HEARD BUT NOT HEEDED: LANGUAGE SURVIVES IN A VLA'S REPRESENTATIONS BUT NOT ITS ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A fine-tuned Vision-Language-Action (VLA) model that scores 96.85% on the LIBERO manipulation benchmark stops following language instructions when given a scene outside its training distribution. We diagnose this failure through three measurements: (1) the model still internally represents which object was named (82.8% accuracy vs 25% baseline), (2) attention roughly doubles when the object is mentioned, yet (3) the policy grasps the wrong object. This dissociation understanding language and attending correctly, but choosing the wrong action shows that instruction-following failures need not stem from language misunderstanding or visual misidentification. We validate this dissociation with a within-distribution control experiment. Attention allocation to the target object is virtually identical in both settings ($\approx 1.6\%$ vs $\approx 1.7\%$ of total attention), yet language-driven attention shifts are five times stronger when behavior fails. This suggests that insufficient attention magnitude cannot account for the behavioral failure. The evidence points to a breakdown at the interface where the model translates visual and language information into motor commands. We introduce object-level behavior metrics (recording which object was actually grasped, rather than only whether the task succeeded) and show that benchmark structural properties make instruction-following failures invisible to standard evaluation metrics.

1 INTRODUCTION

Current Vision-Language-Action (VLA) models work by fine-tuning pretrained models on demonstrations of specific scenes, robots, and task types. This focused training produces strong performance on familiar scenarios, but raises a critical question: what happens when the model encounters a *new* scene it has never seen during training? In real-world deployment, robots will routinely encounter unfamiliar environments. If a model stops following instructions outside its training distribution, it not only acts incorrectly it can appear to succeed when the robot's default backup behavior coincidentally accomplishes the goal. Standard success metrics then fail to detect the failure. This paper documents one such case and investigates where, mechanistically, the model's decision-making breaks down. Understanding these failure modes is essential: it lets us identify and fix the specific architectural or training components responsible, rather than simply retraining on more data.

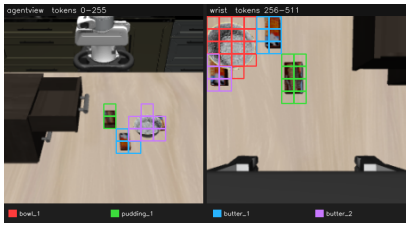


Figure 1: Ground-truth object masks (unseen by the model) on agentview and wrist views, used as probe and attention labels.

Standard evaluations report a model's success rate on benchmark tasks. $\pi_{0.5}$ Intelligence et al. (2025), a state-of-the-art VLA model fine-tuned on LIBERO, achieves 96.85% average success. A metric like this typically suggests the model reliably follows instructions. However, this paper reveals a critical blind spot: move to a single new scene, and the model stops following instructions entirely. Remarkably, this failure is completely invisible to standard success rate metrics and to failure-detection methods that monitor trajectory statistics. The core issue is that success metrics only ask "did the task complete?" but not "did the robot follow the instruction?" When an unfamiliar scene happens to have a default strategy that coincidentally solves the task, the model's failure to follow instructions goes undetected.

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.

We make this explicit by distinguishing two failure modes that standard metrics conflate: (1) reaching for the *wrong object* the model ignored the instruction versus (2) fumbling the *right one* the model understood but lost control. By logging which object the robot actually grasps, we can see that apparently "50% success" actually means the robot executed one fixed strategy repeatedly, succeeding only when that strategy coincidentally matched three different requested goals.

In summary, this paper makes the following contributions.

1. *A controlled demonstration of OOD instruction-following failure.* We show that a fine-tuned VLA policy reliably follows instruction overrides in-distribution ($19/20$ trials) but largely ceases to condition its object selection on language in an out-of-distribution scene, despite retaining substantial task competence.
2. *A representation–attention–action dissociation.* We demonstrate that, under the same OOD condition, the named referent remains linearly decodable from internal representations (82.8% vs. 25% chance), and naming an object reliably increases attention to its image regions, yet the policy still selects the wrong object. These measurements constrain the failure to occur downstream of simple referent recoverability and coarse attention allocation.
3. *Object-level behavioral diagnostics.* We introduce evaluation based on the identity of the object actually grasped, rather than task success alone, revealing cases in which a fixed policy behavior appears partially successful only because it coincides with the requested goal.
4. *A structural analysis of LIBERO's language dependence.* We show that approximately three-quarters of the fine-tuning task distribution can identify the target from scene structure without requiring the instruction, exposing a benchmark property that can mask failures of language-conditioned behavior.

2 Related Work

Vision-Language-Action models are trained on instruction–image pairs and scored by task success Zhou et al. (2025); Black et al. (2024); Liu et al. (2023). That metric asks whether the task completed, not whether the instruction was *used*; the two diverge whenever a scene admits multiple valid strategies.

Language Grounding and Shortcut Learning in VLA Models. High benchmark success often comes from visual scene recognition: when appearance strongly predicts the task (e.g., a stove is always associated with stove tasks), models can succeed without reading instructions Zhou et al. (2025); Liu et al. (2023). Recent work documents this: LIBERO-Plus Fei et al. (2025) and LangGap Hou and Zhao (2026) expose instruction-following failures and typically attribute them to failures in language understanding or visual perception. These failures occur when benchmarks permit shortcut learning, allowing policies to ignore instruction content and rely instead on visual scene correlations or task identity.

