LANCE: LEARNING WHAT TO SEE IN VISION-LANGUAGE-ACTION POLICIES VIA LANGUAGE-ANCHORED ACTION QUERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) policies have advanced robotic manipulation, yet they still struggle with reliable instruction following. These policies tend to exploit vision-action correlations instead of consistently using language to identify task-relevant visual evidence for action prediction. To address this issue, we propose **LANCE** (**Language-Anchored Action Queries**), an action-query framework that explicitly couples scene-conditioned action representations to task semantics derived from language instructions. LANCE employs two sets of learnable action queries: one reads only the language tokens to construct a *language anchor*, while the other reads both language and visual tokens to form a scene-conditioned action representation. To explicitly couple the scene-conditioned representation with the language anchor, *Reciprocal Alignment* aligns their latent distributions. Crucially, alignment alone does not ensure that the language anchor captures the action distinctions between instructions. We therefore devise *PriorAction*, an auxiliary action prediction loss conditioned on the language anchor. Jointly, the two objectives steer visual processing to prioritize task-relevant evidence for instruction-consistent action prediction. Experiments across diverse unseen settings show that LANCE improves instruction-consistent target selection. Moreover, under zero-shot composition of learned skills, LANCE achieves higher semantic coverage and subtask completion rates.

1 INTRODUCTION

Vision-language-action (VLA) policies (Brohan et al., 2023b;a; Kim et al., 2024; Black et al., 2024; Bjorck et al., 2025) have rapidly advanced language-conditioned robotic manipulation by directly mapping natural-language instructions and visual observations to continuous, closed-loop robot actions. Despite this progress, reliable instruction following remains challenging (Kim et al., 2026; Fei et al., 2025; Lian et al., 2026) under unseen instructions and over compositional tasks.

Recent studies (Jia et al., 2026; Zha et al., 2026) suggest that reliable instruction following depends not only on fitting demonstrated trajectories, but also on identifying task-relevant information for action prediction. To examine whether current VLA policies achieve this, we analyze how language instructions influence visual processing during action generation. As illustrated in Fig. 1(a), the action pathways in several VLA policies (Bjorck et al., 2025; Intelligence et al., 2025; Lian et al., 2026; Xu et al., 2026) exhibit diffuse visual attention that extends to instruction-irrelevant regions. These observations suggest that existing VLA policies may not effectively use language to identify and prioritize task-relevant visual evidence, motivating a central question: *How can language actively guide the selection of task-relevant visual evidence for action generation?*

To answer this question, we propose **LANCE** (**Language-Anchored Action Queries**), an action-query framework that explicitly couples scene-conditioned action representations to task semantics derived from language instructions (Fig. 1(b)). Specifically, LANCE introduces two sets of learnable action queries within a shared vision-language backbone. The first set attends to only language tokens and extracts language-only features, while the other attends to both language and visual tokens and aggregates multimodal context. These features parameterize a scene-independent distribution $p(u \mid \ell)$ and a scene-conditioned distribution $q(u \mid o_t, \ell)$ over a latent action-conditioning space. We use $p(u \mid \ell)$ as a *language anchor* to represent task-level action semantics that should be shared

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.

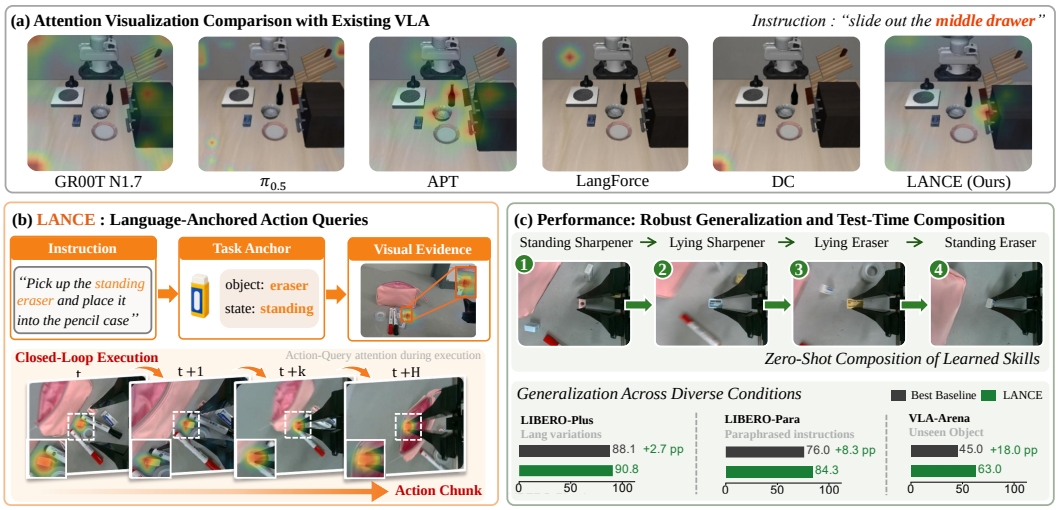


Figure 1: (a) Comparison of visual attention across VLA policies under the same observation and instruction (b) LANCE: Language-Anchored Action Queries. (c) Simulation benchmark results and real-world test-time skill composition.

between the language-only and multimodal action representations induced by the given instruction. The scene-conditioned distribution $q(u \mid o_t, \ell)$ grounds these semantics using the evidence available in the current scene. To explicitly couple the two representations, we introduce *Reciprocal Alignment*, which bidirectionally aligns the scene-conditioned distribution with the language anchor through KL regularization. Thus, language anchoring regularizes the action representations so that the visual evidence they aggregates remains consistent with the task-level semantics.

Crucially, alignment alone does not ensure that the language anchor captures action-relevant distinctions between instructions. We therefore devise *PriorAction*, an auxiliary loss that supervises action prediction conditioned on samples from the language anchor. The auxiliary action loss shapes an action-relevant language anchor, while Reciprocal Alignment couples the scene-conditioned distribution with this language anchor. Jointly, the two objectives encourage the scene-conditioned representation to remain consistent with the task-level semantics while incorporating visual evidence required for execution, thereby supporting instruction-consistent action generation.

We evaluate LANCE under diverse instruction and scene variations in both simulation and real-robot settings (Fig. 1(c)). Under language variation and paraphrasing, LANCE achieves 90.8% success on LIBERO-Plus Language and 84.3% on LIBERO-PARA. On the VLA-Arena UnseenObjects subset, LANCE achieves 63% average success. Real-robot experiments further show more reliable instruction-consistent target selection across most evaluated settings, including cross-condition generalization, visual distractors, and unseen objects, containers, and instructions. Moreover, under zero-shot composition of learned skills, LANCE achieves higher semantic coverage and completes more requested subtasks than the baseline when following previously unseen composite instructions. Our contributions are threefold:

- We find that existing VLA policies do not consistently use language to prioritize task-relevant visual evidence during action generation.
- We propose LANCE, an action-query framework that constructs a language anchor for visual evidence selection. PriorAction shapes this anchor with distinctive action semantics, while Reciprocal Alignment uses it to regularize the action representations .
- We demonstrate improved language generalization and stronger instruction grounding across diverse cross-condition, visual distractor, and zero-shot skill-composition settings.

2 Related Work

Vision-Language-Action Models. Vision-language-action (VLA) policies integrate language-conditioned perception with low-level robot control. Early systems such as RT-1 (Brohan et al., 2023b) learned task-conditioned transformer controllers from large multi-task robot datasets,

while RT-X (Collaboration et al., 2023) and Octo (Octo Model Team et al., 2024) scaled robot-policy pretraining across datasets and embodiments. RT-2 (Brohan et al., 2023a) and OpenVLA (Kim et al., 2024) instead adapted pretrained vision-language models (VLMs) to control by representing robot actions as discrete tokens, transferring broad visual-semantic knowledge to action prediction. Since naive tokenization and autoregressive decoding can limit precision and throughput in high-frequency control, recent approaches decode continuous action chunks through regression, diffusion, or flow matching, including OpenVLA-OFT, CogACT, RDT-1B, π_0, and GR00T-N1 (Kim et al., 2025; Li et al., 2024; Liu et al., 2024; Black et al., 2024; Bjorck et al., 2025). While these advances (Zheng et al., 2025; Zha et al., 2026; Yang et al., 2026a; Qu et al., 2025; Shen et al., 2025; Ding et al., 2026) improve motor skills and adaptability, they still do not guarantee that the policy actually relies on the language instruction.

