Semantic Anchoring for Robotic Action Representations

Yuan Xu[*,1], **Youheng Shi**[*,1], **Chengyang Li**[2,3], **Wentao Zhu**[2], **Yizhou Wang**[1]

[1]Peking University [2]Eastern Institute of Technology, Ningbo [3]Shanghai Jiao Tong University

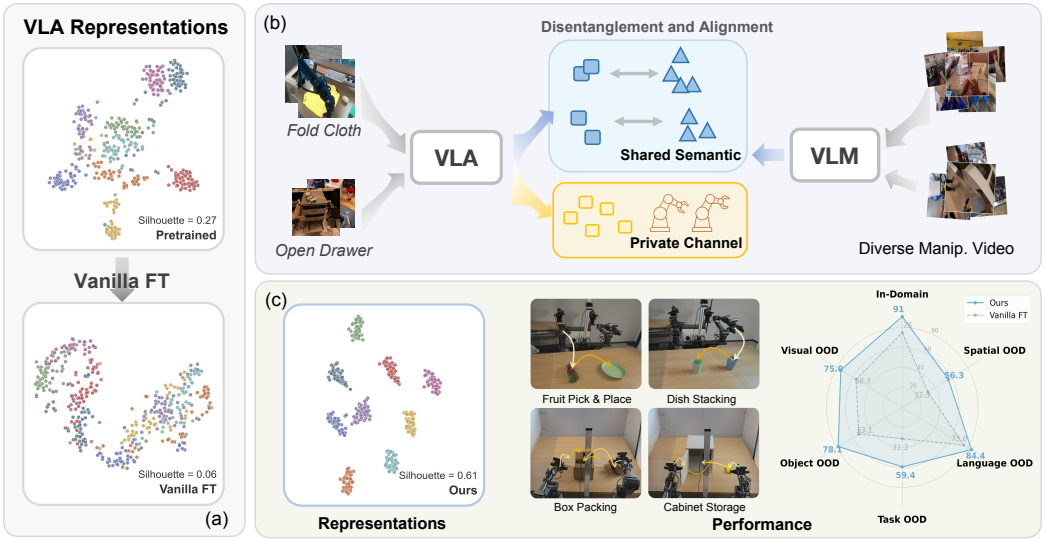

Figure 1: **Semantic anchoring for robotic action representations. (a)** Vanilla action-only fine-tuning destroys the semantic structure inherited from pretraining: t-SNE of π_0's layer-10 features on LIBERO-Goal [1], colored by task. **(b)** Our plug-and-play method anchors action representations to a semantic manifold via contrastive alignment with shared/private decomposition. **(c)** Repaired representations recover semantic structure and consistently improve both in-distribution and OOD performance on a real bimanual platform.

Abstract: Vision-Language-Action (VLA) models inherit rich semantic representations from pretrained Vision-Language Models, yet fine-tuning on limited robot demonstrations degrades this structure and undermines generalization. A fundamental question therefore arises: *what constitutes a good action representation?* Inspired by the mirror neuron theory's insight that observation and execution share an intention-level encoding, we examine whether a robot's action representations preserve the semantic structure captured by pretrained encoders. Systematic probing confirms that this structure erodes during fine-tuning, and that its quality synchronizes with both task success and out-of-distribution generalization. We further introduce a plug-and-play method that anchors action representations to a semantic manifold while decomposing representations into a shared semantic channel and a private channel, all discarded at inference, leaving the deployed model unchanged. Validated on different VLA backbones across simulation and real-world benchmarks, our method yields up to $+18.7\%$ on real-world in-distribution tasks and $+21.5\%$ on out-of-distribution generalization.

1 Introduction

Vision-Language-Action (VLA) models build generalist robot policies by fine-tuning pretrained Vision-Language Models (VLM) on robot demonstrations, inheriting a richly structured representation space that, in principle, enables generalization across novel objects, scenes, and instructions [2, 3, 4]. In practice, however, demonstration data are orders of magnitude smaller and narrower than the

[*]Equal contribution.

VLM's pretraining corpus, and action-only supervision drives the model toward shortcut mappings that degrade the inherited semantic structure [5, 6, 7]: Fig. 1a shows that the clear task-level clustering present in pretrained representations collapses after fine-tuning. Existing efforts mitigate this by scaling up data [8, 9] or redesigning architectures [10, 11, 12], but they do so without a clear picture of what is being lost. A more fundamental question precedes both strategies: *what constitutes a good action representation, and how can we tell when it is degrading?*

What makes a motor representation support flexible generalization has been studied extensively in cognitive neuroscience. The mirror neuron theory [13, 14, 15] reveals that in primates, the same neurons fire when observing a goal-directed action and when executing it [16, 17]. Crucially, these neurons are organized around *intentions*: they respond differently to identical motor acts embedded in different goals, forming an intention-level "motor vocabulary" [18, 19]. This implies that biological motor representations encode not only movement patterns but also *what task the agent intends to accomplish*, and it is this intention-level organization that supports flexible generalization to novel situations.

We draw an analogy to robot learning. A VLA's action representations do not merely encode motor commands; they also carry the model's understanding of the current task, e.g., whether the robot is picking up a spoon, stacking a dish, or closing a cabinet. If this understanding remains well-organized, the model can transfer knowledge across similar tasks; if it degrades, the model falls back on superficial cues and loses generalization. Under the mirror neuron analogy, pretrained vision-language encoders can be viewed as a concrete instance of the "observation side": trained on large-scale multimodal data, they compress diverse visual and linguistic experience into semantic manifolds in which task-relevant concepts are well organized. The Platonic Representation Hypothesis [20] suggests that such manifolds reflect a convergent semantic organization rather than an idiosyncratic encoding. These manifolds provide a reference for examining the intention-level organization of VLA action representations on the "execution side."

We design a probing framework (§2) to test this idea. Tracking the action–semantic alignment across π_0's full fine-tuning trajectory, we find that action-only fine-tuning progressively erodes the intention-level structure inherited from pretraining, and that the surviving structure's quality synchronizes with both out-of-distribution generalization and per-trajectory rollout success.

Motivated by these findings, we propose a training-time method (§3) that contrastively anchors VLA mid-layer action representations to a well-structured semantic manifold (Fig. 1b). Because semantic alignment rewards invariance to surface variation while action prediction requires sensitivity to physical detail, we decompose each representation into a *shared* channel aligned to the semantic manifold and a *private* channel that preserves execution-specific detail. All auxiliary modules are discarded at inference, leaving the deployed model identical to the action-only baseline.