From Behavioral Failure to Mechanistic Diagnosis. We identify a sharper dissociation than prior work: the model successfully *encodes* the instruction, correctly *attends* to the named object, yet still selects the wrong action. This indicates the failure occurs downstream, at the interface where language understanding and visual grounding combine into motor control. We establish this with three non-invasive measurements on the frozen model representation probing, attention analysis, and object-level behavior metrics without any intervention that could alter the model's computation. This localization strategy moves beyond establishing that language can be ignored; instead, it traces where the understanding survives and where behavior breaks down.

Representation Probing, Attention, and Causal Analysis. Representation probing and attention analysis are standard diagnostic tools in interpretability. Probing tests whether information is recoverable from representations Hewitt and Liang (2019); Belinkov (2022); attention analysis measures which regions the model processes Bibal et al. (2022). However, neither method establishes causation. Jain and Wallace Jain and Wallace (2019) and Wiegreffe and Pinter Wiegreffe and Pinter (2019) debate the interpretability of attention weights, while Elazar *et al.* Elazar et al. (2021) demonstrate that properties can be highly decodable yet causally irrelevant to the output. Our findings present

the inverse case: language encoding, visual grounding, and attention all function correctly, yet behavior remains incorrect. This dissociation is precisely what these non-causal methods can reveal: information survives through the model's pipeline, but action selection still fails. Establishing true causation would require intervention via activation patching Meng et al. (2022) or causal mediation analysis Vig et al. (2020), which we reserve for future work. Related work by Feng and Steinhardt Feng and Steinhardt (2024) explores binding entities to attributes through added mechanisms; whether similar principles explain our dissociation remains open.

3 METHODS

To localize where language-conditioned behaviour fails, we employ a two-stage diagnostic pipeline. First, we establish a clear behavioural dissociation using controlled in-distribution and out-of-distribution tasks that differ only in whether the named object is the policy's default action. Second, we decompose the three-stage information flow language encoding, visual grounding, and action selection by measuring what information is *present* in the model's key/value cache and where the model *attends*, without any intervention that could alter behaviour. This forward-pass-only approach preserves the frozen network's original computation and rules out explanations progressively by elimination.

3.1 BACKGROUND AND PLATFORM

3.1.1 THE MODEL ARCHITECTURE AND LANGUAGE INTERFACE

The model $\pi_{0.5}$ has a modular architecture: an 18-layer stack containing two specialized components that process different types of information. A language-vision component (the prefix expert) processes images and text, while an action component (the action expert) converts that information into robot commands. Critically, these two components communicate through a single, well-defined channel: a cache of processed information (the key/value cache) that gets updated at each layer.

This architectural choice is fortunate for our investigation: if the action expert can access information about which object was named, that information must pass through this cache. It cannot bypass this channel. Therefore, measuring whether object identity information is present in this cache directly answers the question: "does the action expert have access to the instruction's meaning?"

A practical side note: the model has no separate language output at inference time. During evaluation, the language-processing component is executed, but its outputs are discarded except for this communication cache. We cannot simply ask the model "what instruction did you receive?" and check whether it answers correctly. This communication-channel-only design is part of why language-following failures can be invisible to standard debugging methods.

3.1.2 BENCHMARK AND CONTROLLER

We evaluate the model using LIBERO Liu et al. (2023), a standard benchmark suite for robot manipulation. The model was trained on four task collections totaling roughly 50 tasks, but we test on LIBERO's 90-task suite to ensure out-of-distribution evaluation. The model runs on a physics simulator (MuJoCo Todorov et al. (2012)) with a standard robot controller (OSC, operational-space control) that maps desired end-effector poses to joint torques. The controller operates at 20 Hz and executes actions in chunks. Inference time is 80–133 ms per chunk, much faster than the physics simulation, so timing is not a constraint.

To localize the failure, we deploy two non-invasive measurements, both applied to the frozen, trained model without any retraining, intervention, or rollout collection:

- *Measurement 1: What is encoded?* We train a linear decoder to read the model's internal representations and predict which object was named. If it succeeds, the model understood the instruction. If it fails, the model never encoded the meaning.
- *Measurement 2: Where does it attend?* When the robot looks at an image, its attention mechanism assigns weights to different image regions. We measure: does attention increase when an object is mentioned? If yes, the model is routing visual attention correctly to language. If no, weak attention routing might be the problem.

Together, these measurements answer the diagnostic pipeline: (1) Did the model understand language? (2) Did the model look at the right place? (3) If both are true, why did it choose the wrong action? The detailed procedures, control experiments, and limitations are in the Appendix.

4 EXPERIMENT

4.1 THE BEHAVIOURAL DISSOCIATION

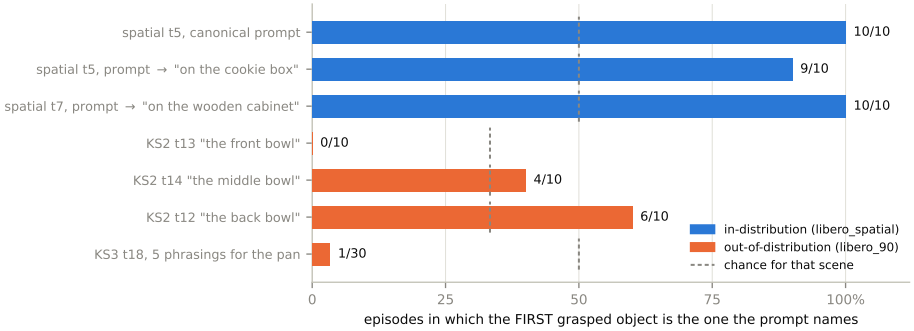


Figure 2: First object grasped, by condition. Blue: in-distribution (prompt moves the target). Orange: OOD (prompt does not). Dashed ticks mark chance (33%). Success omitted (zero by construction).

