SPURDIAG-BENCH: DIAGNOSING SPURIOUS CORRELATIONS IN VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite strong benchmark performance, there is growing consensus that many Vision-Language-Action (VLA) models still fail in real-world scenarios, suggesting that benchmark success overstates their real-world generalization. While most previous work attributes it to a data-scale problem, we observe that it can also stem from a **spurious correlation** phenomenon: through shortcut learning, models associate targets with incidental cues rather than desirable semantics and task logic. To systematically diagnose this phenomenon, we introduce **SpurDiag-Bench**. SpurDiag-Bench distinguishes itself from existing VLA benchmarks through diagnostic depth: it revisits known classic VLA tasks at fine granularity and reveals that **VLA failures do not require harder/broader tasks** – simple changes in task configuration alone are sufficient to expose failures hidden behind strong benchmark performance. Precisely, we introduce a chain analysis that decomposes the model along its perception-reasoning-action chain and applies causal interventions to separately examine the grounding of visual-language and actions. Across experiments on 12 state-of-the-art open-source VLAs, we report that spurious correlations **are pervasive** (even substantial for the best-performing model, $\pi_{0.5}$). We further fine-tune on representative models, investigating the **mitigation** of spurious correlations: some models become less reliant, while others retain persistent shortcut reliance. These findings caution against assuming that shortcut reliance can be readily corrected through additional training. Key properties are also identified in robust VLA models featured for countering spurious correlations (e.g., action-expert architecture, diverse pretraining), providing guidance for future model design for the community. The benchmark and source code will be released, and the leaderboard is available at `https://vla-page.netlify.app`.

1 INTRODUCTION

Vision-Language-Action (VLA) models pursue a central ambition of embodied intelligence: turning knowledge acquired through large-scale pretraining into generalizable, goal-directed behavior in the physical world (Kim et al., 2024; Zitkovich et al., 2023; Black et al., 2025b;a; Cen et al., 2025; Sun et al., 2026). Recent advances have yielded increasingly strong performance on standard robotic manipulation benchmarks (Liu et al., 2023; Mu et al., 2021), with several leading models reporting near-perfect success rates on established suites such as LIBERO (e.g., Zheng et al., 2026).

Despite these impressive results, there is broad consensus within the community that current VLAs still struggle to generalize reliably in real-world settings. As shown in Fig. 1, previous VLA benchmarks increasingly expand the complexity and scope of the evaluation, but still rely largely on task-level performance to examine models' generalizability (Zhang et al., 2026; Fei et al., 2026; Li et al., 2022). These benchmarks converge on the same implicit assumption that generalization is primarily a data-scale problem, addressable through broader task and data coverage.

In this work, we suggest that VLA failures are not merely a consequence of insufficient data scale, but can also arise from **spurious correlations**, an issue that inherently exists in vision-language models: models associate targets with incidental cues, such as object positions, colors, background layouts, or stereotyped trajectories, instead of grounding them in object semantics and task logic. Related symptoms have been studied from different perspectives, including language priors that override visual evidence in multimodal models (Wang et al., 2025a), "shortcut learning" (Geirhos et al., 2020),

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.

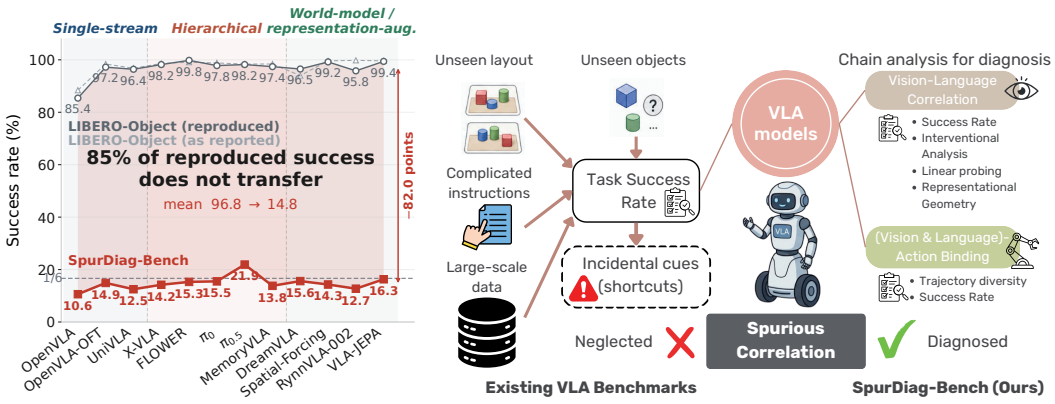


Figure 1: **(Left)** Performance Comparison between LIBERO-Object and SpurDiag-Bench. Across 12 VLA models with various architectures, performance drops from a mean success rate of 96.8% on LIBERO-Object to 14.8% on SpurDiag-Bench, revealing that high task success can mask substantial reliance on spurious correlations. **(Right)** Existing VLA benchmarks merely evaluate task success rate and neglect the potential shortcuts. SpurDiag-Bench uses chain analysis for deeper diagnosis.

Benchmark	Sim.	Breadth						Depth: mechanistic diagnosis				
		Real	Obj	Spat	Vis	Lang	Task	Stage	Probe	Causal	Traj.	Ref. GT
COLOSSEUM (Pumacay et al., 2024)	RLBench	✓	●	○	●	○	○	○	○	○	○	○
AGNOSTOS (Zhou et al., 2025a)	RLBench	✓	●	○	○	○	●	○	○	○	○	○
INT-ACT (Fang et al., 2025)	ManiSkill	×	●	●	◐	●	◐	◐	○	○	○	○
LIBERO-Plus (Fei et al., 2026)	LIBERO	×	●	●	●	●	○	◐	○	○	○	○
LIBERO-PRO (Zhou et al., 2025b)	LIBERO	×	●	●	◐	●	○	○	○	○	○	○
GemBench (Garcia et al., 2025)	RLBench	✓	●	●	○	●	●	●	○	●	○	○
VLA-Arena (Zhang et al., 2026)	RoboSuite	✓	●	◐	●	●	●	◐	◐	○	◐	◐
SpurDiag-Bench (Ours)	LIBERO	✓	●	●	○	◐	○	●	●	●	●	●

Table 1: **VLA benchmark comparison.** Existing VLA benchmarks scale *breadth* and evaluate mostly at the task-outcome level. Only a few recent works begin to probe *depth* along one or two axes. SpurDiag-Bench is the first benchmark that delivers the full mechanistic-diagnosis stack (stage localization, quantitative representation probing, internal causal intervention, trajectory metrics, and reference-trajectory baselines), marked ● full, ◐ partial, ○ none.

instruction insensitivity and visual shortcuts in VLAs (Fei et al., 2026; Darabi & Trivedi, 2026), concept and spatial *lock-in* after low-data post-training (Huang et al., 2026), and spatial overfitting or action-pattern memorization (Li, 2026; Ni et al., 2026). Despite their different terminology, these observations reveal a shared failure mechanism: models can achieve apparent task success by exploiting statistically predictive but non-causal cues. Diagnosing spurious correlations helps move beyond *whether* a task is completed to *how* it is completed and *where* the model loses grounding.

We introduce **SpurDiag-Bench**, a diagnostic benchmark for identifying and analyzing spurious correlations in VLA models. Unlike any existing VLA benchmarks that focus on expanding task diversity/difficulty or measuring success rates under new generalization splits, SpurDiag-Bench distinguishes itself by revisiting the fine-grained execution details of known tasks without requiring harder/broader tasks. Its goal is to examine whether models have learned causally grounded vision-language-action reasoning, rather than merely reproducing statistically correlated behaviors. Inspired by prior information-flow analyses in large vision-language models (Zhang et al., 2025b), we design a **chain analysis for diagnosis**, which decomposes a VLA along its perception-reasoning-action chain and diagnoses two modality bindings: a *vision-language* (V-L) binding that tests whether the model grounds the instructed target to its true physical entity, and a *vision-language to action* ((VL)-A) binding that tests whether the action module acts on that entity. This decomposition enables a principled diagnosis of where and why VLA models fail.

The experimental results on SpurDiag-Bench suggest that VLA models frequently rely on *spurious correlations* during task execution that cannot be simply addressed by scaling the training alone. Even

when models achieve high success rates on unseen objects, novel layouts, or paraphrased instructions, they may still rely on superficial co-occurrences learned from training data. Such a phenomenon therefore reflects apparent generalization rather than true vision-language-action grounding. Scaling data may improve coverage over these shortcuts, but it does not by itself ensure that models learn the correct causal correspondence. Lastly, our fine-tuning experiments verify that not all the VLA models can recover from shortcut reliance even after additional training, and further identify which model characteristics and architectural design may easily counter spurious correlations.

Our main contributions are summarized as follows:

- We introduce **SpurDiag-Bench**, the first diagnostic benchmark that revisits established VLA tasks to examine the **causal basis of task success**, enabling systematic assessment of reliance on spurious correlations beyond aggregate task success.
- We propose a **chain analysis for diagnosis**, combining causal interventions with carefully designed metrics to isolate the two modality-binding stages, V-L and (VL)-A, along the VLA computational chain, to precisely pinpoint where failures arise from.
- We evaluate 12 state-of-the-art VLA models and provide empirical evidence that reliance on **spurious correlations is widespread**, even without introducing harder or broader tasks.
- Lastly, we conduct systematic fine-tuning experiments on representative VLAs to investigate **the mitigation of spurious correlations**. We further summarize common characteristics of VLA models that exhibit stronger recovery from spurious correlations, providing insights for future VLA architecture design and training for the community.

2 Related Work

Vision-Language-Action Models. Vision-Language-Action (VLA) models connect pretrained vision-language representations to robot control through different action interfaces. Token-based policies cast quantized controls as classification or autoregressive prediction, as in RT-2 (Zitkovich et al., 2023) and OpenVLA (Kim et al., 2024). Continuous-action policies instead employ diffusion or flow matching, as in π_0 (Black et al., 2025b), FLOWER (Reuss et al., 2025), and X-VLA (Zheng et al., 2026). Other lines enrich this interface with structured inductive biases, including implicit spatial representations (Li et al., 2026a), explicit trace reasoning against memorization (Ni et al., 2026), temporal memory (Shi et al., 2026), and latent-action or future-state prediction (Bu et al., 2025; Zhang et al., 2025a; Cen et al., 2025; Sun et al., 2026).

Generalization Benchmarks for VLA. Simulation suites such as ManiSkill (Mu et al., 2021), LIBERO (Liu et al., 2023), and BEHAVIOR (Li et al., 2022) standardize robot-learning tasks and demonstrations. Recent benchmarks further probe controlled generalization shifts. COLOSSEUM studies environmental perturbations (Pumacay et al., 2024). GemBench and AGNOSTOS test novel configurations and tasks (Garcia et al., 2025; Zhou et al., 2025a). INT-ACT separates intention understanding from motor execution (Fang et al., 2025). LIBERO-Plus, LIBERO-PRO, and VLA-Arena diagnose robustness to visual, linguistic, initial-state, and task variations (Fei et al., 2026; Zhou et al., 2025b; Zhang et al., 2026).

Failure Analysis for VLA Models. Recent studies diagnose VLA failures beyond aggregate success. Representation analyses show that action fine-tuning can degrade visual features (Kachaev et al., 2026), ViT features remain useful throughout the hierarchy (Li et al., 2025), and viewpoint brittleness often stems from spatial misalignment (Li et al., 2026b). Linear probes find object and action states linearly decodable, though decodability does not imply use (Lu et al., 2025), while activation interventions expose spurious spatial representations (Grant et al., 2026). Behavioral studies identify spatial overfitting (Li, 2026), concept lock-in (Huang et al., 2026), and instruction insensitivity (Fei et al., 2026). ProGAL-VLA and SemanticVLA address related shortcuts through explicit grounding or semantic reasoning (Darabi & Trivedi, 2026; Ni et al., 2026).

3 A Diagnostic Method for Spurious Correlations

This section introduces our **chain analysis** for diagnosing spurious correlations in VLA models to precisely pinpoint the failures through two key binding relationships, V-L and (VL)-A.

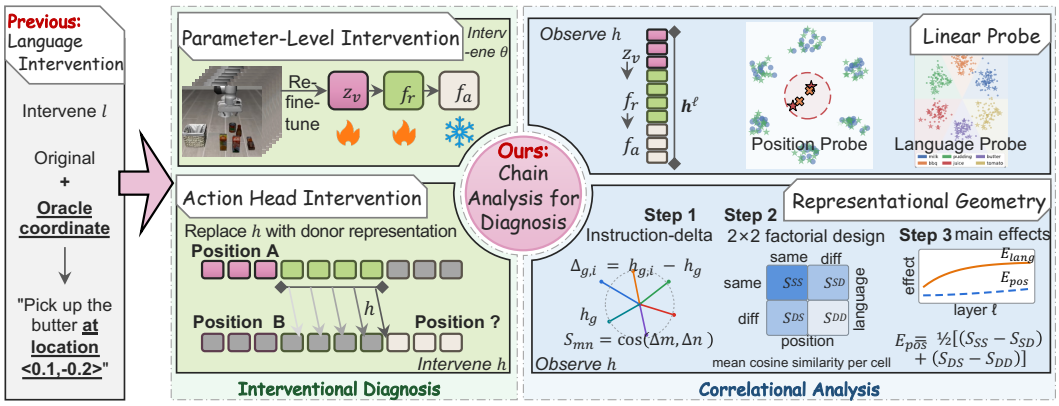


Figure 2: Left: previous language-space intervention. Right: our chain analysis for diagnosis. Green: interventional diagnosis, at the parameters and at h. Blue: correlational analysis, a linear probe and a representational geometry. Thus, the binding of V-L in f_r and (VL)-A in f_a are separately analyzed.

