IMPROVING THE ROBUSTNESS OF VLAS TO SPATIAL OBJECT PERTURBATION VIA ATTENTION-HEAD AMPLIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) policies struggle under small perturbations, which limits robotic deployment. We focus on *spatial object perturbation*. When the target object is displaced from its training-time position, the policy fails to manipulate it and often reaches toward the training-time position. **With activation patching we show that this failure reflects non-use of the object's position rather than its absence, and we propose a training-free method that mitigates it by amplifying the few attention heads that carry that position to the action tokens.** On OpenVLA, OpenVLA-OFT, π_0, $\pi_{0.5}$, and GR00T N1.7 on LIBERO-Object, the new object location is present in a few object patches of the input visual tokens and is consumed in a limited band of layers. We then test every attention head one at a time. We run the policy on the scene with the moved object, but feed that head the output it would produce if the object were still at its training-time position, and measure how much the planned action stops following the object. We found that in every policy a small number of heads carry the displacement. Amplifying the input-dependent deviation of these heads raises paired closed-loop success, at 10 cm from 12.3% to 17.9% in OpenVLA-OFT and from 29.1% to 41.8% in π_0, and at 15 cm from 42.1% to 50.5% in $\pi_{0.5}$. The gain reproduces on the test states and transfers to LIBERO-Pro.

1 INTRODUCTION

Vision-Language-Action (VLA) models couple a pretrained vision-language backbone with an action decoder and are trained by imitation on large robot datasets. Successive generations have changed how the action is produced, from discrete action tokens in OpenVLA (Kim et al., 2024) to continuous chunk regression in OpenVLA-OFT (Kim et al., 2025) and to flow matching in π_0 (Black et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and GR00T N1.7 (Bjorck et al., 2025), and each generation has raised success on the standard suites (Liu et al., 2023; Mees et al., 2022; Li et al., 2024; Tao et al., 2025). Current models exceed 97% on the four LIBERO suites (Kim et al., 2025) and complete 4.5 of 5 tasks on CALVIN ABC→D (Reuss et al., 2025).

Most of these results, however, measure in-distribution inference in environments that differ little from training. In perturbed environments performance drops substantially, and even a camera-angle change can cost up to 90.9 percentage points (Fei et al., 2025). In particular, VLAs tend to shortcut-learn the absolute positions of objects, so the success rate drops sharply when an object is located away from its training-time position (Fei et al., 2025; Zhou et al., 2025; Tang et al., 2026).

Existing remedies retrain or fine-tune the policy, with re-collected data (Tang et al., 2026; Wang et al., 2025), reinforcement learning (Liu et al., 2025; Huang et al., 2025), or added 3D input (Qu et al., 2025), and each requires new data or training. A remedy that needs no training must instead know where inside the policy the failure arises. Recent interpretability work on VLAs shows that object position can be decoded from hidden states (Buurmeijer et al., 2026; Zhang et al., 2026a), but a variable that can be read out is not necessarily the one the policy uses. Whether the new object location is used, and through which components it reaches the action, has not been tested causally.

We answer this with tools from language-model interpretability, activation patching (Vig et al., 2020; Meng et al., 2022) and head-level intervention (Li et al., 2023), adapted to a closed-loop policy in

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.

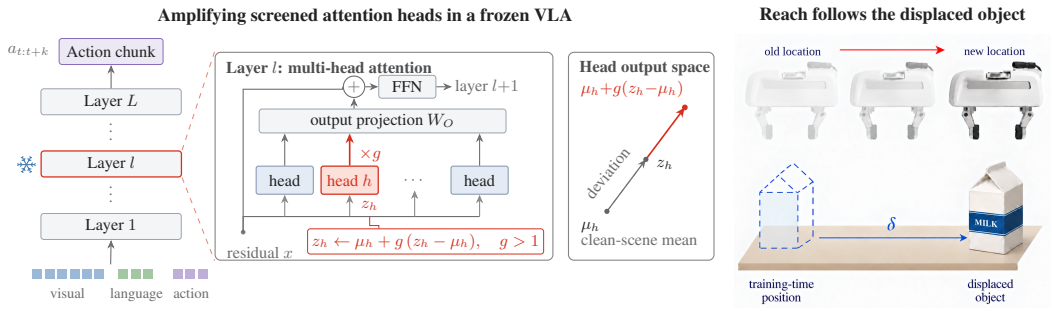

Figure 1: **Overview.** Left: a frozen VLA policy maps visual, language, and action tokens to an action chunk. Inside one layer, the output z_h of a screened attention head h is amplified before the output projection. The clean-scene mean μ_h is kept and only the input-dependent deviation $z_h - \mu_h$ is multiplied by $g > 1$ (inset), with no parameter changed. Right: when the object is displaced by δ from its training-time position, the base policy reaches the old location. With the amplified heads the reach shifts by δ and follows the object.

three ways. The contrast pair is physical, two scenes with the same initial state and the same replayed action prefix that differ only in the object location. The readout is the planned action, and each head is scored by the slope of its response to the displacement, which is invariant to the constant offsets that mislead zero ablation. The same head output that the screen measures is then amplified, by scaling only its input-dependent deviation, and judged in closed loop against random, deviation-matched, and energy-matched controls.

We call manipulating an object located away from its training-time position the spatial object perturbation task. Our answer is that the information is present but under-used. In all five policies the new object location enters through a few object patches of the visual tokens, is consumed in a limited band of layers, and is carried into the action tokens by a small number of heads. Amplifying those heads without training raises paired closed-loop success in OpenVLA-OFT, π_0, and $\pi_{0.5}$, harms no direction, reproduces on test states, and transfers to LIBERO-Pro. It fails in OpenVLA and GR00T for reasons the same measurements identify, a state-specific offset and one-sided tracking, so the lever is real but conditional. Figure 1 summarizes the method.

These results rest on the following contributions.

- We show that in four of the five policies the dominant failure under displacement is a reach to the training-time position, and that the accuracy of this reach does not depend on the displacement, so the policy replays a fixed target rather than misestimating the current one.
- We propose a head screen for closed-loop policies that replaces one head's output with its clean-scene value at a matched initial state and regresses the plan change on the displacement, and we show that it is invariant to constant offsets.
- We trace the causal chain with one primitive and show that the displacement is localized to a few visual patches and that the amplified heads account for nearly all of the layer-level consumption in OpenVLA-OFT, four fifths in $\pi_{0.5}$, and about half in the other three.
- We show that amplifying the screened heads without training raises success in three policies and that random, deviation-matched, and energy-matched heads give none.

2 Related Work

Mechanistic interpretability and causal intervention. Mechanistic interpretability studies how model components causally contribute to behavior. Work on language models has used causal mediation analysis, causal tracing, and inference-time intervention to localize and modify internal computations (Vig et al., 2020; Meng et al., 2022; Li et al., 2023). Activation patching and interchange intervention provide a general way to test whether replacing an internal activation changes the output while holding the rest of the computation fixed (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024; Geiger et al., 2022). Attribution patching approximates such a sweep with gradients so that

all components can be scored at once (Nanda, 2023; Syed et al., 2024; Kramár et al., 2024). We use these tools to ask where the displaced object position is present, where it is consumed, which heads route it to the action tokens, and whether strengthening those heads changes closed-loop behavior.

Representation observability and intervention in VLAs. Recent work studies internal representations in VLAs with probes, sparse features, activation steering, and activation injection. Häon et al. (2025) and Swann et al. (2026) identify interpretable or steerable VLA representations. Buurmeijer et al. (2026) show that low-level state and action variables can be decoded from hidden states. Miao et al. (2026) argue that VLA failures can arise when internally present information is not exploited. Grant et al. (2026) show that VLA features are not used uniformly. Zhang et al. (2026a) use decoded object positions to guide a training-free control filter. We ask the next question, whether a present variable is the one the policy reads, with activation patching rather than decoding.

VLA robustness under spatial perturbation. Robustness studies show that VLA policies are vulnerable to changes in object position and scene configuration (Fei et al., 2025; Zhou et al., 2025; Wang et al., 2026). Existing remedies improve spatial generalization by collecting new data (Tang et al., 2026; Wang et al., 2025), fine-tuning with reinforcement learning (Liu et al., 2025; Huang et al., 2025), or adding explicit spatial information (Qu et al., 2025). These methods change the training data, the objective, or the model input. We instead study a frozen policy and test whether a targeted inference-time intervention raises paired closed-loop success without retraining.

Attention routing and head-level control. Several studies connect VLA behavior to attention routing. Jeong et al. (2026) analyze attention heads as path-deviation detectors. Zhang et al. (2026b) recalibrate attention at inference time to restore linguistic grounding. Jia et al. (2026) train action-attention heads to specialize to task-relevant factors. We instead identify naturally occurring heads in a frozen policy by their causal response to object displacement. The screened heads are not assigned role labels by supervision and are not chosen from attention weights alone. They are chosen because replacing their output with the clean value changes the plan response to displacement.

3 METHOD

Setting. Each task asks the policy to pick up a named object and place it in a basket. In the training data the target object always sits at one fixed position for its task. At test time we move only the target object by a planar vector δ (Figure 2). We call the scene with the object at its training-time position the *clean scene* and the scene after the displacement the *moved scene*. The instruction, the other objects, the basket, and the initial robot pose stay unchanged, and the instruction names the object but not its location. If the policy still reaches toward the old location, that behavior must come from the policy itself. We evaluate on LIBERO-Object (Liu et al., 2023) with the benchmark's original success criterion, using the public checkpoints of OpenVLA (Kim et al., 2024), OpenVLA-OFT (Kim et al., 2025), called OFT in figure labels, π_0 (Black et al., 2025), $\pi_{0.5}$ (Physical Intelligence et al., 2025), and GR00T N1.7 (Bjorck et al., 2025), called GR00T below, without any fine-tuning.