4.1.1 IN DISTRIBUTION, THE PROMPT SELECTS THE OBJECT

All ten `libero_spatial` tasks share one scene and differ only in which black bowl the instruction names. We overwrote the transmitted instruction of tasks 5 and 7 with the *verbatim training string* of a task naming a different bowl. The image is unchanged; only the sentence moves:

- t5, canonical ("…on the ramekin…") → `bowl_1`, $10/10$;
- t5 → "…on the cookie box…" → `bowl_2`, $9/10$;
- t7 → "…on the wooden cabinet…" → `bowl_2`, $10/10$.

The arm follows the sentence, $19/20$. Note that this is a stronger test than success rate on the same suite, which we show in Sec. 4.6 is not evidence of instruction following at all.

4.1.2 OUT OF DISTRIBUTION, IT DOES NOT

KITCHEN_SCENE2 of `libero_90` holds three identical black bowls distinguished only by position, named "front", "middle" and "back" by three different tasks in the same scene. KITCHEN_SCENE3 holds a frying pan and moka pot; the same scene and initial state are paired with five different instructions naming one object or another. Figures show the actual scenes:

Ten episodes each for KITCHEN_SCENE2 (Table 1):

Averaged over the three tasks this reads as 33% success: "partially working". It is not. It is *one* behaviour (go for the back bowl, $20/30$) scored against three different labels, and the named-object rate $10/30$ is precisely chance. Success tracks only whether the fixed default happens to coincide

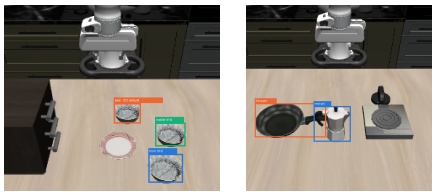


Figure 3: OOD evaluation scenes. Left: KITCHEN_SCENE2 (three identical bowls). Right: KITCHEN_SCENE3 (frying pan vs. moka pot).

Task	Instruction	Named	`bowl_3`*	Success
13	front bowl	0/10	8/10	0/10
14	middle bowl	4/10	6/10	4/10
12	back bowl	6/10	6/10	6/10
	total	**10/30**	**20/30**	10/30

Table 1: KITCHEN_SCENE2. Named-object rate matches three-way guessing. *`bowl_3` is the default.

Task	Prompt transmitted	Succ.	Pan	Moka	Neither
18	"...frying pan..."	1	1	4	1
18	"...pan..."	0	0	5	1
18	"...skillet..."	0	0	4	2
18	"...black pan..."	0	0	6	0
18	"...moka pot..."	0	0	6	0
19	"...moka pot..."	**6**	0	6	0

Table 2: KITCHEN_SCENE3. Same scene, same two objects, same initial states; only the transmitted noun varies. Task-18 rows can succeed only by placing the pan. $n = 6$ per row.

with the goal. The 33% average is a measurement artifact, not partial competence; Sec. 4.6 shows the same structure across the benchmark.

4.1.3 A PAIRED CONTROL: SAME SCENE, ONE NOUN CHANGED

KITCHEN_SCENE3 tasks 18 ("put the frying pan on the stove") and 19 ("put the moka pot on the stove") use the same scene, the same two manipulable objects (`chefmate_8_frypan_1`, `moka_pot_1`) and the same initial states. Only the noun differs. The distributional asymmetry is the point: `libero_10`, a fine-tuning suite, contains *two* moka-pot-on-stove tasks, and no training suite anywhere mentions a frying pan.

Table 2 gives the result. Task 19 scores $6/6$, so the scene, the grasp and the physics are within competence. Across the five task-18 conditions the policy takes the moka pot in 25 of 30 trials under every phrasing, including "black pan", whose colour cue separates a black pan from a silver pot unambiguously. Asking for the moka pot and asking for the black pan produce *identical* behaviour ($6/6$ moka each): the instruction is not merely weighted down, it is not differentiating. At 256×256 the rendered observation shows an unmistakable black pan with a handle beside a visually distinct silver pot, so this is not an asset-legibility problem.

With $n = 6$ per row the single pan grasp under the true prompt is within noise; the defensible contrasts are $25/30$ versus $6/6$ and $1/30$ versus $6/6$.

4.2 DIAGNOSTIC PIPELINE: THREE QUESTIONS, THREE MEASUREMENTS

The rest of this section asks three questions in order: (1) does the model *encode* the named object (linear probe on prefix K/V)? (2) does it *attend* to that referent (paired post-softmax mass)? (3) does it *act* on either (which object is grasped)? Encoding and attention survive distribution shift; action selection does not. This pattern constrains the failure to stages downstream of simple referent decodability and attention allocation.