3.1 Problem Formulation

As discussed in Sec. 1, we study whether VLA inferences are grounded in task-relevant visual and linguistic information or depend on incidental regularities in task configurations, namely spurious correlations. Given a visual observation $\mathbf{x}_t$ and a language instruction l, a VLA policy predicts an action $\mathbf{a}_t$ toward the instructed goal. To organize our diagnosis, we describe action prediction through the following functional decomposition:

$$\mathbf{a}_t = f_a\big(f_r(z_v, l)\big), \quad z_v = f_v(\mathbf{x}_t) \in \mathbb{R}^{N\times d}, \tag{1}$$

where f_v is the visual encoder producing N tokens of width d, f_r is the reasoning module, and f_a is the action head. [1]

The functional decomposition above highlights two relationships essential to task execution: the instruction must be associated with the relevant entities in the observed scene, and the resulting task interpretation must guide action generation. We refer to these task-relevant cross-modal associations as *bindings*. Accordingly, the diagnosis is decomposed into two complementary stages: vision-language (V-L) binding and vision-language-to-action ((VL)-A) binding.

V-L binding examines whether the language instruction is associated with the target entity in the visual scene. Given the broad scope of this concept and the difficulty of exhaustively testing all possible spurious associations, we focus on spatial grounding as a concrete diagnostic case: whether the instructed target object and its physical position can be decoded from h:

$$h = f_r(z_v, l), \quad z_v = f_v(\mathbf{x}_t), \tag{2}$$

that is, whether f_r can bind the instructed target to its visual features z_v and thereby infer the target's physical position. Note that the same approach can also be applied to other attributes, such as color.

(VL)-A binding aims to diagnose the action head f_a in Eq. 1: given the correct entity representation f_r directly from the vision-language layer (i.e., its position is known), whether the current action head could perform the correct move (i.e., grasp the object) or be *locked* in the shortcut trajectory.

3.2 Interventions for Diagnosing and Mitigating Spurious Correlations

In previous work, a common input-level diagnostic appends the ground-truth target coordinate to the instruction, $\hat{l} = l \oplus \mathbf{p}^*$. That test only asks whether the model can use a coordinate written in the prompt. It cannot separate a failure in f_r from a failure in f_a.

We therefore intervene inside the model, at the trainable parameters and at intermediate activations in f_r in two fashions. Specifically, we apply interventions to examine whether shortcut reliance

[1] Note that for ease of exposition, we adopt a unified abstraction of the VLA architecture. In practice, VLA models vary substantially in architectural design, auxiliary inputs, and output formulations, and the boundaries between f_v, f_r, and f_a may not always be explicit. Model-specific configurations are detailed in Appendix E.2.

can be mitigated by 1) fine-tuning models at different stages gradually and 2) implementing further activation interventions that hold the model weights fixed and replace intermediate representations to test their influence on downstream target selection.

Parameter-level intervention: frozen-head re-fine-tuning. Standard re-fine-tuning updates the weights of the whole VLA and then reports success rate, so a gain cannot be attributed to a specific stage. We collect demonstrations with a scripted controller, not a policy, so the data is not biased by the model's own prior. On each layout the controller reads the ground-truth target position, grasps the instructed object there, and places it in the basket. We re-fine-tune on these demonstrations under a module freeze. The action head f_a is frozen, so the update falls only on the vision-language pathway and any repair is attributed to it. We train the visual encoder f_v and adapt the reasoning module f_r with LoRA (Hu et al., 2022), while the base weights of f_r stay frozen. On the same demonstrations we also train an unfrozen-head counterpart that updates f_a. Later sections compare the base, frozen-head, and unfrozen-head checkpoints.

Activation substitution. Activation patching substitutes an intermediate representation and reads the downstream behavior (Vig et al., 2020; Meng et al., 2022). This is a counterfactual on h: two instructions give two computations of $h = f_r(z_v, l)$, and only that representation changes. The prompt remains fixed. From layer ℓ onward we replace both the image and the text tokens with those of the donor instruction, and we rebuild the substitution at every replan. We also replace the residual write of a single layer, leaving later layers to recompute. The score is whether the closed-loop action follows the unpatched donor run. We further mix the two representations and read the output action.

3.3 Correlational Analysis of Binding Before and After interventions

To characterize the representational changes associated with shortcut mitigation, we analyze V–L binding before and after the fine-tuning intervention. Linear probes assess whether target entity and physical position are decodable from the hidden representation. Representational geometry further examines how instruction-induced differences are interpreted with respect to the target entity.

Linear Probing

Standard linear probes (Alain & Bengio, 2017) test whether a quantity is decodable from a hidden state. As illustrated in Sec. 3.1, we choose to probe two task-relevant quantities in the (pooled) hidden representation $h^{(\ell)} \in \mathbb{R}^d$ at layer ℓ of f_r: the target object and its physical position. A *language* probe classifies the target object $o_i \in \mathcal{O}$. A *position* probe regresses the target's 2D coordinate $\mathbf{p}$. For each target object $o_i \in \mathcal{O}$, the position probe is a ridge regression (Hoerl & Kennard, 1970),

$$\hat{\mathbf{p}}_i^{(j)} = W_i h^{(\ell,j)} + b_i, \qquad (W_i, b_i) = \arg\min_{W,b} \sum_j \left\| \mathbf{p}_i^{(j)} - W h^{(\ell,j)} - b \right\|^2 + \lambda \|W\|_F^2, \tag{3}$$

where $\mathbf{p}_i^{(j)}$ is the ground-truth 2D coordinate of o_i at trial j and λ is the ridge coefficient. The language probe is a linear classifier trained analogously. We score the position probe with R^2_{pos} and the language probe with R^2_{lang} on a one-hot encoding of o_i.

Representational Geometry

Representational geometry (Kriegeskorte & Wei, 2021) tests the utility of the given information. In our benchmark, it examines whether the instruction-induced representation is bound to the entity itself in a counterfactual way by altering the entity's current physical position. The design varies both the instruction and the object-to-slot layout. A resolved bind follows the actual state.

Step 1: Instruction-delta and cosine similarity. For a layout g with six instructions $\{l_i\}_{i=1}^6$, let $h_{g,i}^{(\ell)}$ be the layer-ℓ representation and $\bar{h}_g^{(\ell)} = \frac{1}{6}\sum_{j=1}^6 h_{g,j}^{(\ell)}$ their per-layout mean. To eliminate discrepancies caused by the scene and the layout of objects, we subtract this mean and normalize to unit length, computing

$$\Delta_{g,i}^{(\ell)} = h_{g,i}^{(\ell)} - \bar{h}_g^{(\ell)}, \qquad \hat{\Delta}_{g,i}^{(\ell)} = \Delta_{g,i}^{(\ell)} / \|\Delta_{g,i}^{(\ell)}\|. \tag{4}$$

The subtraction cancels everything the six instructions share under the same scene, such as the background and the object layout. Therefore the instruction-delta reflects the differences induced by

the language-named target object and by its physical position. Writing $m = (g_m, i_m)$ for a sample, we take the cosine similarity $S_{mn} = \langle \hat{\Delta}_m^{(\ell)}, \hat{\Delta}_n^{(\ell)} \rangle$ over all cross-layout pairs $g_m \neq g_n$.

Step 2: Factorial design. We use a 2×2 factorial design to separate the two factors, language and position. Grouping each pair by same or different language, and by same or different position, gives four cells SS, SD, DS and DD (language first, position second). We write $\bar{S}_{SS}, \bar{S}_{SD}, \bar{S}_{DS}, \bar{S}_{DD}$ for the mean cosine similarity in each cell.

Step 3: Two-factor main-effect analysis. We read language and position as the two main effects,

$$E_{\text{lang}} = \tfrac{1}{2}\big[(\bar{S}_{SS} - \bar{S}_{DS}) + (\bar{S}_{SD} - \bar{S}_{DD})\big], \qquad E_{\text{pos}} = \tfrac{1}{2}\big[(\bar{S}_{SS} - \bar{S}_{SD}) + (\bar{S}_{DS} - \bar{S}_{DD})\big]. \quad (5)$$

E_{pos} is the main effect of the target's actual state, here its current position. Larger is better: the instruction-delta follows where the named object now is. E_{lang} is the main effect of what the instruction names, after that position factor is aligned. Smaller is better: when the target occupies the same place, the named object should not still split the instruction-delta. A large E_{lang} is organization by the name, not a resolved V-L bind. We compute both at one readout layer per model (Table S7).

4 SpurDiag-Bench: A Spurious-Correlation Diagnosis Benchmark

4.1 Task Setup

SpurDiag-Bench keeps the official LIBERO-Object task and changes only the object-to-slot assignment. The robot still picks the object named in the instruction and places it in the basket, in the same scene and with the same objects (Fig. 1). The skill and the visual domain do not get harder. Published checkpoints already exceed 85% success on the unperturbed layout, yet a permutation of slots is enough to expose a position shortcut that the official success rate cannot detect.

The arena is a fixed $2{\times}3$ grid of six slots, occupied by six objects $\mathcal{O} = \{o_1, \ldots, o_6\}$. A **position order** assigns objects to slots. We evaluate on 180 such layouts, balanced so that every object appears in every slot approximately evenly frequent. A genuinely grounded model should redirect its actions and trajectory to follow the target whenever the layout or the instruction changes.

To expose the VLA failures without increasing task complexity, we retain the same objects, scene, and pick-and-place objective, while varying only the spatial configurations. Particularly, we test **out-of-distribution (OOD) generalization** in two ways: **interpolation** within the training distribution, and **extrapolation** beyond the distribution. Success rate is reported on these tests. Interpolation is the interior circle of radius 0.05 m held out of training (Sec. 4.2), with 36 evaluation scenes. Extrapolation places only the target outside the axis-aligned box of the six slot centers. Three disjoint levels isolate confounds (Fig. 3). In far_a, the position is the only factor that changes: the distance to the robot base and both camera views stay within their training ranges. In far_b, the distance to the robot base leaves its training range, while both camera views stay within theirs. In far_c, at least one camera sees the target worse than at any training position. The extrapolation split contains 162 evaluation scenes. Details are in Appendix A.

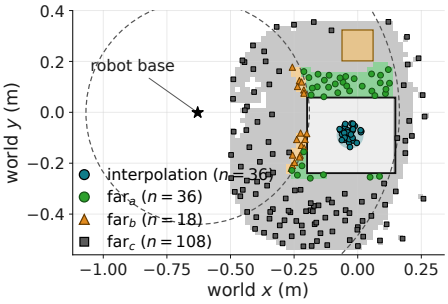


Figure 3: Interpolation and the three extrapolation levels. The teal disk is the held-out circle. Only the target moves.

4.2 Re-fine-tuning Data Collection

To test whether the binding failure is repairable, we collect decorrelated demonstrations for re-fine-tuning. The scripted controller of Sec. 3.2 reads the ground-truth target position and places the object in the basket. Object positions are sampled at random rather than on the grid of Sec. 4.1, with a minimum pairwise distance of 0.07 m. Every object appears across the whole workspace, decorrelating identity from position. We hold out the interior circle of Sec. 4.1: demonstrations with the target inside it are removed from training, leaving the circle empty. The set is balanced across the six objects. Full phases and dataset sizes are in Appendix A.

4.3 Evaluation Metrics

Instead of evaluating each model with SR for general task completeness performance, we also design tailored metrics so that V-L and (VL)-A binding can be carefully quantitatively measured (Sec. 3). Table 2 lists each matrix by diagnostic target, range, and role. $D^{\dagger}_{\text{var}}$ is a diversity readout of the six instruction-conditioned motions. Appendix I splits each failed rollout by task stage. The probe and geometry constructions are in Sec. 3.3. Computation details are in Appendix H.

Table 2: Evaluation metrics on SpurDiag-Bench, ordered by diagnostic target. ↑ higher is better, ↓ lower is better. † $n = 6$ instruction-conditioned motions per scene.

Diagnostic target	Metric	Range	Role
Task Completion	SR ↑	$[0, 100]\%$	Fraction of episodes that complete the instructed task.
V-L binding	R^2_{pos} ↑	$(-\infty, 1]$	Whether h contains the target's position.
	R^2_{lang} ↑	$(-\infty, 1]$	Whether h contains the named object.
	E_{pos} ↑	$[-2, 2]$	Degree to which the instruction-delta is affected by the target's actual position.
	E_{lang} ↓	$[-2, 2]$	Degree to which the instruction-delta is affected by what the instruction names, after position is aligned.
(VL)-A binding	$D^{\dagger}_{\text{var}}$	$[1, n]$	Distinctness of motions across instructions.

5 Experimental Results

The evaluation of the VLA models can be summarized as three sorts. We first 1) directly test the released checkpoints (Appendix E.2) of 12 state-of-the-art VLAs on SpurDiag-Bench, without additional fine-tuning, to assess their reliance on spurious correlations. Then we conduct fine-tuning on some representative models, including 2) fine-tuning only the vision-language pathway while freezing the action head, to diagnose the spurious correlation at the V-L stage, and 3) unfreezing and updating the action head on the same demonstrations to examine the contribution of the (VL)-A stage. The demonstrations follow the setting in Sec. 4.2, the intervention setup is described in Sec. 3.2, and training hyperparameters are provided in Appendix F.