Figure 2: A LIBERO-Object scene with the target object at its training-time position (white) and at the new location after a planar displacement δ (red). Everything else stays unchanged.

Overview. Our claim has three parts. The displacement information is present in the visual tokens. A small number of attention heads carry it into the action tokens. Amplifying the contribution of those heads reduces the failure. Every part is tested on the same pair of scenes, the clean scene and the moved scene. Section 3.1 tests the first part by replacing visual token rows with their clean-scene values, and finds the depth at which the policy consumes the information. Section 3.2 tests the second part by replacing one head output at a time with its clean-scene value and screening the heads that respond, and Section 3.3 checks that the screened heads account for the consumption found in Section 3.1. Section 3.4 tests the third part by scaling the deviation of the

same head outputs. Because the screen and the intervention act on the same quantity, the heads we find are the heads we modify.

3.1 LOCATING THE DISPLACEMENT INFORMATION

Section 4.1 shows that the policies reach the old object location. One explanation is that the new location never enters the network. Another is that it enters but is not carried to the action. To separate the two we ask where the information enters and at which depth the policy consumes it, with activation patching (Vig et al., 2020; Meng et al., 2022) between two scenes that share the same initial state and the same replayed prefix, one clean and one with the object moved by δ. For each initial state we build an eight-direction grid of displacements at fixed magnitudes and replay the first k actions of the clean-scene episode in every grid cell before the query, so the robot pose is identical across cells and only the object differs. Every measurement in this section is *open loop*. The policy runs once on a fixed observation and we read the planned action without executing it. For chunk policies the *readout* $v \in \mathbb{R}^2$ is the sum of the planar displacements in the chunk. For the discrete policy we fix the clean-scene tokens and read the expected planar action from the logits, because the argmax hides small changes. For flow-matching policies we fix the noise per initial state, so the base and the patched passes use the same random numbers. Let v_{clean} and v_δ be the readouts of the two scenes. The *base response* $v_\delta - v_{\text{clean}}$ tells how far the planned action follows the object, and is the reference for every patch. For any patched forward pass we report the *transfer fraction*, the part of the base response that the patch moves. Let $u = (v_\delta - v_{\text{clean}})/\|v_\delta - v_{\text{clean}}\|$ be the unit vector along the base response in each grid cell. When clean-scene values are inserted into the moved scene and v_{patch} is the resulting readout,

$$T = \frac{\sum (v_\delta - v_{\text{patch}}) \cdot u}{\sum (v_\delta - v_{\text{clean}}) \cdot u}, \tag{1}$$

where the sums run over the grid cells of an initial state, so that a value of one means the patch removes all of the response and zero means none. When moved-scene values are inserted into the clean scene instead, as in the row test below, the numerator is $(v_{\text{patch}} - v_{\text{clean}}) \cdot u$, so that one means the patch carries all of the response (Appendix A).

To locate the information we look at the visual tokens at the input of the transformer, where each image patch occupies one row of the activation, for example 512 rows for two cameras of 16×16 patches in OpenVLA-OFT. We insert the moved-scene values of only a subset of visual token rows into the clean scene and measure the transfer fraction. Rows are ranked by the size of their difference between the two scenes. Controls use the same number of random rows, the remaining rows, and each camera separately.

To find the depth we replace all visual token rows at the output of layer l of the moved scene with their clean-scene values and write $T(l)$ for the transfer fraction. $T(l)$ starts at one and drops where the policy has already read the information out of the visual rows. The layers where $T(l)$ falls are the *consumption window*.

3.2 SCREENING HEADS BY DISPLACEMENT RESPONSE

If the information is present and consumed, some computation carries it from the visual rows to the action. We want the heads whose output carries it. In a transformer layer, each attention head h produces an output $z_h \in \mathbb{R}^d$ at every token position. The outputs of all heads are concatenated, multiplied by the output projection W_O, and added to the residual stream. We act on z_h right before this projection, in the screen below and in the intervention of Section 3.4 alike. This changes only what head h contributes to the residual stream, and that contribution is linear in z_h. The attention weights, the other heads, the MLP, and all parameters stay untouched. For each head we ask how much of the base response disappears when we remove the part of that head's output that depends on δ. On the same scene pair as Section 3.1 we run the moved scene again with z_h replaced by its clean-scene value at every token position (Wang et al., 2023; Zhang & Nanda, 2024) and obtain $v_{\delta,h}$. The *head response* is

$$w_h(\delta) = v_\delta - v_{\delta,h}, \tag{2}$$

the part of the base response that head h carried. The clean-scene value of z_h comes from the same robot pose and the same inputs except the object location. Write $z_h = c_h + d_h(\delta)$, where c_h collects everything that does not depend on δ. The replacement changes the head's contribution by

$d_h(\delta) - d_h(0)$ only, and c_h cancels. Through W_O, c_h adds a push to the action that is the same whatever the object location, which we call the head's *constant offset*, and w_h is unaffected by it. Zero or mean ablation removes c_h as well, so a head with a large constant offset and no displacement response would look like a head that responds to the displacement.

We compute w_h on the eight-direction grid of Section 3.1 and fit $w_h(\delta) \approx S_h\delta$ by least squares without intercept, where $S_h \in \mathbb{R}^{2\times 2}$ is the linear map from the displacement to the head response. With $g_h = \mathrm{tr}(S_h)/2$, the mean gain of the head over the two axes, and g_{base} the same quantity fitted from the base response $v_\delta - v_{\mathrm{clean}}$, the score is the *relative contribution*

$$G_h = g_h / g_{\mathrm{base}}. \tag{3}$$

A head with a large G_h carries a large share of the plan change caused by the object displacement. A head is kept only if its response has a consistent sign across directions and training slots and passes a permutation test. Kept heads are ranked by g_h divided by its standard error, and G_h reports their size. Details are in Appendix B.

3.3 Routing: the share carried by the screened heads

The screen says which heads respond, but every head of a layer reads the visual rows, so a screened head sitting at the depth where $T(l)$ drops is only a necessary sign. The evidence that a head carries the information is a *head patch* measured against the layer drop. To connect the two we replace only z_h of the screened heads inside the consumption window, in the same scene pair. We compare the transfer fraction of this head patch with the drop of $T(l)$ across the window, using the same denominator. We also split the patch by token rows, so that we can tell whether the head passes the information into the action rows, the visual rows, or the text rows.

3.4 Amplifying the screened heads

Activation addition and inference-time intervention (Turner et al., 2023; Li et al., 2023) add a fixed direction to the residual stream or to the outputs of selected heads, and recent VLA work reweights attention at inference without training (Zhang et al., 2026b) or trains attention heads to specialize (Jia et al., 2026). Our intervention keeps the head-level, training-free form but changes what is modified. It acts on the heads selected by the screen of Section 3.2 and on the same quantity z_h. For each such head we set

$$z_h \leftarrow \mu_h + g\,(z_h - \mu_h), \qquad g > 1. \tag{4}$$

Here μ_h is the mean of z_h over clean scenes of the same task at the same token position. The mean is kept as it is and only the input-dependent deviation from the mean is scaled by g. In the space of head outputs this moves z_h away from μ_h along its own deviation direction by the factor g (Figure 1, inset). Scaling the whole output instead, $z_h \leftarrow g\,z_h$, would also multiply the mean and add $(g-1)\,\mu_h$, a term that does not depend on the input and therefore shifts the action by a fixed amount regardless of δ. Keeping the mean removes this term. By the linearity noted in Section 3.2, this scales the deviation of the head's contribution to the residual stream by the same factor. Nothing is added that the head would not produce on its own, and the policy parameters are untouched.

One variant scales only the component of the deviation that lies in the displacement subspace of the head. This subspace is estimated by principal component analysis of the difference between the moved-scene and clean-scene values of z_h collected on the grid of Section 3.1, and Appendix C gives the procedure. It is the setting chosen for $\pi_{0.5}$ in Section 4.4.

