LOOK: LATENT KNOWLEDGE IS OVERLOOKED, NOT OVERWRITTEN, IN VISION-LANGUAGE-ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Failures of vision-language-action (VLA) models on concepts unseen in robot demonstrations are often attributed to semantic damage during action training. We find that target knowledge can be weakened yet remain useful for control: the surviving signal can be overlooked by the action pathway. We study this possibility with CrossBind, which measures target identification and closed-loop behavior separately. Across three action-pretrained backbones, an unfitted likelihood readout identifies placement targets on 0.56 to 0.69 of episodes against a $1/3$ chance level. Yet same-checkpoint pairing in two task-adapted policies reveals only weak associations between correct identification and correct placement. This discrepancy motivates a semantics-to-action access bottleneck. RECALL tests this account by giving the retained signal an explicit path to action. With the backbone frozen, demonstration target indices and the original action loss train the action expert and interface; the inherited readout selects targets at inference. At fixed policy weights, replacing a location-only condition with readout-selected targets raises success from 0.215 to 0.438 on $\pi_{0.5}$, from 0.056 to 0.722 on InstructVLA, and from 0.049 to 0.514 on X-VLA. These results show why preserving semantics must be accompanied by attention to its use: retained target knowledge can support behavior when given an effective route to action.

1 INTRODUCTION

Vision-language-action (VLA) models are typically initialized from pretrained vision-language models (VLMs) (Brohan et al., 2023; Kim et al., 2024; Black et al., 2025) to carry open-world understanding into robot control. Yet action-trained VLAs frequently ignore instructions or fail on concepts unseen in robot demonstrations, motivating explanations based on damage to pretrained semantics during action adaptation (Dey et al., 2025; Fang et al., 2025; Driess et al., 2025). This account leaves open a second source of failure: useful semantics may survive adaptation yet remain poorly accessed by action generation. Protecting what a model knows does not by itself ensure that its policy will use it.

Closed-loop success cannot distinguish these possibilities. A language-conditioned policy must identify the instruction's referents and execute the motion, and failure at either stage looks the same. Conversely, success need not come from understanding, because fixed layouts, instruction templates and initial states can let a policy bypass vision-language binding (de Haan et al., 2019; Xing et al., 2025). Closed-loop benchmarks (Mees et al., 2022; Liu et al., 2023) therefore need complementary measurements to determine whether task-relevant knowledge remains recoverable and how that knowledge relates to behavior.

We study this question with CrossBind, where the robot places an instructed block onto an instructed target image. Training and test use disjoint colors and target images. Separate readout and behavior measurements, crossed destination arrangements and shortcut audits support a controlled study of semantic binding. Three findings progressively develop and test the access-bottleneck account.

The semantics survives action pretraining. We score candidate regions with the model's own generative likelihood, without fitting a labeled probe. Read this way, $\pi_{0.5}$ (Physical Intelligence et al., 2025), InstructVLA (Yang et al., 2026) and X-VLA (Zheng et al., 2026) identify the target on

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.

0.56 to 0.69 of episodes against a $1/3$ chance level (Figure 1). These lineages inherit PaliGemma (Beyer et al., 2024), Eagle2 (Li et al., 2025b) and Florence-2 (Xiao et al., 2024), respectively, and span distinct ways of coupling language and action. Even X-VLA, built without a language decoder, yields a readout of 0.56 through an unfitted ancestor-decoder graft. Thus, action pretraining weakens the signal but leaves useful target knowledge recoverable.