5.1 Performance of State-of-the-Art VLAs on SpurDiag-Bench

As shown in Fig. 1, across the 12 evaluated state-of-the-art VLAs, performance on SpurDiag-Bench drops consistently across all three architectural families. Despite achieving 85.4–99.8% success on LIBERO-Object, their success rates fall to only 10.6–21.9% on SpurDiag-Bench (14.8% on average), revealing **a substantial gap between task success and robustness to spurious correlations**.

5.2 V-L and (VL)-A binding

V-L binding. On each VLA model (Table 3a), R^2_{pos} and R^2_{lang} show that the location is linearly decodable from h, and that the name of the object to grasp is in h. V-L binding still requires that this name be tied to the target's actual state. It is not: the closed-loop rollout still goes to the official slots, and E_{pos} is near zero. SR remains 11–22% (Fig. 1), including 21.9% on $\pi_{0.5}$.

Fine-tuning on the same demonstrations does not mitigate spurious correlations uniformly. On $\pi_{0.5}$ and π_0, frozen-head fine-tuning reverses the geometry, and SR follows (21.9% to 95.5% and 15.5% to 89.5%). VLA-JEPA is only half repaired on E_{pos} and SR. UniVLA and MemoryVLA raise E_{pos} far more than SR. X-VLA raises E_{pos} with a small SR gain. High E_{pos} is required for high SR. It is not sufficient. A model can look grounded in h and still fail the task (Appendix I). Extra training does not, by itself, clear shortcut reliance.

(VL)-A binding. Whether the repaired binding reaches the action is read from the object the arm approaches (Fig. 4; Appendix I). With the action head frozen, wrong-target rollouts fall from 73–83% on the base checkpoints to at most 5% on five of the six models. VLA-JEPA keeps 15%, and 16% after unfreezing. The shortcut is in V-L, not in the action head. The remaining failures are (VL)-A failures: the arm reaches the named object and does not lift it. They reach 69% on X-VLA and about

Table 3: Representatives on SpurDiag-Bench, grouped by family. Left: V-L geometry and SR after frozen-head fine-tuning. Right: D_{var} and SR across the three checkpoints. OpenVLA has no detachable action head, and $*$ marks the missing frozen-head value. R^2_{pos} and R^2_{lang} can be negative: $R^2 < 0$ is worse than predicting the mean. E_{lang} and E_{pos} are cosine-similarity contrasts in $[-2, 2]$. SR is the 180×6 basket success rate. $D^{\dagger}_{\text{var}}$ is the mean per-scene FAKTUAL $\exp(H)$ on end-effector xyz commands ($n = 6$, Appendix H.3). ↑ higher is better, ↓ lower is better.

(a) *base→frozen-head.* (b) *base→frozen-head→unfrozen-head.*

Model	V-L Binding					(VL)-A Binding	
	R^2_{pos} ↑	R^2_{lang} ↑	E_{lang} ↓	E_{pos} ↑	SR (%) ↑	$D^{\dagger}_{\text{var}}$	SR (%) ↑
			Single-stream				
OpenVLA	0.930→∗	0.997→∗	0.672→∗	0.013→∗	10.6→∗	3.78→∗→4.36	10.6→∗→58.1
UniVLA	0.926→0.975	0.979→0.980	0.428→0.011	0.016→1.043	12.5→58.4	2.55→3.89→3.67	12.5→58.4→82.6
			Hierarchical				
X-VLA	0.649→0.849	0.998→0.998	0.467→0.129	0.022→0.972	14.2→24.6	1.17→1.09→1.08	14.2→24.6→25.9
π_0	0.846→0.890	0.960→0.862	0.958→0.085	0.001→0.942	15.5→89.5	2.98→3.85→3.82	15.5→89.5→89.6
$\pi_{0.5}$	0.852→0.927	0.978→0.987	0.240→0.100	0.107→0.855	21.9→95.5	2.92→4.26→3.83	21.9→95.5→96.0
MemoryVLA	0.638→0.912	0.993→0.997	0.919→0.056	0.001→0.935	13.8→68.6	2.77→3.83→3.71	13.8→68.6→59.8
			World-model				
VLA-JEPA	0.653→0.868	0.984→0.989	0.548→0.197	0.001→0.596	16.3→50.2	2.55→4.07→3.84	16.3→50.2→70.2

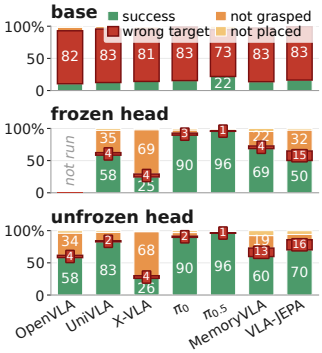


Figure 4: Failure decomposition per checkpoint.

Table 4: SR (%) on the interpolation and extrapolation splits of Sec. 4.1, as *base→frozen-head→unfrozen-head*. n=36/36/18/108 episodes per column. $*$: no frozen-head checkpoint.

Model	interpolation	extrapolation		
		far$_a$	far$_b$	far$_c$
		Single-stream		
OpenVLA	2.8→∗→63.9	0.0→∗→33.3	0.0→∗→22.2	0.0→∗→2.8
UniVLA	11.1→38.9→77.8	0.0→11.1→36.1	0.0→0.0→50.0	0.0→2.8→1.9
		Hierarchical		
X-VLA	13.9→0.0→0.0	0.0→5.6→0.0	0.0→0.0→0.0	0.0→0.9→0.0
π_0	19.4→88.9→91.7	0.0→50.0→44.4	0.0→66.7→66.7	1.9→5.6→2.8
$\pi_{0.5}$	66.7→86.1→94.4	5.6→63.9→75.0	16.7→72.2→100.0	3.7→13.0→27.8
MemoryVLA	2.8→52.8→58.3	0.0→33.3→25.0	0.0→55.6→38.9	0.9→4.6→0.0
		World-model		
VLA-JEPA	0.0→88.9→91.7	0.0→22.2→27.8	0.0→38.9→33.3	0.0→1.9→1.9

a third on VLA-JEPA and UniVLA. Except on X-VLA, $D^{\dagger}_{\text{var}}$ rises with the frozen head, so the six instructions lead to more distinct motions.

Updating the action head is not a second, reliable fix. It reduces the (VL)-A failures on UniVLA and VLA-JEPA, leaves X-VLA unchanged under open-loop absolute control, and returns MemoryVLA's wrong-target rollouts to 13%. $\pi_{0.5}$ and π_0 are already above 87% and barely move.

5.3 Interpolation and Extrapolation under Spatial OOD

Section 5.2 diagnoses both bindings on the training layouts. We next test the same chain off that support: interpolation inside the held-out circle, and extrapolation to farther positions. Fine-tuning still raises $\pi_{0.5}$ on interpolation and on far$_a$, but it does not do so for every model. On interpolation, $\pi_{0.5}$ goes from 66.7% to 86.1% to 94.4%. Six base checkpoints are at 0 on far$_a$, and $\pi_{0.5}$ is at 5.6%. Fine-tuning raises $\pi_{0.5}$ to 75.0%. The difference is whether position was learned. It is not a precision gap. X-VLA interpolation does not rise, so its far zeros are not an extrapolation failure.

On $\pi_{0.5}$, far$_b$ is 100% against 75.0% on far$_a$. The reachable ring cuts the outward half. What remains is about 3 cm closer to the base. This split does not rank kinematic difficulty. On far$_c$ all but one checkpoint stay at or below 13%. The highest is $\pi_{0.5}$ at 27.8%. Those successes have no target pixels on the wrist camera at $t = 0$. The reach is guided by the third-person camera, and the wrist recaptures on the way. The split measures recovery after that camera fails. It is not a total loss of perception.

5.4 Activation substitution: localizing V-L and (VL)-A binding

We next localize where f_r uses the V-L binding, and whether f_a follows it. On $\pi_{0.5}$, we keep the weights, the scene, and the prompt of instruction A, and replace activations in f_r with those computed for instruction B (Sec. 3.2; Fig. 5). Substitution from layer 14 onward follows the donor grasp in 84.1% of trials, against a donor ceiling of 91.6%; from layer 15 onward it falls to 27.1%. Starting the substitution at any layer from 0 through 12 stays at that ceiling, so the switch is not from patching more layers. Editing a single residual write is strongest at layer 12 (38.3%), with smaller effects at layers 1 and 3–9. Neither a random displacement of the same magnitude nor substitution at the first inference alone switches the grasp. The main split selects scenes by the position of the unpatched donor run, on one instruction pair. Where both unpatched instructions grasp their named object, the same substitution still switches the grasp, so the switch is not only a change of slot.

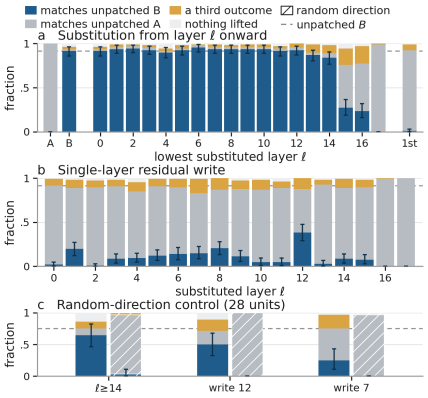


Figure 5: Closed-loop activation substitution on released $\pi_{0.5}$ ($n = 107$ scenes).

Mixing the two representations does not produce an intermediate grasp: the mixing weight decides which grasp occurs, and the arm lifts nothing and stops between the two objects in at most 1.9% of trials. Substituting h therefore selects between the two closed-loop grasps rather than writing a new binding, which is why Sec. 3.2 also intervenes on θ.

5.5 Real-world validation and Architecture Design insights.

Real-world deployment on SO-100 (Knight et al., 2024) and UR5 (Universal Robots, 2021) arms is conducted to rule out potential rendering artifacts introduced by the simulation environment. We repeat the diagnosis on a real SO-100 arm with $\pi_{0.5}$. The results reproduce the same diagnostic pattern: frozen-head fine-tuning restores correct target selection and raises SR from 13.9% to 66.7%, with little further gain from updating the action head (Appendix B). UR5 experiments further extend the diagnosis to changed instructions and same-category objects (Appendix C). **Design insights.** We also analyze the potential VLA architecture design that better counters the spurious correlations, suggesting that dedicated action experts, heterogeneous co-training, and knowledge insulation may facilitate shortcut mitigation. See Appendix D for a detailed discussion.

6 Conclusion and Limitations

In this work, we introduce **SpurDiag-Bench** to diagnose spurious correlations in Vision-Language-Action (VLA) models. It exposes the failure of VLA without introducing extra distribution shift or task difficultly, yet 12 VLAs drop from 85.4–99.8% to 10.6–21.9% in success rate. Chain analysis shows that the object name and position are decodable, but they are linked by a superficial association. Activation substitution localizes target selection to intermediate language-model layers. Re-fine-tuning reduces wrong-target failures but does not fully repair them: some models still pick the wrong object, and part of the remaining failures occur in the action stage. The spatial OOD splits and a real SO-100 arm show the same pattern. Success rate alone therefore cannot certify grounding. Our results also indicate the potential characisitcs that a robust VLA needs to counter the spurious correlations. We hope to inspire the community to look beyond whether a VLA succeeds and ask what evidence drives its actions—a crucial step toward trustworthy embodied intelligence.

Limitations and Broader Impact. Our study shows that even a focused analysis of modality binding (physical position) can expose how spurious correlations shape VLA decisions; other cues, such as colour and shape, can also be explored. Results suggest that the language model can be a main source of spurious correlations in the models we tested, motivating stronger LMs paired in VLA. By revealing failures hidden by task-success scores, our diagnostics can guide more reliable VLA evaluation, though passing these tests alone does not establish real-world safety.

AI USE STATEMENT

We used generative AI tools in the following capacities during this work. **Language polishing.** An LLM assistant was used to polish the manuscript text. All final wording reflects the authors' judgment and review. We did not use generative AI to draft sections of the paper. **Literature discovery.** We periodically queried an LLM with web-search capabilities to scan recent related work. All cited references were verified by the authors. **Code development.** An LLM coding assistant accelerated the implementation of experiment infrastructure and analysis scripts. AI-generated code was manually reviewed by the authors to ensure correctness before use in reported experiments. We did not use generative AI for proving mathematical claims. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

An anonymous project page at `https://vla-page.netlify.app` provides the leaderboard of Table 3. The released checkpoints and the evaluation interface of every model are listed in Appendix E.2. The task and data setup is in Appendix A, the fine-tuning hyperparameters are in Appendix F, and the metric computations are in Appendix H.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017. URL `https://openreview.net/forum?id=HJ4-rAVtl`.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL `https://arxiv.org/abs/2511.21631`.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024. URL `https://arxiv.org/abs/2407.07726`.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0:

A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025b. URL https://www.roboticsproceedings.org/rss21/p010.html.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to act anywhere with task-centric latent actions. In *Robotics: Science and Systems (RSS)*, 2025. URL https://www.roboticsproceedings.org/rss21/p014.html.

Jun Cen, Siteng Huang, Yuqian Yuan, Kehan Li, Hangjie Yuan, Chaohui Yu, Bohan Hou, Yuming Jiang, Jiayan Guo, Xin Li, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. RynnVLA-002: A unified vision-language-action and world model. *arXiv preprint arXiv:2511.17502*, 2025. URL https://arxiv.org/abs/2511.17502.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788*, 2016. URL https://arxiv.org/abs/1603.03788.

Nastaran Darabi and Amit Ranjan Trivedi. ProGAL-VLA: Grounded alignment through prospective reasoning in vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Findings Track*, pp. 9013–9022, 2026. URL https://openaccess.thecvf.com/content/CVPR2026F/html/Darabi_ProGAL-VLA_Grounded_Alignment_through_Prospective_Reasoning_in_Vision-Language-Action_Models_CVPRF_2026_paper.html.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

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

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025. URL https://arxiv.org/abs/2506.09930.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: A progressive robustness benchmark for visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Fei_LIBERO-Plus_A_Progressive_Robustness_Benchmark_for_Visual-Language-Action_Models_CVPR_2026_paper.html.