Our contributions are as follows:

- We propose the semantic structure of action representations as a measurable proxy for representation quality, and show via systematic probing that action-only fine-tuning erodes this structure while its quality predicts task success and out-of-distribution generalization.
- We propose a plug-and-play method that restores this structure at zero inference cost, combining contrastive alignment to a semantic manifold with shared/private decomposition.
- We validate on architecturally distinct VLA backbones across simulation and real-world benchmarks, showing up to $+18.7\%$ on real-world in-distribution tasks and $+21.5\%$ on OOD generalization.

2 Diagnostic: Action–Instruction Alignment Erosion

2.1 Probing setup

Action-only fine-tuning may erode the semantic structure inherited from pretraining. To track whether this occurs, we measure the alignment between action representations and instruction embeddings from a separately pretrained encoder across the full fine-tuning trajectory of π_0 [4] on LIBERO [1],

probing at checkpoints from the pretrained backbone through step 30k, and examining whether the trend predicts downstream performance.

Prior work [7, 21] shows that the instruction-relevant semantic structure peaks around the early-to-mid layers. We extract the layer-k=10 hidden states $\mathbf{h}^{(k)} \in \mathbb{R}^d$, out of N=18 backbone layers, and mean-pool over the action-token positions into $\bar{\mathbf{h}}^{(k)}$. Following the Platonic Representation Hypothesis [20], we use a separately pretrained encoder, the text encoder of Qwen3-VL-Embedding ($\mathbf{e} = g_{\text{probe}}(\ell)$), as a stable semantic reference, since the VLA's own language pathway also degrades during fine-tuning. To quantify agreement we adopt the alignment-probing protocol of Zhang et al. [22]: freeze both sides and train lightweight projection heads $\mathcal{T}_a, \mathcal{T}_t$ ($\hat{\mathbf{z}}_a = \mathcal{T}_a(\bar{\mathbf{h}}^{(k)})$, $\hat{\mathbf{z}}_t = \mathcal{T}_t(\mathbf{e})$) into a shared d_p-dim space on task-disjoint LIBERO pairs, using 8 tasks for training and the remaining 2 tasks for evaluation within each of the 4 suites, with the bidirectional InfoNCE objective [23],

$$\mathcal{L}_{\text{align}} = -\frac{1}{2B}\sum_{n=1}^{B}\left[\log\frac{\exp\big(\text{sim}(\hat{\mathbf{z}}_a^{(n)}, \hat{\mathbf{z}}_t^{(n)})/\tau\big)}{\sum_{m=1}^{B}\exp\big(\text{sim}(\hat{\mathbf{z}}_a^{(n)}, \hat{\mathbf{z}}_t^{(m)})/\tau\big)} + \log\frac{\exp\big(\text{sim}(\hat{\mathbf{z}}_t^{(n)}, \hat{\mathbf{z}}_a^{(n)})/\tau\big)}{\sum_{m=1}^{B}\exp\big(\text{sim}(\hat{\mathbf{z}}_t^{(n)}, \hat{\mathbf{z}}_a^{(m)})/\tau\big)}\right], \tag{1}$$

with sim cosine similarity, B the batch size, and τ a learnable temperature. We report the bidirectional Recall@1 under the trained heads as the action–instruction alignment at layer k.

2.2 Alignment dynamics over fine-tuning

We probe seven checkpoints at 5k-step intervals from the pretrained backbone to step 30k, measuring the layer-k alignment at each. For the six checkpoints with trained policies, steps 5k–30k, we additionally measure in-distribution success on LIBERO [1] and out-of-distribution success on LIBERO-Pro [24]. $\mathcal{T}_a, \mathcal{T}_t$ are retrained at every checkpoint.

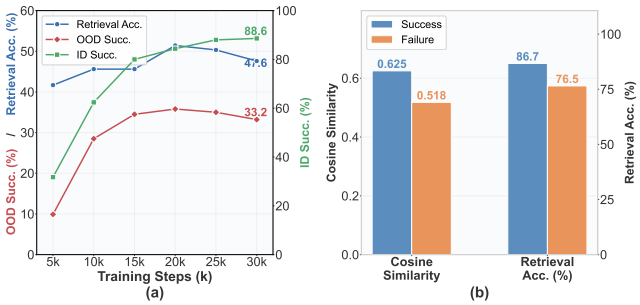

Figure 2: **Alignment probing results on LIBERO [1]. (a)** Action–instruction alignment (retrieval accuracy), ID success, and OOD success tracked over the π_0 fine-tuning trajectory. OOD success synchronizes with alignment, while ID success rises monotonically. **(b)** Per-trajectory cosine similarity and retrieval accuracy at step 30k, separated by rollout outcome; both metrics are systematically higher for successful rollouts.

Relative to the pretrained backbone, the alignment drops sharply early in fine-tuning, recovers only partially, declines again late in training, and never returns to its pretrained level of 62.74% (Fig. 2a, blue); the intention-level semantic structure inherited from pretraining is progressively eroding. In-distribution success, by contrast, rises almost monotonically (Fig. 2a, green). The two curves diverge because LIBERO's training and evaluation distributions are nearly identical, so in-distribution success can be attained by fitting distribution-specific shortcuts and does not faithfully reflect representation quality. Out-of-distribution success on LIBERO-Pro tells a different story: it follows the same rise-then-fall trajectory as the alignment (Fig. 2a, red; Spearman ρ=0.964). In summary, action-only fine-tuning progressively erodes the intention-level semantic structure inherited from pretraining, and the surviving structure's quality synchronizes with out-of-distribution generalization.

2.3 Per-trajectory alignment and rollout outcome

Beyond the trajectory-level trend, we ask whether alignment also predicts individual rollout outcomes. Using the step-30k policy on LIBERO, we compute per-trajectory cosine similarity between action features and the instruction embedding. Fig. 2b confirms that both cosine similarity and retrieval accuracy are systematically higher for successful than for failed rollouts. Together with the longitudinal trend above, these results motivate an explicit alignment objective: if semantic structure both tracks generalization and reflects per-episode success, actively preserving it during fine-tuning should yield better policies.