Instruction Generalization in VLA Policies. Recent studies (Fei et al., 2025; Kim et al., 2026; Zhang et al., 2025) show that VLA policies may exploit visual shortcuts and degrade under paraphrases, counterfactual task changes, unseen objects, or compositional instructions. Data-centric methods co-train robot trajectories with Internet-scale vision-language data, as in RT-2 (Brohan et al., 2023a), $\pi_{0.5}$ (Intelligence et al., 2025), and InstructVLA (Yang et al., 2025); Knowledge Insulation (Driess et al., 2025) and TwinBrainVLA (Yu et al., 2026) complement this direction by protecting pretrained semantic representations from action optimization. Action pretrain methods construct stronger priors before full VLA learning: BayesVLA (Xu et al., 2025) and APT (Xu et al., 2026) learn a vision-action prior before language alignment, and Qwen-VLA (Wang et al., 2026) and LA4VLA (Lin et al., 2026) begin with text-action pretraining before full VLA training. CogVLA (Li et al., 2026), Otter (Huang et al., 2025) and GuidedVLA (Jia et al., 2026) explicitly guide visual routing or specialize action attention to capture task-relevant information. LangForce (Lian et al., 2026) contrasts a vision-only prior with a vision-language posterior to maximize conditional action-instruction mutual information. In contrast to these approaches, LANCE learns a scene-independent language anchor through auxiliary action supervision. It jointly aligns the vision-language representation with this action-relevant anchor to guide visual evidence selection.

3 METHOD

3.1 PRELIMINARIES

Vision-language-action (VLA) models learn to map visual observations and language instructions directly to robot actions. Given a language instruction ℓ, multi-view observations o_t, and the current proprioceptive state s_t, a policy predicts an H-step action chunk $\hat{a}_{t:t+H-1}$:

$$\hat{a}_{t:t+H-1} = \pi_\theta(c_t, s_t), \qquad \hat{a}_{t:t+H-1} \in \mathbb{R}^{H \times d_a}, \tag{1}$$

where c_t denotes the vision-language context and d_a is the action space dimension. In VLA policies, c_t is typically extracted by a VLM backbone F_Θ. Following the design of modern VLMs, modality-specific tokenizers first map the ℓ and o_t into tokens $\mathbf{L} \in \mathbb{R}^{N_l \times d}$ and $\mathbf{V}_t \in \mathbb{R}^{N_v \times d}$, where N_l and N_v denote the respective token counts and d is the hidden dimension. The transformer backbone F_Θ then processes these tokens to extract contextualized vision-language representations. These representations form the conditioning context c_t, either directly or through aggregation via learnable action queries. Conditioned on c_t and s_t, the action expert predicts the action chunk in Eq. 1. The policy is optimized via behavior cloning using an action objective, denoted by ℓ_{act}.

3.2 LANCE: LANGUAGE-ANCHORED ACTION QUERIES

Overview. As illustrated in Fig. 2, LANCE augments a shared VLM backbone with two context-differentiated action-query banks. Language-only queries, restricted to language tokens, produce features z_l, while vision-language queries produce scene-conditioned features z_{vl} from both language and visual tokens. The two representations parameterize a scene-independent language-anchor distribution $p(u \mid \ell)$ and a scene-conditioned distribution $q(u \mid o_t, \ell)$, respectively. LANCE couples the two representations through *Reciprocal Alignment*, which bidirectionally aligns the scene-conditioned distribution with the language anchor, thereby regularizing the scene-conditioned representation and encouraging the vision-language queries to aggregate visual evidence consistent with the task-level semantics. LANCE further introduces a complementary training objective, *PriorAction*, to shape the language anchor so that it captures distinct task semantics across different

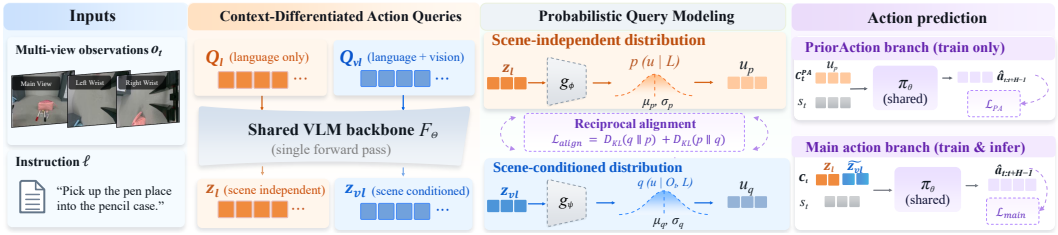


Figure 2: Overview of LANCE. Language-only and vision-language action queries parameterize scene-independent and scene-conditioned latent distributions, respectively. *Reciprocal Alignment* couples the two distributions, while *PriorAction* shapes the language anchor with action-relevant semantics through the shared action expert.

instructions. For action prediction, a latent sampled from the scene-conditioned distribution q is residually fused with z_{vl} to preserve scene-specific information and then concatenated with z_l to form the action condition. The shared action expert predicts the action chunk from this condition and the robot's proprioceptive state.

Context-Differentiated Action Queries. To derive language-only and scene-conditioned action-conditioning features, we differentiate action queries by the context they can access. Specifically, we introduce two learnable query banks, Q_l and Q_{vl}, within the shared backbone F_Θ, and jointly compute their features in a single forward pass under a structured attention mask M:

$$(z_l, z_{vl}) = F_\Theta\left(Q_l, Q_{vl}; \mathbf{L}, \mathbf{V}_t, M\right). \tag{2}$$

Here, M restricts Q_l to the instruction tokens, while allowing Q_{vl} to access both instruction and visual tokens. The resulting language-only features $z_l \in \mathbb{R}^{K\times d}$ encode instruction information, while the vision-language features $z_{vl} \in \mathbb{R}^{K\times d}$ aggregate the current multimodal context, where K is the number of queries in each bank.

Probabilistic Query Modeling. To represent task-level action semantics and their scene-conditioned grounding in a latent action-conditioning space, we use z_l and z_{vl} to parameterize two latent distributions. Two lightweight heads map each query feature to a mean and a log-variance:

$$[\mu_p, \rho_p] = g_\phi(z_l), \qquad [\mu_q, \rho_q] = g_\psi(z_{vl}), \tag{3}$$

where $\mu_p, \rho_p, \mu_q, \rho_q \in \mathbb{R}^{K\times d}$. Each head projects the feature dimension from d to a hidden dimension d_b and then to $2d$. We split the output along the feature dimension into the mean and log-variance, with $\rho_p = \log\sigma_p^2$ and $\rho_q = \log\sigma_q^2$.

These parameters define two diagonal Gaussian distributions over a latent tensor $u \in \mathbb{R}^{K\times d}$:

$$p(u \mid \ell) = \mathcal{N}\left(\mu_p, \mathrm{diag}(\exp(\rho_p))\right), \tag{4}$$

$$q(u \mid o_t, \ell) = \mathcal{N}\left(\mu_q, \mathrm{diag}(\exp(\rho_q))\right). \tag{5}$$

We refer to p and q as the *scene-independent* and *scene-conditioned* distributions, respectively. Conditioned only on the instruction, p serves as a learnable language anchor for q. During training, we sample $u_q \sim q$ through reparameterization:

$$u_q = \mu_q + \exp\left(\frac{1}{2}\rho_q\right) \odot \epsilon_q, \qquad \epsilon_q \sim \mathcal{N}(0, I), \tag{6}$$

where $\epsilon_q \in \mathbb{R}^{K\times d}$ is sampled from a standard Gaussian distribution. To retain a direct path for scene-specific information alongside the aligned latent, we fuse u_q with the original vision-language features z_{vl}: $\widetilde{z_{vl}} = (1-\alpha)z_{vl} + \alpha u_q$. The fused features are then concatenated with the language features to form the action condition: $c_t = [\mathrm{sg}(z_l); \widetilde{z_{vl}}]$. Here, $\alpha \in [0,1]$ is the mixing weight, $\mathrm{sg}(\cdot)$ denotes stop-gradient, and $c_t \in \mathbb{R}^{2K\times d}$. We stop gradients through z_l in the main branch to preserve the language-only information extracted by the language queries. The action expert then predicts the action chunk from c_t and the proprioceptive state as $\hat{a}_{t:t+H-1} = \pi_\theta(c_t, s_t)$.

Reciprocal Alignment. To explicitly couple the scene-conditioned action representation with task semantics derived from language, we align its latent distribution with the language anchor:

$$\mathcal{L}_{q\to p} = D_{\mathrm{KL}}\left(q(u \mid o_t, \ell) \,\|\, p(u \mid \ell)\right). \tag{7}$$

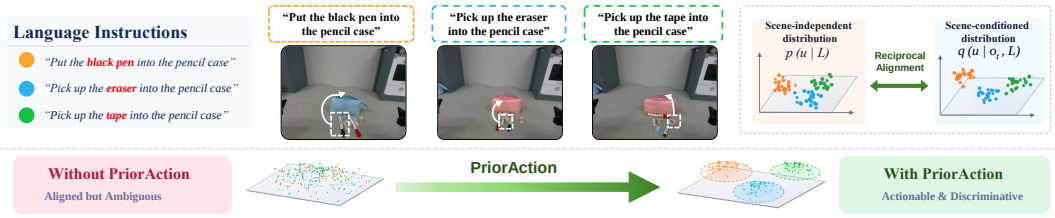


Figure 3: **Effect of Reciprocal Alignment and PriorAction.** Reciprocal Alignment couples the scene-independent and scene-conditioned distributions (top), while PriorAction injects action-relevant semantics into the language anchor (bottom).

