA	3/4	1/4	2/4	4/4	10/16	8/10	18/26
ReGuide	4/4	4/4	3/4	4/4	15/16	8/10	23/26

Planner waiting time and cost. Harness has a median episode wall time of 184 s, including 156 s spent waiting for the planner. A planner response takes a median of 6.1 s, a 90th percentile of 12.5 s, and a maximum of 37.4 s, with a median of 20 waits per episode. API cost averages $1.84 per episode.

ReGuide timing protocol. We measure the runtime of ReGuide with oracle grounding and the bare policy on the same 26 seed-0 initial states. Both use one dedicated RTX A6000 policy server and a Xeon w5-3435X CPU, with no other jobs or video recording. Replanning latency includes the policy query, websocket communication, and client computation, excluding simulator physics and rendering. Timing is collected in separate runs from the success comparison in Table 22.

Table 23: **Runtime of oracle-grounding ReGuide and the bare policy.** Timings in milliseconds after the first decision. Each entry reports median / 90th percentile unless otherwise stated.

Measurement	Bare $\pi_{0.5}$	ReGuide
Policy query	127.2 / 128.1	126.7 / 127.7
Replanning latency	129.0 / 129.9	129.2 / 130.3
Client computation per step	1.60 / 1.81	2.14 / 2.61
Client computation (99th percentile)	2.04	3.49
Simulator step (median)	10.3	10.1

Runtime results. Table 23 reports steady-state latency. Initialization, including retrieval-encoder loading, takes a median of 2.2 s versus 28 ms for the bare policy. These timings use oracle grounding and exclude GroundingDINO. Planner waits and policy replanning are measured per event at their respective frequencies.

D Real-World Details

D.1 Hardware and Control

The robot is a UFACTORY xArm6 with the xArm gripper. Two Intel RealSense D435i cameras provide aligned RGB-D streams at 640×480 and 30 fps, one fixed at the base and one on the wrist, and the policy receives 480×480 crops of both. Control runs at 10 Hz with absolute joint targets streamed to the controller. A workstation with two RTX 4090 GPUs runs the policy on one GPU and the detector on the other. A safety layer below the wrapper limits the joint step to 0.30 rad per tick and the joint speed to 2.0 rad/s, keeps the joints 2° inside the limits of the controller, and enforces a floor of the tool above the table.

D.2 Policy

$\pi_{0.5}$ is fully fine-tuned from the base checkpoint with AdamW, gradient clipping at 1.0, a learning rate warmed up over 1600 steps to 5×10^{-5} and then kept constant, an exponential moving average of 0.999, batch size 128 and 16k steps on one H200, which took about 21.5 hours. The final checkpoint is deployed. The observation consists of the two image crops and a 7-dimensional state of six joint angles and the gripper, and the action is a chunk of 10 absolute joint and gripper targets, of which 8 are executed per replan while the next chunk is requested two ticks ahead. Median inference time is 94 ms with ReGuide and 92 ms without. Observations are two ticks old when chunk execution begins.

