Reading and Steering a Vision Language Action Policy in English with Action-Grounded Supervision

Bill Cai **Minsoo Khang** **Quinn Cheong** **Xiaogang Wang** **Yash Shah**
Amazon
{billcaiy,mskhang,quinncsh,xiaogawa,syash}@amazon.com

Abstract

Vision language action (VLA) policies map images and instructions to robot commands, but their internal representations are difficult to inspect and modify. We build a language interface for a VLA policy whose weights remain fixed. A *verbalizer*, a separate language model, converts an internal activation into a structured description, and a *reconstructor* maps the original and edited descriptions back into activation space. Their difference provides a steering vector. We train both modules using descriptions generated from images, task instructions, and the policy's predicted action sequences. Including these actions in the labeling prompt increases mentions of predefined direction words from 6 of 400 held-out frames to all 400. For $\pi_{0.5}$, generated direction words agree with commanded motion on 77.2% of evaluated axes, compared with 54.6% after shuffling frame pairings. Action-conditioned labels also improve scene descriptions, although substantial hallucination remains. Applying direction-word edits throughout simulated episodes steers $\pi_{0.5}$ on three of four tasks and, in a separate control suite, along both tested axes, with larger effects than controls matched for perturbation magnitude. A second policy, GR00T, shows significant steering in one of eight task–reconstructor conditions. Stronger steering reduces task success. These results demonstrate language-based inspection and steering of a frozen VLA and measure its limits in descriptive accuracy, cross-policy transfer, and task preservation.

1 Introduction

Vision language action (VLA) policies map images and instructions to robot commands. Their internal activations, the numerical representations computed during inference, are difficult for a person to inspect or modify. Generated explanations alone do not solve this problem: they can be unfaithful to the computation that produced the output [38]. We study an interface that describes these activations in language and uses edits to those descriptions to steer a frozen policy, whose weights remain fixed.

Activation steering changes behavior by adding vectors to internal representations [37, 43]. Other methods decode activations into text, including interfaces that use the decoded text to steer language models [10, 30]. For VLAs, prior work predicts actions and scene properties from activations or intervenes on selected features [9, 23, 35]. We connect these operations through an editable language interface and test what supervision makes its descriptions track the policy's actions.

Our interface (Figure 1) uses a *verbalizer* to map an activation to a structured description and a *reconstructor* to map text back into activation space. Both are instances of a separate model, Qwen3-4B [40], rather than the policy's own backbone: natural-language autoencoders for language models have the interpreted model decode its own activations [13], but these policies emit actions rather than text, so a model from a different family reads their activations through one learned projection. Changing direction words, such as *forward*→*backward*, produces a difference between

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Workshop: VLM4RWD.

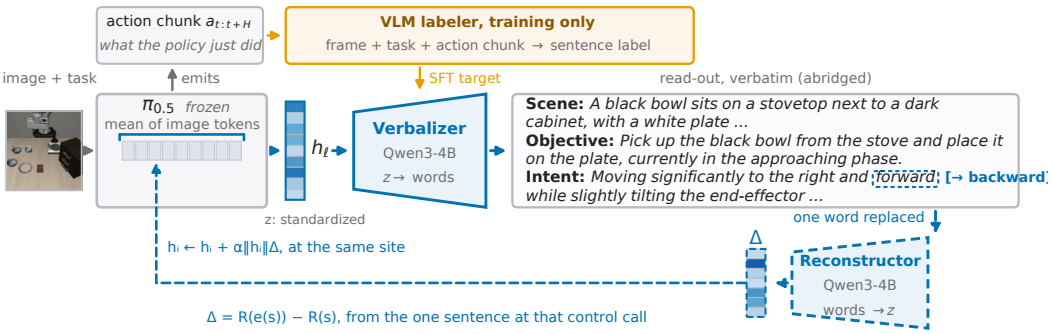


Figure 1: **A two-way language interface to a frozen VLA.** At the deployed $\pi_{0.5}$ read–write site (layer 17), the image-token activations are averaged into h_ℓ and standardized. The verbalizer turns that standardized activation alone into a structured sentence. Training targets are written from the frame, task, and action chunk emitted by the policy (top band); §3.1 withholds that action. Editing direction words in the sentence produced at that control call yields Δ, which is mapped back and added at the same site on every control call.

reconstructed activations that we inject into the policy. Both modules learn from *action-grounded labels*: descriptions generated using the image, task instruction, and the policy's predicted actions.

A useful interface must describe the policy accurately and change behavior when edited. We test these properties separately. We score motion descriptions against the policy's commands and scene descriptions against images, using label quality as a reference rather than ground truth. We then test steering in *closed loop*: the intervention is applied repeatedly as the policy observes and acts during an episode.

Contributions.

- **Action-grounded labels supply editable direction words.** Showing the labeler the emitted action raises direction mentions from 6 to all 400 held-out frames and improves prediction of commanded translation from the generated text (§3.1).
- **Separate motion and scene evaluations reveal different levels of accuracy.** Direction words agree with commanded motion on 77.2% of evaluated axes, compared with 54.6% after shuffling frame pairings. Grounding removes 1.24 hallucinated objects per frame, but scene descriptions remain inaccurate (§3.2).
- **Direction-word edits steer the policy in closed loop.** The edits steer three of four $\pi_{0.5}$ tasks and both tested axes, with larger effects than four magnitude-matched controls. Appending the direction phrase to the instruction does not steer the policy in these experiments. On GR00T, a vertical-word edit reaches $+0.028$ m in one of eight task–reconstructor conditions (§4–5).

Related work. Prior work interprets representations by predicting output tokens from intermediate layers [2], explaining individual neurons [3], decomposing activations into sparsely active features [7], prompting models to describe hidden representations [14], and recovering text from embeddings [27]. LatentQA [30] and SelfIE [10] steer language models through decoded activations. Natural-language autoencoders [13] come closest: they obtain a steering vector as the difference between reconstructions of an original and an edited explanation, the construction of Eq. 1. Our setting differs in four respects: the interface controls a VLA, its verbalizer is a separate model rather than the interpreted one, its labels describe the action the policy actually emitted, and its causal test edits direction words in closed-loop robot episodes. Our controls also account for two known confounds: steering vectors derived from prompt differences can owe much of their effect to spurious prompt features [36], and weights that predict an output from an activation need not change that output when injected into the model [18, 25]; conversely, steerable directions need not be decodable [28].

For robot policies, predictors trained on activations recover object and action state [23], reasoning sequences describe planned behavior before acting [41], and activation directions associated with

behavior change speed and motion [17]. Recent work exposes steerable sparse features [35], controls linear features [9], transplants task activations between episodes [15], steers activation components associated with task success [26], and predicts failures from internal features [16]. These methods operate through vectors or subspaces rather than an English sentence a person can inspect and edit. Language has also conditioned robot policies [24], corrected robots through their inputs [8, 11, 12, 34, 42], supported help requests [32], and explained failures after an episode [22]; our interface instead edits an internal activation of the frozen policy.

2 The interface and the protocol

Read and write. A *control call* is one policy inference that predicts an *action chunk*, a sequence of future robot commands. We average the activations of the image tokens, which represent the visual input, at one layer into a single 2048-dimensional vector h_ℓ per call. The verbalizer turns this vector into a structured English description. To steer, we edit its direction words, a reconstructor maps the original and edited sentences back to vectors, and we add their difference back at the same layer. Both modules are Qwen3-4B [40], and at deployment the verbalizer receives h_ℓ and nothing else. Steps (1)–(7) below follow the order of the work: rollouts, site selection, splits, labels, SFT, GRPO, editing.

Setup. We attach to two frozen policies with different action-generation architectures: $\pi_{0.5}$ [5, 6] and GR00T-N1.7 [4]. We build the full read–write interface on $\pi_{0.5}$ and use GR00T as a cross-family test. Closed-loop steering runs on four `libero_10` tasks of LIBERO [21]: basket placement (0), bowl placement and drawer closing (3), book placement (5), and moka-pot placement (8). Full instructions are in Appendix A; read-side evaluation uses held-out LIBERO splits. Every result in this paper is in simulation, obtained on a single H100 in about 600 GPU-hours. We measure steering along world axis 1, which points away from the base, and on the control suite also along world axis 2, the vertical. Commanded axes align with world axes. Their natural ranges of movement differ, so we report each axis separately.

(1) Rollouts. We roll out the frozen policy and record, at each control call, the frame, task string, emitted action chunk, and activations at every layer (Appendix A).

(2) Site selection by causal leverage. We fixed the sites before any reported intervention and chose them by causal leverage, the change in the emitted action when the activation at a site is replaced (*pinned*) with another frame's, among layers with similar *decodability*, the held-out R^2 of a linear model predicting commanded translation from the activation. Layers 15–17 of $\pi_{0.5}$ all decode commanded translation at $R^2 \approx 0.90$, but pinning changes the emitted action seven to nine times more at layer 17 than at layers 16 and 15 (Appendix C). We therefore use layer 17 for every deployed $\pi_{0.5}$ read and write result, and layer 16 appears only as a low-leverage ablation (Appendix C). GR00T uses layer 3, the highest-leverage site measured for that policy.