To penalize mean differences relative to the variances of both distributions, we also include the reverse KL divergence:

$$\mathcal{L}_{p\to q} = D_{\mathrm{KL}}\left(p(u \mid \ell) \,\|\, q(u \mid o_t, \ell)\right). \tag{8}$$

The two terms form the Reciprocal Alignment objective $\mathcal{L}_{\mathrm{align}} = \mathcal{L}_{q\to p} + \mathcal{L}_{p\to q}$. Through bidirectional KL regularization, the predicted distribution scales adaptively modulate the penalty for deviations between the two representations. Because p and q are parameterized by z_l and z_{vl}, respectively, Reciprocal Alignment couples the language-only and scene-conditioned query representations in the shared latent action-conditioning space. This coupling regularizes the scene-conditioned representation toward the task-level semantics encoded by the language anchor, thereby encouraging the vision-language queries to aggregate task-relevant visual evidence for action prediction. The closed-form Gaussian KL expressions and additional analysis are provided in Appendix A.

PriorAction: Action-Shaped Language Anchor. Crucially, without direct action supervision, distribution alignment alone does not ensure that the language anchor captures the action distinctions between instructions (Fig. 3). Even when the two distributions are well aligned for each task, latent representations from different tasks may overlap rather than form task-discriminative clusters. To provide this supervision, we introduce *PriorAction*, an auxiliary loss that replaces the multimodal condition c_t with a sample from p:

$$c_t^{\mathrm{PA}} = \mu_p + \exp\left(\frac{1}{2}\rho_p\right) \odot \epsilon_p, \qquad \epsilon_p \sim \mathcal{N}(0, I). \tag{9}$$

The shared action expert receives condition c_t^{PA} through cross-attention, while the proprioceptive state s_t remains unchanged. We supervise this with the same action objective as the main branch and denote it by $\mathcal{L}_{\mathrm{PA}}$:

$$\mathcal{L}_{\mathrm{PA}} = \ell_{\mathrm{act}}\left(\pi_\theta(c_t^{\mathrm{PA}}, s_t), a^\star_{t:t+H-1}\right). \tag{10}$$

This auxiliary loss shapes the language anchor to encode action-relevant distinctions across instructions rather than scene-specific execution details. Reciprocal alignment uses this action-relevant anchor to couple the scene-conditioned distribution. Jointly, the two objectives encourage the scene-conditioned representation to remain consistent with the task-level semantics while incorporating visual evidence required for execution, thereby supporting instruction-consistent action generation.

Training and inference. The main action objective supervises the prediction from the context c_t: $\mathcal{L}_{\mathrm{main}} = \ell_{\mathrm{act}}\left(\pi_\theta(c_t, s_t), a^\star_{t:t+H-1}\right)$. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{main}} + \lambda_{\mathrm{PA}}\mathcal{L}_{\mathrm{PA}} + \lambda_{\mathrm{align}}\mathcal{L}_{\mathrm{align}}. \tag{11}$$

where λ_{PA} and λ_{align} control the weights of the two auxiliary objectives.

At inference, we replace sampling with the deterministic mean $u_q = \mu_q$ and construct c_t using the same residual fusion and concatenation as during training. The action expert predicts an action chunk from c_t and s_t. The language-only features z_l remain part of c_t, but the scene-independent distribution head and the PriorAction branch are no longer used. No additional backbone or action-expert forward pass is required at inference.

4 Experiments

4.1 Experimental Setup

Simulation benchmarks and baselines. We evaluate instruction generalization on LIBERO-Plus Language (Fei et al., 2025), which tests instruction variations across LIBERO-Goal, Spatial, Ob-

Table 1: Real-robot single-task evaluation under cross-condition, visual-distractor, and Unseen conditions settings. We report Semantic Accuracy and Task SR (%).

Dimension	Setting	Trials	$\pi_{0.5}$		GR00T		**LANCE (Ours)**	
		$\pi/G/L$	Sem. Acc.	Task SR	Sem. Acc.	Task SR	Sem. Acc.	Task SR
Cross-condition	Attribute-Near	20/20/20	**100.0**	90.0	85.0	40.0	**100.0**	**95.0**
	Attribute-Far	20/20/20	85.0	60.0	75.0	25.0	**100.0**	**90.0**
	State-Lying	20/20/20	85.0	**85.0**	85.0	60.0	**95.0**	**85.0**
	State-Standing	20/20/20	70.0	65.0	**100.0**	55.0	95.0	**85.0**
Visual distractor	Same Category (Attr.-State)	40/40/40	77.5	75.0	90.0	55.0	**95.0**	**85.0**
	Stacking / Clutter	20/20/20	75.0	**75.0**	75.0	35.0	**90.0**	**75.0**
	Partial Occlusion	10/10/10	80.0	60.0	**90.0**	40.0	**90.0**	**80.0**
Unseen conditions	Unseen Object	10/10/10	40.0	40.0	60.0	30.0	**90.0**	**90.0**
	Unseen Container	10/10/10	80.0	80.0	80.0	50.0	**100.0**	**100.0**
	Unseen Background	10/10/10	60.0	60.0	**80.0**	**80.0**	70.0	60.0
	Unseen Instruction	10/10/10	**100.0**	**100.0**	70.0	70.0	**100.0**	**100.0**

Table 2: Real-robot closed-loop composition of individually trained atomic skills. We report Semantic Coverage, Task Coverage, Ordered Progress, and Full SR. N and H denote rollouts and requested items per rollout, respectively. Entries show counts with percentages in parentheses.

Task	Model	N	H	Sem. Cov.↑	Task Cov.↑	Ord. Prog.↑	Full SR ↑
Multi-target composition							
Multiple Black Pens	GR00T	5	3	13/15 (86.7)	2/15 (13.3)	-	0/5 (0.0)
	$\pi_{0.5}$	5	3	**15/15 (100.0)**	**12/15 (80.0)**	-	**3/5 (60.0)**
	LANCE	5	3	**15/15 (100.0)**	**12/15 (80.0)**	-	**3/5 (60.0)**
Multiple State-conditioned Objects	GR00T	5	3	14/15 (93.3)	9/15 (60.0)	-	2/5 (40.0)
	$\pi_{0.5}$	5	3	14/15 (93.3)	13/15 (86.7)	-	3/5 (60.0)
	LANCE	5	3	**15/15 (100.0)**	**14/15 (93.3)**	-	**4/5 (80.0)**
Composite-instruction composition							
Black Pen → Standing Sharpener → Tape	GR00T	5	3	5/15 (33.3)	2/15 (13.3)	0/15 (0.0)	0/5 (0.0)
	$\pi_{0.5}$	5	3	8/15 (53.3)	6/15 (40.0)	4/15 (26.7)	0/5 (0.0)
	LANCE	5	3	**12/15 (80.0)**	**10/15 (66.7)**	**9/15 (60.0)**	**2/5 (40.0)**
Standing Sharpener → Lying Sharpener → Lying Eraser → Standing Eraser	GR00T	5	4	5/20 (25.0)	4/20 (20.0)	4/20 (20.0)	0/5 (0.0)
	$\pi_{0.5}$	5	4	12/20 (60.0)	10/20 (50.0)	4/20 (20.0)	1/5 (20.0)
	LANCE	5	4	**17/20 (85.0)**	**14/20 (70.0)**	**10/20 (50.0)**	**2/5 (40.0)**

ject, and Long, and LIBERO-PARA (Kim et al., 2026), which replaces the original instructions with semantically equivalent paraphrases while keeping the tasks unchanged. VLA-Arena UnseenObjects (Zhang et al., 2025) evaluates object generalization across three difficulty levels (L0-L2). We report task success rate (SR) on all benchmarks and the SR drop from the original instructions on LIBERO-PARA. Following the StarVLA (Ye et al., 2026) evaluation protocol, we train on the standard LIBERO data for LIBERO-Plus and LIBERO-PARA and only on the L0 training split for VLA-Arena. Our simulation comparisons include OpenVLA-OFT (Kim et al., 2025), $\pi_{0.5}$ (Intelligence et al., 2025), ABot-M0 (Yang et al., 2026b), Xiaomi-Robotics-0 (Cai et al., 2026), Qwen-based StarVLA variants (Ye et al., 2026), LangForce (Lian et al., 2026), and APT (Xu et al., 2026). We additionally use a Direct Concatenation (DC) baseline that directly concatenates the language-only and vision-language query features without probabilistic modeling, PriorAction, or Reciprocal Alignment. Implementation details and hyperparameters are provided in AppendixB.1, with additional simulation results in Appendix B.5.

Real-robot evaluation setup. We compare $\pi_{0.5}$ (Intelligence et al., 2025), GR00T-N1.7 (Bjorck et al., 2025), and LANCE on an ARX5 robot. The single-task evaluation covers changes in object attributes and states, target distance, visual clutter and occlusion, and unseen objects, containers, backgrounds, and instruction paraphrases. We report Semantic Accuracy and task success rate (SR), separating instruction-consistent selection from successful physical execution. All policies are fine-tuned on the same single-task demonstrations; no multi-target or composite trajectories are used for training. Detailed training data and single-task evaluation protocols are provided in Appendix B.2 and Appendix B.3.