D.3 DEMONSTRATIONS

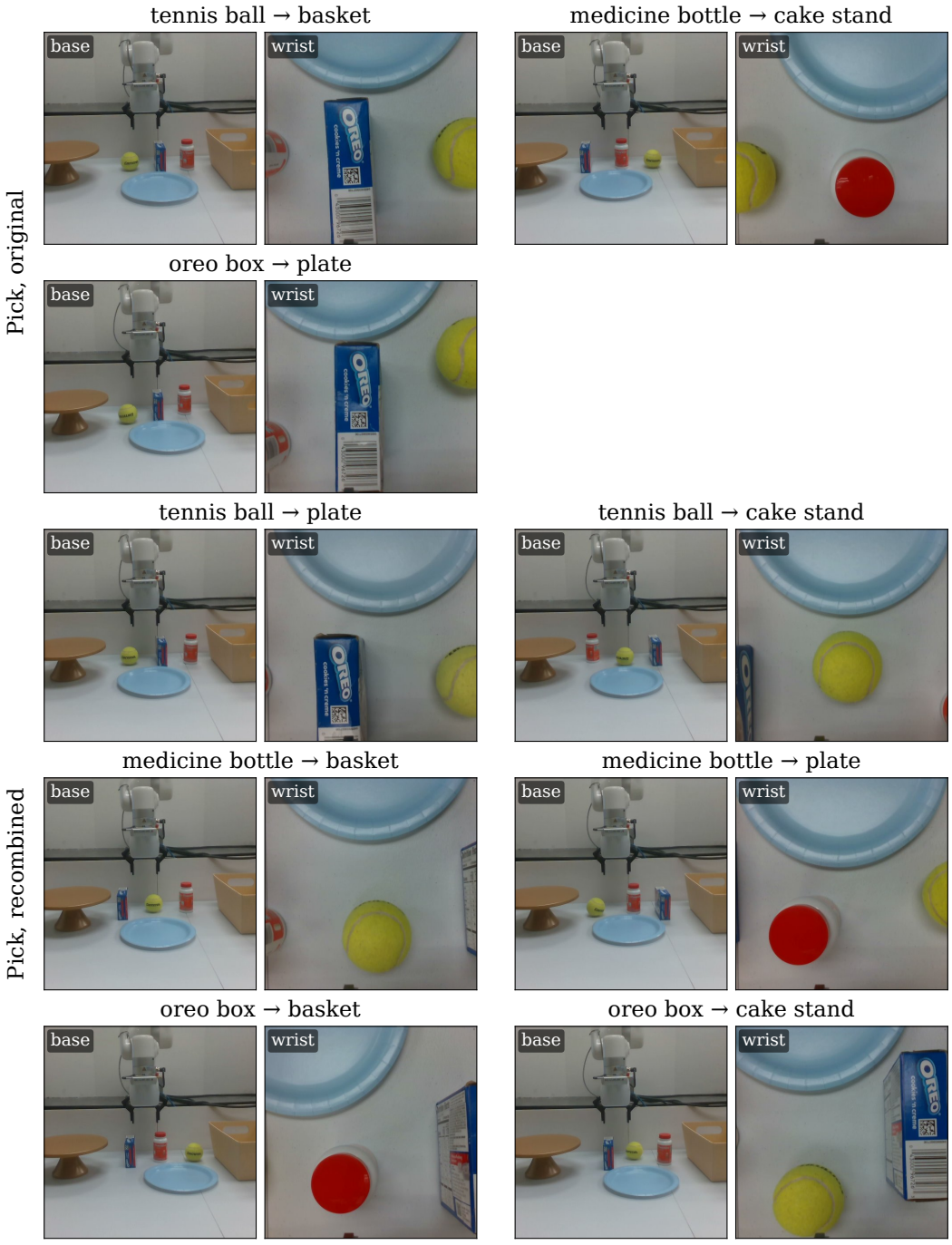

Figure 19: One evaluation layout per pick-and-place cell with trained objects, as seen by the base camera (left) and the wrist camera (right) at the start of the episode. Rows group the original and the recombined pairings.

A scripted demonstrator records 50 demonstrations for each of five tasks, 250 episodes and 59,167 frames at 10 Hz. The pick-and-place tasks are tennis ball to basket, medicine bottle to cake stand and oreo box to plate, executed as hover, descend, close, lift, carry and place. The push tasks are tennis ball toward the cake stand and medicine bottle toward the basket, executed with a half-open gripper