The intervention has three choices, the number of heads K, the factor g, and whether to scale the full deviation or its subspace component. We fix these per policy by a two-stage procedure. The first stage is open loop and uses the grid of Section 3.1. The second stage is *closed loop*, where the full episode runs with the intervention active at every step and we record whether the task succeeds. Heads and settings are chosen on one set of initial states, which we call the *validation states*, and evaluated on a second set from disjoint environment seeds that is never used for any choice, which we call the *test states*. A candidate setting is kept only if it increases the planned displacement toward the object in most directions and changes the clean-scene plan by at most a fixed fraction of the base response. The surviving settings are then compared closed loop on the validation states, and one is chosen. Because this choice uses closed-loop outcomes, the closed-loop gain on the validation states is biased upward by the selection, and only the test states give an unbiased estimate

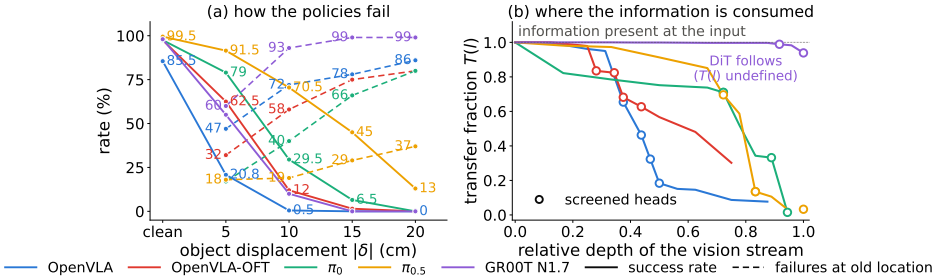

Figure 3: (a) Success rate under object displacement (solid) and, among the failed episodes, the fraction in which the gripper reaches the old object location (dashed), on the same 1,000 initial states for every policy. In the old-location failures the closest distance between the end effector and the old location is 0.6 to 2.1 cm for every policy and does not grow with the displacement. Labels that would overlap are omitted and all values are given in Table D.1. (b) Transfer fraction $T(l)$ when all visual token rows at depth l of the moved scene are replaced by their clean-scene values, against relative depth of the vision stream. Every policy starts at $T = 1$, so the displacement is fully present at the input, and drops in a limited depth range. Open circles mark the layers of the heads screened in Section 4.3. The GR00T curve covers its language backbone and vision-language blocks only, because its diffusion transformer reads the visual tokens through cross-attention and $T(l)$ is not defined there. The remaining drop from 0.94 to zero happens inside the diffusion transformer.

of the gain. Different policies end up with different settings, and we report this as an outcome of the procedure rather than a tuned choice. The candidates, the gate values, and the reason for each choice are given in Appendix B.

4 Experiments

4.1 Failure modes

We first ask how the five policies fail when the target object is moved. Figure 3(a) reports the success rate for each displacement magnitude and, among the failed episodes, the fraction in which the gripper reaches the old location of the object.

Three observations follow. First, reaching the old location is the dominant failure at 15 and 20 cm for four of the five policies, and it is present in all five. Second, the closest distance to the old location stays within about 2 cm at every displacement. If the failures came from a noisy estimate of the object position, this distance would grow with the displacement. It does not, so the policy reproduces a fixed target rather than misestimating the current one. Third, $\pi_{0.5}$ is the exception in kind and not only in degree. Its dominant failure at every magnitude is reaching the new location and then failing to grasp, and its old-location fraction never exceeds 37%.

4.2 Presence and consumption of the displacement information

We apply the row and layer patching of Section 3.1 to the five policies. Figure 3(b) gives the layer curve.

Two facts follow. First, the new location is fully present at the input, since $T(0) = 1$ in every policy, and it is local to a few object patches, under the wrist camera where one exists, with the same number of random rows carrying almost nothing (Appendix E). Second, the information is consumed in a limited depth range whose shape differs by architecture, a staircase in OpenVLA-OFT, a cliff in $\pi_{0.5}$, a slow decline followed by a drop in π_0, and a mid-network band in OpenVLA. In GR00T the language backbone consumes nothing and the drop is delayed to the last vision-language blocks and the diffusion transformer, which the curve cannot show because $T(l)$ is not defined for a cross-attention reader. **The old-location failure therefore occurs while the new location is fully present in the network, so what fails is the use of the information, not its presence.**

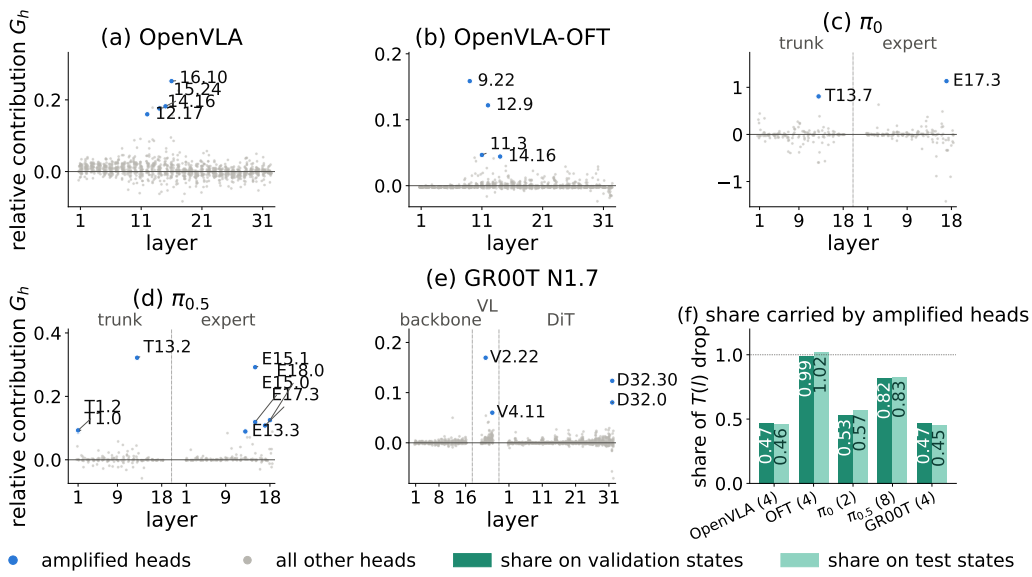

Figure 4: (a)–(e) Relative contribution G_h of every attention head, plotted against its layer, for the five policies. Gray points are all heads. Blue points are the heads amplified in Section 4.4, four in OpenVLA, OpenVLA-OFT, and GR00T, two in π_0, and eight in $\pi_{0.5}$. G_h is the fraction of the base displacement response that passes through head h, so a value of 0.16 means that replacing that head's output with its clean-scene value removes 16% of the response. The amplified heads lie in the middle layers of OpenVLA and OpenVLA-OFT, in trunk layer 13 and the late expert layers of π_0 and $\pi_{0.5}$, and in the vision-language and last DiT layers of GR00T. For π_0 the top head alone exceeds the base response. (f) Fraction of the drop of $T(l)$ that is explained by replacing, together, the amplified heads, on the validation and on the test states. The drop is taken over the depth range spanned by that head set, from the input of its shallowest head to the output of its deepest head, and the number of heads is given in parentheses. Values with the same four heads for every policy are in Table E.1.

4.3 Screening results and the share carried by the screened heads

We run the screen of Section 3.2 on every attention head of each policy to find the computation that carries the information out of the visual rows. Figure 4(a)–(e) shows the result. In every policy the displacement information is carried by a small number of heads that stand far above the rest, and their depths coincide with the drop of $T(l)$ in Figure 3(b). The stability of the ranking across initial states is high for $\pi_{0.5}$ and low for OpenVLA and GR00T, where the top heads are real but their order depends on the initial states used (Appendix F).

The screen also returns heads with the opposite sign, whose removal should increase the response. Their contribution is a few percent of the base response, and removing them in closed loop is indistinguishable from removing random heads in all five policies (Appendix G).

Coincidence in depth is only a necessary sign, so we measure the share of the drop that the screened heads account for with the head patch of Section 3.3. Figure 4(f) gives the result. The transfer carried by each screened head alone follows the same ranking as its screening score, so **the screen and the depth curve measure the same flow from two sides.** In OpenVLA-OFT the amplified heads account for essentially all of the drop over the depth range they span, in $\pi_{0.5}$ for about four fifths, and in OpenVLA, π_0, and GR00T for about half, with the rest spread over other heads of the same layers. Where the share is partial, adding more screened heads changes little, because the heads overlap in what they carry. In every policy the heads write into the action rows, and the text rows receive nothing, except in OpenVLA where the action tokens are predicted at the end of the prompt (Appendix E). All shares reproduce on the test states. **The failure is therefore located at the routing step.** The displacement enters the network, a few heads carry it to the action, and the

Table 1: **Screened heads and their amplification in the five policies.** Heads are written {stream}{layer}.{head} (T/E = π trunk/expert, V/D = GR00T vision-language/DiT) with their G_h. The four heads ranked highest by g_h over its standard error are listed. π_0 amplifies only the first two and $\pi_{0.5}$ these four plus the next four, scaling only the displacement-subspace component (Section 3.4). Success is paired closed-loop success (%) at the listed displacement with the change from base in parentheses. Amplified pools validation and test states, and test is the unbiased estimate on the test states alone. Random and matched are the controls of the protocol in this section, and matched reports the deviation-matched control where such heads exist and the energy-matched control otherwise. Changes are computed from unrounded rates. An asterisk marks McNemar $p < .05$ in either direction, and bold marks a significant gain.

Policy	screened heads (G_h)				setting, displacement (δ)	closed-loop success (%)				
	1	2	3	4		base	amplified	test	random	matched
OpenVLA	16.10 0.25	12.17 0.16	14.16 0.18	15.24 0.18	4 heads $\times 2$ 5 cm	20.3	23.4 (+3.1)	23.8 (+1.9)	16.9 (−3.4)	14.7 (−5.6)*
OpenVLA-OFT	9.22 0.16	12.9 0.12	14.16 0.04	11.3 0.05	4 heads $\times 2$ 10 cm	12.3	**17.9 (+5.6)***	14.9 (+3.7)	12.7 (+0.4)	11.9 (−0.4)
π_0	E17.3 1.13	T13.7 0.81	E16.2 0.49	E13.5 0.37	2 heads $\times 1.5$ 10 cm	29.1	**41.8 (+12.7)***	**35.8 (+10.4)***	30.2 (+1.1)	31.7 (+2.6)
$\pi_{0.5}$	T13.2 0.32	E15.1 0.29	E18.0 0.13	E15.0 0.12	8 heads $\times 1.5$ subspace, 15 cm	42.1	**50.5 (+8.3)***	**56.0 (+14.7)***	42.6 (+0.5)	43.5 (+1.4)
GR00T N1.7	V2.22 0.17	D32.30 0.12	D32.0 0.08	V4.11 0.06	4 heads $\times 2$ 5 cm	53.4	52.5 (−0.9)	55.0 (+3.8)	54.3 (+0.9)	54.3 (+0.9)

policy still reaches the old location, so **what is missing is not the information but the size of its contribution.**

4.4 Head amplification results

We amplify the screened heads with the intervention of Section 3.4.