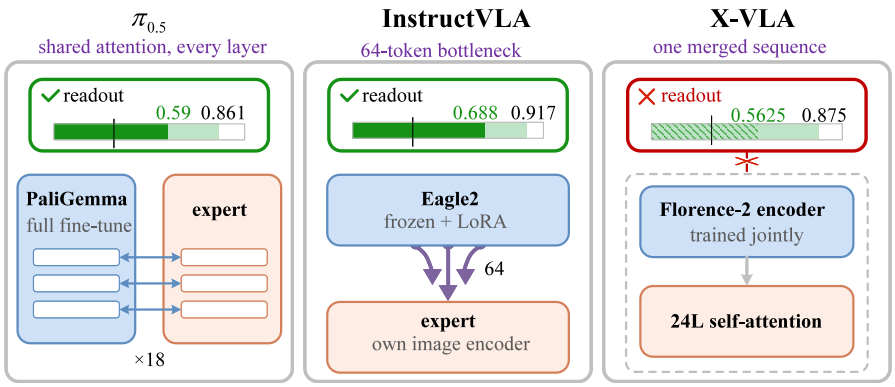


Figure 1: Three VLA lineages retain above-chance placement readouts after action pretraining. Pale bars: ancestor VLM; solid bars: action-pretrained backbone; black ticks: $1/3$ chance. The diagrams show the action interfaces. X-VLA is read through an unfitted ancestor-decoder graft (cross and hatching).

Recoverable knowledge is weakly reflected in behavior. Pick accuracy on unseen colors largely survives, while placement remains more difficult. On the same task-adapted checkpoints and episodes, we pair readout correctness with placement outcomes for $\pi_{0.5}$ and InstructVLA. Correctly reading the target has little predictive power for correct placement. This discrepancy motivates a semantics-to-action access bottleneck: recoverable target knowledge need not effectively guide the action pathway.

An explicit access path improves behavior. To test whether the retained signal can support control, RECALL routes the visual evidence selected by the readout to the action expert. The backbone stays frozen, while demonstration target indices train the expert and interface under the original action loss. At inference, the inherited readout selects the targets. Within each trained policy, this condition outperforms one that supplies candidate locations without identifying the requested targets. The comparison keeps policy weights fixed while changing the target information available to action generation. Thus, the retained signal has control value when the policy learns to access it.

Together these findings show how pretrained semantics can be both weakened and overlooked. Preserving that knowledge addresses semantic retention; the access-bottleneck account directs attention to the path through which the surviving knowledge can guide action.

- **Measurement.** CrossBind combines semantic readout and closed-loop evaluation with destination crossing and shortcut audits to examine competing explanations for failure.
- **Diagnosis.** An unfitted readout recovers target semantics across three lineages, while same-checkpoint pairing reveals a weak link to the original policies' behavior.
- **Intervention.** RECALL gives the retained signal a learned route to action; same-policy condition replacements test the control value of the selected targets.

2 Related Work

Semantic inheritance in VLAs. Building robot policies on pretrained VLMs is now the dominant paradigm (Brohan et al., 2023; Kim et al., 2024; Black et al., 2025; Physical Intelligence et al., 2025), and existing studies report that action-trained VLAs may ignore the language condition under distribution shift (Fang et al., 2025) and drift away from the original VLM's visual representations (Dey et al., 2025). The remedies insulate the VLM weights during action training (Driess et al.,

2025), pull representations back toward the pretrained encoder (Dey et al., 2025), or co-train with image-text data (Brohan et al., 2023; Physical Intelligence et al., 2025; Yang et al., 2026). We complement these approaches by connecting a recoverable target judgment to a same-policy intervention on the information supplied to the action expert.

Dataset shortcuts in robot learning. Imitation policies exploit regularities that correlate with the task without causing it (Geirhos et al., 2020): confounded factors in the observation (de Haan et al., 2019), copies of past actions (Wen et al., 2020), and, at robot scale, dataset organization that confounds task identity with scene, viewpoint and object placement (Xing et al., 2025; Saxena et al., 2025), so that generalization depends more on how environmental factors combine than on the number of demonstrations (Gao et al., 2024; Lin et al., 2025). CrossBind uses an enumerable environment to distinguish the shortcut structure of the evaluated training recipe from the guarantees of a full-factorial design.

Grounding conditions and action generation. Supplying an explicit intermediate grounding improves control, whether as verbalized locations and subtasks (Zawalski et al., 2024; Huang et al., 2026), overlaid end-effector traces (Zheng et al., 2025; Li et al., 2025a), or predicted points and affordances (Yuan et al., 2024). In each case the grounding may come from an external model, from extra supervision (Song et al., 2026), or from a newly trained module, so a gain cannot establish that the VLA backbone retained the semantics. RECALL reuses the action-pretrained backbone's likelihood readout, with the inherited decoder restored for X-VLA, and separates selected target identity from candidate location with control arms.