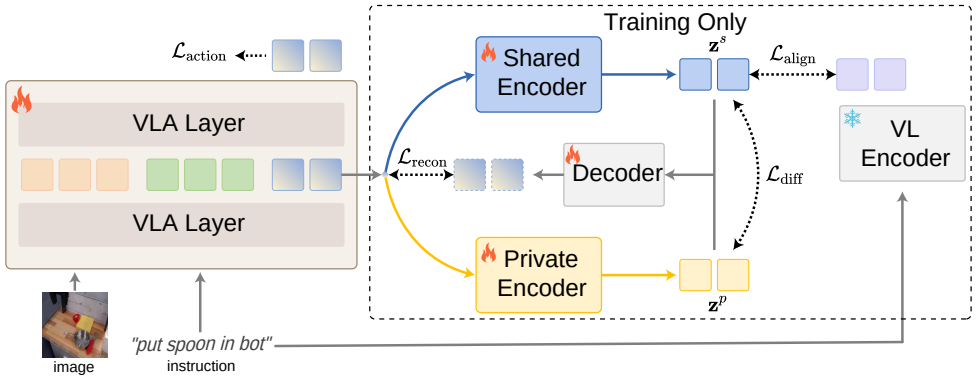

Figure 3: **Method overview.** A pretrained VLA maps visual observations and language instructions to actions. At a mid-network layer, we decompose each action-token representation into a shared component that encodes intention-level semantics and a private component that preserves execution-specific detail. The shared components are attention-pooled and contrastively aligned to the instruction embedding from a frozen encoder. All modules inside the dashed box are discarded at inference, leaving the deployed model unchanged.

3 Method

3.1 Problem formulation

A pretrained VLA f_θ maps a visual observation $\mathbf{o}_t$ and a natural-language instruction ℓ to an action $\mathbf{a}_t$. Standard post-training on demonstrations $\mathcal{D} = \{(\mathbf{o}_t, \ell, \mathbf{a}_t)\}$ minimises an action loss

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{\mathcal{D}}\Big[\, c\big(f_\theta(\mathbf{o}_t, \ell),\ \mathbf{a}_t\big)\Big], \tag{2}$$

where c is the backbone's native per-sample action cost, i.e., cross-entropy over discretised action tokens for autoregressive VLAs or a flow-matching regression cost for continuous VLAs [4]. As §2 shows, this objective provides no mechanism to preserve the inherited semantic structure. We therefore add an alignment objective whose gradient is channeled separately from the action loss, so as not to interfere with action prediction.

We extract the T action-token hidden states $\{\mathbf{h}_i^{(k)}\}_{i=1}^T$ at backbone layer k, each $\mathbf{h}_i^{(k)} \in \mathbb{R}^d$. The alignment target is the instruction embedding $\mathbf{e} = g_{\text{enc}}(\ell)$ from the text encoder of a frozen vision-language model EgoHOD [25], whose output space serves as a concrete proxy for the semantic manifold introduced in §1 [20]. We denote this encoder as g_{enc}; §4.3 ablates the encoder choice.

3.2 Anchoring action features to the semantic manifold

We aggregate the action-token hidden states into a sentence-level representation via cross-attention with a learnable query $\mathbf{q} \in \mathbb{R}^d$:

$$\alpha_i = \frac{\exp(\mathbf{q}^\top \mathbf{h}_i^{(k)}/\sqrt{d})}{\sum_{m=1}^T \exp(\mathbf{q}^\top \mathbf{h}_m^{(k)}/\sqrt{d})}, \qquad \bar{\mathbf{h}}^{(k)} = \sum_{i=1}^T \alpha_i\, \mathbf{h}_i^{(k)}. \tag{3}$$

Two-layer MLP heads $\mathcal{T}_a, \mathcal{T}_t$ project $\bar{\mathbf{h}}^{(k)}$ and $\mathbf{e}$ into a shared d_p-dimensional space: $\hat{\mathbf{z}}_a = \mathcal{T}_a(\bar{\mathbf{h}}^{(k)})$, $\hat{\mathbf{z}}_t = \mathcal{T}_t(\mathbf{e})$. We apply the same bidirectional InfoNCE loss $\mathcal{L}_{\text{align}}$ as in Eq. 1; the alignment gradient flows through $\mathcal{T}_a$ into the backbone at layer k, actively shaping the representation. As presented, the attention pools all per-token features; the next section refines this by routing only the intention-level component through the attention and into $\mathcal{L}_{\text{align}}$.

3.3 Shared–private decomposition

Semantic alignment rewards *invariance* to surface variation, while action prediction requires *sensitivity* to physical detail, opposing requirements that loss weighting alone cannot resolve. Inspired by the separation of goal-level and effector-specific coding in the mirror-neuron circuit [14, 15], we decompose each action-token feature *before* attention pooling into a *shared* channel for intention-level semantics and a *private* channel that captures the execution-specific residual, including embodiment

geometry, timing, and motor detail, needed for action prediction but orthogonal to task semantics. The decomposition follows Domain Separation Networks [26].

Concretely, two-layer MLP encoders decompose each per-token feature $\mathbf{h}_i^{(k)}$ into a d_s-dimensional shared representation $\mathbf{z}_i^s$ and a d_s-dimensional private representation $\mathbf{z}_i^p$; a decoder produces the reconstruction $\mathbf{h}_{\mathrm{rec}\,i}$ additively:

$$\mathbf{z}_i^s = E_s(\mathbf{h}_i^{(k)}), \qquad \mathbf{z}_i^p = E_p(\mathbf{h}_i^{(k)}), \qquad \mathbf{h}_{\mathrm{rec}\,i} = \mathrm{Dec}(\mathbf{z}_i^s + \mathbf{z}_i^p). \tag{4}$$

Only the shared channels enter the attention pooling of Eq. 3, with $\mathbf{z}_i^s$ replacing $\mathbf{h}_i^{(k)}$ and the query now $\mathbf{q} \in \mathbb{R}^{d_s}$ matching the shared-channel dimension, so the alignment loss $\mathcal{L}_{\text{align}}$ receives an intention-level aggregate free of private-channel detail:

$$\alpha_i = \frac{\exp(\mathbf{q}^\top \mathbf{z}_i^s/\sqrt{d_s})}{\sum_{m=1}^{T}\exp(\mathbf{q}^\top \mathbf{z}_m^s/\sqrt{d_s})}, \qquad \bar{\mathbf{h}}^{(k)} = \sum_{i=1}^{T} \alpha_i\, \mathbf{z}_i^s. \tag{5}$$