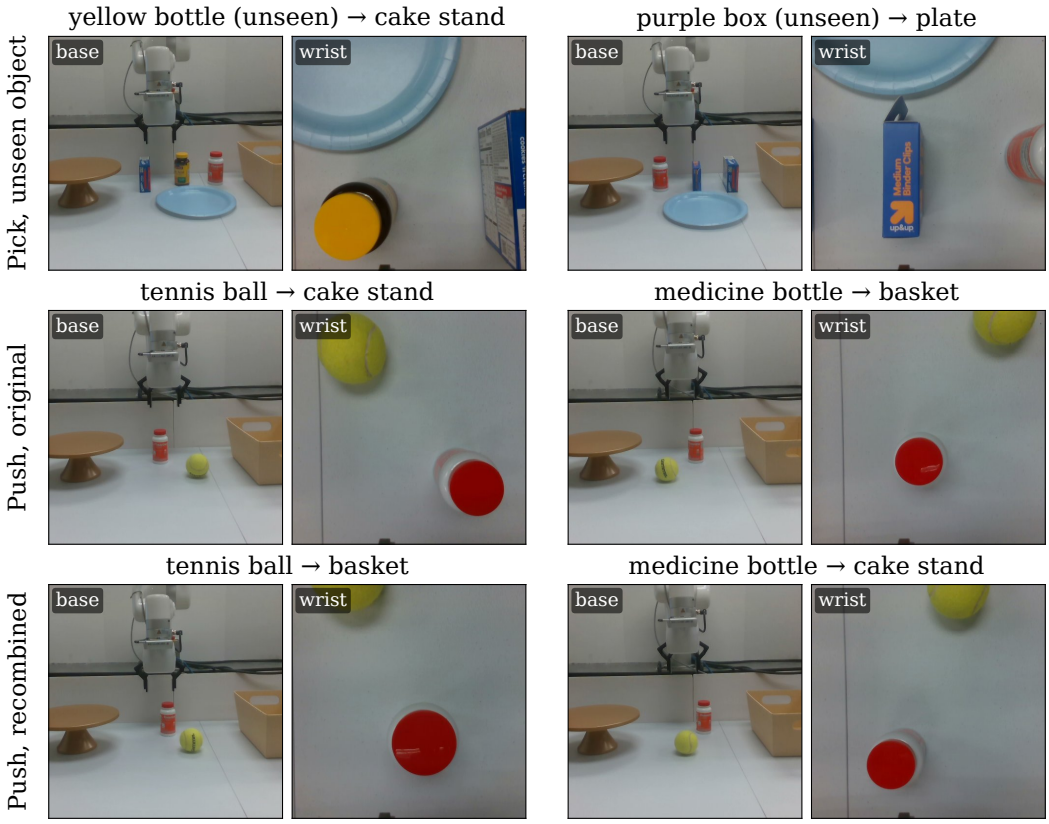

Figure 20: One evaluation layout per unseen-object and push cell, same views as Fig. 19. In the unseen-object cells one training object is replaced by the unseen one.

that descends behind the object and pushes it by 5 cm along the line from the object to the destination. Events and poses are exported at 10 Hz from the recorded observations and are the only input of the statistics of ReGuide, which are mined with the same code as in simulation. One demonstration of each of the five tasks is included in the supplementary videos.

D.4 Evaluation Protocol

The 15 cells of Table 24 are evaluated under 10 layouts each with both methods, which gives 300 episodes. Both methods run the same client, and the bare policy is obtained by switching the wrapper off. Both methods are evaluated on the same paired initial layouts in each cell. In a cell with an unseen object, one randomly chosen training object is replaced by the unseen one, so three objects are on the table. An episode ends when the operator stops it and gives the verdict, at the step budget, or at a stop of the safety layer, and the last two count as failures. The budget is 600 policy steps after the 10 settle ticks in the pick-and-place cells (610 ticks in total) and 600 ticks including the settle ticks in the push cells (590 policy steps); tick counts reported for the robot include the settle ticks. Figs. 19 and 20 show one layout per cell from both cameras.