Reading latent knowledge out of foundation models. Reading knowledge from a model's distribution or states is a standard analysis tool (Alain & Bengio, 2016; Belinkov, 2022; Kadavath et al., 2022; Belrose et al., 2023). Probe capacity and the distinction between decodability and behavioral relevance require explicit controls (Hewitt & Liang, 2019; Elazar et al., 2021). We use the model's generative likelihood without fitting a labeled probe, and then test the readout's control value on primitives excluded from the manipulation targets.

3 CrossBind: Controlled Evaluation of Semantic Binding

3.1 Task and Measurement Objective

CrossBind instantiates the measurement requirements of Section 1 as a minimal tabletop benchmark (Figure 2), with task setting and scene assets adapted from the WISER semantic pick-and-place benchmark (Li et al., 2026) and data generation rebuilt around the controls below. Each episode contains four colored blocks and three target images, and the instruction asks the robot to place a specified block onto a specified image. Binding is discrete and two-level: *pick* maps a color description to one of four blocks (chance $1/4$), *place* maps a target description to one of three images (chance $1/3$). Target concepts span concrete objects, abstract symbols and instructions requiring simple reasoning about image content, covering 24 categories. A separately trained position-conditioned policy reaches 99.65% on training scenes (Appendix F); this provides a task-feasibility reference.

Training and test are disjoint at the level of semantic primitives: the splits use different block colors, and target-image identities and their referring expressions do not overlap at all. Test colors and images therefore never appear as pick or place targets in any demonstration, so the evaluation tests semantic transfer beyond the demonstrated manipulation targets.

3.2 Data Construction and Shortcut Audits

If the initial state, the visual layout or the language template predicts the task target on its own, closed-loop success cannot be attributed to semantic binding. On an initial version of the data generated with WISER's original pipeline, all three channels turn out to be exploitable without any joint instruction-image understanding (Figure 3a).

CrossBind addresses these channels through a full-factorial training set. Every episode starts from the same initial pose, and scene and language factors are crossed: 24 target-image triples $\times$ all $4!$ block arrangements $\times$ all $3!$ image arrangements give 3,456 configurations, each containing all 12

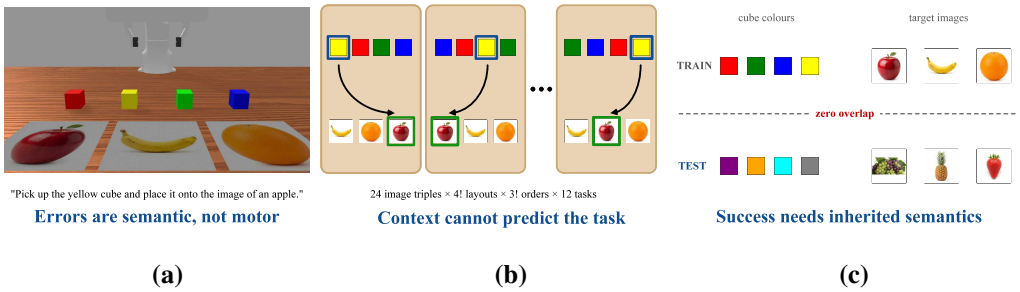


Figure 2: CrossBind design aims: (a) reduce motor demands, (b) remove contextual task cues through full-factorial crossing, and (c) test transfer to unseen primitives. Panel headings express design intentions; execution feasibility is assessed separately on training scenes.

pick–place tasks, for 41,472 training demonstrations. Phrasing variants are assigned stratified by image identity and place slot, independently of block arrangement. Thus each complete instruction occurs equally often with its referenced block in each of the four slots and its referenced image in each of the three, putting instruction lookup at chance on both levels ($1/4$ and $1/3$). These controls act on complementary channels. Fixing the reset state removes the initial-pose cue; enumerating all tasks within each scene prevents the first frame from identifying which task is requested. Crossing each complete instruction with block and image positions then removes the phrasing-to-slot association. The construction therefore makes the joint instruction–image relation the informative signal for selecting both referents, while preserving the same pick-and-place action space. Figure 3a audits this construction; Appendix F gives the channel protocols.