(3) Splits. For each policy, each of the two label sets of step (4) covers the same $10{,}000$ records, of which only the $4{,}304$-record **fit** split is used for training. The remaining records hold out unseen episodes of trained tasks and eight unseen tasks, a discovery split, and a reserved set unused here. Read-side numbers use the unseen-task split unless stated otherwise (Appendix A).

(4) Labels. A vision–language model (Gemma-4-26B-A4B, temperature 0) writes every label. Each label has three fields: SCENE lists what is on the table, OBJECTIVE names the task and the phase it has reached, and INTENT describes how the arm is commanded to move. The name INTENT refers to commanded motion, not an inferred goal or mental state. We edit only its direction words. Both corpora are built over the same records and conditioned on the frame and the task string; the *grounded* arm also shows the labeler the action chunk the frozen policy emitted at that frame and the *plain* arm does not. The preamble, INTENT instruction, and inserted action block differ. The labels themselves differ in most frames because the prompt context does, so §3.1 compares two complete label sets rather than one edited field.

(5) Supervised fine-tuning (SFT). Both modules are trained on activation–description pairs. Reconstruction uses principal component analysis (PCA) to retain the 32 leading directions of

activation variation, denoted PCA-32 (§4). Let z be the 2048-dimensional activation after coordinate-wise standardization and unit normalization, and let u be its projection onto the 32 leading principal components fitted on the fit split. The verbalizer receives z through a trainable projection W that produces one soft-prompt token, a learned input embedding placed before a fixed instruction q. For a labeled pair (z, y), we fine-tune the full model by $-\sum_t \log \pi_\theta(y_t \mid y_{<t}, Wz, q)$. The reconstructor R_ϕ uses the same backbone with a linear head and is trained on the reversed pairs by $\|\widehat{R_\phi(y)} - u\|_2^2$, where $\widehat{v} = v/\|v\|$. The inverse PCA projection and coordinate rescaling return reconstructed differences to activation space when writing.

(6) GRPO refinement. Matching a target sentence does not reward accurate reconstruction. We therefore refine the verbalizer with group relative policy optimization (GRPO) [33], a reinforcement-learning method that compares rewards among sentences sampled for the same activation, following Fraser-Taliente et al. [13]. Each sentence receives reward $-\log\|u - \widehat{R_\phi(s)}\|_2^2$. A length penalty and a Kullback–Leibler (KL) divergence penalty discourage long outputs and departures from the SFT model. The reconstructor is also trained on the sampled sentences, but receives no gradient through the reward (Appendix B).

(7) Editing, dose, and the steering metric. We never hand-design a direction. For the sentence s the verbalizer produced at a control call, its direction-word edit $e(s)$, and the reconstructor R, the steering direction is

$$\Delta = \delta/\|\delta\|, \qquad \delta = R(e(s)) - R(s), \tag{1}$$

computed from the single sentence produced at that call. Here R is R_ϕ followed by the inverse PCA projection and rescaling, so Δ is a direction in activation space. For $\pi_{0.5}$, at every control call we add to each image-token activation h_i at the site a vector along Δ with norm proportional to $\alpha\|h_i\|$, so α is a dimensionless steering strength, which we call the *dose*. GR00T instead receives the vector at all tokens in raw activation units; its dose is calibrated separately (Appendix F.1). Every control we report is *magnitude-matched*: it uses the same norm, dose, site, and control calls as the edit it replaces. Our primary steering metric is the paired contrast $d^+ - d^-$: the difference in net gripper displacement along the tested axis between opposite edits from the same starting state, measured in meters. Generating and reconstructing a description takes $3{,}667$ ms; injection alone adds 5–7% to a control call (Appendix B).

Statistics. These policies are stochastic, and evaluation outcomes of learned control policies vary across runs [1, 19]: re-running the same code from the same states changes about a fifth of the outcomes, and baseline task-5 success varies by more than 0.1 across otherwise identical sessions. We therefore pair comparisons on shared starting states and use two-sided Wilcoxon signed-rank tests. We set decision thresholds on disjoint runs; the significant $\pi_{0.5}$ forward/backward effects replicate across sessions and reconstructors, while the significant GR00T condition and the vertical-axis suite are single sessions. Holm–Bonferroni correction controls the probability of any false rejection within each declared test family. Appendix H reports minimum detectable effects for the earlier GR00T experiment and dose sweep.

3 Reading: accuracy of motion and scene descriptions

3.1 Action-grounded labels supply direction words

The read-out can only be faithful about motion if it names motion. Both verbalizers train on the same $4{,}304$ fit records, so the labels are the only difference between the two interfaces.

The plain-label interface almost never names a direction. Under the fixed narrow word list, the plain layer-17 interface names a direction in 6 of 400 held-out INTENT fields, against all 400 for the grounded interface (Figure 2a); only four of the six plain occurrences use the word for motion rather than location. Both are fluent, but the plain interface usually describes a goal relative to an object rather than the robot's motion (Figure 3; twelve more frames in Appendix D.3). A broader list raises the plain count to $45/400$ but leaves the grounded agreement result unchanged (Appendix D.1).

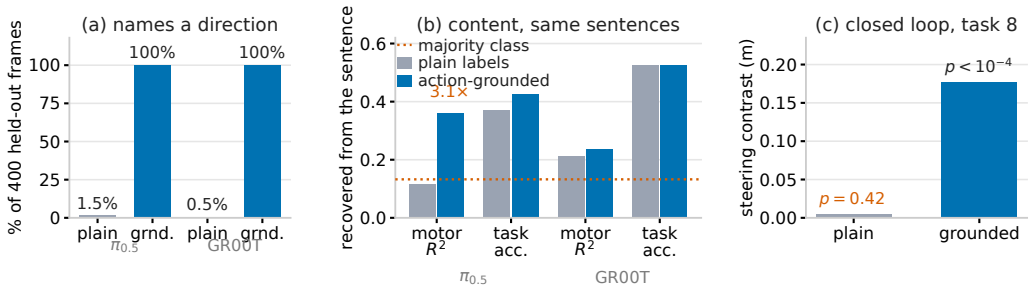
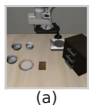


Figure 2: **The policy's action supplies editable direction words at layer** 17**.** *(a)* Under the fixed word list, the plain interface names a robot-relative direction on $6/400$ frames for $\pi_{0.5}$ ($2/400$ for GR00T), whereas the grounded interface always does. *(b)* For $\pi_{0.5}$, R^2 for predicting commanded translation from text, and task accuracy, under each label set. *(c)* On task 8, the matched layer-17 plain-label interface produces no closed-loop contrast.



(a)

task: "pick up the black bowl next to the plate and place it on the plate"

read-out: Moving significantly to the right and forward while slightly tilting the end-effector downward, maintaining a closed gripper.

says **right** **forward** **down** commanded r+7.0 f+6.2 u-1.9

plain-label interface: Move the gripper toward the chocolate pudding container to prepare for a grasp.



(b)

task: "pick up the milk and place it in the basket"

read-out: Moving the gripper left, backward, and significantly downward while rotating clockwise and closing the gripper.

says **left** ✗ **backward** **down** commanded r-0.5 f+0.2 u-2.0

plain-label interface: Move the gripper toward the ketchup bottle to prepare for a grasp.

Figure 3: **Layer-**17 **read-outs on two held-out frames.** Each row gives the frame, instruction, and verbatim INTENT field decoded from that activation alone. *says* lists the direction word on each axis, blue where it agrees with the commanded sign and orange with $\times$ where it does not; *commanded* is the net commanded translation on the right (r), forward (f) and up (u) axes in normalized units; the gray line is the matched layer-17 plain-label interface. Across the two examples, five of six scorable axes agree.

Grounding adds motor information and does not reduce task information. We score both properties on the same layer-17 sentences with episode-grouped folds. Motor information is held-out R^2 for predicting commanded translation from sentence words; task information is 8-way task accuracy (majority class 0.133). Grounding raises motor R^2 from 0.116 to 0.360 and task accuracy from 0.370 to 0.425 (Figure 2b).

3.2 Field-wise faithfulness

We evaluate motion descriptions against commanded actions and scene descriptions against images; the OBJECTIVE field's task-phase claims are not evaluated.

Motion: the sentence tracks the policy's command. We compare each direction word with the sign of the corresponding commanded translation. We score axes whose translation magnitude exceeds 0.15 normalized units; this threshold excludes near-zero commands. Pairing sentences with actions from different frames gives a shuffled baseline. We also score the labeler's descriptions as a reference. At the deployed $\pi_{0.5}$ layer-17 site, the verbalizer agrees on 0.772 of 972 scorable axes, above the 0.546 ± 0.016 shuffled baseline and below the 0.982 label agreement (Figure 4a). On the 857 axes scorable at both sites, layer 17 improves over layer 16 by 0.044 (episode-bootstrap 95% CI $[0.009, 0.079]$). GR00T layer 3 reaches 0.634 against a 0.563 ± 0.014 baseline.