A reconstruction loss ensures the two channels jointly preserve $\mathbf{h}_i^{(k)}$, combining MSE with a scale-invariant term [26]:

$$\mathcal{L}_{\text{recon}} = \frac{1}{T}\sum_{i=1}^{T}\left[\, \tfrac{1}{d}\big\|\mathbf{h}_{\mathrm{rec}\,i} - \mathbf{h}_i^{(k)}\big\|_2^2 + \tfrac{1}{d^2}\Big(\sum_{j=1}^{d}\big(h_{\mathrm{rec},ij} - h_{ij}\big)\Big)^2 \right]. \tag{6}$$

An angular decorrelation penalty keeps the two channels from re-entangling:

$$\mathcal{L}_{\text{diff}} = \tfrac{1}{BT}\sum_{n=1}^{B}\sum_{i=1}^{T}\Big(\tfrac{\langle \mathbf{z}_{n,i}^s, \mathbf{z}_{n,i}^p\rangle}{\|\mathbf{z}_{n,i}^s\|\,\|\mathbf{z}_{n,i}^p\|}\Big)^2. \tag{7}$$

3.4 Training objective

The full objective combines the action loss with the three alignment terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} + \lambda_{\text{align}}\,\mathcal{L}_{\text{align}} + \lambda_r\,\mathcal{L}_{\text{recon}} + \lambda_d\,\mathcal{L}_{\text{diff}}, \tag{8}$$

with λ_r=0.01, λ_d=0.075. λ_{align} is backbone-dependent because autoregressive and flow-matching action heads produce gradients at different scales; concrete values are given in §4. All terms are computed over the demonstrations $\mathcal{D}$ and require no additional data at training time. At deployment all auxiliary modules are dropped, so the inference graph is identical to the action-only baseline [27, 28].

4 Experiments

We design experiments to answer: (RQ1) Does the alignment objective deliver consistent gains across simulation (§4.1) and a real bimanual robot (§4.2)? (RQ2) Does the gain transfer across architecturally distinct VLA backbones (§4.1)? (RQ3) Which design choices drive the gain (§4.3)?

4.1 Simulation Experiments

Setup. We evaluate on two simulation benchmarks that together cover small-data in-domain capability and large-scale cross-domain transfer. LIBERO [1] contains four task suites with 50 demonstrations per task and tests near-distribution performance. SimplerEnv [29] fine-tunes backbones on BridgeData V2 [30], a large-scale, multi-source real-robot dataset, and evaluates WidowX manipulation tasks under the Visual Matching protocol.

We test on two architecturally distinct backbones: π_0 [4], a flow-matching VLA that we fully fine-tune, and SpatialVLA [11], an autoregressive VLA that we fine-tune with LoRA. We follow the official recipe of each backbone for the action objective and add only our alignment objective. The frozen alignment target is the text encoder of EgoHOD [25], applied at layer k=10 for both π_0 (18 layers) and SpatialVLA (27 layers). The alignment weight λ_{align} is ≈ 0.1 for π_0 and ≈ 0.5 for SpatialVLA. All alignment modules are discarded at inference, so the deployed policy is identical to the action-only baseline. Detailed hyperparameters are in the supplementary material.

Method	Spatial	Object	Goal	Long	Avg
OpenVLA [3]	84.7	88.4	79.2	53.7	76.5
Octo [10]	78.9	85.7	84.6	51.1	75.1
π_0-FAST [31]	96.4	96.8	88.6	60.2	85.5
SmolVLA [32]	90.0	96.0	92.0	71.0	87.3
π_0 [4]	94.0	96.5	90.0	76.5	89.3
π_0 + Ours	**96.5**	**98.5**	**92.5**	**82.0**	**92.4**$_{\uparrow 3.1}$

(a) LIBERO [1]. **Bold**: improvement over the baseline.

Method	spoon	carrot	stack	eggplant	Avg
Octo [10]	47.2	9.7	4.2	56.9	30.0
π_0-FAST [31]	29.1	21.9	10.8	66.6	32.1
π_0 [4]	37.5	33.3	25.0	45.8	35.4
π_0 + Ours	**45.8**	**37.5**	**29.2**	**54.2**	**41.7**$_{\uparrow 6.3}$
SpatialVLA [11]	20.8	25.0	29.2	100.0	43.8
SpatialVLA + Ours	**29.2**	**41.7**	**33.3**	**100.0**	**51.0**$_{\uparrow 7.2}$

(b) SimplerEnv [29]. **Bold**: improvement over the respective baseline.

Table 1: Simulation success rates (%) on LIBERO [1] and SimplerEnv [29]. Semantic anchoring improves performance across benchmarks and VLA backbones.

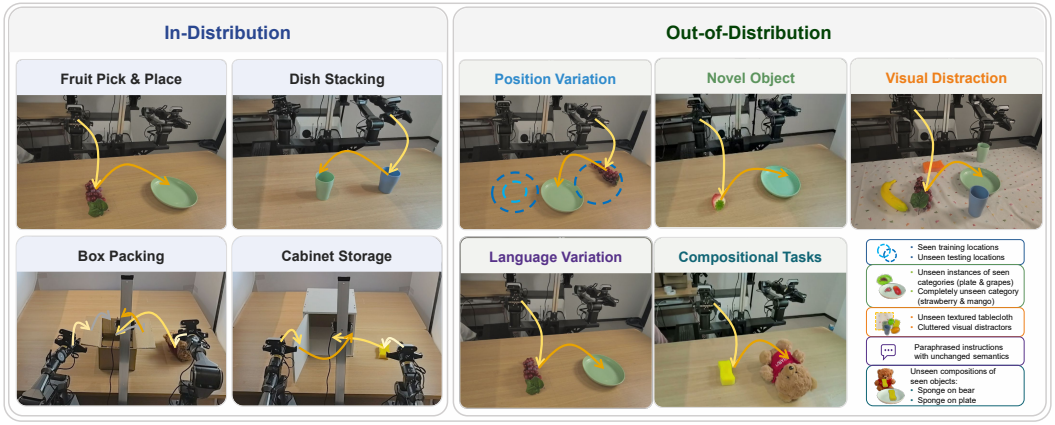

Figure 4: Real-robot experimental setup. **Left:** four in-distribution task families (two single-stage, two with articulated closure). **Middle:** five out-of-distribution axes built on the pick-and-place family. **Right:** average success rates; our method improves over the π_0 baseline by $+18.7\%$ (ID) and $+21.5\%$ (OOD).