The policy comparison uses the earlier placement-crossed recipe, which balances destination positions (Appendix A). Figure 3b shows a $\pi_{0.5}$ train–test gap of 0.65 on the initial data and 0.04 on the placement-crossed recipe. Placement also stalls on the training split, at a conditional accuracy of 0.333 at the selected baseline step. Thus the difficulty that Section 4 studies is not restricted to unseen primitives.

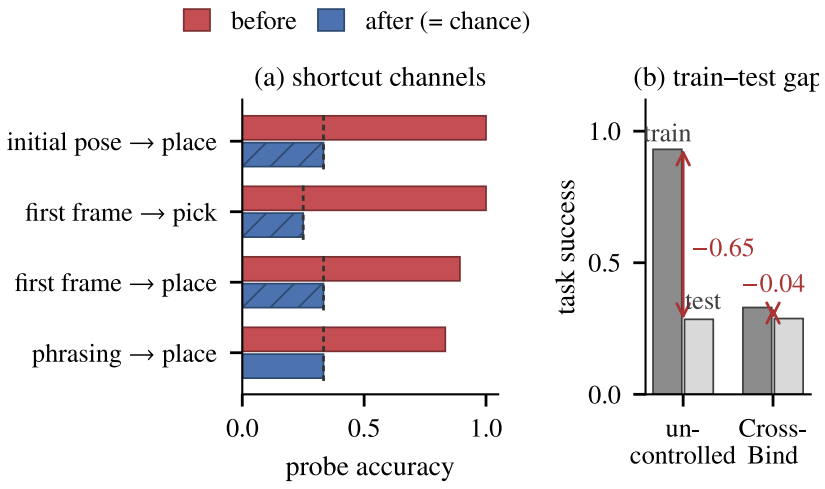


Figure 3: Shortcut audits and policy behavior. (a) Probe accuracy before construction and chance rates after full-factorial crossing; hatched bars denote construction-level guarantees. (b) $\pi_{0.5}$ train–test performance on the initial and placement-crossed recipes.

CrossBind supplies a setting in which target readout and action outcomes can be measured separately. Section 4 uses that setting to examine what the backbone retains and how that measurement relates to behavior.

4 DIAGNOSIS: A SEMANTICS-TO-ACTION ACCESS BOTTLENECK

4.1 CLOSED-LOOP BINDING ON UNSEEN PRIMITIVES

We evaluate $\pi_{0.5}$, InstructVLA and X-VLA on CrossBind using the training variants detailed in Appendix A. The 288 held-out episodes are 24 configurations $\times$ 12 environments. For baseline and condition-replacement summaries, we select checkpoints on configurations 0–11 and report configurations 12–23, with 144 episodes per half. Both halves use the same unseen colors and target images. The two-level metrics locate the difficulty after object selection (Table 1). Pick accuracy on unseen colors is 0.993, 0.833 and 0.639, all above the $1/4$ chance level, whereas place accuracy conditioned on a correct pick is 0.280, 0.550 and 0.337. Every model places well below its own pick level, with $\pi_{0.5}$ and X-VLA near the $1/3$ destination-choice reference. Conditional placement uses the successful grasps as its denominator: 143, 120 and 92, respectively.

Binding color words to blocks remains possible even when those colors never appear as training pick targets. What stalls is the place level, and two accounts remain open. The relevant information may have been lost, or it may survive while action generation does not use it. Closed-loop success cannot separate them, so Section 4.2 measures what the backbone retains and Section 4.3 tests its episode-level association with the placement outcome.

Table 1: Object selection and conditional placement on unseen primitives. Reported half of the held-out episodes, $n = 144$ per row. *Place* is conditioned on a correct pick, so Pick $\times$ Place = Task.