Long-horizon closed-loop composition setup. We also evaluate test-time composition of individually trained atomic skills. Multi-target tasks require manipulating several objects that satisfy the same instruction condition, whereas composite instructions combine three or four ordered subtasks.

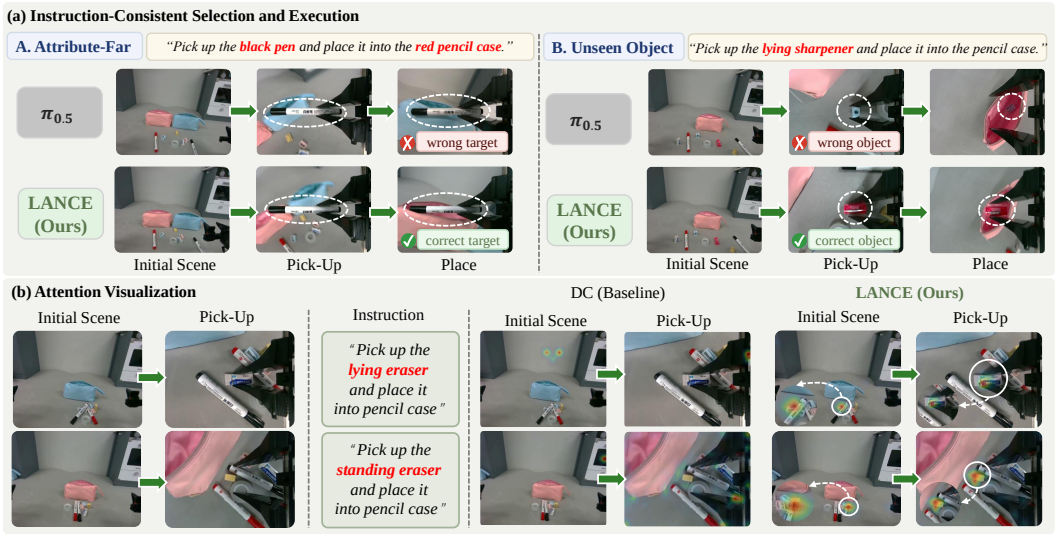


Figure 4: (a) Example rollouts on Attribute-Far and Unseen Object tasks. (b) Real-robot attention visualization under different instructions in the same scene.

Each policy receives the complete instruction once and executes continuously in closed loop. We evaluate five rollouts per task and report Semantic Coverage, Task Coverage, Ordered Progress, and Full SR. Detailed protocols and metric definitions are provided in Appendix B.4.

4.2 Real-robot results

Single-task results. LANCE achieves the strongest performance across most single-task settings (Table 1). For the baseline models, GR00T exhibits a substantial gap between Semantic Accuracy and Task SR. Meanwhile, $\pi_{0.5}$ frequently selects nearby or familiar objects instead of the instruction targets under Attribute-Far and Unseen Object (Fig. 4(a)), where LANCE substantially improves both Semantic Accuracy and Task SR.

Long-horizon results. Table 2 evaluates zero-shot composition of individually trained skills. On multi-target tasks, LANCE slightly exceeds $\pi_{0.5}$, while larger gains emerge on composite instructions. For the three- and four-subtask instructions, LANCE achieves semantic coverage of 80.0% and 85.0%, respectively, with higher Task Coverage and Ordered Progress than both baselines. These results show that LANCE maintains more reliable instruction-consistent target selection as preceding actions change the scene. Ordered Progress nevertheless remains below Task Coverage, indicating that completing requested subtasks does not yet guarantee execution in the specified order.

Real-robot attention visualization. As shown in Fig. 4(b), we change the language instruction while keeping the real-robot scene unchanged. DC shows similar and diffuse attention across different instructions, whereas LANCE shifts its attention to the task-relevant visual regions. This difference remains visible in both the initial scene and the pick-up views.

4.3 Simulation Results

Instruction generalization on LIBERO-Plus. Table 3(a) reports success rates on LIBERO-Plus Language. LANCE achieves the highest success rate among the compared methods (90.8%), exceeding the best baseline, LangForce (Lian et al., 2026), by 2.7 percentage points. Results on the remaining LIBERO-Plus subsets are provided in Appendix B.5.

Robustness to language paraphrases. On LIBERO-PARA, LANCE achieves 98.8% on the original instructions and retains 84.3% under paraphrased instructions (Table 3(b)). At the same original-task success rate of 98.8%, Xiaomi-Robotics-0 drops to 76.0%, corresponding to a larger performance drop of 22.8 versus 14.5 percentage points for LANCE. These results indicate stronger paraphrase robustness without sacrificing original-task performance.

Table 3: **Language generalization results.** Results on LIBERO-Plus Language and LIBERO-PARA. Drop denotes Goal minus PARA.

(a) LIBERO-Plus Language.

Method	SR (%) ↑
OpenVLA-OFT	79.5
$\pi_{0.5}$	83.3
ABot-M0	86.4
Qwen2.5-VL-GR00T	86.3
Qwen3-VL-OFT	87.0
APT	77.6
LangForce	<u>88.1</u>
LANCE (Ours)	**90.8**

(b) LIBERO-PARA.

Method	Goal	PARA	Drop ↓
OpenVLA-OFT$_{\text{goal}}$	97.9	64.7	33.2
OpenVLA-OFT$_{\text{mixed}}$	96.1	63.7	32.4
$\pi_{0.5}$	97.6	71.4	26.2
$\pi_{0.5}$ (expert-only)	78.6	39.1	39.5
X-VLA	97.8	62.1	35.7
VLA-Adapter	98.2	46.3	51.9
Xiaomi-Robotics-0	**98.8**	<u>76.0</u>	<u>22.8</u>
LANCE (Ours)	**98.8**	**84.3**	**14.5**

Table 4: Component ablation on LIBERO-Plus Language. Query denotes the DC query interface; Dist., PA, and Align. denote Gaussian parameterization, PriorAction, and Reciprocal Alignment, respectively. All values are success rates (%).

Query	Dist.	PA	Align.	Goal	Spatial	Object	Long	Avg. ↑
✓				81.5	92.8	87.3	92.2	88.4
✓	✓			75.1	95.9	90.1	**95.8**	89.0
✓	✓	✓		78.5	**96.2**	89.0	91.6	88.7
✓	✓		✓	78.5	93.8	88.7	90.3	87.7
✓	✓	✓	✓	**84.1**	95.6	**90.4**	93.2	**90.8**

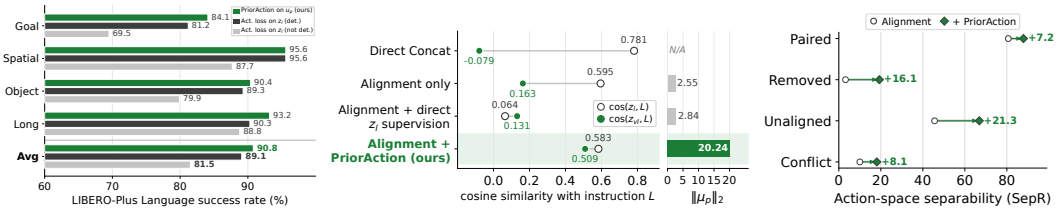
4.4 Ablation Study

Component ablation. We ablate PriorAction and Reciprocal Alignment in Table 4. Adding either component alone does not outperform the Gaussian-only variant, whereas combining both yields the highest average success rate of 90.8%. This pattern reflects their complementary roles: Reciprocal Alignment alone does not make the language anchor action-relevant, while PriorAction alone does not couple the scene-independent and scene-conditioned distributions.

Placement of action supervision. We compare action supervision on the language-only features z_l and on samples $u_p \sim p(u \mid \ell)$ from the scene-independent distribution (Fig. 5a). When action supervision is applied to z_l, stopping its gradients performs better than allowing the action loss to update it directly. PriorAction, which supervises the sampled u_p, achieves the highest average success rate. These results support placing action supervision in the scene-independent distribution, rather than directly on the upstream language query features.

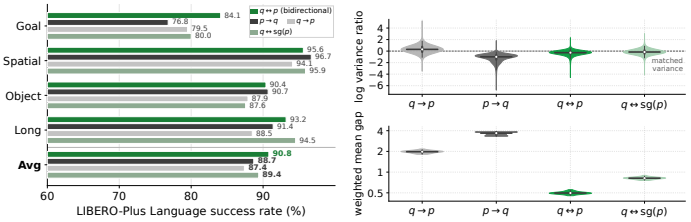
Representation analysis. We further examine how the placement of action supervision affects the instruction alignment of the two query representations, measured by their cosine similarity to the instruction (Fig. 5(b)). Direct action supervision on z_l reduces its instruction alignment to 0.064. In contrast, PriorAction largely preserves the instruction alignment of z_l, while increasing that of z_{vl} from 0.163 to 0.509 relative to alignment alone. Thus, PriorAction preserves the instruction information extracted by the language queries while increasing the instruction alignment of the vision-language features.