Simulation results. Table 1 summarizes the results. On LIBERO, our alignment objective improves accuracy on every suite despite only 50 demonstrations per task, indicating that the alignment objective does not impair action prediction and provides complementary gains even in near-distribution settings. The t-SNE visualizations on LIBERO-Goal (Fig. 1a,c) confirm this. On SimplerEnv, which evaluates under a real-to-sim domain shift, gains are larger on both backbones, likely because BridgeData V2's richer task diversity gives the alignment more semantic material to anchor onto. We further evaluate on LIBERO-Pro [24], an extended perturbation benchmark built on LIBERO, and find that our method consistently improves π_0 on every perturbation axis; full results are in the supplementary material.

4.2 Real-Robot Experiments

Setup. The platform is a bimanual AgileX Cobot Magic V2 with two 6-DoF arms, parallel grippers, and three RGB cameras, including one front-view and two wrist-mounted. We design four task families with two variants each (Fig. 4, left): *fruit pick-and-place* and *dish stacking* are single-stage tasks, while *box packing* and *cabinet storage* each chain object placement with an articulated closure by closing the box flaps or the cabinet door, requiring longer-horizon coordination.

We evaluate generalization along five axes on the pick-and-place family (Fig. 4, middle): *Language*, which uses paraphrased instructions; *Position*, with unseen layouts; *Object*, introducing novel objects; *Visual*, adding new backgrounds and distractors; and *Task*, which tests unseen source–target combinations of objects seen during training.

We collect 200 teleoperated trajectories per task and fully fine-tune a single multi-task π_0 policy; our method shares this protocol and adds the alignment objective at the same mid-network layer as in simulation. Baselines include the same multi-task π_0 trained under the same protocol, as well as per-task ACT [33] and Diffusion Policy [34], trained individually because their multi-task variants did

Method	In-Distribution					Out-of-Distribution						
	Fruit Pick&Place	Dish Stacking	Box Packing	Cabinet Storage	Avg	Orig.	Lang.	Pos.	Obj.	Vis.	Task	Avg
ACT	82.5	45.0	25.0	47.5	50.0	82.5	–	7.5	40.0	45.0	–	–
DP	80.0	47.5	32.5	42.5	50.6	80.0	–	12.5	32.5	35.0	–	–
π_0	75.0	42.5	37.5	50.0	51.3	75.0	75.0	32.5	52.5	57.5	30.0	49.5
π_0 + Ours	**92.5**	**67.5**	**52.5**	**67.5**	**70.0**	**92.5**	**85.0**	**57.5**	**77.5**	**75.0**	**60.0**	**71.0**

Table 2: Real-robot success rate (%, 20 trials each). **Left:** in-distribution across four task families. **Right:** out-of-distribution generalization evaluated on the pick-and-place family.

not converge in our setup. Because ACT and Diffusion Policy are trained per task without language conditioning, the Language and Task axes are not applicable in Table 2. Each task is evaluated for 20 trials. Full hyperparameters are provided in the supplementary material.

Results. Table 2 summarises both in-distribution and out-of-distribution results. Our method consistently improves π_0 across all four task families. For out-of-distribution generalization, improvements are again consistent across all five axes, with the largest gains on Position and Task, the two most challenging axes that demand spatial and compositional reasoning beyond memorised demonstrations. On the Task axis, which requires executing unseen source–target combinations of objects seen during training, π_0 frequently fails to interpret the novel composition, stalling or grasping irrelevant objects, whereas our method correctly identifies the target and selects the appropriate arm; see the supplementary material for qualitative comparisons.

4.3 Ablation and Analysis

We ablate one design choice at a time on SpatialVLA evaluated in SimplerEnv, fixing the training protocol of Table 1b.

Method components. We incrementally add each module to isolate its contribution: contrastive alignment alone vs. the full method with shared/private decomposition. As shown in Fig. 5a, each component contributes incrementally. Applying contrastive alignment alone, + *Align*, already lifts the baseline, confirming that anchoring action features to a semantic reference is beneficial even without explicit channel separation. Adding the shared/private decomposition, + *DSN*, further improves the average, indicating that decomposing intention-level semantics from execution-specific detail provides a structural prior that strengthens alignment beyond what the contrastive loss alone can achieve.

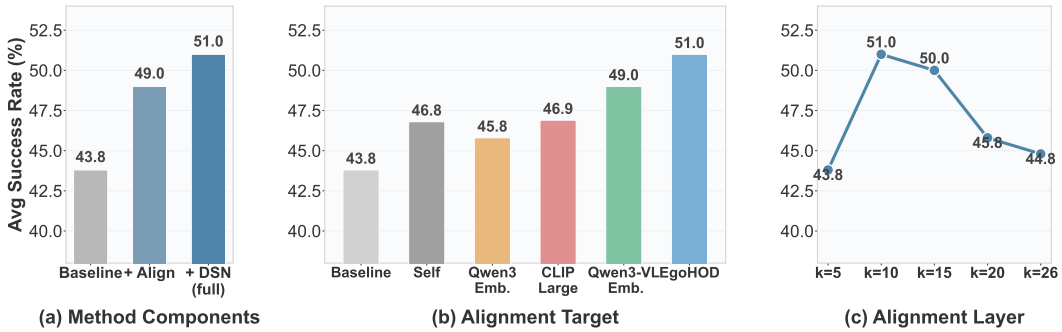

Figure 5: **Ablation study** with SpatialVLA [11] on SimplerEnv [29]. **(a)** Method components: + *Align* applies contrastive alignment without decomposition; + *DSN* adds shared/private decomposition. **(b)** Alignment target: *Self* contrastively aligns same-task action features without an external encoder; the remaining bars use different pretrained encoders. **(c)** Performance by alignment layer k.

Alignment target. We vary the frozen encoder g_{enc} to examine the impact of the alignment target (Fig. 5b). The *Self* variant contrastively clusters same-task action features without an external encoder; it improves over the baseline but falls well below EgoHOD, indicating that anchoring to a semantically rich external space provides substantially stronger guidance than task-identity clustering

alone. Among external encoders, visual grounding is critical: the language-only Qwen3-Embedding underperforms every visually grounded alternative. Within the visually grounded group, CLIP-Large, trained on static image-text pairs, underperforms both Qwen3-VL-Embedding and EgoHOD, which incorporate temporal video structure; since EgoHOD shares the same text encoder as CLIP-Large, the gap isolates the contribution of visual dynamics. Among the two video-grounded encoders, EgoHOD outperforms the larger Qwen3-VL-Embedding, confirming that domain-specific pretraining on manipulation videos matters more than model scale.