4.3 WHAT THE MODEL ENCODES: REFERENT INFORMATION REMAINS RECOVERABLE

4.3.1 DESIGN

Does the model's internal representation encode which object the instruction named? We train a linear classifier on each of the 18 layer representations and ask it to predict the named object's site. The scene is OOD, with four objects permuted over four locations (180 samples from 90 frames). Labels are **A** (named object's site), **B** (the other graspable object), and **C** (a never-mentioned distractor). If identity is decodable only at early layers it came from the image; if it appears later, the model built it by combining language with vision. A linear decoder has a single decision boundary and cannot compute that interaction itself, so success means the network already did.

4.3.2 RESULT

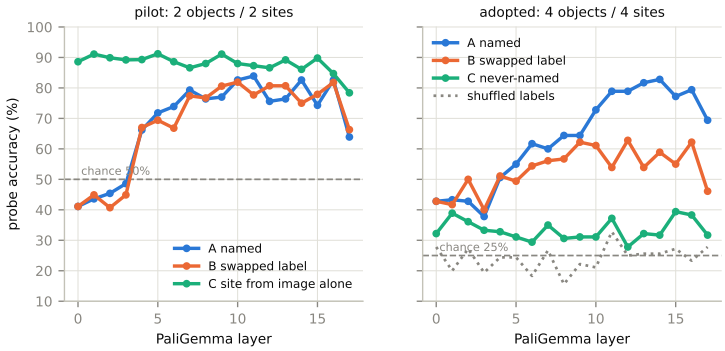


Figure 4: Linear probe by layer. Right: 4-site design. A(named) 82.8% exceeds B(swapped) 62.8%; A and B are equal at layer 0, so the binding is constructed by the network. Left: 2-site pilot, where site is decodable from the image alone (91%).

The decoder PaliGemma Beyer et al. (2024) identifies the named object (A) at 82.8%, against 62.8% for the swapped control (B) and 39.4% for the never-mentioned distractor (C); empirical majority-class baseline is 27%. At layer 0, before any processing, A and B are identical (42.8%). The gap opens around layer 5 and peaks at layers 11–14: the model *constructs* the binding; it is not read off the image. Alternate pooling gives the same pattern (81.1%/63.3%/41.1%).

4.3.3 Control design: ruling out spurious correlations

A first version of this probe read 85.8%; its own controls ruled that reading out, and in doing so determined the design we report (Fig. 4, left). It used two objects at two sites, so their positions were perfectly anti-correlated, one determines the other, and object site was decodable from the image alone at 91%, including 88.6% at *layer* 0, before any attention. The swapped-label control came back at 81.9% against 83.9% named: no referent specificity whatsoever. The probe never needed the prompt.

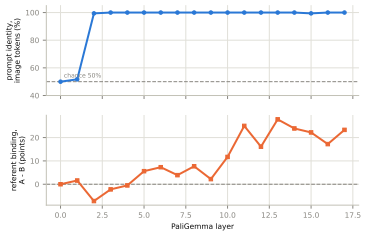


Figure 5: Global prompt identity (top) vs. referent binding (bottom) by layer. Prompt identity is at ceiling by layer 2; referent location separates only from layer 5, so a global prompt shift cannot explain the binding.

Each of those failures specifies a requirement the permutation design meets: more objects than can be pairwise anti-correlated (0.457 of 2.0 bits of residual coupling), plus a never-named distractor. A second confound is that LIBERO jitter is only 3–6 px, so in KITCHEN_SCENE2/3 each target occupies a near-constant image cell and prompt→location is deterministic; those scenes' expressions are positional, so randomising placement would destroy the labels. We therefore use a scene whose expressions name identity, and permute objects over that scene's own sites.

4.3.4 Probe signals at different layers and timescales

A residual objection is that the probe reads a global prompt-conditioned shift and not a referent binding. We decoded *prompt identity* directly (Fig. 5). From text tokens it is 100% at every layer including 0: the validated positive control, since tokens are embedded there. From image tokens it is at chance at layer 0, 99.4% by layer 2, and 100% thereafter. Referent *location* is equal at layer 0 and separates only from ≈layer 5. The confound is present, at ceiling, four layers before the effect exists; it cannot explain the effect.

4.4 Where the Model Attends: Prompt-Conditioned Attention Changes

Sec. 4.3 shows the referent is *present* in the key/value cache. Here we ask: does the action expert *attend* to it? We measure post-softmax attention to the referent's image patches, named versus not-named, paired within object.

4.4.1 RESULT

object	median D	vs. not-named	sign test
bowl	$+4.47 \times 10^{-4}$	$+102.5\%$	$89/90$ (z=9.28)
pudding	$+3.03 \times 10^{-4}$	$+87.0\%$	$80/90$ (z=7.38)
pooled	$+4.11 \times 10^{-4}$		$169/180$ (z=11.78)

Table 3: Attention mass per token on an object, named minus not-named, paired within object on 90 frames.