Dan Friedman and Adji Bousso Dieng. The Vendi score: A diversity evaluation metric for machine learning. *Transactions on Machine Learning Research (TMLR)*, 2023. URL https://openreview.net/forum?id=g97OHbQyk1.

Ricardo Garcia, Shizhe Chen, and Cordelia Schmid. Towards generalizable vision-language robotic manipulation: A benchmark and LLM-guided 3D policy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8996–9002. IEEE, 2025. URL https://ieeexplore.ieee.org/document/11127315.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence (Nat. Mach. Intell.)*, 2(11):665–673, 2020. URL https://www.nature.com/articles/s42256-020-00257-z.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models. In *International Conference on Learning Representations (ICLR), Workshop on Multimodal Intelligence*, 2026. URL https://openreview.net/forum?id=w8Mw9zJt1y.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. URL https://doi.org/10.1080/00401706.1970.10488634.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Suning Huang, Jiaqi Shao, Ke Wang, Qianzhong Chen, Jiankai Sun, Yanjiang Guo, Mac Schwager, and Jeannette Bohg. Breaking lock-in: Preserving steerability under low-data VLA post-training. *arXiv preprint arXiv:2604.23121*, 2026. URL https://arxiv.org/abs/2604.23121.

Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K. Kovalev, and Aleksandr I. Panov. Don't blind your VLA: Aligning visual representations for OOD generalization. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1864–1873. IFAAMAS, 2026. URL https://doi.org/10.65109/PPER9186.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p120.html.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2024. URL https://proceedings.mlr.press/v270/kim25c.html.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems (RSS)*, 2025. URL https://www.roboticsproceedings.org/rss21/p017.html.

Rob Knight, Pepijn Kooijmans, Remi Cadene, Simon Alibert, Michel Aractingi, Dana Aubakirova, Adil Zouitine, Russi Martino, Steven Palma, Caroline Pascal, and Thomas Wolf. Standard open SO-100 & SO-101 arms, 2024. URL https://github.com/TheRobotStudio/SO-ARM100.

Nikolaus Kriegeskorte and Xue-Xin Wei. Neural tuning and representational geometry. *Nature Reviews Neuroscience (Nat. Rev. Neurosci.)*, 22(11):703–718, 2021. URL https://doi.org/10.1038/s41583-021-00502-3.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R. Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio

Savarese, Hyowon Gweon, C. Karen Liu, Jiajun Wu, and Li Fei-Fei. BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pp. 80–93. PMLR, 2022. URL https://proceedings.mlr.press/v205/li23a.html.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial Forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026a. URL https://openreview.net/forum?id=euMVC1DO4k.

Quanyi Li. VLAs are confined yet capable of generalizing to novel tasks. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 7583–7591, 2026. URL https://www.ijcai.org/proceedings/2026/843.

Weiqi Li, Quande Zhang, Ruifeng Zhai, Liang Lin, and Guangrun Wang. VLA models are more generalizable than you think: Revisiting physical and spatial modeling. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35025–35035, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_VLA_Models_Are_More_Generalizable_Than_You_Think_Revisiting_Physical_CVPR_2026_paper.html.

Wenhao Li, Chengwei Ma, Hua Chen, and Weixin Mao. VAT: Vision action transformer by unlocking full representation of ViT. *arXiv preprint arXiv:2512.06013*, 2025. URL https://arxiv.org/abs/2512.06013.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture. In *IEEE International Conference on AI and Data Analytics (ICAD)*, pp. 1–8. IEEE, 2025. URL https://ieeexplore.ieee.org/document/11114062.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Tongzhou Mu, Zhan Ling, Fanbo Xiang, Derek Yang, Xuanlin Li, Stone Tao, Zhiao Huang, Zhiwei Jia, and Hao Su. ManiSkill: Generalizable manipulation skill benchmark with large-scale demonstrations. In *Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS Datasets and Benchmarks)*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/eda80a3d5b344bc40f3bc04f65b7a357-Abstract-round2.html.

Fei Ni, Zhuo Chen, Yifu Yuan, Zibin Dong, Xianze Yao, Shan Luo, Jianye Hao, Jiankang Deng, and Stefanos Zafeiriou. SemanticVLA: Towards semantic reasoning over action memorization via synergistic explicit trace and latent action planning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12237–12247, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Ni_SemanticVLA_Towards_Semantic_Reasoning_over_Action_Memorization_via_Synergistic_Explicit_CVPR_2026_paper.html.

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. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024. URL https://doi.org/10.1109/ICRA57147.2024.10611477.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Physical Intelligence. openpi. GitHub repository, 2025. URL https://github.com/Physical-Intelligence/openpi.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. THE COLOSSEUM: A benchmark for evaluating generalization for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024. URL https://www.roboticsproceedings.org/rss20/p133.html.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. FLOWER: Democratizing generalist robot policies with efficient vision-language-flow models. In *Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3736–3761. PMLR, 2025. URL https://proceedings.mlr.press/v305/reuss25a.html.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=54U3XHf7qq.

Sreevardhan Sirigiri, Nathan Samuel de Lara, Christopher Agia, Florian Shkurti, and Fabio Ramos. Diversity you can actually measure: A fast, model-free diversity metric for robotics datasets. *arXiv preprint arXiv:2603.11634*, 2026. URL https://arxiv.org/abs/2603.11634.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision (ECCV)*, pp. 478–497. Springer, 2026. URL https://link.springer.com/chapter/10.1007/978-3-032-37196-6_26.

Universal Robots. UR5 CB-Series user manual: Software version 3.13, 2021. URL https://www.universal-robots.com/download/manuals-cb-series/user/ur5/313/user-manual-ur5-cb-series-sw313-english-us-en-us/. Official documentation page, updated February 2, 2021.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 12388–12401, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. MLLM can see? Dynamic correction decoding for hallucination mitigation. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=4z3IguA4Zg.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_VGGT_Visual_Geometry_Grounded_Transformer_CVPR_2025_paper.html.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4818–4829, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Xiao_Florence-2_Advancing_a_Unified_Representation_for_a_Variety_of_Vision_CVPR_2024_paper.html.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhai_Sigmoid_Loss_for_Language_Image_Pre-Training_ICCV_2023_paper.html.

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. In *International Conference on Machine Learning (ICML)*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026. URL https://openreview.net/forum?id=qjeCH95rbF.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 24195–24228, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/22d4f952efa13970f0b1ffb22170d416-Abstract-Conference.html.

Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. Cross-modal information flow in multimodal large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19781–19791, 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhang_Cross-modal_Information_Flow_in_Multimodal_Large_Language_Models_CVPR_2025_paper.html.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

Jiaming Zhou, Ke Ye, Jiayi Liu, Teli Ma, Zifan Wang, Ronghe Qiu, Kun-Yu Lin, Zhilin Zhao, and Junwei Liang. Exploring the limits of vision-language-action manipulation in cross-task generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 139899–139927, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/cc92809cd8dfbd035801966ab4896741-Abstract-Conference.html.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025b. URL https://arxiv.org/abs/2510.03827.

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

APPENDIX

A TASK AND DATA SETUP

This appendix specifies the evaluation layouts, the spatial OOD splits, and the re-fine-tuning demonstrations of Sec. 4.1 and Sec. 4.2. Metric definitions are in Appendix H. Fine-tuning hyperparameters are in Appendix F.

A.1 LAYOUT EVALUATION

SpurDiag-Bench places six objects on a fixed 2×3 grid and changes only the object-to-slot assignment. The six objects are always present: milk, bbq sauce, chocolate pudding, orange juice, butter, and tomato sauce. The slot centers are $(-0.119, -0.239)$, $(0.050, -0.100)$, $(-0.149, 0.058)$, $(0.102, -0.198)$, $(0.148, 0.030)$, and $(-0.199, -0.080)$. The first two slots are the positions in which official LIBERO-Object places its targets. The suite contains 180 distinct position orders out of the $6! = 720$ permutations. Each object appears in each slot between 21 and 40 times. Each layout is run once with each of the six instructions, for 1080 episodes. The default instruction stored in the BDDL file is not used.

An episode succeeds if the target's body center lies in the interior region of the basket at the final step. This test is geometrically the same as the LIBERO success predicate. The region is about $12.2\times12.2\times14.9$ cm. The horizon is 400 steps for all models except X-VLA (Appendix E.2). On the same protocol, the scripted controller used for the demonstrations (Appendix A.4) succeeds on 993 of 1080 episodes (91.9%).

A.2 RELATION TO LIBERO-PLUS AND LIBERO-PRO

LIBERO-Plus (Fei et al., 2026) and LIBERO-PRO (Zhou et al., 2025b) also perturb object positions in LIBERO and report the resulting change in success rate. Both change the positions themselves: LIBERO-Plus perturbs the initial position and orientation of the target, and LIBERO-PRO displaces objects from their default configuration. Under such a change, a failure can come from the new position alone. SpurDiag-Bench keeps the six slot positions of LIBERO-Object and changes only which object occupies which slot. All six positions and all six objects appear in the official training scenes, so only the pairing of object and position is new. Each layout is also run with all six instructions, so the instruction is the only input that differs between episodes of the same layout. The effect of new positions is tested separately by the spatial OOD splits below, which control reachability, distance, and visibility.

A.3 SPATIAL OOD

Training targets start inside the axis-aligned box of the six slot centers, inset by 1 cm, and outside a circle of radius 0.05 m centered at $(-0.03, -0.09)$ (Appendix A.4). Interpolation puts the target inside the circle (36 scenes). Extrapolation puts it outside the box (162 scenes), at one of three far levels (Fig. 3). Only the target moves. The other five objects stay inside the box and outside the circle.

The far levels are defined by reachability, distance, and visibility. Every far position is reachable by the arm, checked by inverse kinematics. Distance is the planar distance from the robot base at $(-0.629, -0.001)$. Over the reachable part of the box, it ranges from 0.44 to 0.79 m. Visibility is the pixel count of the target in the instance segmentation of each of the two policy cameras, the third-person view and the wrist view, at the policy's 256×256 resolution, in the first frame the policy receives. The reference for each camera is the smallest count over the six slot centers, with the same object placed at each center in the same scene. In far_a, the distance stays in this range and both cameras see the target at least as well as the reference. In far_b, visibility meets the same condition, but the distance is below this range. In far_c, the wrist camera sees fewer than 100 pixels of the target. No far_a or far_b scene falls below 100 wrist-camera pixels, so the three levels are disjoint.

Far scene counts follow the square root of each level's area, rounded to a multiple of six so the six objects are equally represented. Targets inside a level are chosen by farthest-point sampling. Each

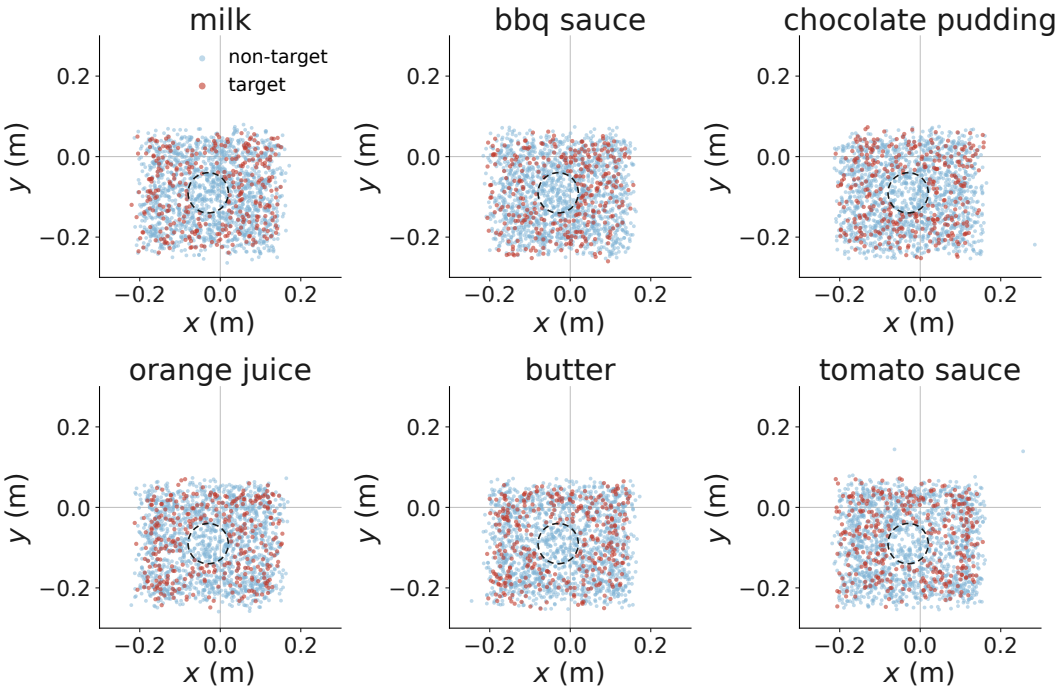


Figure S1: Planar positions of the six objects in the 1897 training demonstrations. Red marks the target. The dashed circle is the same held-out circle as Fig. 3. No target starts inside it.

sampled scene is then checked against its level's criteria with its own six objects. Twenty-nine scenes that failed were redrawn inside the same level.