Scene: grounding improves accuracy and reduces hallucinations. Claude Sonnet 4.5, a different model family from the Gemma-4 labeler, scores each scene description against its frame on a 0–3 rubric and against a different frame to set the baseline (Appendix D.2). Both read-outs score above

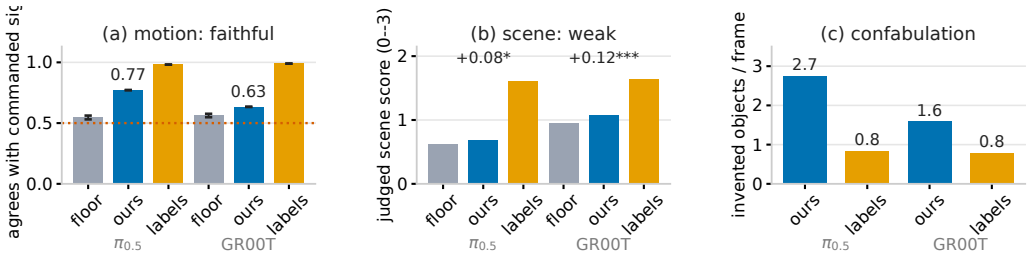


Figure 4: **Direction words track commanded motion, and grounding improves scene descriptions.** *(a)* Layer-17 direction words agree with the commanded sign on 0.772 of 972 scorable axes ($\pi_{0.5}$), against a 0.546 ± 0.016 shuffled-frame baseline over 200 permutations and a 0.982 label agreement. *(b)* On 200 frames (199 scored), scene descriptions score 0.688 against a 0.620 shuffled-frame baseline on a 0–3 rubric; their labels reach 1.61; stars mark paired Wilcoxon $p < 0.05$ (*) and $p < 10^{-3}$ (***). *(c)* Objects named but absent from the image, per frame (lower is better). The dotted line in (a) marks 0.5.

their baselines: the layer-17 $\pi_{0.5}$ read-out scores 0.688 against a 0.620 baseline (on the 199 frames scored in both conditions the baseline mean is 0.613 and the paired difference $+0.075$, Wilcoxon p=0.025) and GR00T scores 1.075 against 0.955 ($+0.121$, $p<10^{-3}$). The matched plain interface scores 0.45 and names 3.99 absent objects per frame, so grounding gains 0.236 scene-score points ($p<10^{-6}$) and removes 1.24 hallucinated objects per frame in paired comparison (Figure 4b,c shows the grounded read-out; the plain-interface values are not plotted).

Scene descriptions remain inaccurate. The $\pi_{0.5}$ interface names 2.74 absent LIBERO objects per frame against 0.84 for its labels, and 4% of its read-outs score at least 2 against 59% of its labels. Grounding improves both fields, but the motion agreement result does not establish the accuracy of the rest of the description. The 0–3 rubric is coarse: its middle grades ("most visible elements") are not defined by a count, so we treat the score as an ordinal summary and rely on the paired baseline and the absent-object count, an object-level false-positive measure, for the claims above. It does not measure recall, the fraction of visible objects a description names; object-level precision and recall against per-frame annotations would be a sharper test (Appendix D.2).

4 Steering: direction-word edits change robot motion

We edit direction words, inject the reconstructed difference, and measure where the gripper moves.

Scoring reconstruction against a mean predictor. Every steering direction is built from the reconstructor, so every result that follows depends on its fidelity. Cosine similarity is uninformative here: two unrelated read-site activations already have cosine 0.985, so a round-trip cosine of 0.991–0.998 is close to chance. We therefore report the fraction of variance explained (FVE) in the standardized PCA-32 coordinates of §2, where 0 matches predicting the mean activation and negative is worse. The reconstructor scores below zero against the full standardized target and above zero against the 32-component target (Table 8, Appendix E). End-to-end evaluation reconstructs the verbalizer's output, rather than a labeler's description. At the deployed site it reaches FVE $+0.119$, above the $+0.010$ of the earlier layer-16 site, and GRPO raises it to $+0.190$. An FVE of this size can still support steering: the write path uses a difference of reconstructions (Eq. 1), so any error shared by the two outputs cancels.

One edited word changes the emitted action. At a single control call, substituting each of the six direction words for its opposite moves the emitted action along the axis that word names, four of the six with 95% paired-bootstrap intervals excluding zero; a magnitude-matched random direction moves the named axis three to twenty times less (Appendix F.2).

Closed-loop steering on four tasks. We now add the edit at every control call across complete episodes. On each of the four tasks we take 48 starting states; from each we run one episode with the *forward* edit and one with the *backward* edit, and Table 1 reports the paired difference in net forward/backward (fore/aft) displacement. The mean displacement contrast follows the

Table 1: **Direction-word edits steer $\pi_{0.5}$ on three of four tasks, under either reconstructor.** Entries are paired forward-minus-backward displacement contrasts in meters over 48 shared starting states, with two-sided Wilcoxon signed-rank p; **bold** marks contrasts significant at the 0.05 level after Holm–Bonferroni correction within the 16-test confirmatory family described below; the dose is α=30. The eight GR00T conditions of the same family (vertical words, α=100) are in Appendix G; one reaches $+0.028$ m (§5).

	SFT reconstructor		after GRPO	
task	Δ (m)	p	Δ (m)	p
0	$+0.019$	0.75	$+0.022$	0.76
3	$\mathbf{+0.313}$	$\mathbf{7\times10^{-15}}$	$\mathbf{+0.341}$	$\mathbf{7\times10^{-15}}$
5	$\mathbf{+0.253}$	$\mathbf{1\times10^{-8}}$	$\mathbf{+0.262}$	$\mathbf{3\times10^{-8}}$
8	$\mathbf{+0.304}$	$\mathbf{1\times10^{-11}}$	$\mathbf{+0.277}$	$\mathbf{2\times10^{-9}}$

Table 2: **The direction-word edit produces the largest signed contrast on both tested axes.** The task-5 control suite compares fore/aft (forward − backward) and vertical (up − down) edits with four magnitude-matched alternatives. Entries are means over shared non-truncated starts (n=17 for the fore/aft edit, 18 elsewhere) with uncorrected two-sided Wilcoxon signed-rank p. Rows marked $\pm$ are positive-minus-negative contrasts; the other three are the control's displacement from the unedited run, so an equal per-arm effect appears up to twice as large in a $\pm$ row. **Bold** marks our edit, not significance.

	fore/aft (axis 1)		vertical (axis 2)	
direction written into the site	Δ (m)	p	Δ (m)	p
random, full activation space	$+0.016$	0.7987	$+0.011$	0.0432
random, in the reconstructor's output subspace	$+0.013$	0.7660	-0.005	0.6705
edit an irrelevant object name	-0.023	0.0182	-0.020	0.0987
decoder weight vector, $\pm$	-0.010	0.3692	$+0.012$	0.0385
edit the direction word, $\pm$ (ours)	$\mathbf{+0.241}$	$\mathbf{0.0017}$	$\mathbf{+0.140}$	$\mathbf{<10^{-4}}$

edited direction on three of the four tasks, under either the supervised reconstructor or its GRPO-refined counterpart; the responsive tasks give contrasts of $+0.253$ to $+0.313$ m under the supervised reconstructor, and the two reconstructors agree to within 0.03 m, so steering does not require the reinforcement-learning stage and every $\pi_{0.5}$ steering result below uses the supervised reconstructor.

Multiplicity and the unresponsive task. These contrasts belong to a pre-specified confirmatory family of 16 tests, the four tasks under {SFT, GRPO} × {$\pi_{0.5}$, GR00T} (fore/aft words for $\pi_{0.5}$, vertical words for GR00T), of which 7 remain significant after Holm–Bonferroni correction at the 0.05 level: tasks 3, 5 and 8 under both $\pi_{0.5}$ reconstructors, and one GR00T condition (§5; all families in Appendix H). Task 0 does not respond under either reconstructor, although its edit was applied on 58 of about 60 control calls; it also has the longest untouched path (1.62 m against 1.01–1.47), which may hide a small change of direction. For tasks 3, 5 and 8 the contrast persists after division by total path length, and steering shortens the path on all four tasks, so the effect is not explained by increased travel.

How the control suite is run. This suite tests whether a vector of the same magnitude at the same site produces a comparable directional effect. For each paired rollout, we force every word naming the axis to its positive pole (*forward* or *up*) or negative pole, reconstruct the two sentences, and write their difference into layer 17 at α=30 on every control call. Each alternative uses the same norm, dose, site, and intervention schedule (Table 2; Figure 10, Appendix F.3).

In this session the direction-word edit produces $+0.241$ m fore/aft (n=17, p=0.0017) and, with the vertical words edited instead, $+0.140$ m vertical (n=18, $p<10^{-4}$, every start in the commanded direction; Appendix F.4). Every matched control has absolute contrast at most 0.023 m, so the edit is at least six times larger on both axes, and at least three times larger once the $\pm$ rows are put on a single-arm scale. Three controls reach uncorrected $p<0.05$, the object edit fore/aft at -0.023 m and the full-space random and decoder vectors vertically at $+0.011$ and $+0.012$ m, each an order of

magnitude below the edit. The effect therefore depends on the content of the edit, not only on its magnitude or its subspace.