Protocol. Closed-loop success is compared between the base policy and an intervention on the same initial state and the same δ. We report the net change over the full episode set with McNemar's test on the paired outcomes. We also report the net change in each half-plane of δ and require that no half-plane is harmed. Because the closed-loop harness is not fully deterministic, we never judge an intervention from a single run. Controls take the same form as the intervention. For amplifying K heads by a factor g, we use three controls, each ruling out one alternative explanation of a gain. The first amplifies K random heads by the same g, and rules out that any head would do. The second amplifies K heads that do not respond to the object displacement but whose outputs vary as much as the amplified heads, again by the same g, and rules out that the size of the deviation is what matters. The third uses the same non-responsive heads but raises their factor above g so that the total change in head output matches that of the amplified heads, and rules out that the amount of injected change is what matters. Every result is reported on the validation and on the test states of Section 3.4.

Object displacement on LIBERO-Object. Table 1 lists, for each policy, the screened heads and the paired closed-loop success before and after amplifying them, next to the matched controls. For $\pi_{0.5}$ the selected setting scales only the displacement-subspace component of the deviation, because scaling the full deviation gave a gain in the $+y$ half-plane only. In OpenVLA-OFT, π_0, and $\pi_{0.5}$ the amplification raises success, harms no half-plane, leaves clean-scene and small-displacement episodes unchanged, and reproduces on the test states. The gain is specific to the screened heads, since random heads at the same factor, inert heads with matched deviation magnitude, and heads with a matched size of the added change are all ineffective in all five policies. The two policies without a gain show that carrying the displacement is necessary for the lever but not sufficient. In OpenVLA the screened heads are real but the same intervention produces a state-specific offset, one that depends on the initial state but not on δ, rather than tracking, and in GR00T the base policy tracks strongly in one direction and weakly in the others, and the screened heads amplify that imbalance. The range of factors that raise success at the main displacement without harming small displacements or clean scenes, which we call the *factor window*, is narrow, and Appendix H reports the sweep.

Table 2: **Transfer to LIBERO-Pro.** The same interventions as in Table 1, applied without change to the LIBERO-Pro perturbations. Cells give base success → amplified success (%) with the change in parentheses. Episodes per cell are 100, except 50 for $-y$ and for every OpenVLA cell (25 for OpenVLA $-y$). The first six rows move the object by 7 cm, the next two by 14 cm, and the last four leave the target in place. Random-head controls change success by at most 4 points in the three policies with a gain and never reach significance in GR00T (-7 to $+9$) or OpenVLA (-4 to $+6$). An asterisk marks McNemar $p < .05$, and bold marks a significant gain.

Perturbation	OpenVLA	OpenVLA-OFT	π_0	$\pi_{0.5}$	GR00T N1.7
$+y$ 7 cm, slot axis	0 → 6 (+6)	0 → 5 (+5)	61 → **75** (+**14**)*	99 → 99 (+0)	15 → 16 (+1)
$\pm x$ 7 cm, slot axis	4 → 4 (+0)	62 → 69 (+7)	59 → **73** (+**14**)*	90 → 89 (−1)	40 → 38 (−2)
$+y$ 7 cm	4 → 8 (+4)	1 → **9** (+**8**)*	54 → **68** (+**14**)*	90 → **99** (+**9**)*	9 → 17 (+8)
$+x$ 7 cm	6 → 8 (+2)	37 → 39 (+2)	51 → 59 (+8)	60 → 70 (+10)	34 → 37 (+3)
$-x$ 7 cm	2 → 2 (+0)	62 → 68 (+6)	75 → 75 (+0)	90 → 96 (+6)	52 → 53 (+1)
$-y$ 7 cm	0 → 4 (+4)	58 → **82** (+**24**)*	48 → 46 (−2)	90 → 88 (−2)	30 → 24 (−6)
$+y$ 14 cm	0 → 2 (+2)	0 → 0 (+0)	3 → **10** (+**7**)*	65 → 73 (+8)	0 → 0 (+0)
$\pm x$ 14 cm	0 → 0 (+0)	0 → 0 (+0)	5 → 7 (+2)	52 → 58 (+6)	1 → 0 (−1)
clean	84 → 78 (−6)	99 → 95 (−4)	95 → 95 (+0)	98 → 100 (+2)	95 → 94 (−1)
language	76 → 80 (+4)	98 → 96 (−2)	98 → 99 (+1)	100 → 98 (−2)	100 → 96 (−4)
task	4 → 0 (−4)	0 → 0 (+0)	10 → 10 (+0)	11 → 9 (−2)	10 → 10 (+0)
object swap	0 → 0 (+0)	4 → **16** (+**12**)*	0 → 0 (+0)	20 → 26 (+6)	0 → 0 (+0)

Transfer to LIBERO-Pro. Table 2 applies the same interventions, unchanged, to LIBERO-Pro, where the effect is specific to object displacement. The intervention raises success on the displaced conditions where the base policy is not saturated, harms none of them significantly, and does nothing under language and task perturbations. The object-swap condition, in which the target sits at another object's slot, is a displacement in disguise and gains in OpenVLA-OFT. The two policies without a lever also show no transfer, since in GR00T every change lies within the range of the random control and in OpenVLA the base policy is at or near zero on every displaced condition.

Appendix I shows that the gain arises in the approach phase in all three policies, and Figure I.1 shows the base policy closing on empty space while the amplified policy lifts the object, and Appendix J compares the intervention with a published activation-steering method under the same protocol.

5 Conclusion

When the target object is moved, five VLA policies reach for the location where it used to be. The policy does not lack the information about the new location. It fails to use it. In every policy the displacement is present in a few visual token rows and is carried into the action tokens by a small number of attention heads. A screen that replaces one head's output with its clean-scene value finds these heads, and scaling their deviation without training raises success under displacement in three of the five policies, with no harm to any direction, reproduction on the test states, and transfer to LIBERO-Pro. The gain does not appear for random or deviation-matched heads, removing heads helps no policy, and amplification fails in OpenVLA and GR00T for reasons the same measurements identify, a state-specific offset in the first and one-sided tracking in the second. The lever is therefore real but conditional, and the conditions can be measured with the same tools that find the heads. The screened heads belong to a checkpoint, so a policy whose weights change through fine-tuning must be screened again before the intervention is applied. Future work will apply the screen and the amplification to such fine-tuned policies, in particular to policies fine-tuned for a physical robot. The screen should also be extended to other perturbation axes.

Limitations. The factor window is narrow and differs by policy, and neither restricting the amplification to a subspace nor conditioning the mean on the query phase widens it (Appendix H). In OpenVLA-OFT the heads ranked third and below have nearly equal scores, so the screen does not fix which of them enter the amplified set. Sets built from other heads of that rank range also gain or stay neutral on the test states, but with a few hundred episodes we cannot tell whether one such set is a few points better than another. All results come from simulation, on two related suites, with a small number of test states, and the harness is not fully deterministic for the flow-matching policies. Finally, we have not yet verified that the screen finds usable heads after fine-tuning.

AI USE STATEMENT

We used a generative AI coding assistant throughout this work. It wrote and edited experiment code under our direction, including the screening, patching, and intervention tools, the closed-loop evaluation harness, the analysis scripts, and the figure scripts, and it launched and monitored many of the runs. It assisted with the writing of this paper, producing draft passages from our outlines, experiment reports, and logs, which we then edited, rewrote, and revised through repeated rounds of comments. It also proposed some of the analyses that we then ran, such as the paired net-change accounting and the matched-deviation controls. All experimental designs, the choice of policies and perturbations, the hypotheses, and the interpretation of the results are ours. We reviewed the AI-written code, checked the reported numbers against the underlying logs, and rewrote or removed drafted claims that we could not verify. It also helped with the related work, drafting that section from papers we selected, checking that every cited work exists, and summarizing cited papers on request, and we verified each cited claim against the original paper. All experimental data come from running the policies in the simulator through the code described above. No generative model produced, altered, labeled, or selected any observation, action, or outcome in that data. The analyses it proposed were checked against the literature by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies and modifies the internal computation of publicly released robot policies in a simulated benchmark. It involves no human subjects, no personal data, and no new data collection. All policies, checkpoints, and evaluation suites are public and were used under their licenses. The intervention we propose changes a policy's behavior at inference time without retraining. Such a change can improve or degrade a policy, and we report the cases where it harms performance as carefully as those where it helps. Any deployment of an inference-time intervention on a physical robot should be validated in closed loop under the conditions of use, because our results show that open-loop measurements do not predict closed-loop outcomes. We see no direct path from this work to harmful applications beyond those that apply to robot manipulation policies in general. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The evaluation protocol, the definitions of the screen and the intervention, and the statistical tests are given in Sections 3 and 4.4. Appendix K lists the hardware, the software versions, and the wall-clock time of every experiment type. Appendix B gives the screening gates, the open-loop gate values, the candidate sets, and the reason for each chosen setting, and Appendix C gives the subspace estimation. The full values behind the figures are in Appendices D and E, and the factor sweep is in Appendix H. All policies are public checkpoints evaluated without fine-tuning on the public LIBERO-Object and LIBERO-Pro suites. The code for the screen, the patching, the intervention, and the paired evaluation will be released with the camera-ready version.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems (RSS)*, 2025.