A.4 Re-fine-tuning demonstrations

Demonstrations are collected with the scripted controller of Sec. 3.2, not a policy. The controller reads the ground-truth target pose and executes eight phases: approach, descend, grasp, lift, align, lower, release, and retreat. The gripper stops above the basket rim and the object drops in. Object positions are sampled uniformly in the axis-aligned box of the six slot centers, inset by 1 cm, with a minimum pairwise distance of 0.07 m, and are set directly in the simulator state. Training positions are continuous and are not tied to the six slots.

The held-out circle is not excluded at collection time. Before training, we remove any demonstration whose target starts inside the circle. We first balance the set to 344 demonstrations per target (seed 0), 2064 in total, then remove the circle, leaving 1897 demonstrations (Fig. S1). All reported fine-tunes use this set.

B SO-100 real-world experiments

This section reports the real-robot counterpart of the $\pi_{0.5}$ experiments. It repeats the chain of Sec. 5.2 on a physical SO-100. A base checkpoint is trained on demonstrations that bind the target object to its slot. It is then re-fine-tuned on decorrelated demonstrations with the action head frozen and unfrozen.

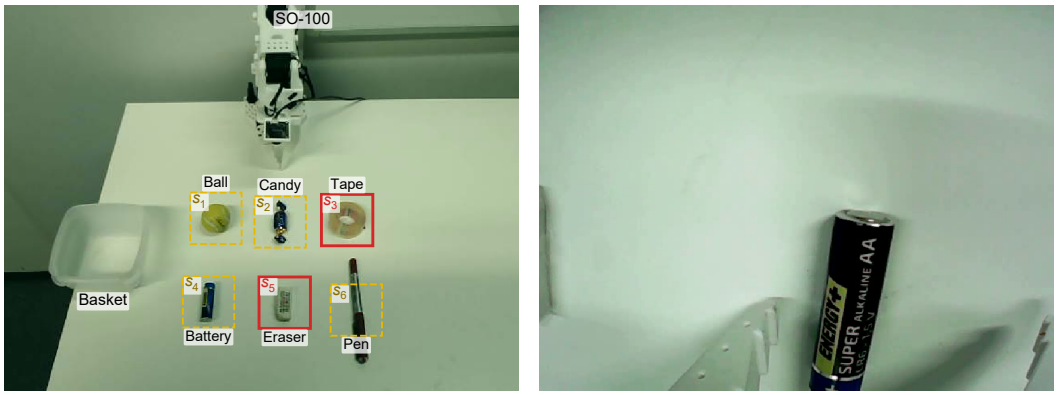
Platform. The platform reproduces the LIBERO-Object arena on a SO-100 arm (Fig. S2, Table S1). Six slots form a 2×3 grid in front of the arm. Slots s_1–s_3 form the row next to the arm and s_4–s_6 the row towards the camera. A fixed basket stands to the left. The six objects are a battery, an eraser, a ball, a pen, a candy and a roll of tape. The instruction is "pick up the {object} and put it in the basket". A base camera faces the arm and the second camera is mounted on the wrist.

Deployment. A laptop runs the control loop and queries the policy on remote GPUs. The loop runs at 30 Hz. $\pi_{0.5}$ returns a chunk of 50 absolute joint targets. A new chunk is requested every 10 executed steps. When it arrives, the steps that elapsed during inference are skipped. Each command passes a rate limit, a joint-range check and a table-height check before it is sent. A rollout starts from a fixed rest pose and lasts at most 45 s. Images reach the server as JPEG and receive the same square crop and color gains as the training data, followed by the policy's own preprocessing.

Base set. The base set copies the layout statistics of LIBERO-Object. We measured these statistics on its BDDL files and its 500 demonstrations. In LIBERO-Object, each instruction has a single layout, repeated in all of its demonstrations. The target stands on only two of the six slots, five tasks on each. Most objects do not move between demonstrations. The base set keeps these properties. Each of the six instructions has one layout and 25 demonstrations. The target stands on s_3 for the ball, the pen and the tape, and on s_5 for the other three objects. As in Sec. 5.2, we call s_3 and s_5 the *official slots*. The official slot of an instruction is the one its target occupies in the base set. Across the six layouts every object occupies every slot once, so a slot alone does not identify an object.

Component	Configuration
Robot	SO-100 (5 joints + parallel gripper), LeRobot driver, position control over USB
Cameras	base camera facing arm and wrist camera, 640×480
Workspace	2×3 slots, 10 cm $\times$ 13 cm spacing, fixed basket
Objects	battery, eraser, ball, pen, candy, tape
Control	30 Hz absolute joint targets, chunk of 50, re-requested every 10 steps
Safety	4°/step rate limit, joint range, table height, operator with emergency stop
Inference	one GH200 per checkpoint, reached from the laptop through an SSH tunnel
Training	8×4 GH200 per run

Table S1: Hardware and deployment of the SO-100 real-world experiments.



(a) Base camera (rest pose): six slots, basket

(b) Wrist camera, just before the gripper closes

Figure S2: SO-100 setup. (a) Base camera at the rest pose. Red: the official slots s_3 and s_5, the only target slots of the base set. (b) Wrist camera just before the gripper closes on a target.

Re-fine-tuning set. The re-fine-tuning set decorrelates object identity from position, as in Sec. 4.2. It uses the same six slots. Five 6×6 Latin squares give 30 layouts, and each layout is recorded once with each of the six instructions. Every object is thus the target on every slot five times. No layout occurs in the base or the test set.

Demonstrations. All demonstrations are scripted, as in simulation. An inverse-kinematics controller moves the arm above the target, lowers it, closes the gripper, lifts, moves over the basket, opens the gripper and returns to the rest pose. Phase durations, hover height and aim point are jittered by $\pm 15\%$, ± 1 cm and ± 3 mm. Failed attempts are discarded. The base set has 150 episodes (92,429 frames) and the re-fine-tuning set 180 episodes (111,495 frames), both at 30 fps.

Training. The three checkpoints follow Sec. 3.2 and Appendix F. The base checkpoint starts from a public $\pi_{0.5}$ checkpoint for this arm, hqfang/pi05-so100_101, and is fine-tuned on the base set. Both re-fine-tuned checkpoints start from base and use the same data, schedule and seed. They differ only in whether f_a, the action expert, is updated.

Evaluation. The evaluation follows Sec. 4.1 and Appendix H at a smaller scale.

- **SR.** The test set is one 6×6 Latin square. Its six layouts are each run with all six instructions, so every object is the target on every slot once. No layout occurs in the base or the re-fine-tuning set. The three checkpoints run each (layout, instruction) item back to back, in shuffled order. A rollout succeeds only if the instructed object ends in the basket. The operator labels the outcome.
- **Selected object.** We read it from the measured tool path, as the object on the slot where the gripper first closes below 7 cm. If the gripper never closes there, we use the slot of the first descent.
- D_{var}**.** It follows Appendix H.3 with one change of scale. On the SO-100, $\pi_{0.5}$ commands absolute joint targets. Forward kinematics maps each command the policy issues to a tool position, which is the absolute-command case of Appendix H.3 ($q_t = u_t$). Appendix H.3 keeps the recorded scale, which here is meters. We express positions in units of 5 cm instead, because an input of 1 to the LIBERO controller moves the target by 5 cm. The score is averaged over the six layouts.
- **V-L metrics.** The probe set is separate from the test set. It has 24 further layouts, from four more Latin squares. Each layout is placed and photographed twice at the rest pose, the first frame of every rollout. Each photo is run with each of the six instructions, which gives 288 forwards per checkpoint through the closed-loop observation path.

Results. The real row follows the direction of the simulated chain on every metric except R^2_{lang} (Table S2). Frozen-head re-fine-tuning raises SR from 13.9% to 66.7%. On paired items it fixes 20 base failures and breaks one (exact McNemar $p = 2 \times 10^{-5}$). Unfreezing the head adds 2.7 points, which is not significant ($p = 1.0$). The V-L geometry reverses as in simulation: E_{pos} rises to 1.11 and E_{lang} falls to 0.02. D_{var} rises after frozen-head re-fine-tuning and does not rise further when the head is updated. R^2_{pos} also rises, but both R^2 values stay below simulation. Each probe has about 30 times fewer training samples, and its position labels are slot centers.

(a) base → frozen-head	$R^2_{\text{pos}} \uparrow$	$R^2_{\text{lang}} \uparrow$	$E_{\text{lang}} \downarrow$	$E_{\text{pos}} \uparrow$	SR (%)↑
$\pi_{0.5}$, LIBERO (sim)	$0.852 \to 0.927$	$0.978 \to 0.987$	$0.240 \to 0.100$	$0.107 \to 0.855$	$21.9 \to 95.5$
$\pi_{0.5}$, SO-100 (real)	$0.457 \to 0.598$	$0.908 \to 0.767$	$0.279 \to 0.017$	$0.216 \to 1.108$	$13.9 \to 66.7$

(b) base → frozen-head → unfrozen-head	$D^{\dagger}_{\text{var}}$	SR (%)↑
$\pi_{0.5}$, LIBERO (sim)	$2.92 \to 4.26 \to 3.83$	$21.9 \to 95.5 \to 96.0$
$\pi_{0.5}$, SO-100 (real)	$2.06 \to 2.85 \to 2.80$	$13.9 \to 66.7 \to 69.4$

Table S2: The SO-100 row of Table 3, with the simulated $\pi_{0.5}$ row for reference. Real SR: 36 rollouts per checkpoint, 95% Wilson intervals $[6.1, 28.7]$, $[50.3, 79.8]$ and $[53.1, 82.0]$. $D^{\dagger}_{\text{var}}$: 6 layouts, positions in 5 cm units (1.07, 1.09 and 1.08 in meters). V-L metrics: 288 forwards per checkpoint. The unfrozen-head checkpoint has $R^2_{\text{pos}} = 0.565$, $R^2_{\text{lang}} = 0.879$, $E_{\text{lang}} = 0.063$ and $E_{\text{pos}} = 0.993$.

Failure modes. The base checkpoint fails by binding, and the re-fine-tuned checkpoints fail by manipulation (Fig. S3). Base selects the wrong object in 28 of 36 rollouts. All 28 wrong selections are on an official slot, and 24 of them are on the instruction's own official slot (0.86, against 0.20 under uniform choice). The arm thus repeats the base demonstration of the instruction, whatever object now stands there (Figs. S4 and S5). Base SR follows the same pattern. It is 4/6 when the target stands on its official slot, 1/6 on the other official slot and 0/24 on the other four slots. After re-fine-tuning, both checkpoints select the named object in all 36 rollouts. Their failures are missed grasps and one failed placement. Unfreezing the head does not reduce the missed grasps.

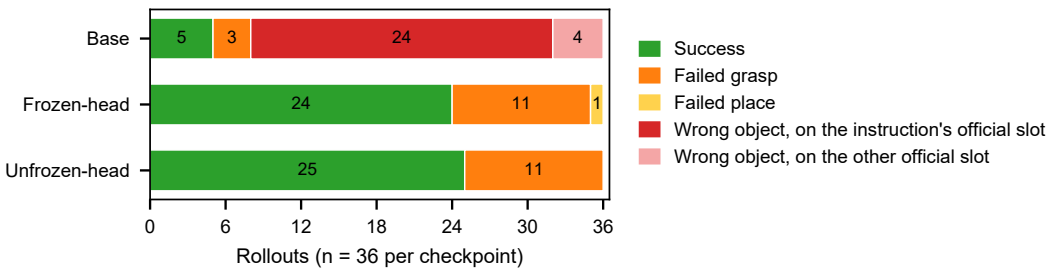


Figure S3: Outcome of every rollout, with wrong selections split by slot.

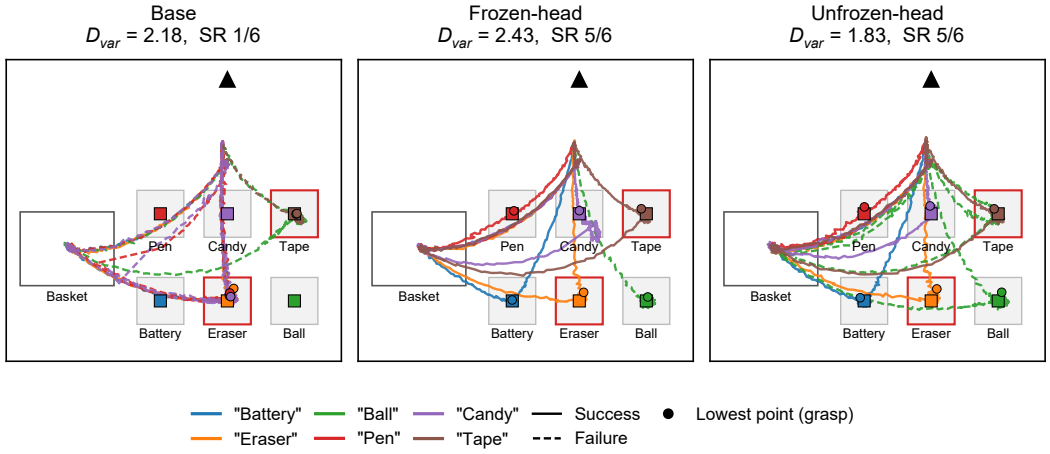


Figure S4: Commanded tool paths in one test layout, seen from above in the orientation of the base camera. These are the paths of D_{var}. Color: the instruction. Red: the official slots. The layout is the one whose base D_{var} is closest to the base mean. Base sends all six instructions to the official slots. After frozen-head re-fine-tuning each instruction reaches its own object.