D.5 Grounding Module

When an instruction is issued, GroundingDINO detects every known object and destination in the base image from a caption that contains their phrases. For each entity the most confident box is kept whose measured top height agrees with the declared height of the entity, and a box is owned by one entity only. During motion, the wrist image refines the position of an entity whenever it is in view, with the image paired to the tool pose at its capture time, and a fix is accepted only within 0.10 m of the base estimate for objects and 0.25 m for destinations. When a transport arrives, the

target is measured once more and corrected by at most 60 mm. The dimensions of the objects and destinations are declared from measurements, and those of the two unseen objects are measured before the evaluation. The 90th percentile of the horizontal error of the refined estimate over 14 recorded episodes is 2.6 cm, and this value floors every horizontal tolerance that the demonstrations would set lower. There is no survey motion before an episode and no additional physical action for perception.

D.6 PUSHING

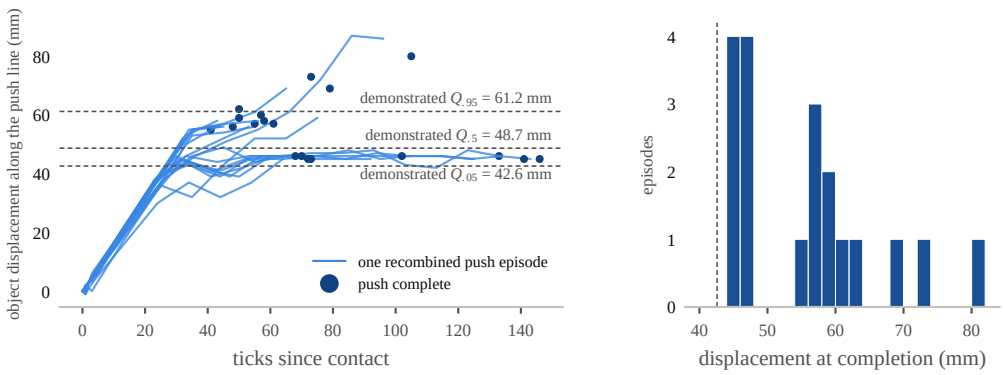

Figure 21: Displacement of the object along the push line after contact in the 20 recombined push episodes (left) and its value when the push is declared complete (right). The dashed lines are the demonstrated quantiles. In 19 episodes the completion line is reached, and one episode is stopped by the operator at 45 mm with the object already past the lower quantile.

For pushing, the instructed object–destination line defines the interaction reference, with longitudinal, lateral and vertical axes. From the 100 push demonstrations, the pre-contact configuration lies 38.2 mm behind the object along the line, 0.3 mm lateral and 29.3 mm below the object centre, with a tolerance of 10.6 mm, a trigger radius of 2.4 mm, a hover altitude of 279.7 mm and a posture tolerance of 0.024 rad. The demonstrated displacement has $Q_{.05}$, $Q_{.5}$ and $Q_{.95}$ of 42.6, 48.7 and 61.2 mm, and a push is complete when the displacement of the object has exceeded the lower value for n ticks. A commitment toward the habitual destination indicates a direction mismatch. After approach transport establishes the pre-contact configuration, carry-side transport advances along the instructed line by the median demonstrated displacement under the step bound. On original cells no transport is issued. During a push the wrapper keeps the tool on the demonstrated line and height. Fig. 21 shows the displacement of the object after contact in the 20 recombined push episodes.

D.7 RESULTS

Table 24 gives the results per cell with the paired outcomes per block, and Fig. 24 the share of trials solved within a given time. On the robot, guidance also supports execution on original tasks. The approach transport starts at a median of tick 67 on trained objects and at tick 10 on unseen objects, where first-decision guidance establishes the transferred configuration. On the original push cells the wrapper issues no transport. Fig. 22 shows the paired outcome of every layout, and Fig. 23 the corrections of the transport targets. The supplementary videos show one paired layout per cell with the two runs of the same layout, the bare policy above and ReGuide below.