Hugo Buurmeijer, Carmen Amo Alonso, Aiden Swann, and Marco Pavone. Observing and controlling features in vision-language-action models. *arXiv preprint arXiv:2603.05487*, 2026.

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.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pp. 7324–7338. PMLR, 2022.

Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models. *arXiv preprint arXiv:2603.19233*, 2026.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models. *arXiv preprint arXiv:2509.00328*, 2025.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

Dongchi Huang, Zhirui Fang, Tianle Zhang, Yihang Li, Lin Zhao, and Chunhe Xia. Co-RFT: Efficient fine-tuning of vision-language-action models through chunked offline reinforcement learning. *arXiv preprint arXiv:2508.02219*, 2025.

Jaehwan Jeong, Evelyn Zhu, Jinying Lin, Emmanuel Jaimes, Tuan-Anh Vu, Jungseock Joo, Sangpil Kim, and M Khalid Jawed. Your vision-language-action model already has attention heads for path deviation detection. *arXiv preprint arXiv:2603.13782*, 2026.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

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.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. AtP*: An efficient and scalable method for localizing LLM behaviour to components. *arXiv preprint arXiv:2403.00745*, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. In *Conference on Robot Learning (CoRL)*, 2024.

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.

Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can RL bring to VLA generalization? An empirical study. *Advances in Neural Information Processing Systems*, 38:97121–97151, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

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.

Miranda Muqing Miao, Subin Kim, Brandon Yang, and Lyle Ungar. Contrastive conceptor activation steering (COAST): Unlocking vision-language-action models through hidden states. *arXiv preprint arXiv:2605.17144*, 2026.

Neel Nanda. Attribution patching: Activation patching at industrial scale. `https://www.neelnanda.io/mechanistic-interpretability/attribution-patching`, 2023.

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.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, Jiayuan Gu, Bin Zhao, Dong Wang, et al. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Robotics: Science and Systems (RSS)*, 2025.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies. In *Conference on Robot Learning (CoRL)*, 2025.

Aiden Swann, Lachlain McGranahan, Hugo Buurmeijer, Monroe Kennedy III, and Mac Schwager. Sparse autoencoders reveal interpretable and steerable features in VLA models. *arXiv preprint arXiv:2603.19183*, 2026.

Aaquib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 407–416, 2024.

Jincheng Tang, Yilong Zhu, Zhengyuan Xie, Jiang-Jiang Liu, and Jiaxing Zhang. The moving eye: Enhancing VLA spatial generalization via hybrid dynamic data collection. *arXiv preprint arXiv:2607.02322*, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. ManiSkill3: GPU parallelized robotics simulation and rendering for generalizable embodied AI. In *Robotics: Science and Systems (RSS)*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

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)*, 2020.

Guodong Wang, Chenkai Zhang, Qingjie Liu, Jinjin Zhang, Jiancheng Cai, Junjie Liu, and Xinmin Liu. LIBERO-X: Robustness litmus for vision-language-action models. *arXiv preprint arXiv:2602.06556*, 2026.

Huanqian Wang, Chi Bene Chen, Yang Yue, Danhua Tao, Tong Guo, Shaoxuan Xie, Denghang Huang, Shiji Song, Guocai Yao, and Gao Huang. Move: A simple motion-based data collection paradigm for spatial generalization in robotic manipulation. *arXiv preprint arXiv:2512.04813*, 2025.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

Fan Zhang, Seongbin Park, Baharan Mirzasoleiman, Shariar Talebi, and Nader Sehatbakhsh. ProbeAct: Probe-guided training-free failure recovery in vision-language-action models. *arXiv preprint arXiv:2606.09740*, 2026a.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024.

Ninghao Zhang, Bin Zhu, Shijie Zhou, and Jingjing Chen. Restoring linguistic grounding in VLA models via train-free attention recalibration. *arXiv preprint arXiv:2603.06001*, 2026b.

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*, 2025.

A TRANSFER FRACTION

The transfer fraction is defined in Section 3.1. The sums in its definition run over grid cells with $\|v_\delta - v_{\text{clean}}\|$ above a small threshold. For the row locality test the clean scene receives rows from the moved scene. For the depth curve and the head decomposition the moved scene receives values from the clean scene. Replacing an activation with its own value gives $T = 0$ and replacing all visual rows at the input gives $T = 1$, which we verified. Confidence intervals come from a bootstrap over initial states. Patch sites differ per policy and are given as the windows in Table E.1.

B SCREENING DETAILS

The sign gate requires the same sign of $w_h \cdot \hat{\delta}$ in at least a fixed number of the eight directions. The slot gate requires the same sign of g_h in both groups of tasks, where the groups are defined by which training slot holds the target. The permutation test shuffles the direction labels within each initial state and recomputes g_h many times. Heads that pass all gates are ranked by g_h divided by its standard error, where the standard error is estimated by leaving out one initial state at a time. For reference, a head with $G_h = 0.2$ carries one fifth of the policy's response to the object displacement. The number of replayed prefix actions k and the number of grid states differ per policy. The number of states used by the attribution score is listed in Table F.1, and the exact screen uses 10 of them.

Selection of the number of heads and the factor. Table B.1 lists, for each policy, the candidate settings of Section 3.4, the ones that pass the open-loop gate, and the one kept after the closed-loop comparison. The gate removes settings before any closed-loop run, but the final choice among the survivors uses closed-loop results on the validation states, so the table records the selection and is not a prediction from the screen alone. The gate requires a positive response in at least seven of the eight directions and a clean-scene plan change of at most half the base response (Table B.2). In OpenVLA the ranking is not reproducible across initial-state sets, so we kept only the four heads that rank in the top eight on both sets and varied the factor alone. The factor that passed the gate, 1.5, harmed the $-y$ half-plane in closed loop, and the factor 2 exceeded the clean limit slightly, so the kept setting for OpenVLA is the one exception to the gate, and the reason is recorded in Table B.1. In OpenVLA-OFT sets of two, four, and sixteen heads pass, and sixteen heads give the largest open-loop gain but no closed-loop gain. In π_0 one head carries more than the whole base response, and two heads suffice. In $\pi_{0.5}$ the response is spread over eight heads. Amplifying only the top two or four moves the plan toward the object in six of the eight directions but away from it when the object is displaced toward $-y$, and adding the next four heads restores those directions, so eight are needed. In GR00T no set passes the direction gate.

Table B.1: Selection of the number of amplified heads K and the factor g. Candidates are the top-K screened heads at each factor. Passing sets satisfy the open-loop gate. The kept setting is the one used in Section 4.4, chosen among the passing sets by closed-loop outcome on the validation states. Full open-loop values are in Table B.2.

Policy	Candidates	Passing the gate	Kept	Reason for the choice
OpenVLA	K=4 (heads in the top eight of both state sets), $g \in \{1.5, 2, 3\}$	g=1.5	K=4, g=2	g=1.5 harmed the $-y$ half-plane in closed loop, g=2 exceeds the clean limit only slightly (0.57)
OpenVLA-OFT	$K \in \{1, \ldots, 64\}$, $g \in \{1.5, 2\}$	K=2 at g=1.5, K=4, K=16	K=4, g=2	K=16 has the largest open-loop gain but no closed-loop gain, K=4 gains at g=2
π_0	$K \in \{1, 2, 4, 8\}$, $g \in \{1.5, 2, 3\}$	all except $K \geq 4$ at g=3	K=2, g=1.5	one head carries 1.1 of the base response, a single head at g=2 is biased to $+y$, larger factors harm 5 cm
$\pi_{0.5}$	$K \in \{1, 2, 4, 8\}$, $g \in \{1.25, 1.5, 2\}$	K=1, K=8	K=8, g=1.5	K=2 and 4 track away from the object for $-y$ displacements, K=1 alone gives a small closed-loop gain
GR00T N1.7	$K \in \{1, 2, 4, 8\}$, $g \in \{1.25, 1.5, 2\}$	none (3 to 5 directions)	K=4, g=2 (default)	no setting is balanced across directions, the pre-specified default is run for completeness

Open-loop candidate sets. Table B.2 gives, for every candidate set, the number of the eight grid directions in which the amplified plan moves further toward the object than the base plan, and the clean-scene plan change as a fraction of the base displacement response. A set passes the gate if

the first is at least seven and the second at most 0.5. Sets are the top-K screened heads, and for OpenVLA only the four-head set of the paragraph above was tested.

Table B.2: Open-loop gate values for every candidate set. Each cell gives the number of grid directions (of 8) with a positive response, followed by the clean-scene plan change relative to the base response. Bold entries pass the gate.