Example layout. Figures S4 and S5 show this contrast in one test layout. Figure S4 draws the commanded tool path of each of the six instructions from above. Base sends all six paths to the two official slots s_3 and s_5. The target stands there only for the eraser and the tape, so the other four instructions end at a wrong object. After re-fine-tuning, every path of both checkpoints ends at the named object, and the only failure is a missed grasp of the ball. Figure S5 follows one instruction, "pick up the battery and put it in the basket", in the base camera. The battery stands on s_4, and the eraser stands on s_5, the official slot of this instruction. Given the same scene and instruction, base lowers the gripper onto s_5, grasps the eraser and puts it in the basket. Frozen-head lowers it onto s_4 and places the battery. Selecting the object on the instruction's own official slot is the most frequent base outcome (24 of 36 rollouts).

R^2_{lang}. It is the only metric whose direction differs from simulation. The drop comes from a weaker readout of the named object, not from its loss. The probe's argmax still selects the named object in 99.7%, 99.0% and 99.7% of forwards for the three checkpoints. Its mean output for the named object falls from 0.86 to 0.68, against an ideal of 1. The drop is not an effect of the probe size. On simulation features subsampled to the real probe size, R^2_{lang} is $0.904 \rightarrow 0.913$. It is also not specific to training length. Both arms are already at this value at half of training. The representation itself changes more than in simulation. The target object's share of the variance of h falls from 31% to 4% on the real arm, against 15% to 13% in simulation.

Layer-wise geometry. Computing E_{lang} and E_{pos} at every layer of f_r locates the change (Sec. 3.3, Fig. S6). In layers 0–4 the three checkpoints nearly coincide, with E_{pos} close to zero. The re-fine-tuned checkpoints separate from base from layer 5 on, and most of the change happens in layers 6–10.

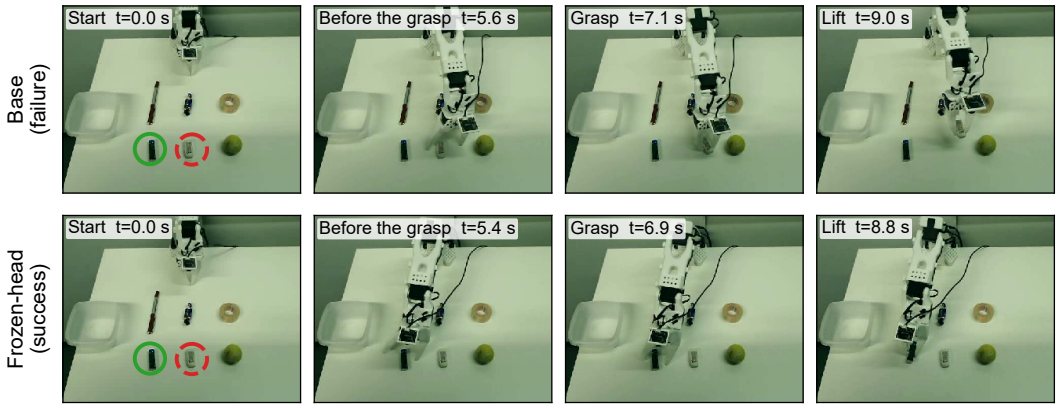


Figure S5: Base (top) and frozen-head (bottom) on the same layout, with the instruction "pick up the battery and put it in the basket". Green: the battery on s_4. Dashed red: s_5, the official slot of this instruction. Base picks up the eraser on s_5 and puts it in the basket. Frozen-head places the battery.

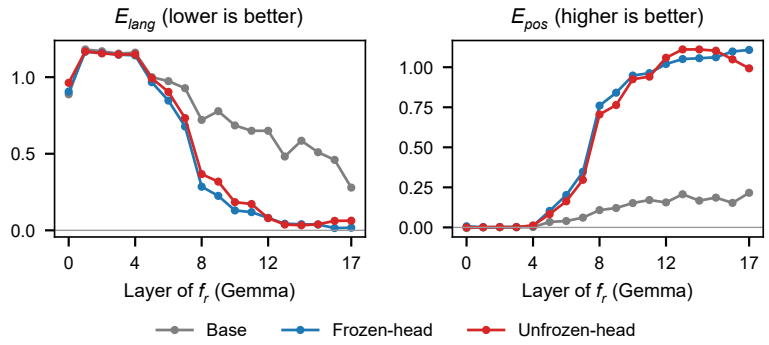


Figure S6: E_{lang} and E_{pos} at every layer of f_r, on the probe set. Table S2 reports the last layer.

The SO-100 experiment reproduces the simulated $\pi_{0.5}$ chain on a physical arm. The comparison is limited by the small test set and one training run per arm.

C UR5 EXTENSION TO OTHER ATTRIBUTES

This section applies the diagnosis of Sec. 5.2 to a second robot, a second policy, and a target attribute other than position. The policy is $\pi_{0.5}$-DROID (Black et al., 2025a; Khazatsky et al., 2024) on a UR5 arm. Besides position and instruction, one test places the target next to objects of the same category. The study reports success rates only. It has no re-fine-tuned checkpoint, so it gives no frozen-head contrast.

Platform. A UR5 arm with an electric parallel gripper faces a tabletop with six objects in a 2×3 layout and a fixed basket (Fig. S7, Table S3). The objects are packaged drinks, a sauce bottle, a chocolate bar and a snack pack. An external camera views the table and a second camera is mounted on the wrist. The instruction is "pick up the {object} and place it in the basket".

Policy and deployment. $\pi_{0.5}$-DROID is fine-tuned on the Berkeley Autolab UR5 dataset of Open X-Embodiment (Open X-Embodiment Collaboration et al., 2024) and on 150 demonstrations recorded on this platform. The policy runs on a local workstation, and a control script sends it the two camera images, the joint and gripper state, and the instruction. Each returned action passes a runtime check before it is sent to the arm. The check rejects invalid values, clips each translation step to 0.03 m and each rotation step to 0.20 rad, keeps the tool inside the tabletop workspace and above a minimum

height, and thresholds the gripper command. An operator supervises every trial with the emergency stop.

Component	Configuration
Robot	UR5 with EPG electric parallel gripper, RTDE/URScript over Ethernet
Cameras	two Intel RealSense D435, external view and wrist view
Workspace	six objects in a 2×3 layout, fixed basket
Policy	$\pi_{0.5}$-DROID, fine-tuned on Berkeley Autolab UR5 and 150 local demonstrations
Safety	0.03 m translation and 0.20 rad rotation per step, workspace and height limits, emergency stop
Inference	local workstation, one RTX 4090-class GPU
Training	$4\times$ V100

Table S3: Hardware and deployment of the UR5 experiments.



(a) Platform (b) Wrist view (left) and external view (right)

Figure S7: UR5 setup. Six objects in a 2×3 layout and a fixed basket.

Tests. Each test changes one factor and keeps the others fixed.

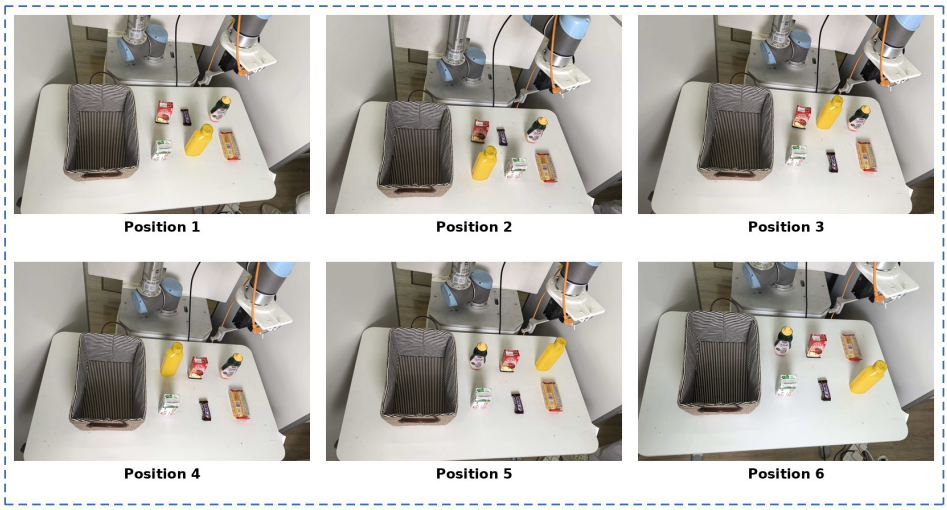
- **Position.** For each target object, the instruction is fixed and the target is swapped with the other objects, so it appears at different positions. The test asks whether the arm follows the target to its new position. Six targets over six layouts give 36 trials.
- **Attribute.** The scene contains objects of the same category, and the instruction names one of them, with a relative position where the category is shared (e.g., "the milk on the right"). The test asks whether the policy binds the instruction to the named object when category alone does not identify it. 18 trials.
- **Instruction.** The layout is fixed and only the target named in the instruction changes. The test asks whether the policy uses the instruction to select the object. 36 trials.

A trial succeeds only if the arm grasps the instructed object and places it in the basket, as in simulation. Figure S8 shows the three tests.

Results. Success stays below one half in all three tests (Table S4). It is 44.4% when the target changes position, 33.3% when the instruction changes, and 22.2% when the target shares its category with a distractor. These rates agree with the simulated finding that a released policy often fails to act on the named entity. Success rates alone do not locate the failure in V-L or in (VL)-A, and the study has one policy, few trials, and no re-fine-tuned contrast. It shows that the protocol transfers to another robot and to attributes other than position.

Test	Changed factor	Success / trials	SR (%)
Position	target position, instruction fixed	16 / 36	44.4
Attribute	same-category distractors	4 / 18	22.2
Instruction	named target, layout fixed	12 / 36	33.3

Table S4: UR5 results with $\pi_{0.5}$-DROID.



(a) Position: six layouts with the instruction fixed





(b) Attribute



(c) Instruction

Figure S8: The three UR5 tests. (b) shows three frames of one rollout for each of two instructions, and (c) three frames of one rollout.

D Design insights for Countering Spurious Correlations

We further examine architectural and training characteristics associated with stronger recovery from spurious correlations, focusing on $\pi_{0.5}$. Architecturally, $\pi_{0.5}$ combines a pretrained vision-language backbone with a dedicated flow-matching action expert Black et al. (2025a). In our experiments, adapting the vision-language pathway while freezing this expert increases success from 21.9% to 95.5%, with only a marginal additional gain to 96.0% when the expert is also updated. This suggests that its existing action-generation capability can support successful execution once the upstream representation is appropriately adapted. Its training recipe offers two plausible explanations for this adaptability. First, heterogeneous co-training combines experience across robots and environments with vision-language supervision, including object localization and semantic subtask prediction. Such supervision may provide transferable semantic and spatial knowledge that facilitates learning the correspondence between instructions and current object locations. Second, the released model uses knowledge-insulated pretraining, which blocks gradients from the continuous action expert into the VLM backbone while retaining discrete action prediction as a representation-learning objective Black et al. (2025a). This design may preserve useful vision-language knowledge while adapting representations to control, making subsequent correction of shortcut reliance more effective. Together, these observations point to **the combination of transferable representations, reusable action-generation capabilities, and training objectives that preserve their compatibility as promising ingredients for mitigating spurious correlations.** These factors remain plausible explanations for the observed recovery, rather than individually established causes, since our experiments do not independently ablate architecture and pretraining.

E EVALUATED MODELS

E.1 VISION FEATURE EXTRACTION IN V-L BINDING

OpenVLA / OpenVLA-OFT / UniVLA / MemoryVLA. These models employ a dual-encoder design for visual processing, combining features from DINOv2 and SigLIP. The extracted representations are concatenated along the feature dimension and subsequently projected into a unified visual embedding space via an image projector. We use these projected embeddings as the visual representations for computing visual binding metrics.

Spatial-Forcing. Spatial-Forcing adopts a vision encoder further enhanced with VGGT-based supervision (Wang et al., 2025b). The resulting visual features are mapped through an image projector to obtain the final visual embeddings, which are used for evaluation in visual binding.

X-VLA. X-VLA utilizes a dual-attention Transformer to encode visual inputs, followed by early fusion with language tokens. We extract the visual features prior to fusion as the representations for visual binding analysis.

DreamVLA. DreamVLA employs a masked transformer diffusion model as its vision encoder. The encoded visual features are projected into embedding space via an image projector. These embeddings are used to measure visual variation in the visual binding task.

RynnVLA-002. RynnVLA-002 follows a world-model-based paradigm, where an image tokenizer converts visual observations into discrete tokens shared between the VLA model and the world model. We directly use these visual tokens for computing visual binding metrics.

FLOWER. FLOWER is built upon a pruned Florence-2-L VLM (Xiao et al., 2024). Visual inputs are processed through the vision transformer encoder and fused with language tokens. We extract the encoder-level visual features for visual binding evaluation.

VLA-JEPA. VLA-JEPA leverages Qwen3-VL (Bai et al., 2025) as its backbone. Its vision encoder follows the SigLIP-2 architecture and is a spatiotemporal Transformer with multi-scale feature extraction, combining learnable and rotary positional encodings along with patch merging for compact representations. These representations are used to compute visual binding metrics. V-JEPA2 encodes future frames only as training targets and is not used at inference.

π_0 / $\pi_{0.5}$. These models adopt PaliGemma-3B (Beyer et al., 2024) as the VLM backbone, which pairs a SigLIP vision encoder with the Gemma-2B language model. We use the output of the projector that maps SigLIP features to the Gemma width as the visual representations for visual binding analysis.

Table S5 summarizes the architectural configurations of the twelve evaluated VLA models, providing the context for our model-specific visual feature extraction in visual binding.