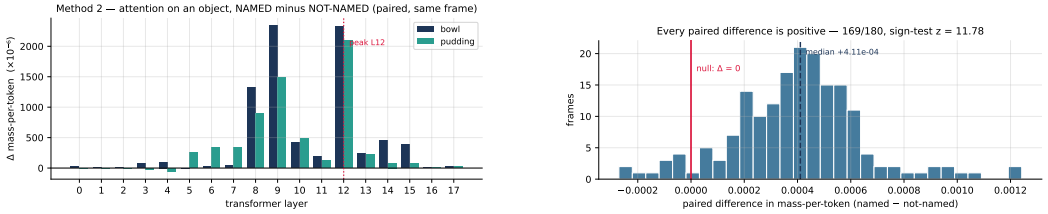


Figure 6: Left: per-layer paired effect (named vs. not). Right: distribution of paired differences; $169/180$ positive ($z = 11.78$). Naming an object reliably increases attention to it.

Naming an object increases attention to it: every paired difference is positive ($p_{10} = +1.02 \times 10^{-4}$, Fig. 6). The effect concentrates in mid-to-late layers (peaking at layer 12, adjacent to the probe's layer-11 peak) and is carried by four of eight heads. Language-driven attention modulation is robust and statistically significant.

4.4.2 CRITICAL CONTROL: IN-DISTRIBUTION ATTENTION IS UNCHANGED

We examine whether weak attention magnitude explains the behavioral failure. Attention allocation where behaviour is correct (in-distribution override, $19/20$) versus where it fails ($0/10$ OOD) is virtually identical: referent 8.05×10^{-4} per token in-distribution vs. 8.17×10^{-4} OOD; margin over background $\approx 7.4\%$ in both (Table 4). This suggests weak absolute attention magnitude is unlikely to be the sole explanation.

	in distribution ($19/20$ correct)	out of distribution ($0/10$)
referent, per token	8.05×10^{-4}	8.17×10^{-4}
background, per token	7.49×10^{-4}	7.60×10^{-4}
margin over background	$+7.4\%$	$+7.4\%$
referent share of mass	$\approx 1.6\%$	$\approx 1.7\%$
paired modulation (median)	$+7.58 \times 10^{-5}$	$+4.11 \times 10^{-4}$

Table 4: Attention allocation is the same where behaviour is correct and where it is not.

4.4.3 MAGNITUDE CAVEAT: ATTENTION IS WEAKER WHERE BEHAVIOR IS CORRECT

Per-token allocation is not the whole story. Against total attention (text is $\approx 40\%$), the referent is only $\approx 1.6\%$ in-distribution and $\approx 1.7\%$ OOD. Against visual attention alone it takes 4.33% of image mass on 3.83% of image tokens.

The paired modulation is $+7.58 \times 10^{-5}$ in-distribution ($73/100$ positive, $z = 4.60$) versus $+4.11 \times 10^{-4}$ OOD ($169/180$, $z = 11.78$): language-driven steering is roughly **five times stronger where behaviour fails**. This suggests attention upweighting alone cannot account for the behavioral difference (Fig. 7). The evidence points to a failure downstream, where the attended information is translated into action.

4.5 THE DISSOCIATION: ENCODING AND ATTENTION WORK, ACTION FAILS

Table 5 gathers the three measurements. Encoding, attention allocation, and language-driven modulation all survive OOD; only grasping fails. The results constrain the failure to a stage downstream of simple referent decodability and coarse attention allocation, consistent with a breakdown at the

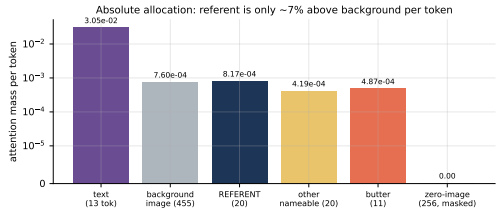


Figure 7: Absolute attention allocation (log scale). The named-object referent receives $\approx 7\%$ more per-token mass than generic background tokens. The masked zero-image block receives exactly zero, validating the token grouping.

vision-language-to-action interface where the model translates attended information into motor commands.

Stage	In-dist. (19/20)	OOD (0/10)	Verdict
Language encoding (probe)		82.8%	Recoverable OOD
Attention to referent	8.05×10^{-4}	8.17×10^{-4}	Preserved
Paired modulation	$+7.6 \times 10^{-5}$	$+4.1 \times 10^{-4}$	Works (stronger OOD)
Object grasped	$19/20$ correct	$0/10$ correct	Fails

Table 5: Three-stage pipeline. Encoding and attention allocation remain intact under distribution shift; only action selection fails. Probe measurements are from OOD scenes; ID encoding is not separately evaluated.

4.6 Why the Benchmark Hides This: Structural Analysis

The natural hypothesis is that LIBERO lets a policy ignore language because the scene determines the task. Read directly from the BDDL definitions (Table 6), three of the four fine-tuning suites are solvable without the instruction: `libero_spatial`'s ten tasks share one scene *and one goal object*, so a layout recogniser scores $\approx 100\%$; `libero_object` gives each scene its own unique target. Only `libero_goal`, whose ten tasks share one object set *and* one identical initial layout (frames are pixel-identical at $t = 0$) makes the instruction strictly necessary. Roughly three quarters of the task distribution does not require reading the instruction to identify the goal, which may explain why the network deprioritises language-conditioned action selection.

Suite	Requires language	Structure
libero_spatial	no	10 tasks, 1 scene, 1 object
libero_object	no	10 distinct object sets
libero_goal	**yes**	1 object set, pixel-identical layout
libero_10	partly	10 tasks over 6 scenes

Table 6: LIBERO suite structures (from BDDL definitions).

Two corrections follow, because they change how the result should be read.