Action-space separability. To evaluate action-relevant distinctions induced by PriorAction, we measure action-space separability on 100 observations from eight LIBERO-Goal tasks (Fig. 5(c)). Following LA4VLA (Lin et al., 2026), we report the Separability Ratio (SepR) to quantify ratio of inter-task to intra-task distances in the predicted actions, with details in Appendix C.2. PriorAction increases SepR across all conditions, including from 3.2 to 19.3 when visual input is removed. The gains persist under unaligned and conflicting observations, indicating stronger action-relevant distinctions between instructions even when visual evidence is absent or misleading.



(a) Language-action supervision. (b) Representation analysis. (c) Action-space separability.

Figure 5: **PriorAction provides behaviorally meaningful language anchoring.** (a) Success rates (%) for different supervision placements on LIBERO-Plus Language. (b) Query-instruction cosine similarity and language-anchor mean norm. (c) Action-space separability (SepR) with alignment alone and with added PriorAction under four visual conditions.



(a) KL direction ablation. (b) Distribution matching.

Figure 6: **Analysis of distribution alignment.** Success rates (%) and distribution statistics under different alignment strategies.

Suite	Deterministic matching	Shared query	**Ours**
Goal	83.7	79.0	**84.1**
Spatial	92.1	91.8	**95.6**
Object	**90.4**	87.0	**90.4**
Long	92.7	92.2	**93.2**
Avg. ↑	89.7	87.4	**90.8**

Table 5: Design ablation on LIBERO-Plus Language.

Alignment ablation. Reciprocal alignment ($q \leftrightarrow p$) achieves the highest average success rate of 90.8% among the tested alignment variants (Fig. 6(a)). Compared with single-direction alignment, Reciprocal Alignment yields more balanced variances and the lowest precision-weighted mean gap (0.50) on 128 static LIBERO observation instruction pairs, which measures mean mismatch relative to the predicted variances (Fig. 6(b)). Stopping alignment gradients through the language anchor ($q \leftrightarrow \mathrm{sg}(p)$) increases this gap to 0.82 and yields a lower average success rate of 89.4%.

Query and distribution design. Table 5 compares two alternative designs. *Deterministic matching* replaces Gaussian distribution alignment with an MSE loss between the language and vision-language query features, while *shared query weights* tie the learnable parameters of the two query banks. Both variants achieve lower average success rates than LANCE.

5 Conclusion

We presented LANCE, an action- query framework that explicitly couples scene-conditioned action representations to task semantics derived from language instructions. LANCE uses language-only queries to construct a scene-independent language anchor and vision-language queries to extract scene-conditioned features for action prediction. *PriorAction* shapes an action-relevant language anchor through action supervision, while *Reciprocal Alignment* couples the scene-independent and scene-conditioned distributions. Together, these designs encourage task-relevant visual evidence selection for instruction-consistent action generation. Across simulation and real-robot evaluations, LANCE improves robustness to language variations and unseen objects, enables more reliable instruction-consistent target selection, and achieves higher semantic and task coverage under zero-shot composition of learned skills.

Our results highlight three broader findings for language-guided robot control. First, language can play a more active role than simply conditioning action prediction by steering the selection of task-relevant visual evidence. Second, the placement of action supervision matters: applying it through the language anchor captures action-relevant distinctions between instructions while preserving the language-only features. Third, instruction-consistent target selection remains reliable as preceding actions change the scene during long-horizon compositional execution. Yet the remaining gap between Task Coverage and Ordered Progress shows that completing requested subtasks does not yet ensure execution in the specified order. This motivates future systems that combine task-relevant visual evidence selection with temporal task context to explicitly track task progress.

AI USE STATEMENT

We used generative AI tools for language polishing and readability improvements, as well as for assisting with the editing and visual refinement of figures and captions. All AI-assisted content was reviewed by the authors.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023b. URL https://arxiv.org/abs/2212.06817.

Rui Cai, Jun Guo, Xinze He, Piaopiao Jin, Jie Li, Bingxuan Lin, Futeng Liu, Wei Liu, Fei Ma, Kun Ma, et al. Xiaomi-robotics-0: An open-sourced vision-language-action model with real-time execution. *arXiv preprint arXiv:2602.12684*, 2026.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, 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, Andrey Kolobov, 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, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, 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, Jay Vakil, 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, Muhammad Zubair Irshad, 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 Mart'in-Mart'in, 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, Shubham Tulsiani, 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, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, 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, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

Zeyuan Ding, Wenhai Liu, Yang Xu, Jiayu Hu, Yinda Chen, Yi Zhang, Yong Dai, Jian Tang, and Xiaozhu Ju. Pelican-vla 0.5: Attending before acting benefits generalization, 2026. URL https://arxiv.org/abs/2607.06655.

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, 2025. URL https://arxiv.org/abs/2505.23705.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

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. *arXiv preprint arXiv:2503.03734*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. π0. 5: a vision-language-action model with open-world generalization, 2025. *URL https://arxiv. org/abs/2504.16054*, 1(2): 3, 2025.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, et al. Guidedvla: Specifying task-relevant factors via plug-and-play action attention specialization. *arXiv preprint arXiv:2605.12369*, 1, 2026.

Chanyoung Kim, Minwoo Kim, Minseok Kang, Hyunwoo Kim, and Dahuin Jung. Libero-para: A diagnostic benchmark and metrics for paraphrase robustness in vla models. *arXiv preprint arXiv:2603.28301*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action models via instruction-driven routing & sparsification. *Advances in neural information processing systems*, 38:137646–137675, 2026.

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. *arXiv preprint arXiv:2601.15197*, 2026.

Tao Lin, Yuxin Du, Yiran Mao, Zewei Ye, Yilei Zhong, Bing Cheng, Yiming Wang, Jiting Liu, Yang Tian, Junchi Yan, et al. La4vla: Learning to act without seeing via language-action pretraining. *arXiv preprint arXiv:2606.27295*, 2026.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. 2025. URL https://openreview.net/forum?id=UPHlqbZFZB.

Qiuyue Wang, Mingsheng Li, Jian Guan, Jinhui Ye, Sicheng Xie, Yitao Liu, Junhao Chen, Zhixuan Liang, Jie Zhang, Xintong Hu, Xuhong Huang, Pei Lin, Junyang Lin, Dayiheng Liu, Shuai Bai, Jingren Zhou, Jiazhao Zhang, Haoqi Yuan, Gengze Zhou, Hang Yin, Ye Wang, Yiyang Huang, Zixing Lei, Wujian Peng, Delin Chen, Yingming Zheng, Jingyang Fan, Xianwei Zhuang, Xin Zhou, Haoyang Li, Anzhe Chen, Tong Zhang, Xuejing Liu, Yuchong Sun, Ruizhe Chen, Zhaohai Li, Chenxu Lü, Zhibo Yang, Tao Yu, and Xionghui Chen. Qwen-vla: Unifying vision-language-action modeling across tasks, environments, and robot embodiments, 2026. URL https://arxiv.org/abs/2605.30280.

Kechun Xu, Zhenjie Zhu, Anzhe Chen, Shuqi Zhao, Qing Huang, Yifei Yang, Haojian Lu, Rong Xiong, Masayoshi Tomizuka, and Yue Wang. Seeing to act, prompting to specify: A bayesian factorization of vision language action policy. *arXiv preprint arXiv:2512.11218*, 2025.

Kechun Xu, Zhenjie Zhu, Anzhe Chen, Rong Xiong, and Yue Wang. Apt: Action expert pretraining improves instruction generalization of vision-language-action policies. *arXiv preprint arXiv:2606.12366*, 2026.

Senqiao Yang, Chengyao Wang, Yuxin Chen, Zixuan Wang, Longxiang Tang, Haokun Gui, Jinhui Ye, Changsheng Lu, Xiaoyang Wu, Mingkang Zhu, Pengguang Chen, Shu Liu, Zhuotao Tian, Hengshuang Zhao, Bei Yu, and Jiaya Jia. Beyond data scaling: Representation-centric continued pre-training for vision-language-action models, August 2026a. URL https://starvla.github.io/VLAct/. Preprint.

Shuai Yang, Hao Li, Yilun Chen, Bin Wang, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*, 2025.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, et al. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026b.

Jinhui Ye, Ning Gao, Senqiao Yang, Jinliang Zheng, Zixuan Wang, Yuxin Chen, Pengguang Chen, Yilun Chen, Shu Liu, and Jiaya Jia. Starvla-alpha: Reducing complexity in vision-language-action systems. *arXiv preprint arXiv:2604.11757*, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Yuliang Wei, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Xinming Wang, Bailing Wang, Cong Huang, and Kai Chen. Twinbrainvla: Unleashing the potential of generalist vlms for embodied tasks via asymmetric mixture-of-transformers, 2026. URL https://arxiv.org/abs/2601.14133.

Lihan Zha, Asher J. Hancock, Mingtong Zhang, Tenny Yin, Yixuan Huang, Dhruv Shah, Allen Z. Ren, and Anirudha Majumdar. Lap: Language-action pre-training enables zero-shot cross-embodiment transfer, 2026. URL https://arxiv.org/abs/2602.10556.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. Vla-arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539*, 2025.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

A Derivation and Analysis of Reciprocal Alignment

This appendix derives the closed-form Reciprocal Alignment objective for the scene-independent and scene-conditioned distributions introduced in Sec. 3.2. The resulting expression shows how the mean discrepancy is weighted by the predicted variances of both distributions.