Alignment layer. We sweep the backbone layer k at which the alignment objective is attached to identify where semantic structure is richest. Fig. 5c shows results across the full depth. Mid-network layers yield the largest gains, with performance peaking at $k{=}10$ and remaining strong at $k{=}15$; both early and late layers diminish the benefit. This is consistent with prior findings that mid-layer representations retain the richest semantic structure, while later layers increasingly specialise for action prediction [35, 7]. We set $k{=}10$ for both backbones.

5 Related Work

Vision-Language-Action Models VLA models fine-tune pretrained VLMs on robot demonstrations, inheriting semantic representations that enable generalization [2, 3, 4]. The paradigm has been extended with generalist diffusion heads [10], flow-matching decoders [4], 3D spatial conditioning [11], and cross-embodiment trunks [12]. A recurring observation is that action supervision on narrow data distorts pretrained features [36, 37], inducing shortcut learning [5, 6, 7]. Existing remedies scale data [8, 9], redesign architectures [11, 10], or add auxiliary objectives [35, 7]. Distinct from these approaches, our method does not scale data or modify the model architecture, but instead establishes alignment with a semantic manifold as a measurable criterion for representation quality and repairs the eroded structure through a training-time-only objective.

Representation Alignment Cross-modal contrastive alignment, exemplified by CLIP [38], has been widely used to learn transferable representations, with the Platonic Representation Hypothesis [20] providing theoretical motivation. REPA [27] aligns diffusion-transformer hidden states to a frozen encoder at training time only; our method inherits this design but adapts it to VLAs with a manipulation-centric encoder and a shared/private decomposition that resolves the gradient conflict between semantic alignment and action prediction. In the embodied domain, contrastive pretraining on egocentric video has yielded reusable representations [39, 40, 41], and EgoHOD [25] captures fine-grained hand–object dynamics that we leverage as a frozen semantic reference. Concurrent work preserves VLA representations via visual-pathway regularisation [7], self-feature contrastive objectives [35], or spatial alignment [28]; our approach differs in targeting action-token representations and anchoring to an external semantic manifold. Zhu et al. [42] share our mirror-neuron motivation in an embodied setting; our work focuses on the VLA fine-tuning regime with diagnostic evidence linking alignment quality to performance.

6 Conclusion, Limitations and Future Directions

We show that action-only fine-tuning of VLAs erodes the intention-level semantic structure inherited from pretraining, and that this structure's quality synchronizes with generalization. Guided by this finding and the mirror neuron theory, we propose a plug-and-play method that anchors action representations to a semantic manifold via shared/private decomposition, all discarded at inference. Experiments on various VLA backbones across simulation and real-world settings validate consistent improvements in both in-distribution and out-of-distribution settings. Despite these promising results, two limitations remain. First, performance is upper-bounded by the quality of the frozen alignment target; a stronger manipulation-centric encoder would likely yield further gains. Second, we apply the objective only during post-training on task-specific demonstrations. Future work will explore extending semantic alignment to the pre-training stage and scaling to broader data sources toward more generalizable robotic manipulation.

References

[1] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36: 44776–44791, 2023.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, pages 2679–2713, 2024.

[4] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[6] Y. Xing, X. Luo, J. Xie, L. Gao, H. Shen, and J. Song. Shortcut learning in generalist robot policies: The role of dataset diversity and fragmentation. In *Conference on Robot Learning (CoRL)*, 2025.

[7] N. Kachaev, M. Kolosov, D. Zelezetsky, A. K. Kovalev, and A. I. Panov. Don't blind your VLA: Aligning visual representations for OOD generalization. *arXiv preprint arXiv:2510.25616*, 2025.

[8] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, pages 17–40, 2025.

[9] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[10] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

[11] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, J. Gu, Z. Wang, Y. Ding, B. Zhao, D. Wang, and X. Li. Spatialvla: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems (RSS)*, 2025.

[12] L. Wang, X. Chen, J. Zhao, and K. He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[13] G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91(1):176–180, 1992.

[14] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

[15] G. Rizzolatti and L. Craighero. The mirror-neuron system. *Annual Review of Neuroscience*, 27: 169–192, 2004.

[16] L. Fadiga, L. Fogassi, G. Pavesi, and G. Rizzolatti. Motor facilitation during action observation: a magnetic stimulation study. *Journal of Neurophysiology*, 73(6):2608–2611, 1995.

[17] C. Keysers and V. Gazzola. Expanding the mirror: vicarious activity for actions, emotions, and sensations. *Current Opinion in Neurobiology*, 19(6):666–671, 2009.

[18] L. Fogassi, P. F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti. Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667, 2005.

[19] G. Rizzolatti and C. Sinigaglia. The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations. *Nature Reviews Neuroscience*, 11(4):264–274, 2010.

[20] M. Huh, B. Cheung, T. Wang, and P. Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pages 20617–20642. PMLR, 2024.

[21] M. Reuss, H. Zhou, M. Rühle, Ö. E. Yağmurlu, F. Otto, and R. Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-action flow policies. In *Conference on Robot Learning (CoRL)*, 2025. arXiv:2509.04996.

[22] L. Zhang, Q. Yang, and A. Agrawal. Assessing and learning alignment of unimodal vision and language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14604–14614, 2025.

[23] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[24] X. Zhou, Y. Xu, G. Tie, Y. Chen, G. Zhang, D. Chu, P. Zhou, and L. Sun. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

[25] B. Pei, Y. Huang, J. Xu, G. Chen, Y. He, L. Yang, Y. Wang, W. Xie, Y. Qiao, F. Wu, and L. Wang. Modeling fine-grained hand-object dynamics for egocentric video representation learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[26] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, 2016.

[27] S. Yu, S. Kwak, H. Jang, J. Jeong, J. Huang, J. Shin, and S. Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations (ICLR)*, 2025.

[28] F. Li, W. Song, H. Zhao, J. Wang, P. Ding, D. Wang, L. Zeng, and H. Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. *arXiv preprint arXiv:2510.12276*, 2025.

[29] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning (CoRL)*, 2024.

[30] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[31] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. FAST: Efficient action tokenization for vision-language-action models. In *Robotics: Science and Systems (RSS)*, 2025.

[32] M. Shukor et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[33] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

[34] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[35] T. Kim, J. Lee, M. Koo, D. Kim, K. Lee, C. Kim, Y. Seo, and J. Shin. Contrastive representation regularization for vision-language-action models. *arXiv preprint arXiv:2510.01711*, 2025.

[36] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang. Fine-tuning can distort pre-trained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

[37] A. Aghajanyan, A. Shrivastava, A. Gupta, N. Goyal, L. Zettlemoyer, and S. Gupta. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*, 2021.