The raw decoder weight has little causal effect. A ridge regression, a linear predictor with a penalty on large weights, predicts commanded displacement from this site's activation at held-out $R^2 = 0.90$ fore/aft and 0.93 vertical. We inject its weight vector as a control (Appendix C). Despite this high decodability, magnitude-matched injections of it produce only -0.010 m fore/aft and $+0.012$ m vertical, compared with $+0.241$ and $+0.140$ m for the text edit. A discriminative decoder can put weight on coordinates that only cancel correlated noise, which improves prediction but has no causal effect [18, 25]. Decodability therefore does not imply controllability. Mapping the same weights through the activation covariance gives the corresponding encoding direction [18], which does steer ($+0.182$ m fore/aft on task 8, $p<10^{-4}$, against $+0.304$ m for the text edit on the same 28 states). A directly computed steering direction therefore also exists. The language interface obtains one through an editable description (Figure 9, Appendix F.3).

Appending the tested direction phrase does not steer the policy. On shared starting states, we either append ", while moving the gripper forward" (or "backward") to the instruction or write the same words through the reconstructor. The instruction contrasts are $+0.002$ m on task 5 (n=24, p=0.73) and -0.018 m on task 8 (n=10, a partial session, p=0.63); the activation edits give $+0.214$ and $+0.466$ m on those states.

5 Ablations and scope

The plain-label interface provides no direction word in the steering trial. In a matched session on task 8 we change only the SFT labels while holding the layer, data, model, edit and dose fixed. Grounded labels yield $+0.177$ m (Wilcoxon signed-rank, $p<10^{-4}$, 34 states); plain labels yield $+0.005$ m (p=0.42, 24 states; Figure 2c). The plain edit is applied on none of about 60 calls per episode because no direction word is available. Supervision therefore supplies the editable vocabulary; this does not test an injected plain-arm direction.

Steering grows with dose and is strongest on the named axis. The displacement contrast rises monotonically with dose (Figure 11a, Appendix F.5) and appears within the first ten control calls at each of the three doses re-run with per-call logging ($p\leq10^{-4}$; Figure 11b). At α=20 the named axis moves $+4.32$ standard deviations, the vertical axis $+2.72$, and the lateral axis -0.12 (no dose moves the lateral axis significantly; smallest p=0.55). The vertical displacement is also substantial; a matched random direction produces no significant effect on any axis. Task success limits the usable dose: at α=5 the aligned edit preserves baseline success, and at the α=30 of Tables 1–2 success falls to 0.444 aligned and 0.056 opposed, from 0.944 untouched (Figure 11d).

Little steering survives at the task-preserving dose. Only α=5 keeps aligned-edit success at the untouched rate, and its whole-episode contrast is small: -0.002 m (p=0.39) in the dose sweep and $+0.026$ m (uncorrected p=0.024) in the re-run session, the latter about a tenth of the $+0.215$ m at α=30. The effect is detectable over the first ten control calls ($+0.017$ m, $p<10^{-4}$). At α=10 the contrast is $+0.075$ m (p=0.12) with aligned success 0.667. No tested dose combines a large contrast with preserved success (Table 10, Appendix F.5).

GR00T shows significant steering in one of eight conditions. With the site, dose and edited axis changed, the grounded interface names a direction on every GR00T frame at 0.634 agreement against a 0.563 shuffled baseline, and FVE rises from $+0.054$ to $+0.142$ after GRPO. Across the same four tasks and two reconstructors, editing the vertical words at α=100, one of eight GR00T conditions survives Holm correction: task 8 under the GRPO reconstructor, at $+0.028$ m with 33 of 48 states in the commanded direction. The other seven are within 0.02 m of zero, one of them from a session cut short at 9 pairs (Appendix G). The intervention changes GR00T's actions: replacing activations at the site has four times $\pi_{0.5}$'s measured effect, and the edit moves the emitted action by 10% of its scale. Yet consistent directional steering remains weak. Grounding also adds less motor information: GR00T's motor R^2 rises by 0.024, compared with 0.244 for $\pi_{0.5}$. This motivates testing improved action-conditioned labels.

6 Limitations and conclusion

Limitations. The experiments use one simulated benchmark, and the large closed-loop steering effects come from one policy. Scene descriptions remain inaccurate, are scored on a coarse ordinal rubric, and the interface never declines to answer. At the main $\pi_{0.5}$ steering dose, task-5 success falls from 0.944 to 0.444 even when the edit is aligned with the task, and at the one dose that preserves success the contrast is about a tenth as large (§5). Two explanations remain open. The reconstructed direction may leave the activation distribution the policy was trained on: the reconstructor scores below the mean predictor against the full standardized target (Table 8), so Δ can carry components unrelated to the edited word. Alternatively, a single 2048-dimensional image-token mean at one layer may be too narrow a site; reading from and writing to several layers or individual tokens could separate motion from task content, at the cost of a larger interface, and is untested. We demonstrate directional steering, not recovery from errors or improved task completion. The labeling ablation shows that grounded labels supply an editable direction word; because the plain interface has no such word to edit, it does not isolate whether grounded semantics also yield a better steering direction than another edit of the plain sentence would. The instruction baseline tests one suffix. A ridge regression on word counts reconstructs held-out activations better than our 4B reconstructor, which overfits (Appendix E); a higher-fidelity reconstructor has not been tested in closed loop.

Conclusion. We introduced a language interface for reading and steering a frozen VLA. Action-grounded labels supply editable direction words. Evaluation against commanded actions and images shows that motion descriptions track the policy's commands while scene descriptions remain inaccurate. Direction-word edits steer $\pi_{0.5}$ on three of four tasks and outperform four matched controls; GR00T shows significant steering in one of eight conditions. Language thus provides an interface for inspecting and changing robot motion without updating policy weights. Future work should turn this control into successful correction by preserving task performance while steering.

Acknowledgments and Disclosure of Funding

All authors are employees of Amazon. The authors received no third-party funding and declare no other competing interests.

References

[1] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Proc. NeurIPS*, 2021.

[2] Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the Tuned Lens. *arXiv:2303.08112*, 2023.

[3] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. OpenAI, 2023. URL `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`.

[4] 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:2503.14734*, 2025.

[5] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proc. CoRL*, volume 305 of *PMLR*, pages 17–40, 2025.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert 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 *Proc. RSS*, 2025.

[7] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL `https://transformer-circuits.pub/2023/monosemantic-features`.

[8] Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. LATTE: LAnguage Trajectory TransformEr. In *Proc. IEEE ICRA*, pages 7287–7294, 2023.

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

[10] Haozhe Chen, Carl Vondrick, and Chengzhi Mao. SelfIE: Self-interpretation of large language model embeddings. In *Proc. ICML*, 2024.

[11] Yuchen Cui, Siddharth Karamcheti, Raj Palleti, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. "No, to the right" – online language corrections for robotic manipulation via shared autonomy. In *Proc. ACM/IEEE HRI*, 2023.

[12] Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. In *Proc. IEEE ICRA*, 2025.

[13] Kit Fraser-Taliente, Subhash Kantamneni, Euan Ong, Dan Mossing, Christina Lu, Paul C. Bogdan, Emmanuel Ameisen, James Chen, Dzmitry Kishylau, Adam Pearce, Julius Tarng, Alex Wu, Jeff Wu, Yang Zhang, Daniel M. Ziegler, Evan Hubinger, Joshua Batson, Jack Lindsey, Samuel Zimmerman, and Samuel Marks. Natural language autoencoders produce unsupervised explanations of LLM activations. *Transformer Circuits Thread*, 2026. URL `https://transformer-circuits.pub/2026/nla/index.html`.

[14] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *Proc. ICML*, 2024.

[15] Bryce Grant, Xijia Zhao, and Peng Wang. Not all features are created equal: A mechanistic study of vision-language-action models. In *Proc. NeurIPS*, 2026. To appear; arXiv:2603.19233.

[16] Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask failure detection for vision-language-action models. In *Proc. NeurIPS*, pages 44801–44836, 2025.

[17] Bear Häon, Kaylene Caswell Stocking, Ian Chuang, and Claire Tomlin. Mechanistic interpretability for steering vision-language-action models. In *Proc. CoRL*, volume 305 of *PMLR*, pages 2743–2762, 2025.

[18] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014.

[19] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proc. AAAI*, 2018.

[20] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[21] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Proc. NeurIPS Datasets and Benchmarks Track*, 2023.

[22] Zeyi Liu, Arpit Bahety, and Shuran Song. REFLECT: Summarizing robot experiences for failure explanation and correction. In *Proc. CoRL*, 2023.

[23] Hong Lu, Hengxu Li, Prithviraj Singh Shahani, Stephanie Herbers, and Matthias Scheutz. Probing a vision-language-action model for symbolic states and integration into a cognitive architecture. In *Proc. IEEE ICAD*, pages 1–8, 2025.

[24] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. In *Proc. RSS*, 2021.

[25] Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Proc. COLM*, 2024.

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

[27] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. In *Proc. EMNLP*, 2023.