A.1 Closed-Form Derivation of the KL Divergences

LANCE models the scene-independent distribution and scene-conditioned distribution as diagonal Gaussians:

$$p(u \mid \ell) = \mathcal{N}\left(\mu_p, \mathrm{diag}(\sigma_p^2)\right), \tag{12}$$

$$q(u \mid o_t, \ell) = \mathcal{N}\left(\mu_q, \mathrm{diag}(\sigma_q^2)\right), \tag{13}$$

where $u \in \mathbb{R}^{K\times d}$ and all variance components are strictly positive. For analytical clarity, we first derive the KL divergence for a single latent dimension u_j. Let $q_j = \mathcal{N}(\mu_{q,j}, \sigma_{q,j}^2)$ and $p_j = \mathcal{N}(\mu_{p,j}, \sigma_{p,j}^2)$.

Forward KL. By definition, $D_{\mathrm{KL}}(q_j \| p_j) = \mathbb{E}_{u_j\sim q_j}[\log q_j(u_j) - \log p_j(u_j)]$. Substituting the Gaussian densities and utilizing $\mathbb{E}_{q_j}[(u_j - \mu_{p,j})^2] = \sigma_{q,j}^2 + (\mu_{q,j} - \mu_{p,j})^2$, we obtain the closed form:

$$D_{\mathrm{KL}}(q_j \| p_j) = \frac{1}{2}\left[\log\frac{\sigma_{p,j}^2}{\sigma_{q,j}^2} + \frac{\sigma_{q,j}^2 + (\mu_{q,j} - \mu_{p,j})^2}{\sigma_{p,j}^2} - 1\right]. \tag{14}$$

Reverse KL. Exchanging the roles of the two distributions yields the reciprocal divergence:

$$D_{\mathrm{KL}}(p_j \| q_j) = \frac{1}{2}\left[\log\frac{\sigma_{q,j}^2}{\sigma_{p,j}^2} + \frac{\sigma_{p,j}^2 + (\mu_{p,j} - \mu_{q,j})^2}{\sigma_{q,j}^2} - 1\right]. \tag{15}$$

Reciprocal Alignment. Summing over all $K \times d$ latent dimensions, the symmetric objective $\mathcal{L}_{\mathrm{Aign}} = D_{\mathrm{KL}}(q\|p) + D_{\mathrm{KL}}(p\|q)$ cancels the logarithmic variance-ratio terms, yielding the unnormalized objective:

$$\mathcal{L}_{\mathrm{Aign}} = \frac{1}{2}\sum_{k,j}\left[\frac{\sigma_{q,kj}^2}{\sigma_{p,kj}^2} + \frac{\sigma_{p,kj}^2}{\sigma_{q,kj}^2} - 2 + (\mu_{q,kj} - \mu_{p,kj})^2\left(\frac{1}{\sigma_{p,kj}^2} + \frac{1}{\sigma_{q,kj}^2}\right)\right]. \tag{16}$$

Unlike a single-direction KL term, Reciprocal Alignment weights the mean discrepancy by the precisions of both distributions. It also penalizes relative variance mismatch in both directions, providing the bidirectional distribution coupling used in the main method.

B Additional Experiment Details

B.1 Implementation Details

Training setup. Our implementation is built on the StarVLA framework (Ye et al., 2026). We use Qwen3-VL-4B-Instruct as the vision-language backbone and a DiT-B action expert. All models are trained for 30,000 steps on 8 NVIDIA A800 GPUs with a global batch size of 128. We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.95$, and clip the gradient norm to 1.0. The learning rates are set to 2.5×10^{-5} for the remaining trainable modules, 1×10^{-5} for the Qwen-VL interface, and 1×10^{-4} for the action expert. We use 2,000 warm-up steps followed by cosine decay to a minimum learning rate of 1×10^{-6}.

LANCE configuration. We use 32 language-only queries and 32 vision-language queries. The PriorAction is applied to a reparameterized sample from $p(u \mid L)$ with a loss weight of 0.1. For Reciprocal Alignment, the $\mathcal{L}_{\mathrm{align}}$ is weighted by $\frac{1}{d}$. The latent-feature mixing coefficient follows a cosine schedule up to $\alpha = 0.5$ and is fixed to 0.5 during inference. The action expert predicts eight-step chunks of 7-D delta joint-position actions and uses four denoising steps at inference. Unless otherwise stated, all ablations retain the same training configuration and modify only the component under study.



Figure 7: Real-world platform, containers, objects, and background variations used in our realworld evaluations.

Table 6: Atomic tasks used for real-robot fine-tuning. Each task variant contains 50 demonstrations.

Training factor	Task variants	Per variant	Total
Object & Receptacle Attributes	Pick-and-place of {*red*, *black*} pens into {*red*, *blue*} pencil cases (4 variants).	50	200
Object-state Grounding	Pick-and-place of {*lying*, *standing*} × {*eraser*, *sharpener*} into pencil cases (4 variants).	50	200
Additional Tasks	Placing tape into a pencil case using either arm (2 variants).	50	100
		Total	**500**

B.2 Real-Robot Training and Evaluation Protocol

Training data and evaluation setup. All models are fine-tuned on the same dataset of 500 teleoperated trajectories collected with the ARX5 robot, with 50 demonstrations for each of ten atomic tasks. As summarized in Table 6, the training set consists of ten atomic manipulation tasks covering the objects and task variations used for real-robot fine-tuning. All models use the same sensory inputs, robot controllers, and software pipeline, with policy inference performed on a single NVIDIA H100 GPU. The training set contains no paraphrased instructions, unseen scene variations, multi-target episodes, or composite long-horizon demonstrations.

B.3 Single-Task Evaluation Protocol.

Evaluation settings. Before each rollout, the workspace is reset to the designated configuration, with the red and blue pencil cases fixed on the left and right sides, respectively. *Near* and *Far* indicate whether the target object is initialized near or far from the container. Each instruction-configuration pair is evaluated over five trials. As shown in Fig. 8, *Attribute-Near* and *Attribute-Far* each contain four pen-container combinations, yielding 20 trials per setting. Similarly, *State-Lying* and *State-Standing* each contain the eraser and sharpener variants, also yielding 20 trials per setting. *Same-category* evaluates target selection among objects from the same semantic category that differ only in attributes or states, comprising 40 trials. *Stacking/Clutter* contains 20 trials, while *Partial Occlusion* and each unseen-condition setting contain 10 trials.

Evaluation metrics. All trials are manually reviewed to compute the metrics reported in Table 1. Selecting an incorrect object, object state, or destination constitutes a semantic failure. If the robot makes an instruction-consistent selection but subsequently fails because of grasping, placement, or collision errors, the trial is counted as correct for Semantic Accuracy but unsuccessful for Task SR.

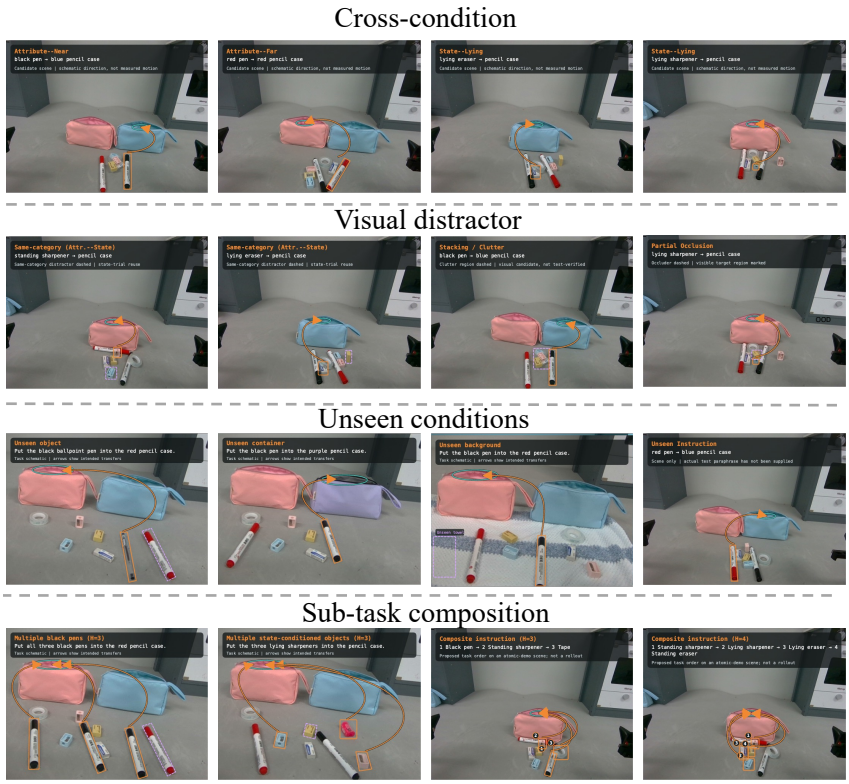


Figure 8: Representative real-world evaluation settings covering cross-condition grounding, visual distractors, generalization to unseen objects, containers, backgrounds, and instructions, and sub-task composition.