[38] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[39] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3M: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, pages 892–909, 2022.

[40] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

[41] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang. Language-driven representation learning for robotics. In *Robotics: Science and Systems (RSS)*, 2023.

[42] W. Zhu, Z. Zhang, Y. Ren, Y. Huang, H. Xu, and Y. Wang. Embodied representation alignment with mirror neurons. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[43] M. Li, Y. Zhang, D. Long, K. Chen, S. Song, S. Bai, Z. Yang, P. Xie, A. Yang, D. Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

A Implementation Details

A.1 Backbone Training Hyperparameters

We follow the official training recipe of each backbone and add only our alignment objective. Table 3 lists backbone-specific hyperparameters for all four experimental settings.

π_0 [4] uses PaliGemma-2B as the vision-language backbone and Gemma-300M as the action expert. All parameters are fully fine-tuned. On LIBERO, we train for 30k steps with a peak learning rate of 2.5×10^{-5} and a batch size of 32. On BridgeData V2, we train for 50k steps with a peak learning rate of 5×10^{-5} and a batch size of 128. For the real robot, we train a single multi-task policy on all eight task variants for 30k steps with a peak learning rate of 5×10^{-5} and a batch size of 64.

SpatialVLA [11] is a 4B-parameter autoregressive VLA. On BridgeData V2, we fine-tune it with LoRA of rank 32 and α=32 applied to all linear layers for 2 epochs, totalling about 22k steps, with a peak learning rate of 10^{-4}. We use DeepSpeed ZeRO Stage-1 for memory efficiency.

All experiments use AdamW with gradient clipping at norm 1.0 and bfloat16 mixed precision.

	π_0 (LIBERO)	π_0 (BridgeV2)	SpatialVLA (BridgeV2)	π_0 (Real)
Fine-tuning	Full	Full	LoRA (r=32)	Full
Backbone LR	2.5×10^{-5}	5×10^{-5}	10^{-4}	5×10^{-5}
Training steps	30k	50k	≈22k (2 ep.)	30k
Batch size	32	128	192	64
β_1, β_2	0.9, 0.95	0.9, 0.95	0.9, 0.999	0.9, 0.95
Weight decay	10^{-10}	10^{-10}	0	10^{-10}
Action horizon	50	5	4	50
VLM frozen	–	–	LoRA only	–
GPUs	2	2	2	2

Table 3: Backbone training hyperparameters.

A.2 Alignment Module

At training time, the alignment module contains shared/private encoders, a reconstruction decoder, a shared-channel attention-pooling block, and lightweight projection heads. Given each backbone feature $\mathbf{h}_i^{(k)} \in \mathbb{R}^d$, the encoders E_s and E_p produce shared and private features $\mathbf{z}_i^s, \mathbf{z}_i^p \in \mathbb{R}^{d_s}$. Both channels use the same subspace dimension d_s, allowing the decoder Dec to reconstruct the original feature from their elementwise sum $\mathbf{z}_i^s + \mathbf{z}_i^p$. Only the shared tokens $\{\mathbf{z}_i^s\}_{i=1}^T$ are passed to the attention-pooling block, where a learnable query $\mathbf{q} \in \mathbb{R}^{d_s}$ aggregates them into a sentence-level representation. The pooled action feature and the frozen 768-dimensional EgoHOD-Large text embedding are then projected into the shared contrastive space by lightweight heads $\mathcal{T}_a$ and $\mathcal{T}_t$. The decorrelation term $\mathcal{L}_{\text{diff}}$ is the mean squared cosine similarity between $\mathbf{z}^s$ and $\mathbf{z}^p$. Both backbones use layer 10 for alignment, a 512-dimensional subspace, a 512-dimensional projection space, reconstruction weight 0.01, decorrelation weight 0.075, weight decay 0.01, and learnable-query attention with eight heads before pooling. The backbone feature dimension is 1024 for π_0 and 2304 for SpatialVLA. The alignment weight is about 0.1 for π_0 and about 0.5 for SpatialVLA, and the alignment learning rate is 10^{-4} for π_0 and 5×10^{-5} for SpatialVLA.

A.3 Diagnostic Probing Protocol

The diagnostic probe uses the text encoder of Qwen3-VL-Embedding [43] as the frozen semantic reference g_{probe}, deliberately choosing a different model from the EgoHOD alignment target g_{enc} to avoid circularity. We extract layer-10 hidden states and mean-pool over the T action-token positions. Two-layer MLP projection heads map the 1024-dimensional action features to a 512-dimensional contrastive space. They are trained with bidirectional InfoNCE on task-disjoint LIBERO pairs, using 8 tasks for training and the remaining 2 tasks for evaluation in each of the four suites. We retrain the heads independently at seven checkpoints: the pretrained backbone and checkpoints from 5k to 30k

steps at 5k intervals. Each probe is trained with AdamW for 1k optimisation steps using learning rate 10^{-4} and weight decay 0.01. We report bidirectional Recall@1 as the alignment metric.

A.4 Real-Robot Platform

The platform is a bimanual AgileX Cobot Magic V2 with two 6-DoF arms and parallel grippers, giving 14 total DoF or 7 per arm including the gripper. Three RGB cameras provide the visual input at 256×256 resolution: one front-view overhead camera and two wrist-mounted cameras. The action space consists of 14-dimensional joint positions, normalised per-dimension using dataset statistics. We collect 200 teleoperated demonstrations per task variant across four families, giving eight variants and about 1,600 trajectories in total. A single multi-task π_0 policy is trained on all variants. Each condition is evaluated over 20 trials.

Baseline training. ACT [33] and Diffusion Policy [34] are trained individually per task, as their multi-task variants did not converge in our bimanual setup. ACT uses a Transformer encoder–decoder with hidden dimension 512, 8 attention heads, 4 encoder layers, 7 decoder layers, a prediction horizon of 50, and dropout 0.1. We train it for 100k steps with batch size 64 and learning rate 10^{-4}. Diffusion Policy is trained for 30k steps with batch size 16 and learning rate 10^{-4}. Its action encoder and decoder both use dimension 14. Both baselines use the same 200 demonstrations per task as π_0.