E.2 PER-MODEL DETAILS

All twelve models are evaluated from released LIBERO checkpoints, without further fine-tuning. Table S6 lists each checkpoint, the LIBERO suites it was trained on, and its success rate on official LIBERO-Object. The reported rate is taken from the original work. The reproduced rate is measured by us with the model's official evaluation code.

Evaluation interface. Each model runs through an adapter that follows its official LIBERO evaluation code. Observation preprocessing, action-chunk execution, and action un-normalization are taken from that code without change. Un-normalization uses the LIBERO-Object statistics stored with the checkpoint. The layouts, instructions, and success test are those of Appendix A. OpenVLA-OFT and Spatial-Forcing run with the transformers fork required by the official OpenVLA-OFT code, which makes attention bidirectional for parallel action decoding. X-VLA predicts absolute end-effector poses, so its controller runs in absolute mode. X-VLA also keeps the 800-step horizon of its official LIBERO client and ends an episode at the first success.

Readout points. The probes and the representational geometry read the pooled representation $h^{(\ell)}$ of f_r (Sec. 3.3) for the seven models of Table 3. Table S7 gives the layer and the tokens of each readout. R^2_{pos} reads $h^{(0)}$, the visual tokens at the input of f_r. R^2_{lang} reads the last layer of f_r. E_{lang} and E_{pos} read the last-layer tokens that f_r passes to f_a. When f_r and f_a share one network, as in OpenVLA,

Model	#Params	VLM Backbone	Vision Encoder	Robot Pretrain Corpus	Action chunk	Backbone update	Cams
			Single-stream				
OpenVLA	7B	Prismatic-7B (Llama-2)	SigLIP+DINOv2	OXE	1	Full FT	1
OpenVLA-OFT	7B	Prismatic-7B	SigLIP+DINOv2	OXE → LIBERO (FT)	8	LoRA	2
UniVLA	7B	Prismatic-7B	SigLIP+DINOv2	OXE + Ego4D + GNM	4	LoRA	1
			Hierarchical				
X-VLA	0.9B	Florence-2-L	DaViT (Florence)	DROID + RoboMind + AgiBot	30	LoRA	2
FLOWER	0.95B	Florence-2-L (pruned)	DaViT (Florence)	OXE-soup (8 datasets)	10	Full FT	2
π_0	3.3B	PaliGemma-3B	SigLIP-So400m	Pi internal + OXE	50	Full FT	2
$\pi_{0.5}$	3.3B	PaliGemma-3B	SigLIP-So400m	Pi mobile + OXE + Web	50	Full FT	4
MemoryVLA	7.3B	Prismatic-7B	SigLIP+DINOv2	OXE → Bridge + RT-1 + LIBERO	16	Full FT	1
			World-model / representation-augmented				
DreamVLA	0.6B	GPT-2 medium (Seer)	MAE + CLIP-text	CALVIN + DROID	3	Partial freeze	2
Spatial-Forcing	7B / 3.3B†	OpenVLA-OFT / π_0 base	inherits base	LIBERO / RoboTwin	8 / 50	LoRA	2
RynnVLA-002	∼7B	Chameleon	VQ-GAN tokenizer	LIBERO + SO100 (no pretrain)	5–10	Full FT	2
VLA-JEPA	∼2B	Qwen3-VL-2B	SigLIP-2 (V-JEPA2 train only)	SSv2 (video) + DROID	7	Full FT	3

Table S5: Configuration of the twelve evaluated VLAs, grouped by family (row groups, as in Fig. 1): OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025), UniVLA (Bu et al., 2025), X-VLA (Zheng et al., 2026), FLOWER (Reuss et al., 2025), π_0 (Black et al., 2025b), $\pi_{0.5}$ (Black et al., 2025a), MemoryVLA (Shi et al., 2026), DreamVLA (Zhang et al., 2025a), Spatial-Forcing (Li et al., 2026a), RynnVLA-002 (Cen et al., 2025), and VLA-JEPA (Sun et al., 2026). OXE denotes Open X-Embodiment (Open X-Embodiment Collaboration et al., 2024), and DROID denotes the dataset of Khazatsky et al. (2024). Model-name color marks the action-head type: **AR-token** (autoregressive discretized actions), **Continuous** (diffusion or flow-matching head), **Latent-pred** (latent-action or future-frame prediction). **Action chunk**, **Backbone update**, and **Cams** report future actions per forward pass, VLM update strategy, and camera count. Within cells, + combines components, → marks pretrain-to-finetune, / separates alternative variants. Only $\pi_{0.5}$ co-trains on internet VL data. No model applies explicit object-reposition or action-side augmentation.†Spatial-Forcing adds +12M params on the OpenVLA-OFT base and +8M on the π_0 base.

Model	Released checkpoint	Training suites	Reported	Ours
OpenVLA	`openvla/openvla-7b-finetuned-libero-object`	Object	88.4	85.4
OpenVLA-OFT	`moojink/openvla-7b-oft-finetuned-libero-object`	Object	98.4	97.2
UniVLA	`qwbu/univla-7b-224-sft-libero` (`univla-libero-object`)	Object	96.8	96.4
X-VLA	`2toINF/X-VLA-Libero`	All four	98.6	98.2
FLOWER	`mbreuss/flower_libero_object`	Object	99.1	99.8
π_0	openpi `pi0_libero`	All four	98.8	97.8
$\pi_{0.5}$	openpi `pi05_libero`	All four	98.4	98.2
MemoryVLA	`shihao1895/memvla-libero-object`	Object	98.4	97.4
DreamVLA	`WenyaoZhang/DreamVLA` (`libero_obj_94.pth`)	LIBERO-90, then Object	94.0	96.5
Spatial-Forcing	`haofuly/spatial_forcing_7b_finetuned_libero_object`	Object	99.6	99.2
RynnVLA-002	`Alibaba-DAMO-Academy/RynnVLA-002` (`VLA_model_256/libero_object`)	Object	99.8	95.8
VLA-JEPA	`ginwind/VLA-JEPA` (`LIBERO/VLA-JEPA-LIBERO.pt`)	All four	99.6	99.4

Table S6: Released checkpoints of the twelve evaluated VLAs. Checkpoints are Hugging Face repositories, with the file or subfolder in parentheses. The π_0 and $\pi_{0.5}$ checkpoints are released in openpi (Physical Intelligence, 2025). **Reported** and **Ours** are success rates (%) on official LIBERO-Object. **Ours** uses each model's official evaluation code with 10 tasks and 50 episodes per task, or 20 per task for DreamVLA as fixed in its official code.

this is the state at the position that emits the first action token. The other five models enter only the success-rate comparison, and their f_v, f_r, and f_a follow Table S5.

F Fine-tuning hyperparameters

Official recipe. Each representative uses that model's official LIBERO fine-tuning recipe. We first reproduce the official LIBERO-Object fine-tune, from the pretrained checkpoint to the released LIBERO-Object checkpoint. On official LIBERO-Object, this reproduction matches the released

Model	R^2_{pos}	R^2_{lang}	E_{lang}, E_{pos}
OpenVLA	projector output	LLM last layer, first-action position	
UniVLA	projector output	LLM last layer, latent-action tokens	
X-VLA	Florence-2 image encoding	Florence-2 encoder output	last block, 30 action tokens
π_0, $\pi_{0.5}$	projector output	Gemma layer 17 (last), main-camera image tokens	
MemoryVLA	projector output	cognition token	
VLA-JEPA	projector output	Qwen3-VL last layer, text tokens	32 embodied-action tokens

Table S7: Readout points of the linear probes and the representational geometry for the seven models of Table 3. A merged cell means that R^2_{lang} and the geometry share one readout point. R^2_{pos} uses the main camera only. Token sets are mean-pooled.

checkpoint under the same evaluation harness. We then start from that released checkpoint and apply the same hyperparameters to the demonstrations of Sec. 4.2. Whether the update is LoRA or full, and the learning rate, batch, and schedule, follow the official recipe. They are not chosen per model for this benchmark. Four deviations are listed below.

Starting weights and the two arms. Both arms start from that released checkpoint. The frozen-head arm freezes f_a and updates the vision-language pathway. The unfrozen-head arm uses the same data and the same hyperparameters, and also updates f_a. The two arms differ only in whether f_a is updated. Table S8 names the frozen module. OpenVLA has no detachable action head. Actions are discrete tokens from the autoregressive VLM. It therefore has only the unfrozen-head arm. Freezing is verified at training start. $\pi_{0.5}$ runs a parameter-wise freeze check before training and also freezes the Gemma-2B base. UniVLA asserts in the training script that the action decoder has zero trainable parameters and that the upstream pathway has a nonzero count. X-VLA writes the four parameter-group learning rates to the log every 20 steps. MemoryVLA prints the number of frozen parameters at launch. π_0 uses the openpi (Physical Intelligence, 2025) `freeze_filter` path regex. VLA-JEPA records the freeze in the config key `freeze_modules`.

Data. All seven representatives use the 1897 demonstrations of Sec. 4.2 (Appendix A.4). Target positions are sampled at random over the full workspace. Demonstrations whose target falls inside the held-out circle of Sec. 4.2 are removed.

Model	Frozen f_a	Update	LR	Batch	Steps	Demos
OpenVLA	–	LoRA r=32	5×10^{-4}	128	20k	1897
UniVLA	action decoder	LoRA r=32	3.5×10^{-4}	128	30k	1897
X-VLA	action head	full, grouped LR	10^{-4} / 10^{-5}	128	20k	1897
π_0	action expert	full	2.5×10^{-5}	32	50k	1897
$\pi_{0.5}$	action expert	LoRA r=16	2.5×10^{-5}	16	6k	1897
MemoryVLA	DiT-L	full	2×10^{-5}	256	14k	1897
VLA-JEPA	action model	full, grouped LR	10^{-5} / 10^{-4} / 3×10^{-5}	256	22k	1897

Table S8: Fine-tuning hyperparameters of the checkpoints in Sec. 5.2. Steps are the reported checkpoint. X-VLA uses 10^{-4} on core and head, 10^{-5} on VLM and soft prompts. VLA-JEPA uses 10^{-5} on the Qwen interface, 10^{-4} on the action head, and 3×10^{-5} on the remaining trainable parameters.

Effective batch is $4\times16\times2$ on OpenVLA, $8\times8\times2$ on UniVLA, 4×32 on X-VLA, and 8×32 on MemoryVLA and VLA-JEPA. Five PyTorch models use bf16 mixed precision. For π_0 and $\pi_{0.5}$, computation is in bf16, trainable parameters remain fp32, and frozen parameters are cast to bf16. Images are 224 px. The VLA-JEPA video branch uses 256. Seeds are 42 for π_0, $\pi_{0.5}$, and VLA-JEPA, 0 for X-VLA, and each repository default otherwise. In total, 13 runs and about 1.1×10^3 GH200 hours. Optimizer details not listed in Table S9 are as follows. LoRA on OpenVLA is all-linear. LoRA on $\pi_{0.5}$ is attn+ffn with α=16. UniVLA uses weight decay 10^{-3}. X-VLA uses AdamW (Loshchilov & Hutter, 2019) β=$(0.9, 0.95)$, weight decay 0.01, and cosine decay to $0.1\times$. π_0 uses weight decay 10^{-10} and EMA 0.99. $\pi_{0.5}$ turns EMA off. MemoryVLA uses weight decay 0 and gradient clip 1.0.

Model	Init	Schedule	Hardware
OpenVLA	released Object	AdamW, constant, no warmup	4×GH200, 7.0 h
UniVLA	univla-libero-object	AdamW, StepLR $\times 0.1$ at $0.8T$	8×GH200, 13.5 / 13.0 h
X-VLA	X-VLA-Libero	cosine, freeze 1k, warmup 1k	4×GH200, about 6 h
π_0	pi0_libero	cosine, warmup 1k, decay 30k	4×GH200, 4.1 / 3.1 h
$\pi_{0.5}$	pi05_libero	cosine, warmup 200, decay 8k	1×GH200, 1.2 h
MemoryVLA	memvla-libero-object	constant, no warmup, FSDP	8×GH200, 10.2 h
VLA-JEPA	VLA-JEPA-LIBERO	cosine, warmup 5k	8×GH200, 36 h limit

Table S9: Initialization, schedule, and wall-clock of the same runs.

VLA-JEPA uses AdamW $\beta{=}(0.9, 0.95)$, weight decay 10^{-8}, gradient clip 1.0, and cosine decay to 10^{-6}.

Reported checkpoints. Checkpoints are not selected by evaluation score. Table S8 lists the reported step. X-VLA is reported at 20k of a 60k official schedule. MemoryVLA hit a disk-quota failure at 15.6k steps. The step-15000 file was corrupt, and the last clean checkpoint is 14k, where the training loss had reached 0.002. Both VLA-JEPA arms were stopped at the 36-hour wall-clock limit, at 22k steps. For π_0, the 30k checkpoint was rotated out by the retention policy, leaving the round checkpoints from 5k to 50k, so all metrics are locked at 50k.

Deviations from the official recipe. Relative to each model's official LIBERO recipe, there are four deviations. UniVLA's official fine-tuning script randomly initializes the action decoder. Freezing a randomly initialized head is not meaningful, so we load that head from the released weights and then freeze it. MemoryVLA's training script has no switch to freeze the action head. We added a freeze call gated by an environment variable and left every other hyperparameter unchanged. For π_0 we set `extra_delta_transform=False`, because LIBERO actions are already deltas. π_0 is trained to 50k, past its 30k cosine schedule. From 30k onward the learning rate stays at the floor of $2.5{\times}10^{-6}$.