	Pick	Place	Task
Chance	0.250	0.333	0.083
$\pi_{0.5}$	0.993	0.280	0.278
InstructVLA	0.833	0.550	0.458
X-VLA	0.639	0.337	0.215

4.2 INHERITED SEMANTICS REMAINS RECOVERABLE

To measure the task semantics a backbone retains we use a candidate-region likelihood readout. Given candidate regions $r_1, \ldots, r_N$ on the episode's initial frame and a phrase $q = (q_1, \ldots, q_{|q|})$, we crop r_i out as the sole visual condition and compute the length-normalized log-likelihood under a fixed captioning prompt c,

$$s(q, r_i) = \frac{1}{|q|} \sum_{j=1}^{|q|} \log p_\theta(q_j \mid q_{<j},\, r_i,\, c)\,, \tag{1}$$

taking $\hat{\imath} = \arg\max_i s(q, r_i)$ as the backbone's identification of the referent. Readout accuracy is the fraction of episodes in which $\hat{\imath}$ hits the ground-truth region, with $N = 3$ at the place level and $N = 4$ at the pick level. For a fixed phrase, length normalization leaves the ranking of its candidate crops unchanged.

The measurement uses the inherited generative interface, without parameter updates or a labeled probe. X-VLA uses the unchanged ancestor decoder graft. Scoring is restricted to candidate crops on the $t = 0$ reset frame; readouts cover the held-out episodes and measure target recoverability under this prompting context. Appendix B gives the templates and repeated-item structure.

After large-scale action pretraining the backbones still read out the instruction's target well above chance (Table 2): three backbone families and three pretraining recipes give similar magnitudes of loss and endpoints between 0.563 and 0.688 against a $1/3$ chance level. Whether this retained knowledge guides an adapted policy is a separate question. X-VLA, whose language interface is deleted at construction time, is read through a zero-fit graft of the ancestor decoder and still reaches 0.563, showing that action-pretrained representations remain compatible with an informative inherited readout. Action pretraining weakens the signal while leaving useful target information recoverable.

Table 2: Placement-target readout through three training stages. Accuracy remains above the $1/3$ chance level after action pretraining; task adaptation affects the three lineages differently. X-VLA uses the ancestor-decoder graft at every stage. The complete pick/place table and scoring protocol are in Appendix B.

Lineage	Ancestor VLM	Action-pretrained	Task-adapted
$\pi_{0.5}$	0.861	0.590	0.597
InstructVLA	0.917	0.688	0.635
X-VLA	0.875	0.563	0.375

Task adaptation affects the readout differently across lineages. For $\pi_{0.5}$ it changes little, from 0.590 to 0.597, while InstructVLA reaches 0.635 and X-VLA falls to 0.375 (Appendix B). Specific final-layer $\pi_{0.5}$ tensors have no action-loss gradient path, although upstream representations change (Appendix F.3). The pairing below therefore uses task-adapted readouts from the same checkpoints as the behavior; RECALL instead freezes the action-pretrained stage.

4.3 Recoverable Semantics and Behavioral Association

Does that readable semantics participate in action generation? Aggregate comparison leaves one account open: the policy might place correctly on exactly the episodes the backbone reads correctly, so that the gap merely reflects an imperfect readout. We therefore pair readout and closed-loop outcome episode by episode, on the same checkpoint and the same episodes (Table 3). The observed placement-probability differences are $+0.026$ for $\pi_{0.5}$ and $+0.066$ for InstructVLA, with binary association coefficients of 0.03 and 0.06. Under this readout, correct identification has little predictive power for the policy's placement outcome. The InstructVLA records use its mixed-object-clause protocol; X-VLA is excluded from this pairing because its task-adapted graft readout approaches chance. Decodability and behavioral relevance are distinct in probing analyses (Elazar et al., 2021); we examine their relationship here for embodied control.

Table 3: Readout correctness weakly predicts placement at the same policy checkpoint. Each block pairs 288 held-out episodes: $\pi_{0.5}$ at step 5670 and InstructVLA at step 4860. Fisher tests describe episode-level associations within each block; Appendix D.2 details the pairing protocol.