D.8 CORRESPONDENCE BETWEEN SIMULATION AND ROBOT

The robot instantiates referent monitoring and geometric guidance through its sensing and control interfaces. Entity poses come from the detector rather than simulator state, with measured perception error flooring the tolerances. Robot joint targets are converted by forward kinematics with unit gains, whereas simulation uses end-effector increments with fitted gains. Control runs at 10 rather than 20 Hz, with event windows matched in seconds. Held and contact predicates use the gripper and tool

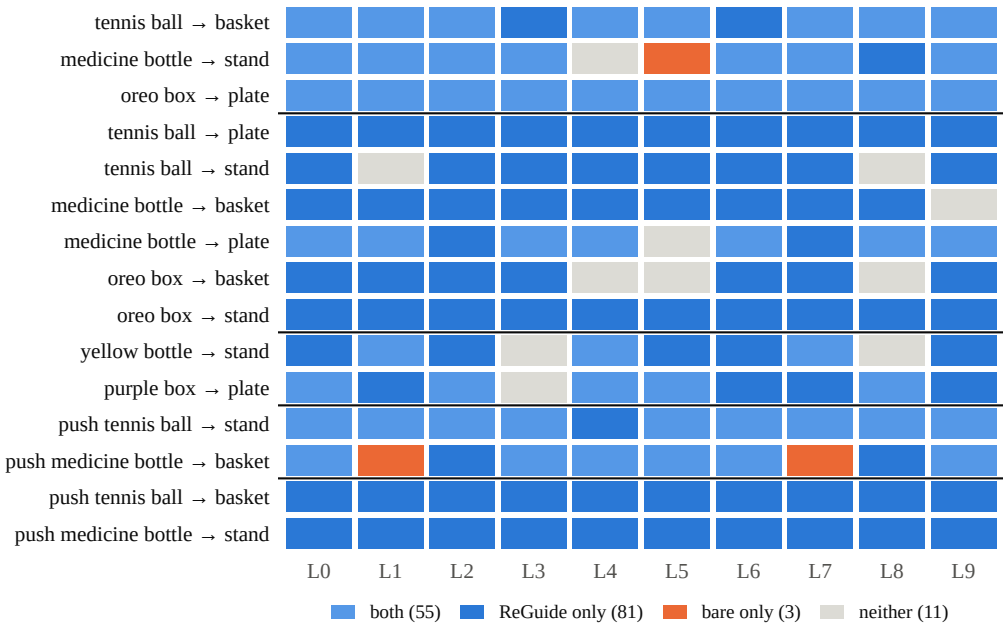

Figure 22: Outcome of the 150 paired layouts of the real-robot study. Each square is one layout of one cell, coloured by whether both methods, only ReGuide, only the bare policy or neither succeeded.

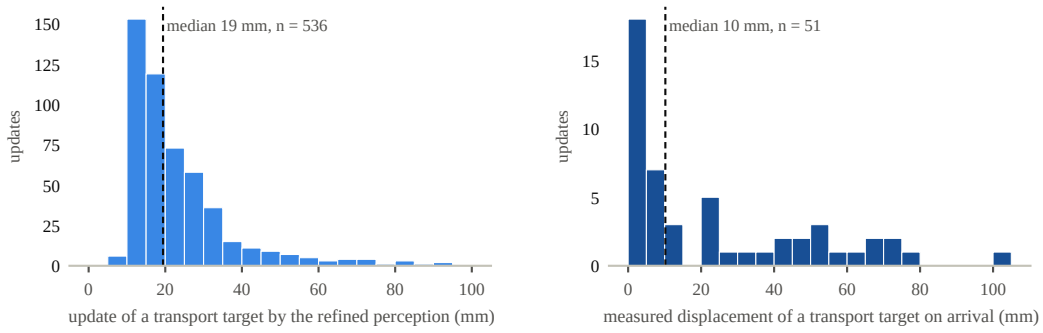

Figure 23: Left, the update of a transport target by the refined perception, in motion and on arrival, 536 updates over the 110 pick-and-place episodes of ReGuide with a median of 19 mm. Right, the displacement of the target measured by the arrival refinement, 51 measurements.