Policy	K	$g = 1.25$	$g = 1.5$	$g = 2$	$g = 3$
OpenVLA	4	—	**8, 0.45**	7, 0.57	8, 1.50
OpenVLA-OFT	1	—	6, 0.07	5, 0.12	—
OpenVLA-OFT	2	—	**7, 0.09**	6, 0.15	—
OpenVLA-OFT	4	—	**7, 0.11**	**7, 0.21**	—
OpenVLA-OFT	8	—	6, 0.15	6, 0.28	—
OpenVLA-OFT	16	—	**7, 0.17**	**7, 0.34**	—
OpenVLA-OFT	32	—	6, 0.25	6, 0.57	—
OpenVLA-OFT	64	—	5, 0.50	5, 1.25	—
π_0	1	—	**8, 0.10**	**8, 0.18**	**8, 0.33**
π_0	2	—	**8, 0.11**	**8, 0.21**	**8, 0.41**
π_0	4	—	**8, 0.15**	**8, 0.28**	8, 0.56
π_0	8	—	**8, 0.17**	**8, 0.36**	8, 0.81
$\pi_{0.5}$	1	**8, 0.03**	**8, 0.06**	**7, 0.12**	—
$\pi_{0.5}$	2	6, 0.06	6, 0.12	6, 0.23	—
$\pi_{0.5}$	4	6, 0.11	6, 0.20	6, 0.37	—
$\pi_{0.5}$	8	**7, 0.13**	**7, 0.23**	**7, 0.42**	—
GR00T N1.7	1	5, 0.03	5, 0.06	3, 0.16	—
GR00T N1.7	2	4, 0.10	4, 0.19	4, 0.42	—
GR00T N1.7	4	5, 0.14	4, 0.28	4, 0.57	—
GR00T N1.7	8	5, 0.15	5, 0.29	4, 0.55	—

C Intervention details

For the subspace variant, we collect $d = z_h^{\delta} - z_h^{\text{clean}}$ at every token position for every grid state and direction and run principal component analysis on the pooled vectors. The top r components form an orthonormal basis U_h of the head's channel space. The intervention then scales only $U_h U_h^{\top}(z_h - \mu_h)$ and leaves the orthogonal remainder unchanged. We also report the fraction of the deviation variance that lies in U_h. The open-loop gate requires a positive projection of the plan change onto the displacement direction in at least a fixed number of the eight grid directions, and a clean-scene plan change below a fixed fraction of the base response at the smallest displacement.

D Failure modes, full table

Table D.1: Success rate (%) under object displacement and, in parentheses, the fraction (%) of failed episodes in which the gripper reaches the old object location. The same 1,000 initial states are used for every policy. These are the values plotted in Figure 3.

Displacement	OpenVLA	OpenVLA-OFT	π_0	$\pi_{0.5}$	GR00T N1.7
clean	85.5	97.5	97.5	99.5	98.0
5 cm	20.8 (47)	62.5 (32)	79.0 (17)	91.5 (18)	55.0 (60)
10 cm	0.5 (72)	12.0 (58)	29.5 (40)	70.5 (19)	10.0 (93)
15 cm	0.0 (78)	1.5 (75)	6.5 (66)	45.0 (29)	0 (99)
20 cm	0 (86)	0 (80)	0 (80)	13.0 (37)	0 (99)

E Causal chain, full table

Presence and locality of the displacement information. Before asking which heads carry the displacement, we checked that it is present at the input and where it is located. Replacing all visual token rows at the input of the moved scene with their clean-scene values removes the whole response in every policy, so $T(0) = 1$ and the new location is fully available to the network. Replacing only

the rows whose activations differ most between the two scenes shows that the information is local. Thirty-two rows, the patches of the object (under the wrist camera where one exists), out of 512 in OpenVLA-OFT and $\pi_{0.5}$, 256 in OpenVLA, and 128 in GR00T, carry 0.63 to 0.95 of the transfer, while the same number of random rows carries at most 0.10 and the selected and remaining rows together account for the full transfer. π_0 is the one policy in which the information is spread more widely and 64 rows are needed for the same transfer. Where two cameras exist the wrist camera supplies 0.79 to 0.99 of the transfer. The old-location failure of the main text therefore occurs while the new location is fully present in the visual tokens.

Full values. Table E.1 lists the locality, the consumption window, the head shares, and the destination rows for all five policies. The head share is given both with the same four heads in every policy and with the amplified set of Section 4.4, whose size differs by policy. Adding the third and fourth screened heads of π_0 raises the share by 0.01 on the validation states and 0.03 on the test states, and the eight heads of $\pi_{0.5}$ overlap strongly, since their joint transfer is 0.58 of the sum of their individual transfers. The text rows receive nothing except in OpenVLA, where the discrete action tokens are predicted at the end of the prompt and the prompt rows therefore receive part of the transfer.

Table E.1: Causal chain in the five policies, full values. Locality is the transfer fraction carried by the visual token rows with the largest clean-to-moved difference, with the fraction of that transfer that comes from the wrist camera. Share (4) is the fraction of the drop of $T(l)$ across the consumption window that is explained by replacing the top four screened heads together. Share (amplified) uses the head set amplified in Section 4.4, with the drop taken over the depth range spanned by that set. Both are given on validation states / test states. Destination is the token rows into which the heads write. Random rows of the same count carry at most 0.10 in every policy.

Policy	Locality (rows, transfer)	Wrist share	Share (4), window	Share (amplified), K, window	Destination
OpenVLA	32 of 256, 0.93 to 0.95	single camera	0.38 / 0.45, L8–20	0.47 / 0.46, 4, L10–16	action and prompt rows
OpenVLA-OFT	32 of 512, 0.74	0.93	0.84 / 0.87, L8–16	0.99 / 1.02, 4, L8–14	action rows 0.89
π_0	64 of 512, 0.87	0.99	0.54 / 0.60, L9–17	0.53 / 0.57, 2, L12–17	expert action rows 0.81
$\pi_{0.5}$	32 of 512, 0.63	0.79	0.83 / 0.87, L9–15	0.82 / 0.83, 8, L0–17	action rows 0.80, visual rows 0.58
GR00T N1.7	32 of 128, 0.89 to 0.93	0.93	0.50 / 0.47, VL out–end	0.47 / 0.45, 4, VL 1–end	VL image and DiT action rows

F ATTRIBUTION APPROXIMATION

The exact screen needs one patched forward pass per head and per scene. We use first-order attribution patching (Syed et al., 2024; Kramár et al., 2024) to rank heads with one backward pass per scene. Let $\hat{\delta}$ be the unit displacement direction. We take the gradient of the readout component $v \cdot \hat{\delta}$ with respect to z_h in the moved scene. The head response along $\hat{\delta}$ is approximated by

$$w_h(\delta) \cdot \hat{\delta} \approx \Big\langle z_h^{\delta} - z_h^{\text{clean}},\ \frac{\partial (v \cdot \hat{\delta})}{\partial z_h} \Big\rangle, \tag{F.1}$$

where the inner product runs over all token positions. This is a first-order approximation to the quantity measured by the exact screen. We use it only to rank heads, then verify the ranking with the exact screen. Agreement with the exact screen over all heads is given per policy in Table F.1. No head selection or closed-loop result in the main text depends on it, and only the ranking-stability statement of Section 4.3 uses its split reproducibility. The only place where the approximation is used on its own is the per-channel screen of Appendix I.

G HEAD REMOVAL DOES NOT HELP

The screen also returns heads with the opposite sign, whose removal should increase the response. In every policy their contribution is a few percent of the base response. Table G.1 shows that removing them in closed loop is indistinguishable from removing random heads. There is no head whose removal makes the policy follow the object.

A different class of head explains why ablation studies can look successful. Some heads have no displacement response but add a large constant offset to the action. Removing them shifts the first

Table F.1: Agreement between the attribution score and the exact screen over all heads. Heads and states give the number of heads and the number of initial states of the attribution score, and the exact screen uses 10 of these states. Recovery is the fraction of the exact top 5% that appears in the attribution top 5%. Split reproducibility is the Spearman correlation of the attribution score between two disjoint sets of initial states, with the overlap of the top 20 heads in parentheses.

Policy	Heads, states	Spearman / Pearson	Top 5% recovered	Split reproducibility
OpenVLA	1024, 20	0.66 / 0.75	0.71	0.10 (1/20)
OpenVLA-OFT	1024, 20	0.83 / 0.98	0.63	0.62 (15/20)
π_0	288, 10	0.72 / 0.71	0.64	0.60 (13/20)
$\pi_{0.5}$	288, 10	0.98 / 1.00	1.00	0.95 (19/20)
GR00T N1.7	1408, 10	0.58 / 0.83	0.34	0.26 (4/20)

Table G.1: Closed-loop net change in success (percentage points) when the heads with the largest opposite-sign score are replaced by their clean-scene mean, compared with replacing the same number of random heads. Paired over the same initial states and displacements. None of the differences is significant.

Policy	Heads	Displacement	Screened heads	Random heads
OpenVLA	8	5 cm	-3.4	-3.4
OpenVLA-OFT	4	10 cm	$+0.7$	-1.5
π_0	8	10 cm	-3.7	0.0
$\pi_{0.5}$	8	10 cm	$+4.5$	$+2.6$
GR00T N1.7	8	5 cm	$+2.8$	-2.5

gripper closure by a fixed vector regardless of δ. Success then rises in the half-plane that the shift points into and falls in the opposite half-plane and at small displacements. We observed this pattern in OpenVLA-OFT, π_0, and $\pi_{0.5}$, the three policies for which we ran a zero-ablation screen. A rescue rate computed only on failed episodes would report the rising half-plane as a gain. The paired net change over all episodes does not.