[28] Mohammed Suhail B. Nadaf. Steerable but not decodable: Function vectors operate beyond the logit lens. *arXiv:2604.02608*, 2026.

[29] Ellis B. Page. Ordered hypotheses for multiple treatments: A significance test for linear ranks. *Journal of the American Statistical Association*, 58(301):216–230, 1963.

[30] Alexander Pan, Lijie Chen, and Jacob Steinhardt. LatentQA: Teaching LLMs to decode activations into natural language. In *Proc. ICLR*, 2026.

[31] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proc. CoRL*, 2023.

[33] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

[34] Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. Yell at your robot: Improving on-the-fly from language corrections. In *Proc. RSS*, 2024.

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

[36] Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. In *Proc. NeurIPS*, pages 139179–139212, 2024.

[37] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv:2308.10248*, 2023.

[38] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Proc. NeurIPS*, 2023.

[39] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[40] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv:2505.09388*, 2025.

[41] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Proc. CoRL*, 2024.

[42] Lihan Zha, Yuchen Cui, Li-Heng Lin, Minae Kwon, Montserrat Gonzalez Arenas, Andy Zeng, Fei Xia, and Dorsa Sadigh. Distilling and retrieving generalizable knowledge for robot manipulation via language corrections. In *Proc. IEEE ICRA*, 2024.

[43] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, et al. Representation engineering: A top-down approach to AI transparency. *arXiv:2310.01405*, 2023.

Table 3: The 10,000 labeled records per policy per arm.

split	records	role
fit	**4,304**	trains the verbalizer and the reconstructor
discovery	1,713	held out; direction discovery and sweeps
id_confirm	1,797	held out; episodes unseen, tasks seen
held_task	1,489	held out; **8 tasks never trained on**
lockval	697	held out; reserved, unused here
total	10,000	

Table 4: The four `libero_10` tasks used for closed-loop steering.

index	instruction
0	put both the alphabet soup and the tomato sauce in the basket
3	put the black bowl in the bottom drawer of the cabinet and close it
5	pick up the book and place it in the back compartment of the caddy
8	put both moka pots on the stove

A Data and labels

This appendix gives the data behind steps (1)–(3) of §2: the splits, the tasks, what a rollout records, and the two labeling prompts. Training and hyperparameters are in Appendix B, the attachment-site sweep in Appendix C, reading-side scoring in Appendix D, reconstruction scoring in Appendix E, and the path from an edit to the policy in Appendix F.

(1) Splits. The paper's first claim is that action-grounded labels determine whether the interface produces editable direction words, so we state exactly what was held fixed. For each policy we build two corpora with the same vision–language model (`gemma-4-26b-a4b`) at temperature 0 over the same rollout records. The two arms are verified to cover an identical set of `doc_id`s, so no frame appears in one arm and not the other, and both arms share the split assignment in Table 3. Only the **fit** split trains anything. The held-out data is held out at three levels: `id_confirm` withholds episodes of trained tasks, `held_task` withholds eight tasks entirely, and `lockval` is reserved and used by no number in this paper.

The steering tasks. Closed-loop steering runs on four tasks of the LIBERO `libero_10` suite, referred to by their suite index throughout (Table 4).

(2) What a rollout records. Each frozen policy is served over HTTP, and at every control call we record the third-person frame, the task string, the action chunk the policy emitted, and hidden states at every layer under three token pools: for $\pi_{0.5}$ the mean over its 512 non-padding image tokens, a middle image token, and the mean over language tokens, at 19 prefix layers; for GR00T the image-token mean, the text-token mean, and the last token, at 16 backbone layers. The interface uses the image-token mean, one 2048-dimensional vector per call. The same records supply the action chunks the grounded labeler sees in step 3, the layer sweep that chooses the site (Appendix C), and the paired starting states every closed-loop contrast is computed on.

(3) The two labeling prompts. The plain arm's prompt shows the frame and the task string and asks for three fields:

```
You are an expert robot manipulation analyst.  Given an image of a
robot scene and a task instruction, produce a structured 3-section
description that captures the semantic content the policy must
encode to act correctly.
Task instruction:  "{task}"
Scene:  <one-line description of objects and layout in the image>
Objective:  <one-line description of what the policy needs to
accomplish, and the current phase>
Intent:  <one-line description of the immediate next-step
manipulation goal, in declarative motion language>
```

The grounded arm's prompt differs in exactly three places. (i) The preamble also announces the action: "Given an image of a robot scene, a task instruction, *and the motion the policy is actually commanding right now…*". (ii) A block carrying the policy's own action chunk is inserted after the task instruction, rendered as labeled text rather than raw arrays:

```
Commanded motion over the next {horizon} control steps, measured
from the policy's own action output (normalized units):
translation:  right axis +7.185  [right(+)/left(-)]
translation:  forward axis +1.620  [forward(+)/backward(-)]
translation:  up axis -7.506  [up(+)/down(-)]
rotation roll/pitch/yaw:  -0.182 / +0.123 / -0.013
gripper:  starts -1.00, ends -1.00  [-1=open ..  +1=closed] -> held
```

(iii) The INTENT instruction is redirected from the goal to the commanded motion: "describe the ACTUAL commanded movement given above — its direction, its relative magnitude, and what the gripper is doing — not what you think the task ideally requires. If the commanded motion disagrees with what the task would suggest, describe the commanded motion," plus a final line, "Do not restate the numbers; render them as motion language."

Two consequences follow. The SCENE and OBJECTIVE instructions are identical across arms, so any difference in those fields comes from the changed prompt context rather than from a changed instruction. The fields do differ in most frames: the two arms produce an identical SCENE field on 353 and an identical OBJECTIVE field on 101 of the 10,000 $\pi_{0.5}$ records. The grounding comparison of §3.1 is therefore between two complete label sets, which is why task accuracy can move as well as motor content. The grounded labeler is also told to describe the commanded motion even when it disagrees with the task, so the labels describe the commanded motion rather than restating the instruction; whether the read-out does is what §3.2 measures.

B Training and hyperparameters

The training pair and the objectives. At training time the pair is (activation, label). The activation is the mean over image tokens at the read layer, a single 2048-dimensional vector, mapped by a trainable linear projection into one soft-prompt token followed by the fixed instruction "`Describe the robot's scene, objective, and intent:`", identical for every example and at inference. The loss is next-token cross-entropy on the label tokens only. At inference the model receives the soft-prompt token and this instruction and nothing else. The reconstructor consumes the same pairs in reverse and regresses onto the standardized PCA-32 target of §2, the 32-dimensional principal-component space that Table 8 selects.

Hyperparameters. Both modules are Qwen3-4B. The verbalizer is fully fine-tuned for 8 epochs on the 4,304 fit pairs with one soft-prompt token ($2048 \to 2560$ projection) and next-token loss on the label tokens; the matched plain-arm verbalizer uses the same architecture, layer, fit frames and epoch budget. The reconstructor is the same backbone with a linear head on the last token, trained by normalized regression onto the standardized PCA-32 target. GRPO uses G=6 samples per activation, KL coefficient β=0.05, and learning rates 10^{-6} (verbalizer) and 10^{-5} (reconstructor); the reconstructor is updated on sampled sentences in the same step but held fixed inside the reward, so no gradient flows into it through the reward. Each steering direction is computed from the one sentence the verbalizer produces at that control call (Eq. 1); no averaging over verbalizer samples is performed.

Differences from the published method. We follow Fraser-Taliente et al. [13] in the metric, the objective and the injection convention, and differ in four ways that affect comparison with their numbers. *Verbalizer*: their autoencoder is the interpreted language model reading its own activations; ours is a separate Qwen3-4B reading a 2048-dimensional activation of a policy with which it shares no weights, because neither served policy decodes text. This cross-model verbalizer reaches 0.772 direction agreement (Table 7) and supplies every steering direction in §4, so a verbalizer need not be the interpreted model for either reading or writing. *Metric*: their FVE is $1 - \mathbb{E}\|h - \mathrm{AR}(z)\|^2 / \mathbb{E}\|h - \bar{h}\|^2$ on unit-normalized activations; ours is the same expression in the coordinates the reconstructor predicts in, which we state explicitly because in this activation geometry the choice moves the number by about 0.3 (Table 8). *Budget*: their open-model runs use 3,000 GRPO steps at batch 128 with

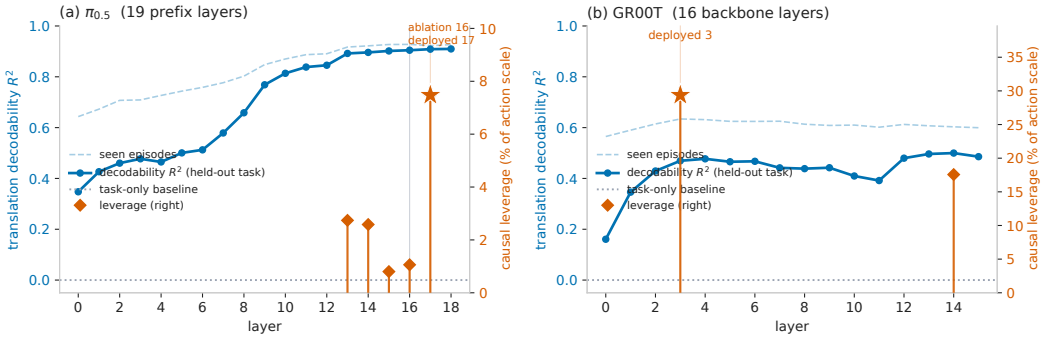


Figure 5: **Decodability plateaus over a wide band of layers; leverage does not.** For every layer of each policy a RIDGECV linear probe decodes the commanded translation from the image-token activation (blue: held-out-task R^2; dashed: the easier held-out-episode split), against a task-only baseline near zero (dotted). Causal leverage (orange markers, right axis, with per-panel scales) is measured only at the candidate deployment layers. In (a) the marker labeled *ablation 16* is the layer-16 site of Table 6; the deployed read–write site is the starred layer 17. Exact values for layers 15–17 and GR00T are in Table 5.