B.4 Long-Horizon Closed-Loop Evaluation Protocol

Long-horizon protocol. We evaluate zero-shot composition using models trained only on the single-task dataset described above, where the atomic skills are observed individually but never demonstrated together as a complete trajectory. The *multi-target tasks* require manipulating three objects that satisfy the same instruction condition ($H = 3$), with no prescribed execution order. The *composite-instruction tasks* require the following ordered sequences:

$$\text{Black Pen} \rightarrow \text{Standing Sharpener} \rightarrow \text{Tape}, \qquad H = 3, \tag{17}$$

and

$$\text{Standing Sharpener} \rightarrow \text{Lying Sharpener} \rightarrow \text{Lying Eraser} \rightarrow \text{Standing Eraser}, \qquad H = 4. \tag{18}$$

Here, H denotes the number of requested targets or subtasks. Each task is evaluated over five continuous closed-loop trials. The complete instruction is provided only once at the beginning of each episode, with no scene resets, intermediate language updates, or human interventions.

Long-horizon metrics. For trial i, let s_i denote the number of requested targets or subtasks selected consistently with the instruction, e_i the number of subtasks physically completed, and p_i the length of the longest completed prefix that follows the specified order. For N trials with horizon H, we compute:

$$\text{Sem. Cov.} = \frac{\sum_{i=1}^{N} s_i}{NH}, \qquad \text{Task Cov.} = \frac{\sum_{i=1}^{N} e_i}{NH}, \tag{19}$$

$$\text{Ord. Prog.} = \frac{\sum_{i=1}^{N} p_i}{NH}, \qquad \text{Full Completion SR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[e_i = H]. \tag{20}$$

Table 7: Zero-shot robustness evaluation on LIBERO-Plus. We report success rates (%) across seven perturbation dimensions. **Bold** highlights the best result in each column.

Method	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
OpenVLA-OFT	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
OpenVLA-OFT$_w$	10.4	38.7	70.5	76.8	93.6	49.9	69.9	55.8
NORA	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
WorldVLA	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
UniVLA	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
$\pi_{0.5}$	**75.8**	**79.4**	83.3	95.5	95.0	**89.6**	**87.0**	**85.7**
RIPT-VLA	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
OpenVLA-OFT$_m$	55.6	21.7	81.0	92.7	91.0	78.6	68.7	67.9
ABot-M0	60.4	67.9	86.4	96.2	91.6	86.4	82.6	80.5
Qwen2.5-VL-FAST	19.6	27.6	74.5	75.2	71.0	27.4	62.7	48.9
Qwen2.5-VL-GR00T	32.9	50.8	86.3	96.2	85.7	62.0	73.6	66.4
Qwen2.5-VL-OFT	34.4	63.7	82.1	86.9	88.8	53.3	74.6	67.2
Qwen3-VL-OFT	47.0	60.1	87.0	**96.3**	**95.3**	73.1	79.2	75.0
LangForce	30.6	39.5	88.1	89.3	87.8	37.5	68.5	60.5
APT	31.8	63.1	77.6	93.6	92.3	76.0	80.1	71.6
LANCE (Ours)	58.0	63.1	**90.8**	92.9	90.8	60.3	73.1	74.0

Table 8: Success rate on the UnseenObjects split of VLA-Arena.

Method	L0	L1	L2	Avg. ↑
OpenVLA	0.70	0.50	0.00	0.40
OpenVLA-OFT	0.60	0.40	0.20	0.40
π_0	**0.80**	0.50	0.00	0.43
π_0-FAST	0.00	0.00	0.00	0.00
UniVLA	0.50	**0.70**	0.10	0.43
SmolVLA	0.20	0.20	0.00	0.13
Qwen2.5-VL-PI	0.34	0.32	0.00	0.22
Qwen3-VL-PI	0.40	0.60	0.06	0.35
Qwen2.5-VL-OFT	0.36	0.34	0.02	0.24
Qwen3-VL-OFT	0.60	0.24	0.06	0.30
Qwen2.5-VL-GR00T	0.58	0.52	0.20	0.43
Qwen3-VL-GR00T	0.63	0.46	<u>0.26</u>	<u>0.45</u>
LANCE (Ours)	0.72	**0.70**	**0.50**	**0.63**

Ordered Progress is reported only for composite instructions because multi-target tasks have no prescribed order. Each target or subtask receives credit at most once. For composite instructions, completing a later subtask after skipping or reordering an earlier one increases Task Coverage but does not extend the ordered prefix. For example, completing subtasks 1 and 3 while failing subtask 2 yields a Task Coverage of $2/H$ but an Ordered Progress of only $1/H$.

B.5 Additional simulation results

Full LIBERO-Plus results. Table 7 reports results across all LIBERO-Plus perturbations. LANCE achieves its strongest relative performance under language variation, reaching the highest success rate of 90.8%. Its performance on the remaining perturbations is mixed, indicating that the primary gains lie in instruction generalization rather than generic visual robustness.

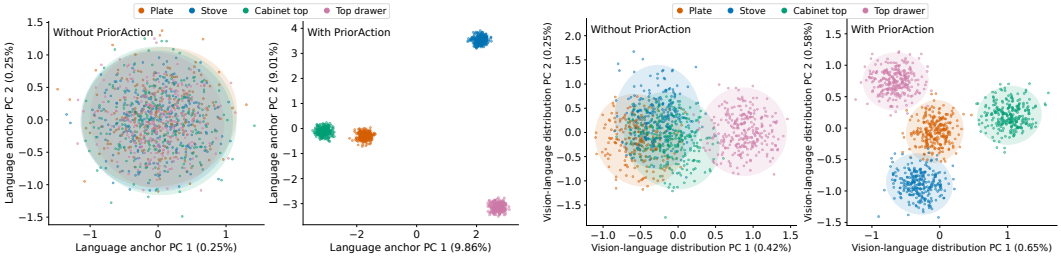
Generalization to unseen objects. Table 8 evaluates object generalization on the VLA-Arena UnseenObjects suit. LANCE achieves the highest average success rate of 0.63, compared with 0.45 for the strongest baseline. The gain is largest on the more challenging L2 split, where LANCE reaches 0.50 compared with 0.26 for the strongest baseline. These results further support LANCE's generalization to unseen objects.

C Additional Ablation Analysis

C.1 Action-Shaped distribution.

To examine how PriorAction shapes the latent distributions, we compare models trained with and without this auxiliary objective. We use four bowl-placement instructions (plate, stove, cabinet top, and top drawer) under fixed visual inputs. For visualization, we average the query-wise Gaussian parameters over the $K = 32$ queries to obtain one Gaussian per instruction and draw 256 samples for PCA projection.

As shown in Fig. 9, without PriorAction, samples from different instructions largely overlap in both the language anchor and scene-conditioned distribution. With PriorAction, the language-anchor samples form clearly separated instruction-specific clusters, while the scene-conditioned distributions also become substantially more separated. This pattern is consistent with PriorAction shaping an instruction-discriminative language anchor and Reciprocal Alignment coupling this structure to the scene-conditioned distribution.



(a) Language Anchor. (b) scene-conditioned distribution.

Figure 9: 2D PCA projections of language and vision language distribution samples across four alternative instructions, comparing models trained with and without PriorAction.

C.2 Action-Space Behavioral Analysis

To examine whether these representation differences are reflected in action predictions, we evaluate action-space separability under four visual conditions. *Paired* uses the observation corresponding to the instruction; *Removed* removes visual information; *Unaligned* uses an observation from a different task; and *Conflict* pairs the instruction with visual evidence corresponding to an incompatible task condition. All other policy inputs remain unchanged.

Standard task success does not directly measure whether different instructions induce distinct action predictions. We therefore follow the action-space analysis of LA4VLA (Lin et al., 2026), adapting its instruction grouping from directional instructions to task-level instructions.

Action representation. For each evaluation instance i, the policy predicts an H-step action chunk. We summarize its translational motion by the cumulative end-effector displacement

$$\hat{\mathbf{p}}_i = \sum_{h=0}^{H-1} \Delta\hat{\mathbf{x}}_{i,h}, \qquad \hat{\mathbf{p}}_i \in \mathbb{R}^3, \tag{21}$$

where $\Delta\hat{\mathbf{x}}_{i,h}$ denotes the predicted Cartesian translation at step h. Rotation and gripper dimensions are not included in the separability calculation.

Let $y_i \in \mathcal{Y}$ denote the task/instruction label associated with sample i, and define

$$\mathcal{I}_k = \{i : y_i = k\} \tag{22}$$

as the set of samples belonging to instruction group k.

Separability ratio. We use the Separability Ratio to measure whether different instructions induce distinct action predictions. To avoid confusion with success rate (SR), we denote this metric by SepR throughout our paper.