H FACTOR SWEEP

The factor has a narrow window. In $\pi_{0.5}$ the factor 2 gives a smaller gain than 1.5 on the test states, which is not significant, and harms one half-plane. Above the window the first gripper closure shifts by a vector that does not depend on δ, and small displacements are harmed. We also tested a query-conditioned mean, in which μ_h is replaced by $\mu_{h,k}$, the clean-scene mean of z_h at the k-th policy query of the episode, with the nearest stored query index not exceeding the current query used at inference. Neither restricting the amplification to the displacement subspace nor conditioning the mean on the query index widens the window in any of the five policies, so the offset is tied to the initial state rather than to the phase of the chunk.

Figure H.1 and Table H.1 give the paired net change in success on the validation states when the amplified head set of each policy is scaled by different factors. In OpenVLA-OFT and π_0 the change at the main displacement rises and then falls with the factor, and the fall is accompanied by harm at 5 cm, a loss of clean-scene success, and a first-close offset that grows with the factor. For $\pi_{0.5}$ the factor 1.25 was not run because it lost to 1.5 open loop. The factor 2 gives $+2.8$ on the validation states and $+7.3$ on the test states against $+1.9$ and $+14.7$ for the factor 1.5, loses three clean-scene episodes on the validation states, and harms the $+x$ half-plane by -11.9 points in the pooled set, so the upper end of the window lies between 1.5 and 2 in this policy as well. Restricting the amplification to the displacement subspace does not move the upper end of the window in π_0, where the subspace factor 3 harms 5 cm by -11.9 as the full factor 3 also does.

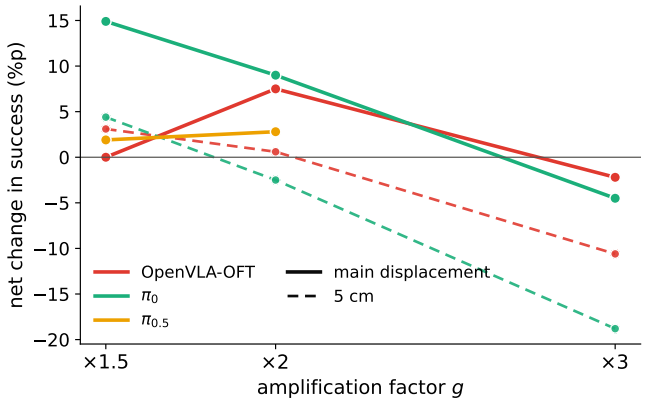

Figure H.1: Net change in success on the validation states against the amplification factor, for the amplified head set of each policy. Solid lines are the main displacement of Table 1 and dashed lines are 5 cm. The gain rises and then falls with the factor, and the fall at the main displacement is preceded by harm at 5 cm.

Table H.1: Net change in success (percentage points) on the validation states as a function of the factor, for the amplified head set of each policy. Main is the displacement of Table 1. Clean is the number of successful clean-scene episodes out of 20 with the intervention (base 20 in every case). Bold marks the kept factor.

Policy	Factor	Main	5 cm	Clean	First-close offset (cm)
OpenVLA-OFT, 4 heads	×1.5	0.0	+3.1	20	—
	×2	**+7.5**	+0.6	20	(−0.7, −0.5)
	×3	−2.2	−10.6	19	(−2.5, −2.3)
π_0, 2 heads	×1.5	**+14.9**	+4.4	19	(+0.3, +0.2)
	×2	+9.0	−2.5	18	(+1.0, +0.9)
	×3	−4.5	−18.8	14	(+2.1, +2.1)
	subspace ×2	+14.9	−3.1	19	(+0.6, +0.1)
	subspace ×3	+1.5	−11.9	18	—
$\pi_{0.5}$, 8 heads, subspace	×1.5	**+1.9**	—	20	(−0.6, +0.5)
	×2	+2.8	—	17	(−1.8, +0.7)

I What the amplification changes

A closed-loop gain says that the intervention helps but not when or through which action channel. We use three decompositions. First, we split each episode at the first step where the gripper command crosses its closing threshold and switch the intervention on only before or only after that step, pairing outcomes with the base episode as in the main protocol. Second, we split the deviation $z_h - \mu_h$ into its component in the displacement subspace U_h and the orthogonal remainder, and amplify each one alone. Third, we repeat the screen at the moment of grasp. The executed actions of a base episode are replayed up to the first close in the moved scene and in a clean scene in which the object sits under the gripper, and the attribution approximation of Appendix F is computed at the query just before and just after that step with the readout set to the planar displacement, the summed vertical displacement, or the summed gripper command of the chunk. All closed-loop numbers in the three decompositions below are net changes in success (percentage points) at 10 cm on the validation states ($n = 314$ per arm). Figure I.1 shows the two outcomes that the first decomposition separates. For one episode per policy it gives the base policy shortly after its first gripper close and the amplified policy after it has lifted the object, from the same initial state and the same displacement.

Three results follow. First, applying the intervention only after the first close does not reach the full gain in any policy. In π_0 and $\pi_{0.5}$ the before-only arm reaches the full gain and moves the first close

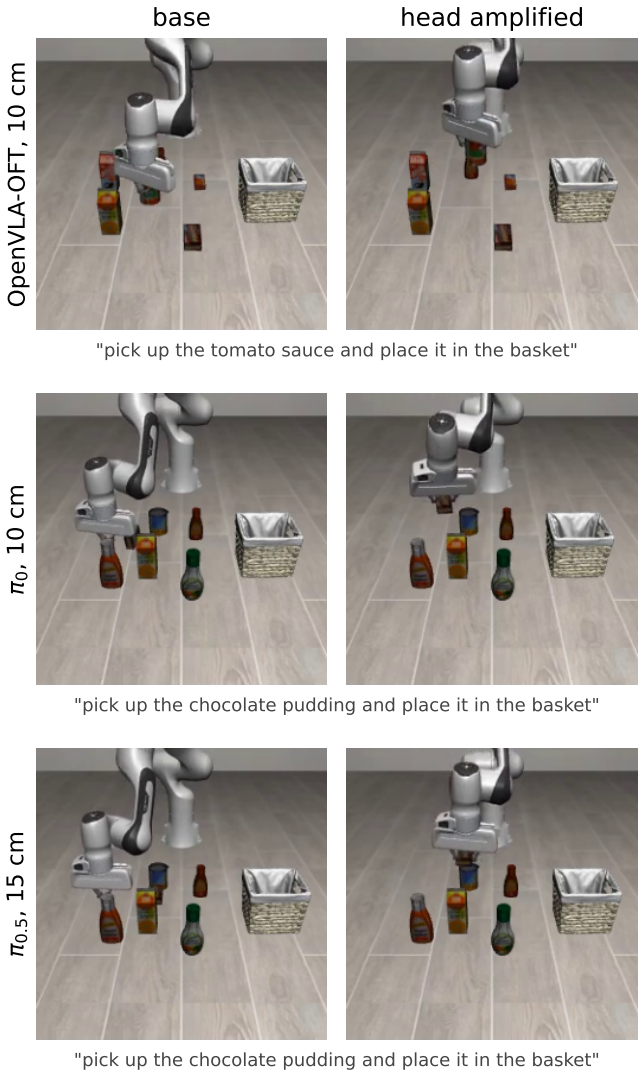

Figure I.1: The first gripper close, base against head amplification. Each row is one episode from the same initial state and the same object displacement, run with the base policy (left) and with the amplified heads of Table 1 (right). The base frame is taken shortly after the first step where the gripper command crosses its closing threshold, and the amplified frame after the object has been lifted. The base policy closes on empty space at the training-time position of the object, and the amplified policy grasps the object at its new location, lifts it, and goes on to place it in the basket. Rows are OpenVLA-OFT at 10 cm, π_0 at 10 cm on a test initial state, and $\pi_{0.5}$ at 15 cm on a test initial state. The corresponding videos are in the supplementary material.

toward the object by the same amount as the full arm. In OpenVLA-OFT the first close does not move at all, the before-only arm recovers less than half of the gain, and the after-only arm recovers none, so the two phases interact. The gain therefore originates in the approach phase in all three policies, even where its visible effect appears later. Second, the component that carries the gain differs. In π_0 the subspace component alone reproduces the full gain, and in $\pi_{0.5}$ it exceeds it while removing the $+y$ bias of the full arm. In OpenVLA-OFT neither component alone has an effect and only the full deviation does. Third, the head ranked first by the approach-phase screen is also ranked first for planar displacement at the moment of grasp in OpenVLA-OFT and π_0 (9.22 and E17.3), and in $\pi_{0.5}$ the eight amplified heads occupy ranks 1 to 9 of the grasp-moment screen. In OpenVLA-OFT and π_0 that head is also among the strongest heads for the vertical and gripper channels at the grasp, whereas in $\pi_{0.5}$ the same heads carry only the planar and vertical channels. Selecting heads by the

Table I.1: Phase gate. Net change at 10 cm when the intervention is applied throughout the episode, only before the first gripper close, or only after it. The last column is the change in the distance from the first-close position to the object relative to base, so a negative value means a close nearer the object.