	Placed	Not placed	P(placed)
$\pi_{0.5}$: readout 0.597, $\phi = 0.03$, $p = 0.69$			
Readout correct	52	120	0.302
Readout incorrect	32	84	0.276
InstructVLA: readout 0.635, $\phi = 0.06$, $p = 0.33$			
Readout correct	87	96	0.475
Readout incorrect	43	62	0.410

Attention allocation on $\pi_{0.5}$ provides a complementary observation: the main-camera patches receive 0.8% of action-query attention against a uniform share of 36% (Appendix F.3). Related work also reports dispersed target attention (Song et al., 2026). Sections 5 and 6 move from these observations to intervention on the target information reaching action generation.

5 RECALL: Routing the Backbone's Readout to Action

RECALL tests whether recoverable semantics can support action generation by holding the backbone fixed and intervening on the target information reaching the action expert. It reuses the inherited readout of Section 4.2 to retrieve visual evidence consistent with the instruction, then turns that evidence into an action condition (Figure 4). This connects diagnosis to control without an independently trained semantic teacher or auxiliary semantic loss.

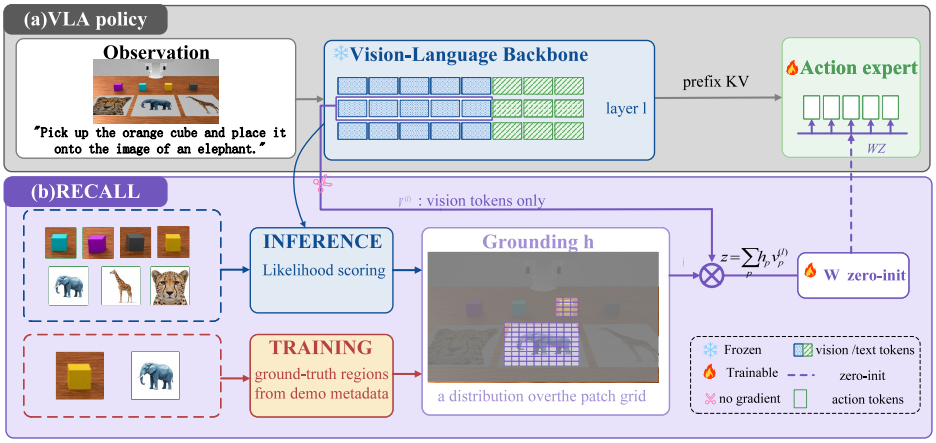


Figure 4: RECALL routes the frozen backbone's readout to the action tokens through a learned conditioning branch. (a) A standard VLA policy. (b) The inherited likelihood selects candidate regions at inference; demonstration target indices select them during training. Either source gives a grounding distribution h. A zero-initialized projection adds its pooled feature $z = \sum_p h_p V_p^{(l)}$ to the action tokens while the backbone remains frozen.

Inherited semantic retrieval. RECALL first needs a candidate set $\mathcal{R} = \{r_i\}_{i=1}^{N}$ covering the regions that may be grasped or serve as placement targets. We mine one fixed candidate set from cross-episode pixel-change statistics and reuse it in every arm. The boxes bound the search space; the readout selects the referents. A pick-and-place instruction has two referents, an object phrase and a placement-target phrase. For a phrase q and candidate r_i the scoring function is exactly the diagnostic readout of Section 4.2, $s_i = s(q, r_i)$, with the same frozen backbone, captioning prompt and length normalization. Reusing one function keeps a single operational definition of "semantics" across the diagnosis and the intervention. The object phrase is scored over the block candidates and the target phrase over the placement candidates (four and three in CrossBind), each resolved by $i^\star = \arg\max_i s_i$. Let Δ^P denote the probability simplex over the P patches of the image-token grid, and let $m^{\text{obj}}, m^{\text{tgt}} \in \Delta^P$ be uniform over the patches the two winning regions cover. The grounding distribution is

$$h = \lambda\, m^{\text{obj}} + (1-\lambda)\, m^{\text{tgt}} \in \Delta^P, \tag{2}$$

with $\lambda = \frac{1}{2}$ for the main comparisons. Hard selection and uniform within-region masks give the same condition representation for self and gt in Section 6. Self predictions are computed on the initial observation and cached for the episode.