`libero_spatial`'s 97% does not certify instruction following. The suite is solvable while ignoring language. The in-distribution evidence is the override test of Sec. 4.1 and `libero_goal`, where identical pixels make language mandatory and the model still scores $\approx 96\%$.

Failure is driven by distribution shift, not vocabulary. The trained noun "moka pot" also fails on task 18, and the KITCHEN_SCENE2 collapse uses black bowls that appear throughout training. The variable is *distribution*, not vocabulary.

4.7 Negative Result: Dynamics Randomisation Does Not Help

Randomising simulated dynamics at evaluation time does not create measurable behavioural change: operational-space control reads the same plant model it perturbs and cancels the intervention (Fig. 8; App. C).

4.8 Implications for Repair

Our diagnosis constrains plausible repair strategies and identifies promising directions for future work.

Retraining on instruction-necessary data. The benchmark analysis (Sec. 4.6) shows that $\approx 75\%$ of fine-tuning tasks are solvable without reading the instruction. This may explain why the readout expert is undertrained. Retraining on data where instructions are strictly necessary such as `libero_goal` and `libero_90` offers a testable hypothesis that would directly validate our diagnosis.

Mechanistic intervention. Since the referent is encoded and attended but not acted upon, the failure likely occurs in how the action expert processes attended information. Activation patching Meng et al. (2022) and causal mediation Vig et al. (2020) can precisely locate and isolate this component. Such approaches are diagnostic first; they become corrective only if the identified locus is editable.

Improving evaluation practice. Beyond any specific repair, our measurements reveal a broader issue: one fixed behavior averaged over three instructions scores 33% on standard metrics. Future benchmarks for instruction-conditioned manipulation should (1) explicitly report referent selection accuracy alongside task success, and (2) include same-scene pairs where different instructions require different objects. Currently, three of LIBERO's four fine-tuning suites cannot distinguish a model that ignores instructions from one that reads them.

4.9 Limitations

We evaluate on a limited number of samples ($n = 6$–10 per condition) in a single out-of-distribution test environment. However, the observed behavioral dissociations are large-effect ($0/10$ vs $19/20$ in-distribution; 25% chance vs 82.8% probe recovery), and the within-subject paired attention analysis achieves statistical significance ($z = 4.60$). We do not claim that $\pi_{0.5}$ universally ignores language: the model correctly follows overridden instructions in-distribution ($19/20$ accuracy). The reported failures are specific to the out-of-distribution condition tested here and may not generalize to other scene distributions or model architectures.

5 Conclusion

We demonstrate that a state-of-the-art vision-language-action model can encode language instructions and route visual attention appropriately, yet still select the wrong action when faced with an out-of-distribution scene. This dissociation reveals a failure mode that standard success metrics cannot detect: one fixed strategy, when averaged over three different instructions, scores 33% on success rate, masking complete instruction-following failure. Control experiments exclude weak visual grounding as the culprit language-driven attention modulation is actually five times stronger in the OOD condition where behavior fails. The evidence points to a failure at the interface between the vision-language encoder and the action selection component.

These findings have broader implications for robot learning benchmarks. When most training tasks can be solved without reading instructions (as with $\approx 75\%$ of LIBERO), a benchmark cannot distinguish models that follow language from those that ignore it. Standard success metrics reward one consistent strategy that happens to work on some tasks, even when instructions are completely disregarded. Future evaluation should explicitly report referent selection accuracy and include same-scene task pairs where different instructions demand different actions. Such measures are essential for certifying that trained policies genuinely follow language instructions.

Mechanistic localization via exclusion provides the foundation for targeted repair: retraining on language-necessary data, intervention via activation patching, or both. Diagnosing failures at this level enables fixing specific components rather than wholesale retraining.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 3889–3900, 2022.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

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

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *International Conference on Learning Representations*, volume 2024, pages 36391–36413, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pages 2733–2743, 2019.

Yuchen Hou and Lin Zhao. Langgap: Diagnosing and closing the language gap in vision-language-action models. *arXiv preprint arXiv:2603.00592*, 2026.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

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. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002/.

Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. *arXiv e-prints*, pages arXiv–2505, 2025.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX

Instrumentation (grasp ground truth, prompt override, perturbation harness with read-back probe, residual logging, monitoring UI), the probe scripts and frame builders, and the anchored patch generators are available with the per-episode logs; every number and figure in this paper regenerates from those logs by a single analysis notebook.

AI USE STATEMENT

We used large language models for three specific tasks in this work: (1) information retrieval to identify relevant papers and citations for contextual background, (2) text refinement assistance to improve clarity and academic phrasing in manuscript revision, and (3) assistance with experiment execution and computational analysis. All text was written by the authors, and any AI-assisted refinements were manually reviewed and edited to ensure accuracy and alignment with the authors' intent.

AI tools were not used in formulating the research question, designing the core dissociation experiment, or developing the linear probe and attention measurement methodologies. While AI assisted with computational and analytical tasks, the mechanistic interpretability reasoning, result interpretation, and derivation of scientific conclusions were conducted directly by the authors and did not rely on AI-generated insights.

We manually verified all AI-assisted content through (a) author review of all text suggestions for accuracy and clarity, and (b) validation of citations and references retrieved via AI against original sources.