Policy	full	before close only	after close only	first-close offset
OpenVLA-OFT	$+7.5$ ($p = .031$)	$+3.0$	-0.7	none
π_0	$+14.9$ ($p = 3 \times 10^{-4}$)	$+12.7$ ($p = .002$)	$+7.5$ ($p = .050$)	-0.41 cm, same with the before-only arm
$\pi_{0.5}$	$+6.7$	$+10.4$ ($p = .020$)	$+2.2$	-0.62 cm, same with the before-only arm

Table I.2: Deviation component. Net change at 10 cm when the full deviation, only its displacement-subspace component, or only the orthogonal remainder is amplified. The subspace rank and factor are given per arm.

Policy	full	subspace only	remainder only
OpenVLA-OFT	$+7.5$ ($\times 2$)	$+0.7$ ($r = 2$, $\times 3$)	$+0.7$ ($r = 2$, $\times 2$)
π_0	$+14.9$ ($\times 1.5$)	$+14.9$ ($r = 4$, $\times 2$)	$+4.5$ ($r = 4$, $\times 1.5$)
$\pi_{0.5}$	$+6.7$ ($\times 1.5$)	$+9.0$ ($r = 4$, $\times 1.5$)	$+4.5$ ($r = 4$, $\times 1.5$)

grasp-moment screen instead of the approach screen does not help. The grasp-selected set harms 5 cm success in OpenVLA-OFT (-8.8, $p = .034$) and in π_0 (-23.7 at 5 cm and -12.7 at 10 cm), and in $\pi_{0.5}$ it is a subset of the approach set and performs the same. The displacement response during the approach is therefore the selection criterion that works in all three.

When amplification works. Carrying the displacement is necessary for the lever but not sufficient. The three policies with a gain are also the ones in which the screened heads write the displacement into the action rows with a balanced response across directions. OpenVLA carries the displacement in real heads but their amplification adds a state-specific offset instead of tracking. Replacing its four screened heads by their clean-scene means harms all four half-planes at 5 cm (-11.9 points on the validation states, $p = 5 \times 10^{-4}$), so the heads do carry the displacement. Amplifying them by 2 gives $+4.4$ overall, but the gain sits in the $+y$ half-plane alone ($+16.7$, $p = .020$) and the first gripper close shifts by $(-0.8, +1.1)$ cm regardless of δ, and the factor 1.5 harms the $-y$ half-plane by -13.3. GR00T tracks strongly in one direction and weakly in the others, and amplification enlarges that imbalance. Its base open-loop tracking is about four times larger toward $-x$ than in the other directions, and this profile matches the closed-loop success rate across the eight directions ($\rho = 0.92$). The imbalance lies inside the displacement subspace of the screened heads, where 0.71 to 0.81 of the deviation variance falls in U_h at $r = 4$, against 0.15 to 0.24 in OpenVLA-OFT and 0.27 to 0.28 in π_0, so the subspace variant has nothing to remove and no rank or factor passes the open-loop gate. The fallback setting in Table 1 changes success by -0.9 at 5 cm and $+0.4$ at 10 cm, the same as random heads.

J Conceptor steering under two accountings

We compared our intervention with conceptor steering (Miao et al., 2026), the only published training-free steering method aimed at task success under perturbation. Its code, checkpoints, and per-task hyperparameters are not public, so we reimplemented it from the paper. The comparison is therefore not a faithful reproduction of the original results, and we report it only to show how the method behaves under our paired protocol. Conceptors were fitted on success and failure activations at 15 cm, and the layer, regularizer, and mixing weight were selected on a separate failure pool, following the published procedure. Two further differences from the original setting should be kept in mind. The base policy here is the fully fine-tuned public checkpoint rather than an early-stopped one, and one global setting of layer, regularizer, and mixing weight is used instead of a per-task search. Table J.1 reports the conceptor under two accountings. Re-running only failed episodes, the accounting used in the original work, makes the conceptor recover a quarter of them. The paired net change over all episodes is not distinguishable from zero ($+0.9$), because the intervention turns a matching number of successes into failures.

Comparison under the paired protocol. Table J.2 runs the reimplementation under our paired protocol on $\pi_{0.5}$ at 15 cm. Its net change is not significant and matches that of random heads, and a direction-free shrinkage of the same layer by the same factor behaves the same way, so the fitted subspace is not the effective ingredient in this setting. Both steering arms gain in the $-x$ and $-y$ half-planes and lose in $+x$ and $+y$, whereas amplification gains in all four. Eight filters with the same eigenvalue spectrum but a random basis give paired net changes from -3.7 to $+7.4$ (mean $+1.5$, standard deviation 3.4). The fitted conceptor sits in the middle of that distribution, so the fitting procedure does not select a better filter than chance. One of the eight draws reaches $+7.4$, gains in all four half-planes, and exceeds the fitted conceptor in direct pairing ($+6.5$, $p = .024$), so a multiplicative filter can help in this setting, but nothing in the published procedure identifies that draw in advance. Head amplification lies two standard deviations above the random-filter mean, no draw reaches it, and it beats the conceptor and the scalar shrinkage in direct pairing on the same episodes. Our test asks a narrower question, whether the same intervention helps a fully trained policy whose failure has a single cause, the displaced object.

Table J.1: Conceptor steering on $\pi_{0.5}$ at 15 cm under failure-conditional and paired accountings. The failure-conditional column is the fraction of 82 failed episodes not used for fitting that succeed under the intervention, and for random bases it is the mean of the two draws evaluated under that accounting. Clean is the number of successful clean-scene episodes out of 40. The paired columns are from the grid protocol of the main text. Random basis rows keep the conceptor's eigenvalue spectrum and rotate its basis at random.

Intervention	Failures recovered (%)	Paired net (validation / test)	Half-planes ($+x$, $-x$, $+y$, $-y$)	Clean
conceptor	24.4	$+0.9$ / $+0.9$	-4.5, $+5.6$, -6.7, $+8.5$	39/40
random basis, 8 draws	26.2 (2 draws)	mean $+1.5$, range -3.7 to $+7.4$	1 of 8 draws positive in all four	39–40/40
scalar $0.7I$	20.7	-2.8 / $+1.8$	-6.0, $+4.4$, -10.1, $+8.5$	39/40
head amplification (ours)	—	$+1.9$ / $+14.7$	$+7.5$, $+4.4$, $+2.2$, $+17.0$	40/40

Table J.2: Conceptor steering versus head amplification on $\pi_{0.5}$ at 15 cm, paired over the same 216 episodes (validation and test states pooled). Conceptor applies $h \leftarrow h(0.7I + 0.3C)^\top$ to the output of one action-expert layer, with C fitted from success and failure activations as in Miao et al. (2026). Scalar applies $0.7I$ at the same layer. The p in parentheses after each success rate is the McNemar test of that arm against head amplification on the same episodes, and the base row's p is that of amplification against base.

Intervention	Success (%)	Net vs. base
base	42.1 ($p = .027$)	—
conceptor steering	43.1 ($p = .081$)	$+0.9$ ($p = .890$)
scalar shrinkage $0.7I$	41.7 ($p = .040$)	-0.5 ($p = 1.000$)
random basis, same spectrum (8 draws)	43.6 (5 of 8 below ours, $p < .05$)	$+1.5 \pm 3.4$ (range -3.7 to $+7.4$)
random 8 heads	42.6 ($p = .043$)	$+0.5$ ($p = 1.000$)
head amplification (ours)	50.5	$+8.3$ ($p = .027$)

K Compute resources and run times

All simulation experiments ran on one server with two Intel Xeon Gold 6254 CPUs (36 cores), 251 GB of RAM, and six NVIDIA A100 80 GB PCIe GPUs, under Ubuntu 22.04 with NVIDIA driver 580. Each policy ran in its own environment, OpenVLA and OpenVLA-OFT on PyTorch 2.2 with CUDA 12.1, π_0 and $\pi_{0.5}$ on PyTorch 2.7 with CUDA 12.6, and GR00T N1.7 on PyTorch 2.9 with CUDA 12.8. Every job used a single GPU, and independent jobs were sharded across three GPUs. Table K.1 gives the wall-clock time of the main experiment types, read from the pipeline logs. The times are indicative, since several jobs shared GPUs with other users.

Table K.1: Approximate wall-clock time per experiment on the server above. A closed-loop arm is one intervention setting run on all 314 episodes of one initial-state set (20 clean, 160 at 5 cm, 134 at 10 cm), split over three GPUs. The exact screen replaces every head once per moved scene, so its cost scales with the number of heads times the number of scenes.

Experiment	Policy	Time
Failure-mode evaluation, 1,000 episodes per magnitude	each policy	2 to 4 h
Exact screen, 1,024 heads, 10 states, 8 directions	OpenVLA-OFT	2.8 h
Exact screen, 288 heads, 10 states, 8 directions	π_0, $\pi_{0.5}$	about 1.5 h (1.0 s per call)
Exact screen, 1,408 heads, 10 states, 8 directions	GR00T N1.7	about 1.5 h (0.24 s per call)
Attribution approximation, all heads, 20 states	each policy	under 1 h
Closed-loop arm, 314 episodes	OpenVLA-OFT	about 55 min
Closed-loop arm, 314 episodes	π_0, $\pi_{0.5}$	about 45 min
Closed-loop arm, 314 episodes	GR00T N1.7	about 25 min
Closed-loop arm, 180 episodes	OpenVLA	about 2 h under GPU sharing
Row, layer, and head patching (Sections 4.2 and 4.3), one policy	each policy	1 to 3 h
LIBERO-Pro transfer, 12 conditions, 3 arms, 100 episodes each	π_0, $\pi_{0.5}$	about 10 h per policy, two policies in parallel