Target-conditioned action interface. Write $V^{(l)} = \{V_p^{(l)}\}_{p=1}^{P}$ for the image-token representations at layer l of the frozen backbone. RECALL pools them with the grounding distribution into $z = \sum_p h_p V_p^{(l)}$. The pooled feature carries the selected regions' appearance and encoded position into the action tokens,

$$\widetilde{a}_k = a_k^{(0)} + (Wz + b), \qquad k = 1, \ldots, K, \tag{3}$$

where $a_k^{(0)}$ is the expert's original embedding of the k-th action token, K the chunk length, and (W, b) a learnable affine projection with zero-initialized weight and bias (Zhang et al., 2023), so the added condition does not change the policy output at the start of training. The same $Wz + b$ reaches all K tokens of a forward pass and is recomputed from each observation. All three architectures use this pooling-and-projection interface; Appendix C gives their feature sources, dimensions and chunk lengths.

Training and inference. RECALL freezes the backbone and optimizes the action expert and the projection, excluding backbone parameters from the optimizer (Appendix E.3). No semantic loss, grounding loss or auxiliary task is added. Each VLA keeps its original action objective, $\pi_{0.5}$ its native flow matching (Lipman et al., 2023; Black et al., 2025). BC and RECALL use different

training budgets, reported in Appendix C.2; the controlled intervention compares conditions within one RECALL policy.

Training and inference use the same candidate regions and differ in how the targets are selected. During training, demonstration target indices specify h_{GT}, and the action loss teaches the expert to use that condition. The readout is neither run nor updated. At inference, the inherited readout selects the regions that define $\hat{h}$. Swapping these conditions at fixed policy weights, instructions and initial evaluation states therefore tests which target information supports closed-loop control.

6 EXPERIMENTS

6.1 SETUP

For $\pi_{0.5}$, InstructVLA and X-VLA, we co-train the target-condition pathway on the same CrossBind training data as their Section 4.1 baselines and evaluate each policy under five grounding sources. *Off* removes the grounding map, *uniform* spreads mass over the whole image, *loc-only* marks all candidates equally to give location without identity, *self* uses the backbone's own readout, and *gt* provides the ground-truth target regions. These are evaluation conditions of one policy, so we fix one checkpoint per architecture and report every arm at it, keeping policy weights, instructions and initial evaluation states identical. Checkpoints are selected by gt success on the validation half to measure how well the policy uses a correctly specified condition. The baseline is a separately trained BC reference. Each training configuration has one run; Appendix D specifies checkpoint selection.

Table 4: Task success under five grounding sources. Reported half, $n = 144$ per cell. The five grounding rows use the same policy weights within each column; BC is a separately trained reference. Checkpoint selection is specified in Appendix D.

Grounding source	$\pi_{0.5}$	InstructVLA	X-VLA
BC baseline	0.278	0.458	0.215
off	0.000	0.000	0.000
uniform	0.167	0.035	0.076
loc-only	0.215	0.056	0.049
self (RECALL)	**0.438**	**0.722**	**0.514**
gt (oracle)	0.854	0.757	0.868

6.2 RECOVERABLE SEMANTICS SUPPORTS CLOSED-LOOP ACTION

The inherited readout improves closed-loop success when supplied as an action condition. Within each column of Table 4, moving from loc-only to self raises success from 0.215 to 0.438 on $\pi_{0.5}$, from 0.056 to 0.722 on InstructVLA, and from 0.049 to 0.514 on X-VLA. Because policy weights are fixed, these gains isolate the effect of replacing undifferentiated candidate locations with readout-selected targets. This within-policy improvement holds across all three architectures; self also exceeds the separately trained BC reference in all three. Together, the intervention results show that the semantics recovered in Section 4 can support closed-loop action.