G=8 on two 8×H100 nodes, reaching 0.75 FVE from a 0.38 warm start; ours is 600 steps at batch 2 with G=6 on one GPU, about 0.2% of their samples, so we treat our GRPO magnitude as indicative and compare our FVE with their warm-start band (0.3–0.4) rather than their final value. *Injection*: their edit direction is added as $h \leftarrow h + \alpha\|h\|\Delta/\|\Delta\|$, which is the $\pi_{0.5}$ convention here up to a constant; GR00T's injection adds in raw units instead, so the two dose scales are not comparable and we calibrate GR00T's by its measured effect on the emitted action (Appendix F.1). All training and all rollouts ran on a single H100, about 600 GPU-hours in total, including the sessions that were interrupted mid-run and are reported as truncated. Latency, measured on the same machine over 15 repetitions: 901 ms per untouched policy call, 938 ms when the call also exports the activation and 950–968 ms when it carries an injected vector; 3,667 ms for a full sentence round trip, of which the reconstructor is 43 ms.

C Attachment-site selection

§2 has to choose where to attach the interface, and Figure 5 shows why decodability does not determine that choice. For every layer of each policy we fit a linear probe, a RIDGECV regression (scikit-learn [31]) standardized on the fit split and evaluated held out, from the image-token activation to the policy's own commanded translation, and report the fraction of variance it explains on the eight held-out tasks. Decodability rises with depth and then plateaus: for $\pi_{0.5}$ it reaches $R^2 \approx 0.90$ from layer 13 onward and stays there, and for GR00T it stays near 0.4–0.5 across its middle and late layers, above the task-only baseline of $R^2 \approx 0$ that predicts the same target from the task label alone. Because a wide band of layers decodes almost equally well, R^2 on its own does not single out a site.

Leverage, the causal effect of a site on the emitted action, differs across these layers and does not track decodability. Leverage is expensive to measure, since it requires live control-call rollouts rather than a held-out probe, so we measure it over the candidate band where decodability has plateaued (layers 13–17 for $\pi_{0.5}$; the deployed site and a mid-backbone layer for GR00T) and overlay it in Figure 5 and Table 5. Across $\pi_{0.5}$'s layers 15–17 decodability moves by less than 0.01 in R^2 while leverage varies ninefold and peaks at the deployed read–write site, layer 17; for GR00T the deployed layer 3 has the highest leverage we measured and steers in one of eight conditions (§5, Appendix G), while layer 14 decodes better (0.500 against 0.469) but has lower leverage. We therefore select the site by leverage among the layers whose decodability has plateaued, not by decodability itself. We measure leverage by pinning another held-out frame's activation at the site for a single control call and recording the mean absolute change in the emitted action chunk, in units of the standard deviation of the translation channels, which Table 5 reports as a percentage (n=40 frames). Pinning the frame's own activation should change nothing, so the residual change it produces is the measurement floor: 0.0003–0.0007 for $\pi_{0.5}$ and 0.001–0.003 for GR00T. The donor/own ratio of Table 5 divides leverage by that floor.

Table 5: Held-out-task translation decodability (R^2) and leverage (mean action change from pinning another frame's activation, the donor, as % of action scale) at the candidate layers; donor/own divides leverage by the floor from pinning the frame's own activation. Deployed sites in bold.

policy	layer	R^2 ↑	leverage↑	donor/own ratio
$\pi_{0.5}$	15	0.902	0.8%	12
	16	0.905	1.1%	18
	17	0.909	**7.5**%	116
GR00T	**3**	0.469	**29.4**%	111
	14	0.500	17.6%	131

Table 6: **Read-site ablation: layer 17 against layer 16, the site used in earlier runs.** Both verbalizers use the same action-grounded labels, architecture, eight-epoch budget, and 400 held-task frames. Agreement is computed on each site's scorable axes; the common-axis row uses the same 857 axes. FVE uses all 1,489 held-task frames under the protocol of Table 8; the layer-17 GRPO entry is a six-seed mean, while layer 16 is a single run. **Bold** marks the deployed site.

quantity	layer 16	**layer 17 (deployed)**
translation decodability, R^2	0.905	**0.909**
causal leverage (% of action scale)	1.1%	**7.5**%
command-sign agreement	0.743 (943)	**0.772** (972)
shuffled-frame baseline	0.541 ± 0.015	0.546 ± 0.016
agreement on 857 common axes	0.740	**0.784**
SFT end-to-end FVE, full standardized	−0.217	**−0.207**
SFT end-to-end FVE, PCA-32	+0.010	**+0.119**
after-GRPO end-to-end FVE, PCA-32	+0.185	**+0.190**

On the common axes, layer 17 improves agreement by +0.044 (episode-block bootstrap 95% CI [+0.009, +0.079]; exact paired p=0.0128). It also raises SFT end-to-end FVE in the PCA-32 target from +0.010 to +0.119, while GRPO reaches similar FVE at both sites. The attachment choice therefore does not trade readability for control: layer 17 is better on the read metric and about seven times stronger under intervention (Table 6). Per axis at layer 17 the same ridge decodes fore/aft translation at held-out R^2=0.90 and vertical translation at 0.93, the two weight vectors used as controls in Table 2.

D Reading-side evaluation

Reading is scored three ways, none of which treats the labels as ground truth: direction-word agreement with the sign of the policy's own commanded action, with a shuffled-frame baseline and the labels' own agreement as a reference (Appendix D.1); recovery of the commanded translation and of task identity from the sentence alone by one fixed estimator family, grouped by episode so neighboring frames cannot leak across the split (§3.1); and a vision-capable judge scoring the SCENE field against the image on a 0–3 rubric, with each sentence also scored against a different frame to set the baseline (Appendix D.2).

D.1 Sensitivity to the direction-word list

The read-side measures count robot-relative direction words in the INTENT field. The list is small and fixed; Table 7 also reports a wider one. *Narrow* is the list used throughout the paper: `right`, `left`, `forward`, `backward`, and up/upward/raising/`lifting` against `down`/`downward`/`lowering`. *Broad* additionally counts `descend(ing)`, `back`, `rising`, `ascending`, `dropping` and plural variants. An axis is scored only when the commanded translation exceeds a deadband of 0.15 normalized units. Both columns are computed from the same generations and action chunks.

D.2 The scene judge

The judge is Claude Sonnet 4.5 (2025-09-29 release), a different model family from the Gemma-4 labeler, so the reference labels are not scored by the model that wrote them. It is shown the frame, the

Table 7: Agreement rates, the basis of the read-side claim, do not change with the word list; the plain arm's direction-word rate does, so the 1.5% figure is list-dependent. Floors are the mean over 200 random pairings, $\pm$ one standard deviation.

quantity (400 held-out frames)	narrow (reported)	broad
$\pi_{0.5}$ grounded: frames naming a direction	$400/400$	$400/400$
$\pi_{0.5}$ plain: frames naming a direction	$6/400$ (1.5%)	$45/400$ (11.3%)
GR00T grounded: frames naming a direction	$400/400$	$400/400$
GR00T plain: frames naming a direction	$2/400$ (0.5%)	$12/400$ (3.0%)
$\pi_{0.5}$ agreement with the commanded sign	$\mathbf{0.772}$	0.772
$\pi_{0.5}$ shuffled baseline	0.546 ± 0.016	0.551 ± 0.015
$\pi_{0.5}$ label agreement	0.982	0.987
GR00T agreement with the commanded sign	$\mathbf{0.634}$	0.638
GR00T shuffled baseline	0.563 ± 0.014	0.555 ± 0.015
GR00T label agreement	0.990	0.991
scorable axes, $\pi_{0.5}$ / GR00T	972 / 983	1,004 / 1,051

task instruction and the candidate sentence in full, and returns two integers from one call: a 0–3 score and a count of hallucinated objects. The rubric is fixed: 3 describes the visible objects and layout accurately, 2 describes most visible elements with minor errors, 1 mentions some visible elements but largely hallucinates, 0 describes a different scene. The prompt lists the LIBERO object vocabulary and asks the judge to count any listed object the description mentions that is not visible, which is the hallucination rate of Figure 4c. Each of the 200 held-out frames is scored under four conditions with that one prompt: the grounded interface, the matched plain interface, the grounded sentence paired with a different frame (the baseline), and the Gemma-4 target description for that held-out frame (the labeler reference). The conditions differ only in the sentence shown or, for the baseline, in the frame it is paired with. Means are over frames, and the p-values in §3.2 are paired Wilcoxon tests over the per-frame scores in `per_doc_scene_score`; one frame returned no score and is dropped from that pairing, leaving 199. The 1.61–1.64 labeler reference score measures the quality of Gemma-4 supervision; it is not an upper bound or a human-annotation score.