Asymmetry of the starting weights. The seven released checkpoints do not cover the same data. OpenVLA, UniVLA, X-VLA, and MemoryVLA start from official LIBERO-Object weights. π_0 and $\pi_{0.5}$ start from the only LIBERO checkpoint released by openpi, which was trained on all four suites. VLA-JEPA's released checkpoint likewise covers all four suites, with 1693 demonstrations. The last three models start from a broader checkpoint than the first four. Each arm is trained once. Differences between arms are not accompanied by multi-seed error bars.

G Position Information in the Visual Encoder

The visual encoder is the entry point for all scene information in a VLA model: any failure in visual representation propagates irreversibly to language understanding and action generation. We examine whether the visual module produces representations that faithfully encode object position and identity. VLA models predominantly utilize ViT-based visual encoders (Kim et al., 2024; Black et al., 2025b), including SigLIP (Zhai et al., 2023) and DINOv2 (Oquab et al., 2024). Visualizing attention maps across encoder depth reveals a consistent pattern: semantic coverage of task-relevant objects peaks at approximately 60% of encoder depth, after which attention narrows progressively to the robot arm and its contact object (Fig. S9).

This behavior raises two competing hypotheses:

H1: *Object representations are discarded in deeper encoder layers.* ***H2:*** *They are retained in deeper encoder layers.*

To discriminate between H1 and H2, we apply a linear probe at each encoder depth. Specifically, we probe each of the 27 layers of the SigLIP encoder of $\pi_{0.5}$. For each object, we fit a linear probe (Alain & Bengio, 2017) by ridge regression (Hoerl & Kennard, 1970) on the first frames of the 180 layouts to predict the 2D ground-truth position of the object. We score it by five-fold cross-validation, so the position orders of test trials also appear in training, and average R^2 over the six objects. We report the best of every fourth image token, and the mean over tokens. We use a linear probe for

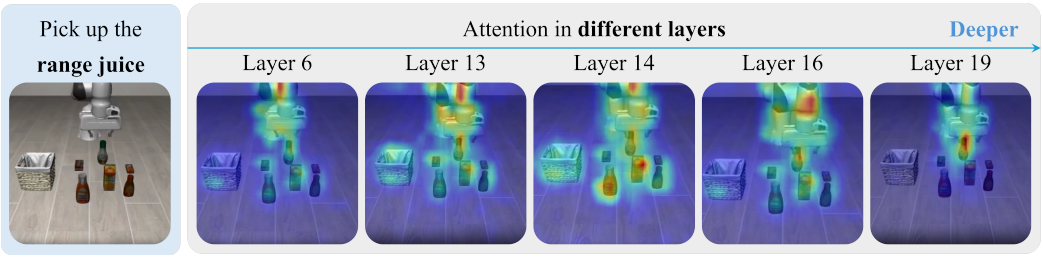


Figure S9: Attention maps at increasing encoder depth (left to right). Semantic coverage of task-relevant objects peaks at ≈60% depth; deeper layers narrow attention to the robot arm.

its conservatism: if positional information is not linearly decodable from a representation, no linear downstream module can recover it. Figure S10 plots R^2 against encoder depth. R^2 rises over the first half of the encoder, peaks at 0.74 at layer 13, and stays between 0.70 and 0.74 up to the last layer, ruling out H1. This complements VAT's layer-skipping results, which find control-relevant features throughout the ViT hierarchy (Li et al., 2025). Decodability should not, however, be conflated with visual alignment: action fine-tuning can still degrade pretrained visual representations (Kachaev et al., 2026), while attention concentration can be distorted by high-norm artifact tokens (Darcet et al., 2024). H2 holds: deeper layers retain the positional information.

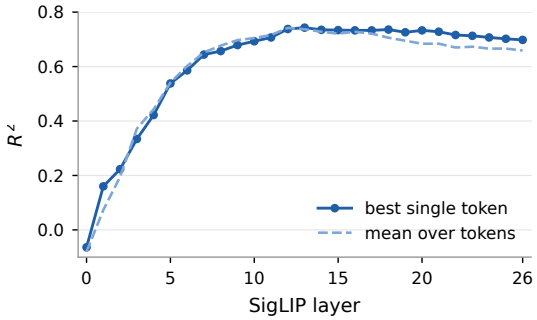


Figure S10: Per-layer linear probe R^2 for the target position in the SigLIP encoder of $\pi_{0.5}$ (five-fold cross-validation, mean over six objects). Solid: best single token. Dashed: mean over tokens.

Replacing the language instruction while holding the visual input fixed leaves attention maps unchanged, confirming that the visual encoder operates independently of language conditioning at this stage.

H Details of Evaluation Metrics

The constructions below are the score definitions used in Sec. 4.3 and Table 2. The probe and geometry procedures remain in Sec. 3.3.

H.1 Linear-probe scores

We score the position probe of Sec. 3.3 with the coefficient of determination

$$R^2_{\text{pos}} = 1 - \frac{\sum_k \|\mathbf{p}_i^{(k)} - \hat{\mathbf{p}}_i^{(k)}\|^2}{\sum_k \|\mathbf{p}_i^{(k)} - \bar{\mathbf{p}}_i\|^2}, \tag{S1}$$

where $\bar{\mathbf{p}}_i$ is the mean ground-truth position of object o_i over the test set. $R^2_{\text{pos}} = 1$ is a perfect readout, $R^2_{\text{pos}} = 0$ matches predicting the mean position, and $R^2_{\text{pos}} < 0$ means the encoded position conflicts with the true one. For the language probe we apply the same formula to a one-hot encoding of o_i and report R^2_{lang}.

H.2 REPRESENTATIONAL-GEOMETRY SCORES

The main effects E_{lang} and E_{pos} are defined in Eq. 5. Each E averages two cosine-similarity differences, so it lies in $[-2, 2]$, not in $[0, 1]$. A value above 1 means same-factor pairs are more aligned, and different-factor pairs more opposed, than one unit of cosine similarity. A grounded representation has high E_{pos} and low E_{lang}.

H.3 TRAJECTORY DIVERSITY DETAILS

This subsection specifies the trajectory diversity score introduced in Sec. 4.3 and reported in Table 3b.

Trajectory Representation. We use the three translation channels of each recorded action sequence, excluding rotation and gripper commands. A sequence contains the commands executed from the first policy step, after the 10 initial no-op steps, until the episode ends or reaches its horizon (400 steps, 800 for X-VLA, Appendix E.2). Let $\mathbf{u}_t \in \mathbb{R}^3$ be the translation command at step t. For models that output delta commands, we form $\mathbf{q}_t = \sum_{s=1}^{t} \mathbf{u}_s$; for X-VLA, which outputs absolute positions, we use $\mathbf{q}_t = \mathbf{u}_t$. Both paths are shifted to their first point,

$$\tau_t = \mathbf{q}_t - \mathbf{q}_1, \qquad t = 1, \ldots, T. \tag{S2}$$

The recorded coordinate scale is retained: delta commands use controller input units, while X-VLA uses absolute position coordinates. These paths describe the model's translation commands, not measured end-effector motion. We append a uniform time coordinate to obtain $\tilde{\tau}_t = (\tau_t, (t-1)/(T-1)) \in \mathbb{R}^4$ and use the piecewise-linear path through these points. Each path starts at the origin.

Signature Kernel. Following the signature construction used in FAKTUAL (Sirigiri et al., 2026; Chevyrev & Kormilitzin, 2016), we concatenate the signature levels through depth $M = 3$,

$$\phi(\tau) = S^{\leq M}(\tilde{\tau}) = \big(1, S^1(\tilde{\tau}), \ldots, S^M(\tilde{\tau})\big), \tag{S3}$$

where $S^k(\tilde{\tau}) \in (\mathbb{R}^4)^{\otimes k}$ is the k-th iterated integral, flattened into a vector. The level-zero constant is included, and all levels have unit weight. We compute the cosine Gram matrix

$$K_{ij} = \frac{\langle \phi(\tau_i), \phi(\tau_j) \rangle}{\|\phi(\tau_i)\| \, \|\phi(\tau_j)\|}, \qquad K_{ii} = 1. \tag{S4}$$

There is no additional level weighting or path-scale normalization.

Variant Diversity. For a group of n paths, let

$$\rho = \frac{K}{\operatorname{tr} K} = \frac{K}{n}, \qquad \lambda_k = \lambda_k(\rho). \tag{S5}$$

The Gram matrix is positive semidefinite with unit diagonal, so $\lambda_k \geq 0$ and $\sum_k \lambda_k = 1$. We report the exponential of its von Neumann entropy (Sirigiri et al., 2026), i.e., the Vendi score (Friedman & Dieng, 2023),

$$H = -\sum_{k=1}^{n} \lambda_k \log \lambda_k, \qquad D_{\text{var}} = \exp(H) \in [1, n], \tag{S6}$$

using natural logarithms and $0 \log 0 = 0$. Numerically, eigenvalues no larger than 10^{-12} are omitted from the entropy sum. This is the effective rank of K. Identical signature features give $D_{\text{var}} = 1$, while $K = I_n$ gives $D_{\text{var}} = n$. Higher values indicate greater diversity in signature space, not task success.

Evaluation Protocol. Each group contains $n = 6$ rollouts from the same scene, one for each target instruction. Successful and failed rollouts are both included. We use the full recorded sequences at their original lengths, without segmentation, subsampling, or truncation to a common length. Only complete groups with six valid sequences of at least two steps are evaluated. We compute the score separately for each scene and then average:

$$\overline{D}_{\text{var}} = \frac{1}{B} \sum_{b=1}^{B} \exp(H_b). \tag{S7}$$

Table 3b reports this mean over all $B = 180$ scenes, separately for each checkpoint. The per-scene group size remains six in every case. Scores are interpreted alongside SR, under the same task, group size, and path construction. Signature features depend on coordinate scale and path extent; in particular, the absolute-position representation of X-VLA limits direct comparisons of its score magnitude with those of delta-command models.

I FAILURE DECOMPOSITION

This appendix assigns each failed rollout to the task stage at which it stops. The task has three stages: reach the named object, lift it, and place it in the basket.

Rollouts. We use the rollouts of each checkpoint from which SR and D_{var} in Table 3b are computed. The outcome of each rollout is taken from the evaluation log. During each rollout, we also record the end-effector position and the positions of the six objects at every step.

Reached object. Let $\mathbf{e}_t$ be the end-effector position, $\mathbf{x}_t^{(i)}$ the position of object o_i, and B the planar footprint of the basket dilated by 2 cm. The decision window ends at the first step at which any object has risen more than $\delta = 3$ cm above its initial height:

$$T^* = \min\{t : \exists i,\ x_{t,z}^{(i)} - x_{1,z}^{(i)} > \delta\}. \tag{S8}$$

If no object rises, the window is the whole rollout. The reaching step is the lowest end-effector position in the window outside B:

$$t^* = \arg \min_{t \le T^*,\ (e_{t,x}, e_{t,y}) \notin B} e_{t,z}. \tag{S9}$$

The reached object is the object closest to the end effector at that step:

$$\hat{g} = \arg\min_i \left\| \mathbf{x}_{t^*}^{(i)} - \mathbf{e}_{t^*} \right\|. \tag{S10}$$

Classes. Each rollout is assigned to the first class it meets, in this order. *Success*: the named object ends in the basket. *Wrong target*: $\hat{g}$ is not the named object. *Not grasped*: $\hat{g}$ is the named object, but it never rises by δ. *Not placed*: the named object rises by δ but does not end in the basket. The four classes are mutually exclusive and cover every rollout. For wrong-target rollouts, we also record whether $\hat{g}$ is the object in the first slot of the layout. Under a uniform choice among the five other objects, this fraction is 0.20.

Results. Figure 4 shows the decomposition for every checkpoint. On the base checkpoints, the failures are wrong-target failures. Wrong target covers 73–83% of rollouts, while not grasped and not placed together cover at most 7%. Among the wrong picks, 41–52% land on the object in the first slot, against 20% under uniform choice. After frozen-head fine-tuning, wrong target falls to at most 15% on every model. The remaining failures are mostly not grasped: 69% on X-VLA, 35% on UniVLA, and 32% on VLA-JEPA. These are (VL)-A failures: the models reach the named object and fail to lift it, which accounts for the gap between E_{pos} and SR in Sec. 5.2. After the action head is unfrozen, the decomposition does not improve uniformly. On MemoryVLA, wrong target rises from 4% to 13%. On X-VLA, not grasped stays at 68%.

Best and worst fine-tuned checkpoints. $\pi_{0.5}$ and VLA-JEPA show the two ends of Fig. 4. On $\pi_{0.5}$, the frozen-head checkpoint lowers wrong target from 73% to 1% and reaches 95.5% SR (Table 3b). The action head is not updated, so the repair is in V-L, and the released action head executes the repaired binding. On VLA-JEPA, the frozen-head checkpoint reaches 50.2% SR, and failures remain at both stages. Wrong target stays at 15%, a residual V-L failure. Not grasped is 32%, a (VL)-A failure. Unfreezing the head raises SR to 70.2% and lowers not grasped to 8%, but wrong target stays at 16%.

A plausible explanation for the strong adaptation of $\pi_{0.5}$ lies in its heterogeneous co-training (Black et al., 2025a) and the knowledge-insulated pretraining of the released model (Driess et al., 2025). Broader pretraining coverage may reduce reliance on fixed object–layout associations, while knowledge insulation may preserve semantic representations for subsequent adaptation to control. These

factors could make targeted fine-tuning more effective at aligning instructions with current object locations rather than memorized scene configurations. This interpretation remains a hypothesis: our results neither establish shortcut-free behavior nor isolate the causal contributions of pretraining data and training mechanisms.