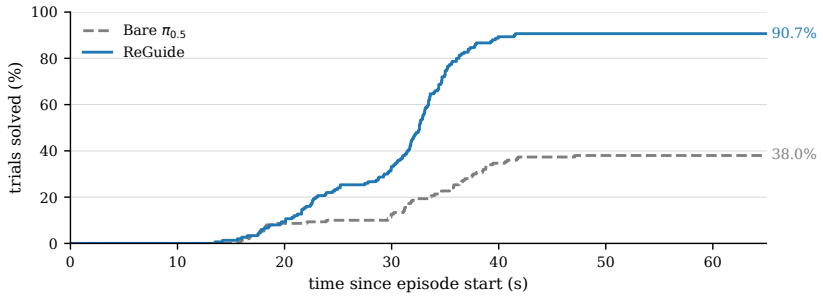

Figure 24: Share of real-robot trials solved within a given time. One successful bare trial on an original cell has no recorded duration and is not drawn.

Table 24: Real-robot successes out of 10 layouts per cell, and paired outcomes per block with the exact McNemar test.

Block	**Object and destination**	**Bare $\pi_{0.5}$**	**ReGuide**
Pick, original	tennis ball to basket	8/10	10/10
	medicine bottle to stand	8/10	8/10
	oreo box to plate	10/10	10/10
paired layouts, both / bare only / ReGuide only / neither		25 / 1 / 3 / 1, $p = 0.625$	
Pick, recombined	tennis ball to plate	0/10	10/10
	tennis ball to stand	0/10	8/10
	medicine bottle to basket	0/10	9/10
	medicine bottle to plate	7/10	9/10
	oreo box to basket	0/10	7/10
	oreo box to stand	0/10	10/10
paired layouts, both / bare only / ReGuide only / neither		7 / 0 / 46 / 7, $p = <0.001$	
Pick, unseen object	yellow bottle to stand	3/10	8/10
	purple box to plate	5/10	9/10
paired layouts, both / bare only / ReGuide only / neither		8 / 0 / 9 / 3, $p = 0.004$	
Push, original	tennis ball toward stand	9/10	10/10
	medicine bottle toward basket	8/10	8/10
paired layouts, both / bare only / ReGuide only / neither		15 / 2 / 3 / 0, $p = 1.000$	
Push, recombined	tennis ball toward basket	0/10	10/10
	medicine bottle toward stand	0/10	10/10
paired layouts, both / bare only / ReGuide only / neither		0 / 0 / 20 / 0, $p = <0.001$	

position on the robot. Success is judged by the operator on the robot and by the goal predicate in simulation.

E LIMITATIONS

ReGuide relies on the frozen policy's local interaction skills after hand-back, so failures in these skills remain a source of error. The failures that remain in Table 12 concentrate on such cells, for example the moka pot of B1, which GR00T N1.7 and OpenVLA-OFT fail in 48 of 50 trials while $\pi_{0.5}$ fails in one. ReGuide also relies on an external grounding module for the identity and the pose of the instructed referents. The main simulation comparison uses oracle grounding to isolate the wrapper from perception errors. Perception-based simulation (Appendix C.6) and real-robot experiments additionally evaluate these errors. The demonstration statistics require the training demonstrations with end-effector and object poses, which not every released policy provides. Our evaluation covers pick-and-place and push tasks in which an instruction resolves to an object and a destination, and every composition cell changes one factor. Articulated or deformable manipulation, compositions that change several factors at once, and relational or ambiguous instructions are not evaluated. An unseen object inherits the statistics of the trained object with the closest shape, which does not cover objects that need a grasp absent from the demonstrations. Finally, the real-robot study uses one arm, one backbone and 150 paired trials.