Scope of the rubric. The grades are ordinal and their middle levels are not defined by an object count: "most visible elements" does not fix a fraction. The scene claims in §3.2 therefore rest on paired comparisons under the same rubric and judge (read-out against shuffled frame, grounded against plain), which do not require the grades to be calibrated, and on the absent-object count, which counts false positives over a fixed object vocabulary. The judge records neither which objects are visible nor which ones a description names correctly, so recall and per-object precision cannot be computed from these runs. Object-level precision and recall against per-frame visibility annotations is the more direct measure and is left to future work.

D.3 Additional read-out examples

Figures 6 and 7 show twelve held-out frames in addition to the two in Figure 3. They are selected by a fixed stride over the sorted held-out `doc_id`s (every 33rd of 400) rather than chosen for appearance. On these twelve the layer-17 direction words agree with the commanded sign on 26 of 31 scorable axes (0.84), above the 0.772 measured over all 400 frames. Five panels show an orange $\times$: four are sign errors on scorable axes, and in one, panel (c), the word names an axis the policy barely moved, below the deadband, which the count excludes. Several panels also describe objects that are not in the frame, the hallucination measured in Figure 4c.

E Reconstruction fidelity in standardized PCA-32 coordinates

This is the derivation behind §4. Every steering direction is a difference of two reconstructor outputs, so the metric that validates the reconstructor must not credit it for structure that is already present before any reconstruction. Raw-space cosine credits exactly that structure here, and correcting for it takes two coordinate changes. Writing is scored by the held-out fraction of variance explained defined below (Table 8) and, closed loop, by the paired displacement contrast of Appendix H.

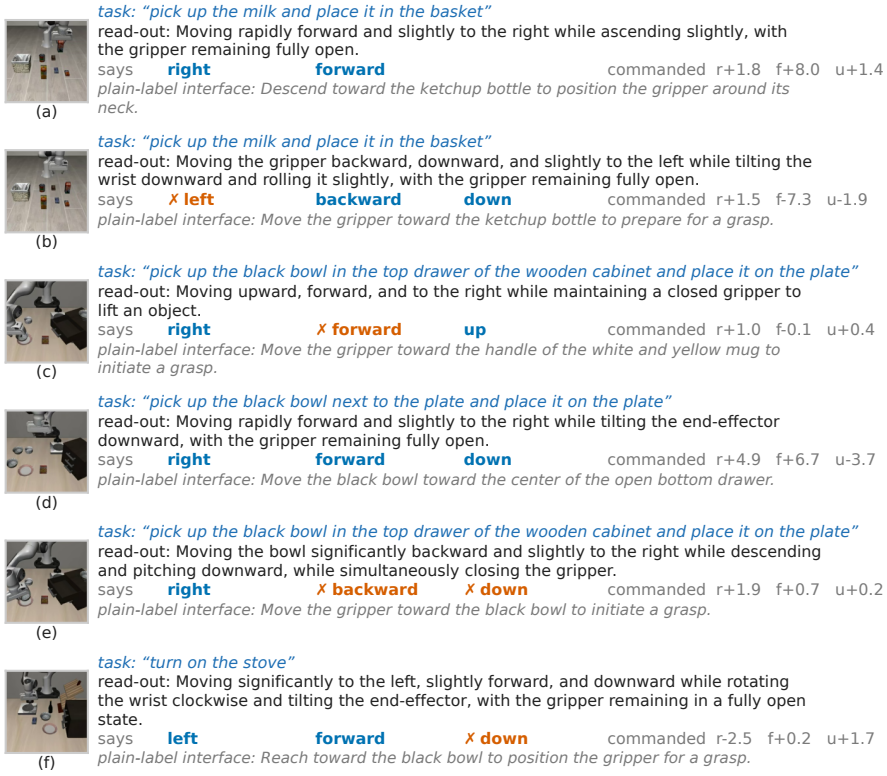


Figure 6: Systematically sampled held-out frames (a)–(f), conventions as in Figure 3: the verbatim read-out, the direction words it uses (blue where the word agrees with the sign the policy commanded, orange with $\times$ where it disagrees), the commanded net translation, and the matched plain-arm sentence.

Table 8: **Reconstruction exceeds the mean predictor only in standardized PCA-32 coordinates.** Held-out FVE (Eq. 2; 0 is no better than predicting the mean activation, negative is worse). *End-to-end* samples the verbalizer and then reconstructs; *reconstructor-only* feeds it the labeler-written target sentence directly, isolating the text-to-activation leg. $\pi_{0.5}$ reads and writes at layer 17, and GR00T at layer 3; the layer-16 ablation is in Table 6. **Bold** marks the $\pi_{0.5}$ reconstructors used for steering. GRPO entries average six seeds for $\pi_{0.5}$ and seven for GR00T. n=1,489 held-out frames.

model	layer	training	target	end-to-end ↑	reconstructor-only
$\pi_{0.5}$	17	SFT	full std.	-0.207	-0.295
$\pi_{0.5}$	17	SFT	PCA-32	$+\mathbf{0.119}$	$+0.063$
$\pi_{0.5}$	17	after GRPO	PCA-32	$+\mathbf{0.190}$	$+0.072$
GR00T	3	SFT	PCA-32	$+0.054$	$+0.218$
GR00T	3	after GRPO	PCA-32	$+0.142$	$+0.199$



(g)

task: "push the plate to the front of the stove"
read-out: Moving significantly forward and slightly to the left while descending slightly, with the gripper remaining fully open.
says **left** **forward** commanded r-8.8 f+0.4 u-5.1
plain-label interface: Lower the black bowl into the interior of the open drawer and then pull the drawer handle to close it.



(h)

task: "put the bowl on the stove"
read-out: Moving the mug forward and slightly to the right while tilting the gripper downward, maintaining a closed grip on the object.
says ✗ **right** **forward** **down** commanded r-2.2 f+3.0 u-6.0
plain-label interface: Move the black bowl toward the center of the open bottom drawer.



(i)

task: "turn on the stove"
read-out: Moving the moka pot leftward and forward while simultaneously lowering it and tilting the gripper downward, while maintaining a closed grip on the object.
says **forward** **down** commanded r-0.0 f+0.5 u-4.8
plain-label interface: Move the gripper toward the black bowl to initiate a grasp.



(j)

task: "put the bowl on top of the cabinet"
read-out: Moving rapidly to the right and slightly forward while descending and tilting the gripper downward with the fingers remaining open.
says **right** **forward** **down** commanded r+7.7 f+0.3 u-4.3
plain-label interface: Descend toward the black bowl to position the gripper around its rim.



(k)

task: "put the bowl on top of the cabinet"
read-out: Moving significantly to the right and slightly backward while descending slightly and rotating the wrist clockwise to position the gripper over the cabinet handle with the gripper remaining open.
says **right** **backward** commanded r+4.6 f-4.9 u-6.2
plain-label interface: Lower the wine bottle vertically onto the flat top surface of the cabinet.



(l)

task: "put the bowl on the stove"
read-out: Moving sharply downward, slightly backward, and to the left while rotating the wrist downward and leftward with the gripper remaining open.
says **left** **backward** **down** commanded r-2.3 f-1.1 u-8.5
plain-label interface: Move toward the salad dressing bottle to prepare for a grasp.

Figure 7: Systematically sampled held-out frames (g)–(l), same selection rule and same conventions. Some disagreements are sign errors, some are claims about an axis the policy barely moved; several panels also describe SCENE objects that are not in the frame, the hallucination measured in Figure 4c.

Notation and coordinates. We use the notation of §2: $h_\ell \in \mathbb{R}^{2048}$ is the mean image-token activation, z is its coordinate-wise standardized and unit-normalized form, and u is its projection onto the leading 32 principal components. All standardization and PCA parameters are fitted on the fit split. The generated description is s, and its reconstructed target is $\hat{u} = \widehat{R_\phi(s)}$, where the hat denotes unit normalization as in the training objective.

Why raw cosine is misleading. Activations from unrelated held-out frames have cosine similarity about 0.985: they already point in nearly the same direction. A reconstruction cosine of 0.991–0.998 therefore provides little evidence that the frame-specific activation was recovered. Standardization subtracts each coordinate's fit-split mean and divides by its fit-split standard deviation. PCA then selects the leading directions of variation for the reconstruction target.

Why report the coordinate system. Reconstruction quality depends on which activation differences the metric weights. The learned reconstructor scores below the mean predictor on the full standardized target and above it on the PCA-32 target (Table 8). These are different evaluation targets; a positive PCA-32 score does not establish accurate reconstruction of the full activation.