A INSTRUMENTATION

All edits to the upstream evaluation script are applied by an idempotent, anchored patch generator (24 anchors) that restores the pristine file first and asserts each anchor occurs exactly once, so a silently partial patch cannot be mistaken for a null result.

Object-level grasp ground truth. Per control step we log which task object is in a closed grasp, via robosuite's `_check_grasp` against each object's contact geoms, and record the first-grasped object, the dominant object by held-step count, and per-object step counts. In a scene with more than one candidate this is a direct read on grounding, independent of whether the episode succeeds.

Prompt override. The instruction transmitted to the policy is decoupled from the instruction that defines success: the client sends an arbitrary string while the BDDL predicate that scores the episode is untouched. An override naming a different object therefore registers as a failure *by construction*;

under an override the informative measure is the grasped object, not success. This is what makes a controlled ablation possible without editing task definitions, and it is why every override row below reports 0 success.

Probe capture. Prefix K/V are captured per layer and mean-pooled over image tokens 0–511 for the probe of Sec. 4.3.

Protocol hygiene. Operator actions that change *which* episodes run set a sticky DIRTY flag, recorded per episode, so an exploratory session cannot be mistaken for a clean reproduction. Pause is protocol-neutral and does not set it.

B METHODOLOGY

Two measurements are deployed on the frozen network, both forward-pass only. Method 1 asks what information is *present* in the key/value cache the action expert reads. Method 2 asks where the expert *looks*. Neither involves rollouts, training or gradient flow into the model.

B.1 METHOD 1: LINEAR PROBING OF THE PREFIX K/V

Linear probing follows Alain and Bengio Alain and Bengio (2016): freeze the network, take activations from an intermediate layer, fit a linear classifier to predict a property of the input, and read its accuracy as evidence about what that layer encodes. We take the value tensors at the 512 real image-token positions and mean-pool them,

$$x^{(L)}(f,p) \;=\; \frac{1}{512}\sum_{t=0}^{511} V_t^{(L)} \;\in\; \mathbb{R}^{256}, \tag{1}$$

where f indexes the frame and p the prompt. The label $y(f,p)$ is the site index, of four, of the object the prompt names. The classifier is multinomial logistic regression with L_2 regularisation; folds are held out *by frame*, never by sample.

Detecting binding through controlled design. Accuracy alone would not establish binding; the experimental design does. Our key insight is that if the representation were merely encoding object positions plus a global prompt marker (additively separable as $x(f,p) \approx g(f) + h(p)$), then a balanced factorial design would not generalize across novel object–location configurations. A linear classifier trained on one set of object–location pairings should fail to generalize when objects are permuted to new sites, unless the network has learned a prompt-dependent referent binding.

Each frame contributes two samples with byte-identical pixels and different labels. In the random-permutation design, the correct site indices satisfy $\mathrm{site}_A(f) \neq \mathrm{site}_B(f)$ across frames with no global permutation pattern. If the model distinguished sites only through image features, a probe would generalize equally well or better on permuted sites. Instead, we find binding emerges at mid-to-late layers, indicating a prompt-conditioned interaction built by the network.

We empirically validate additivity assumptions at layer 0, where representations are additive by construction (token embeddings before attention). We measure the gap at layer L to confirm binding is not present from the start. A nonlinear probe could construct spurious interactions; a linear probe with held-out frames and permuted sites constrains the evidence to genuine binding.

Validating additivity at the input layer. At layer 0 the K/V come from token embeddings before any attention, so the representation is additive by construction and the argument predicts the named and swapped labels are equally predictable. Measured: both 42.8% (mean-pool), both 38.9% (mean+max). The gap opens only once attention has mixed the streams, as shown in the main text (Fig. 4).

B.2 METHOD 2: ATTENTION MASS, PAIRED WITHIN OBJECT

For layer ℓ, head h, action-expert query q and prefix key j,

$$\alpha_{q,j}^{(\ell,h)} = \mathrm{softmax}_j\left(\frac{q^{(\ell,h)} \cdot k_j^{(\ell)}}{\sqrt{256}}\right), \qquad \textstyle\sum_j \alpha_{q,j} = 1. \tag{2}$$

Attention is multi-query ($\mathrm{num_kv_heads} = 1$), so the map from a weight to a prefix token is unambiguous. Averaging over the 10 suffix queries gives $\bar{\alpha}_j$; for a token group G we report total mass $M(G) = \sum_{j \in G} \bar{\alpha}_j$ and mass per token $m(G) = M(G)/|G|$. Only the latter is interpretable as "does the expert look here", as the referent occupies $\approx$20 patches against several hundred for background.

The statistic is, for object o in frame f,

$$\begin{aligned} D(f, o) \;=\; & m\big(P_o \mid \text{prompt names } o\big) \\ & - m\big(P_o \mid \text{prompt names the other}\big), \end{aligned} \tag{3}$$

with P_o the image tokens whose 14×14 patch overlaps o's segmentation mask (Fig. 1). Both terms share a frame, so pixels, P_o and $|P_o|$ are identical: size, contrast, position and patch count are differenced out *exactly*, not controlled for statistically. This matters: the bowl averages 25.7 patches against the pudding's 13.9.

P_o is external ground truth from the simulator's instance rendering, never observed by the model, and α depends on j only through k_j. For mass on precisely those tokens to rise when o is named, the model must itself have computed which tokens correspond to o.