InstructVLA's self success approaches gt, at 0.722 versus 0.757, while larger gaps remain for $\pi_{0.5}$ (0.438 versus 0.854) and X-VLA (0.514 versus 0.868). These gaps measure the effect of replacing correctly specified referents with the inherited readout at fixed policy checkpoints. The readout must resolve both the object and the destination, so errors at either level affect the self condition (Appendix B).

The remaining conditions clarify the role of the learned pathway. Off removes a training input and yields zero success, indicating that the policy depends on the condition. Uniform and loc-only retain the pathway but supply no selected target identity; their lower success shows why the content of the condition matters.

6.3 Identity Semantics vs. Location

The loc-only condition tests whether candidate locations account for the gain. Although it already marks every candidate region, it stays within 0.05 of uniform on all three architectures. Self additionally identifies *which* candidates the instruction refers to and outperforms both controls in each column. Target selection therefore contributes beyond the spatial information in the candidate set.

On $\pi_{0.5}$, episode-level pairing further links readout correctness to placement under self conditioning (Figure 5): the policy places correctly on 0.747 of episodes with a correct readout and 0.025 with an incorrect readout. Uniform and loc-only, whose conditions do not depend on the readout, have placement rates of 0.181 and 0.212 and associations of -0.05 and -0.02; gt shows -0.10 ($p = 0.11$). Thus the strong positive readout–behavior association appears specifically when the policy receives the readout-selected targets. A redirected-target control reinforces this sensitivity: with the policy fixed, changing both selected referents reduces success from 0.858 under gt to 0.010 in the matched harness (Appendix E).

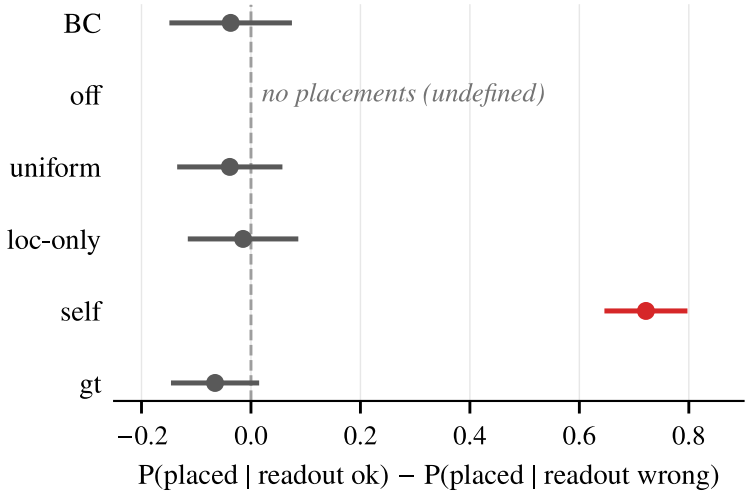


Figure 5: Readout correctness strongly predicts placement under self conditioning. Placement-probability differences and 95% episode-level intervals for $\pi_{0.5}$ BC and five conditions use one readout over 288 episodes. Identity-free conditions do not show the same positive association. Counts and interval construction are given in Appendix D.2.

The two referents also make distinct contributions under the learned $\pi_{0.5}$ interface. In the same 288-episode control harness, retaining only the object component succeeds on 83/288 episodes and retaining only the destination component on 110/288, compared with 247/288 for the equal mixture with correct target indices. Thus neither component alone reproduces the performance of the joint condition. Together with the redirected-target control, this shows that the policy responds to the selected object–destination pair, linking the two levels of semantic binding to the action input (Appendix E).

7 Conclusion

Action training weakens pretrained semantics, but useful target knowledge can remain overlooked. Across three backbones, target readouts remain above chance; in two task-adapted policies, their correctness weakly predicts placement at the same checkpoints. This motivates a semantics-to-action access bottleneck. RECALL provides a learned access path while keeping the backbone frozen. Readout-selected targets improve binding over location-only conditions at fixed policy weights across the three architectures. These findings extend the focus from preserving pretrained knowledge to enabling action generation to use it.