We first compute the average distance between predictions generated under the same instruction:

$$d_{\text{intra}} = \frac{1}{|\mathcal{Y}|} \sum_{k \in \mathcal{Y}} \frac{2}{|\mathcal{I}_k|(|\mathcal{I}_k| - 1)} \sum_{\substack{i,j \in \mathcal{I}_k \\ i<j}} \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\|_2 \,. \tag{23}$$

We then compute the average distance between predictions generated under different instructions:

$$d_{\text{inter}} = \frac{2}{|\mathcal{Y}|(|\mathcal{Y}| - 1)} \sum_{\substack{a,b \in \mathcal{Y} \\ a<b}} \frac{1}{|\mathcal{I}_a||\mathcal{I}_b|} \sum_{i \in \mathcal{I}_a} \sum_{j \in \mathcal{I}_b} \|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j\|_2 \,. \tag{24}$$

The Separability Ratio is

$$\text{SepR} = \frac{d_{\text{inter}}}{d_{\text{intra}} + \epsilon}, \tag{25}$$

where ϵ is a small constant for numerical stability.

A high SepR indicates compact predictions within the same instruction and greater separation across different instructions. SepR therefore measures instruction-dependent structure in the predicted action space rather than task success.

C.3 Execution-Time Distribution Dynamics

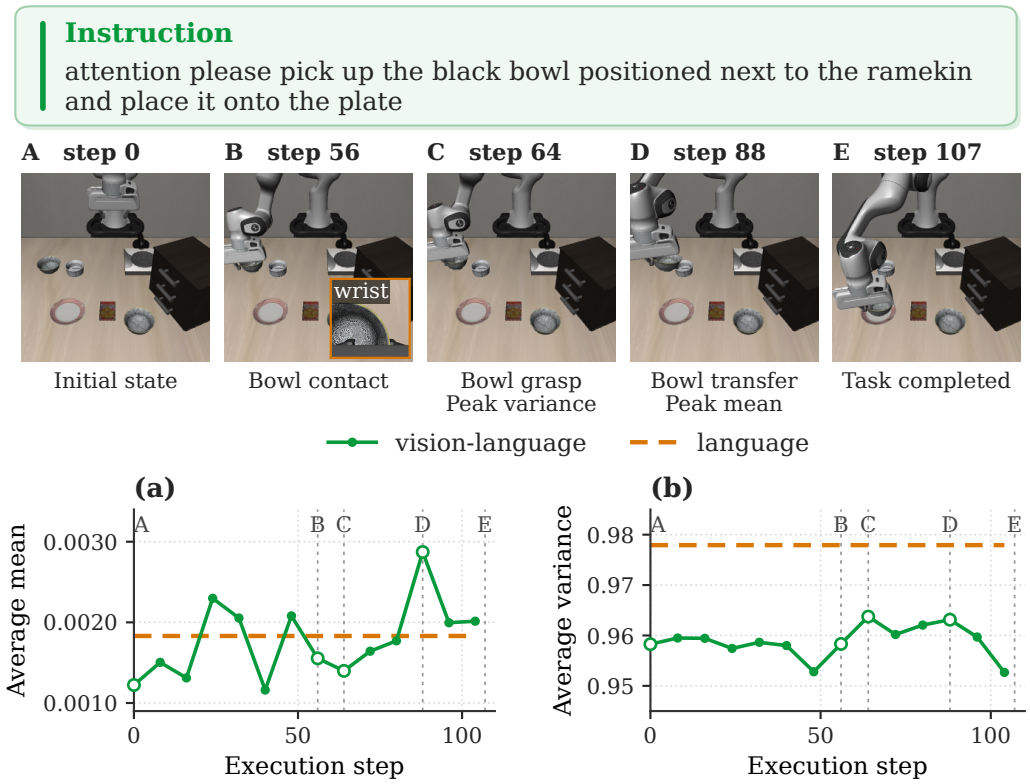


Figure 10: Execution-time dynamics of the language anchor and scene-conditioned distribution. Top: representative frames at key execution stages. Bottom: average latent mean and variance over execution steps.

To further examine how the two distributions behave during execution, we track their average mean and variance for the instruction "attention please pick up the black bowl positioned next to the ramekin and place it onto the plate." As shown in Fig. 10, the language distribution remains relatively stable throughout the rollout, whereas the scene-conditioned distribution changes across execution stages, particularly around grasp and transfer. These dynamics are consistent with the design of LANCE: the language anchor maintains task-consistent action conditioning, while the scene-conditioned distribution adapts to the current observation and execution state.

C.4 Additional Real-Robot Ablations

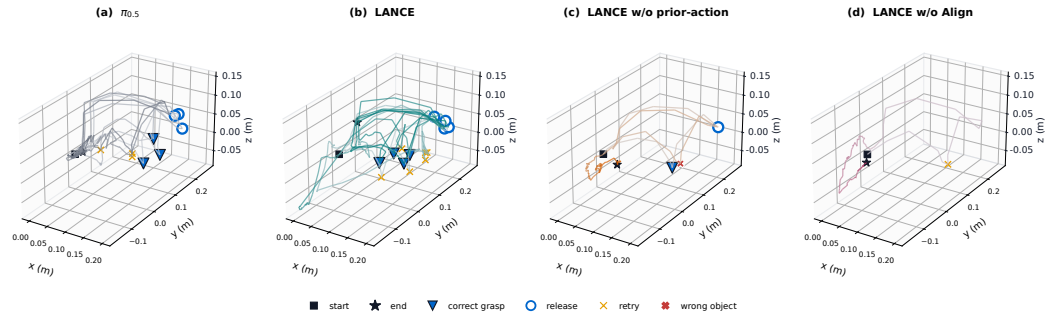


Figure 11: Trajectory-level analysis of the real-robot ablation. Three-dimensional trajectories for (a) $\pi_{0.5}$, (b) LANCE, (c) LANCE without PriorAction, and (d) LANCE without Reciprocal Alignment.

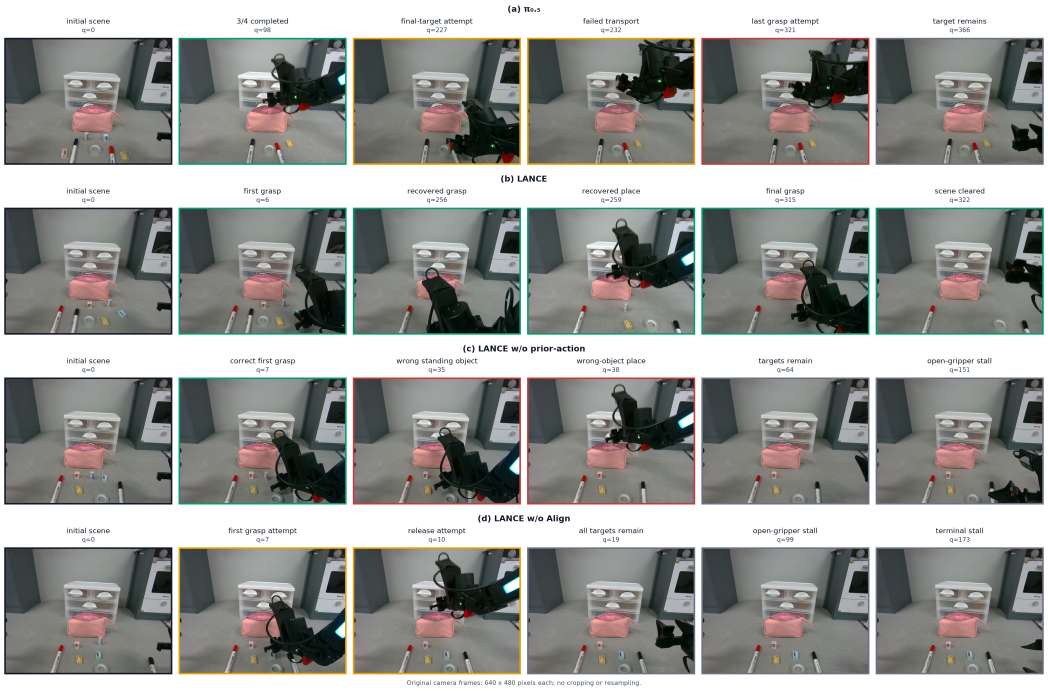


Figure 12: Real-robot ablation on closed-loop skill composition. Representative executions under a fixed four-subtask instruction: Standing Sharpener → Lying Sharpener → Lying Eraser → Standing Eraser into the pencil case.

To isolate the contributions of PriorAction and Reciprocal Alignment during closed-loop execution, we independently remove each objective while retaining the main action loss. Fig. 12 and Fig. 11 show representative rollouts for a four-subtask composite instruction.

The baseline $\pi_{0.5}$ completes the first three subtasks but fails to identify the final target. Without PriorAction, the policy initially selects the correct object but is later distracted by an irrelevant one. Without Reciprocal Alignment, the policy exhibits target-selection failures early in the rollout, leading to unsuccessful grasp–release cycles. In contrast, the full LANCE model completes all four subtasks as preceding actions change the scene.

These qualitative results illustrate the complementary roles of the two objectives: PriorAction shapes an action-relevant language anchor, while Reciprocal Alignment couples it with the scene-conditioned distribution. Together, they support more reliable instruction-consistent target selection during closed-loop execution.