B.3 Validation and sanity groups

Four checks validate the machinery, not the hypothesis. Mass on the 256-token masked zero-image block is *exactly* zero, confirming the token group indices. Text tokens decode prompt identity at 100% from layer 0, confirming the pipeline detects something known to be present. Shuffled labels sit at $\approx 26\%$ against 25% chance. And the layer-0 equality above confirms the additivity premise instead of assuming it.

B.4 What these methods cannot establish

Linear decodability of referents and increased attention mass are correlational evidence, not causal proof. A linear probe shows information is recoverable from the representation, not that the model uses it; attention weights indicate where the model looks, not whether that information causally influences the action. The key distinction, emphasized by Elazar *et al.* Elazar et al. (2021), is that a property can be highly decodable and causally irrelevant, which is why causal inference requires intervention.

Our approach employs exclusion rather than intervention: we verify that encoding and attention routing both continue to operate under distribution shift. Encoding is demonstrable (linear probe: 82.8% vs baseline), and attention routing shows strong language-conditioned modulation (changes by factor of 5 across OOD/ID conditions), yet action is wrong. The conjunction of these observations constrains the failure to a stage where language-guided representations and attention are not yet translated into action. The in-distribution control (Sec. 4.4) rules out magnitude insufficiency, supporting this conclusion. However, this does not fully exclude upstream problems: the representation could encode information in an inaccessible subspace, or attention could point at objects whose token values lack task-relevant information.

Activation patching via causal mediation analysis Meng et al. (2022); Vig et al. (2020) would provide direct intervention evidence and is a natural next step. But exclusionary reasoning testing and eliminating major alternative explanations remains a valid inference strategy in mechanistic interpretability, particularly when effect sizes are large and the excluded mechanisms are independently verifiable.

B.5 Absolute attention allocation

Per-token mass on the named object sits $\approx 7\%$ above generic background image tokens (Fig. 7). The zero-image block is an exact-zero sanity check on the grouping. Despite this selective allocation, the OOD policy still grasps the wrong object, which is why the main text locates the failure downstream of attention (Sec. 4.4).

C NEGATIVE RESULT: OSC CANCELS PLANT PERTURBATION

The following section provides detailed structural analysis of the negative result shown in Fig. 8.

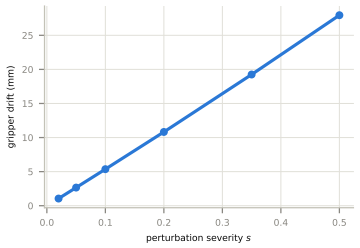


Figure 8: Gripper drift under domain randomisation (log scale). Tracking error stays near zero at every severity: OSC compensates for the same plant model it perturbs. Replays at fixed severity are bit-identical; $s = 0.10$ is ≈ 5 mm.

We first attempted to recover measurement headroom by randomising simulated dynamics at evaluation time, in the spirit of domain randomisation Tobin et al. (2017): object and link mass (with inertia scaled consistently), geom friction, joint damping, and the OSC gain k_p (with k_d rescaled to hold ζ fixed), log-uniformly per episode. The harness applies exactly (values read back from the live simulator divide to the baseline to four significant figures) and produces almost no behavioural change (Fig. 8).

The cause is structural. robosuite's operational-space controller Khatib (1987); Zhu et al. (2020) builds its mass matrix from the simulator's own model and takes gravity/Coriolis compensation from `qfrc_bias`, all read from the same model we perturb. Scaling link mass changes the plant *and* the controller's model of the plant in lockstep, and inverse dynamics cancels it. Ranked by what survives compensation: `mass_robot` is a dead axis; `mass_obj` is weak (a grasped object is an uncompensated contact load, but LIBERO objects are light); `damping` is real, entering `qfrc_passive` which `qfrc_bias` excludes; `friction` is real during contact; `kp` is real, being a genuine controller/plant mismatch. The axes worth perturbing are those the controller has no model of: actuator gain, joint dry friction (additive, since the stock value is zero), and action latency.

Two design points generalise. Perturbations must be applied *after* `reset()` and `set_init_state()`, because robosuite's hard reset rebuilds the model from XML and silently discards earlier mutations; applying them at construction yields an unperturbed run indistinguishable from a genuine null. And the uniform deviate is drawn for every axis even at severity zero, with the RNG keyed on (seed, task, episode), so severity scales the magnitude of a *fixed* direction: a sweep traces a dose–response along one direction instead of resampling an unrelated world at each level.

Implications for the main finding. A perturbation the controller compensates produces a near-zero tracking residual by construction, so a method whose signal is dynamics-consistency is blind to the axes standard randomisation prioritises. The failures we do find are worse for such a method, not better: on task 18, $\approx 83\%$ of failures are wrong-object grasps, and tracking is excellent throughout: the arm smoothly and confidently manipulates the wrong object, so the residual is near zero exactly when the episode is already lost. Semantic failures are structurally invisible to that signal. In a separate diagnostic over successful episodes we also found that $\approx 75\%$ of the raw tracking residual is structural lag, and that an apparent contact-phase residual increase (24 vs. 18 mm) vanishes once normalised by command magnitude: the realised fraction is statistically indistinguishable (0.259 vs. 0.244). Subtracting a constant free-space baseline, the obvious normalisation, would flag every contact phase in every successful episode as anomalous.