A.5 Real-Robot Task Descriptions

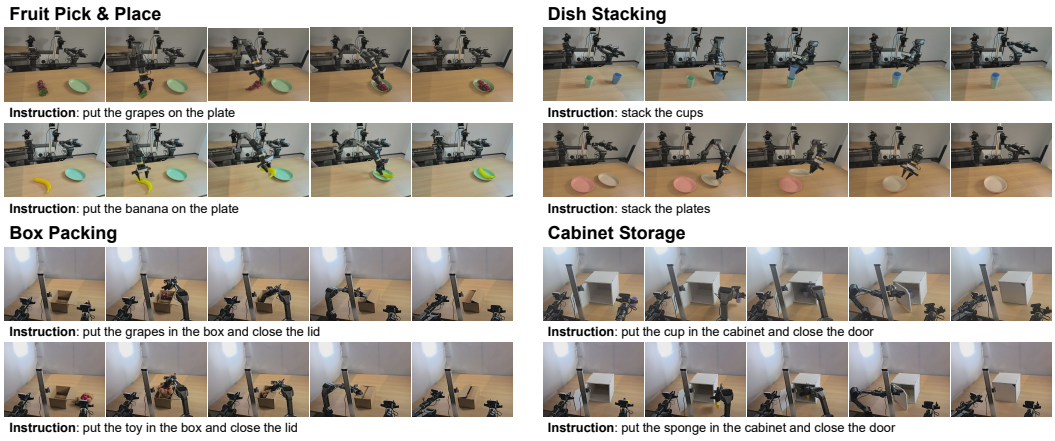

Figure 6: Four real-world bimanual task families with two variants each.

Fig. 6 illustrates the four task families with two variants each. *Fruit pick-and-place* and *dish stacking* are single-stage tasks, while *box packing* and *cabinet storage* each combine object placement with an articulated closure. Table 4 provides detailed task descriptions and evaluation protocols for all eight variants.

Family	Variant	Task Description	Success Criterion
Fruit Pick-and-Place	Banana	Pick up the banana and place it on the plate.	Fruit rests stably on the plate.
	Grapes	Pick up the grapes and place it on the plate.	Fruit rests stably on the plate.
Dish Stacking	Cup	Stack the cup onto the plate.	Cup rests stably on the plate.
	Plate	Stack the plate onto the other plate.	Plate rests stably on the bottom plate.
Box Packing	Grapes	Place the grapes into the box and close both side flaps.	Object inside the box and both side flaps fully closed.
	Bear	Place the toy bear into the box and close both side flaps.	Object inside the box and both side flaps fully closed.
Cabinet Storage	Cup	Place the cup in the cabinet and close the door.	Object inside the cabinet and door closed.
	Sponge	Place the sponge in the cabinet and close the door.	Object inside the cabinet and door closed.

Table 4: Real-world task descriptions and evaluation protocols. Each variant is evaluated over 20 trials with 200 teleoperated demonstrations collected per variant.

A.6 OOD Evaluation Axes

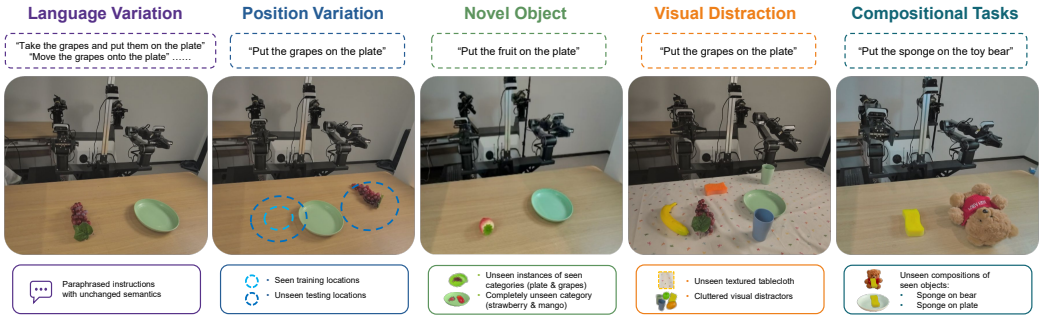

Figure 7: Five out-of-distribution generalization axes on the pick-and-place family.

All five OOD axes are built on the fruit pick-and-place family (Fig. 7). Table 5 details the specific perturbations applied in each axis.

Axis	Description
Language	The task instruction is paraphrased into semantically equivalent but lexically different expressions not seen during training.
Position	Objects are placed at positions beyond the spatial range seen during training.
Object	Both in-category and out-of-category perturbations are applied. In-category: the same object type with different attributes (e.g., plates of different shapes, grapes of different sizes). Out-of-category: entirely novel objects not present in any training demonstration (e.g., mango, strawberry).
Visual	The scene background is changed by placing unseen tablecloths, and distractor objects not present during training are added to the workspace.
Task	The policy must execute unseen source–target combinations of objects seen during training.

Table 5: Detailed descriptions of the five out-of-distribution evaluation axes.

B Additional Analyses

B.1 LIBERO-Pro Results

LIBERO-Pro tests perturbation generalization across five axes: language paraphrase, object swap, task composition, environment change, and object replacement. Table 6 shows the results.

Method	lan	swap	task	env	object	Avg
π_0	61.0	1.2	6.8	44.8	66.0	36.0
π_0 + Ours	**65.0**	**2.0**	**15.2**	**48.2**	**72.2**	**40.5**

Table 6: LIBERO-Pro success rate (%). The same trained checkpoint is evaluated without additional training.

B.2 Qualitative Comparisons

Fig. 8 compares rollouts on the Task generalization axis. Given an unseen source–target combination, the π_0 baseline selects a suboptimal arm and grasps an irrelevant object. Our method correctly identifies the target and chooses the appropriate arm, demonstrating compositional generalization.

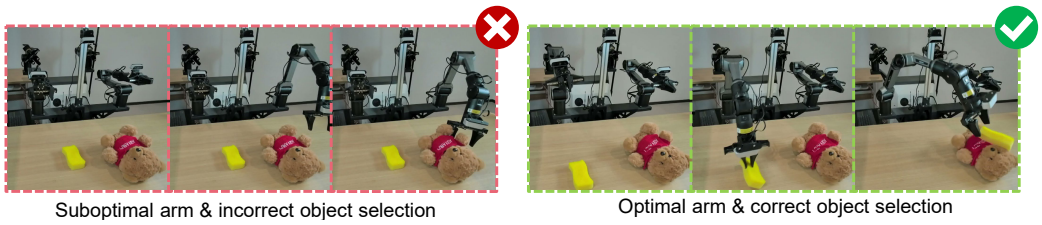

Figure 8: Qualitative comparison on the Task axis. **Left:** π_0 baseline (failure). **Right:** Ours (success).