Scoring. We report the fraction of variance explained (FVE),

$$\mathrm{FVE} = 1 - \frac{\mathbb{E}\|u - \hat{u}\|^2}{\mathbb{E}\|u - \bar{u}\|^2}, \tag{2}$$

where $\bar{u}$ is the mean target over the held-out frames. A score of 0 matches predicting this mean; negative values are worse. We follow the FVE definition of Fraser-Taliente et al. [13] but evaluate it in the reconstructor's target coordinates. The coordinate choice changes the reported value by about 0.3 (Appendix B).

Why a small FVE still steers. An edit is applied as the normalized difference of two reconstructions, $\Delta \propto R(e(s)) - R(s)$ (Eq. 1), where s is the sentence the interface produced and $e(s)$ its direction-word edit. Error components shared by $R(e(s))$ and $R(s)$ cancel in Δ, and the steering experiments test whether the remaining difference produces the intended motion. A reconstructor with modest FVE can therefore still yield a usable steering direction; the steering results of §4 are the test of that.

What Table 8 shows. The comparison in Table 8 isolates the evaluation coordinates for a given model. For SFT, the 32-component score exceeds the full standardized score at both $\pi_{0.5}$ sites (Tables 8 and 6). GRPO raises FVE against the principal-component target, from $+0.119$ to $+0.190$ at the deployed site; no training change we tried, including a frozen backbone and twenty epochs, raised the FVE against the full standardized target above zero; the full-target evaluation remains below the mean predictor. In general, a reconstructor should be scored in coordinates where unrelated states are not already alike; a raw-space cosine near 1 is not evidence of reconstruction.

A bag-of-words baseline for the reconstructor. Given the same label sentences, a ridge regression using word-count features (a bag-of-words representation) reaches held-out reconstructor-only FVE $+0.133$ against the full standardized target and $+0.253$ against the PCA-32 target, above the 4B reconstructor in both. Text-to-activation is therefore learnable at this site, and the deployed module overfits: twenty epochs take its fit-split FVE to $+0.821$ while its held-out FVE against the full standardized target stays negative. The simpler model reconstructs held-out activations more accurately; we have not tested its steering in closed loop.

F Steering

F.1 How an edit reaches the policy

An edit becomes a vector by Eq. 1 and is then added at the write layer, but the two policies' injection sites differ in which tokens they cover and in units, which is why their doses are never compared directly. In $\pi_{0.5}$ the direction is added to each of the 512 non-padding image-token activations, scaled by that token's own norm, so α is dimensionless and relative to the activation it perturbs. In GR00T it is added to the layer-3 backbone output for all tokens in raw activation units. We therefore set GR00T's dose by its measured effect on the emitted action rather than by matching α: α=100 moves the emitted action by 10.0% of its scale against 3.1% at α=30, the $\pi_{0.5}$ dose (Appendix G). The edit is applied at every control call of an episode, and we log how many calls it was applied on, because a null from an edit with no direction word to replace differs from a null from an edit that is applied and has no effect. The confirmatory battery of Table 1 uses α=30 for $\pi_{0.5}$ and α=100 for GR00T.

F.2 Single-step edit grid: all six direction words

Table 9 is the open-loop check referenced in §4: for each of the six direction words we substitute its opposite in the sentence the interface produced, reconstruct, inject for one control call, and measure the change in the emitted action chunk along the axis the word names, in units of that axis's own standard deviation. No episode is run. Brackets are 95% bootstrap intervals and *sign* is the fraction of the 29 to 55 scorable frames that move as commanded. The plain interface has no direction word to edit and therefore no entry at either dose.

F.3 Every steering trial and the control suite

Figure 8 shows every paired starting state from the pre-specified control-suite session, including the four that move opposite to the commanded direction. The mean paired contrast over all of them is the $+0.241$ m reported in §4 (Wilcoxon signed-rank p=0.0017, 13 of 17 in the commanded direction). We show every panel to make clear that the effect is a population effect and not a set of hand-picked episodes.

Figure 10 summarizes the same session as a suite of magnitude-matched controls. The directional text edit produces the largest signed forward-minus-backward contrast. The raw decoder-weight vector and three controls that have no positive/negative pair, and are therefore compared with the unedited run (a random direction in the full activation space, a random direction in the reconstructor's output subspace, and an irrelevant object edit), each have absolute contrast at most 0.023 m.

Table 9: Four of the six edited words move the emitted action along the axis they name in one control call. **Bold** marks the 95% paired-bootstrap intervals that exclude zero; the bottom row is a magnitude-matched random direction.

	α=30		α=60
edited word	Δ (95% CI)	sign	Δ
right	$+\mathbf{0.191}$ $[0.07, 0.32]$	0.70	$+0.173$
left	$+0.101$ $[-0.07, 0.29]$	0.69	$+0.158$
forward	$+\mathbf{0.265}$ $[0.16, 0.39]$	0.82	$+0.304$
backward	$+\mathbf{0.453}$ $[0.29, 0.61]$	0.84	$+0.478$
up	$+0.082$ $[-0.02, 0.18]$	0.65	$+0.058$
down	$+\mathbf{0.084}$ $[0.03, 0.14]$	0.59	$+0.085$
random, matched	$+0.023$ $[0.02, 0.03]$	—	—

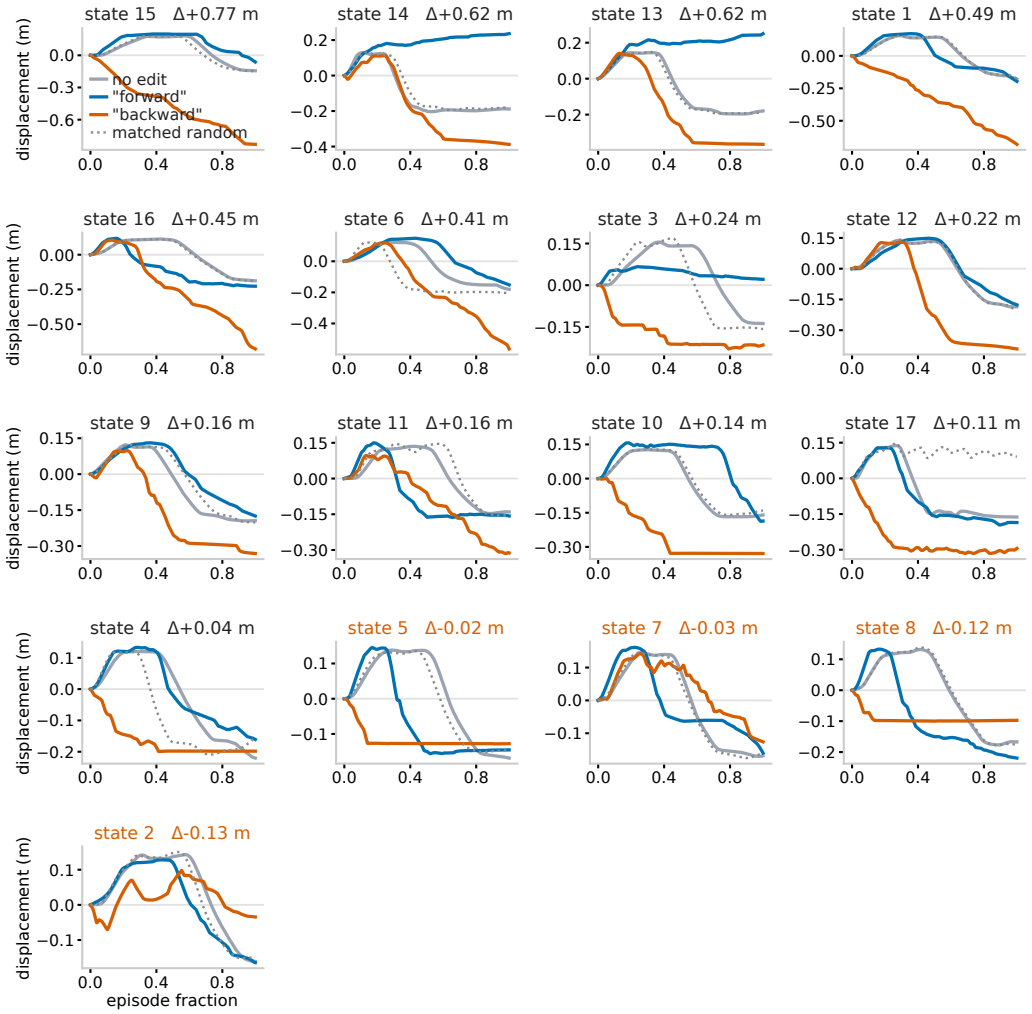


Figure 8: All seventeen paired starting states from the pre-specified control-suite session on task 5. Each panel shows three episodes that differ only in the direction words of the sentence the interface produced: unedited (gray), every forward/backward direction term forced to "forward" (blue), and forced to "backward" (orange). The dotted line is the magnitude-matched random direction where it was run. Titles give the paired contrast; orange titles mark the four states that move opposite to the commanded direction.

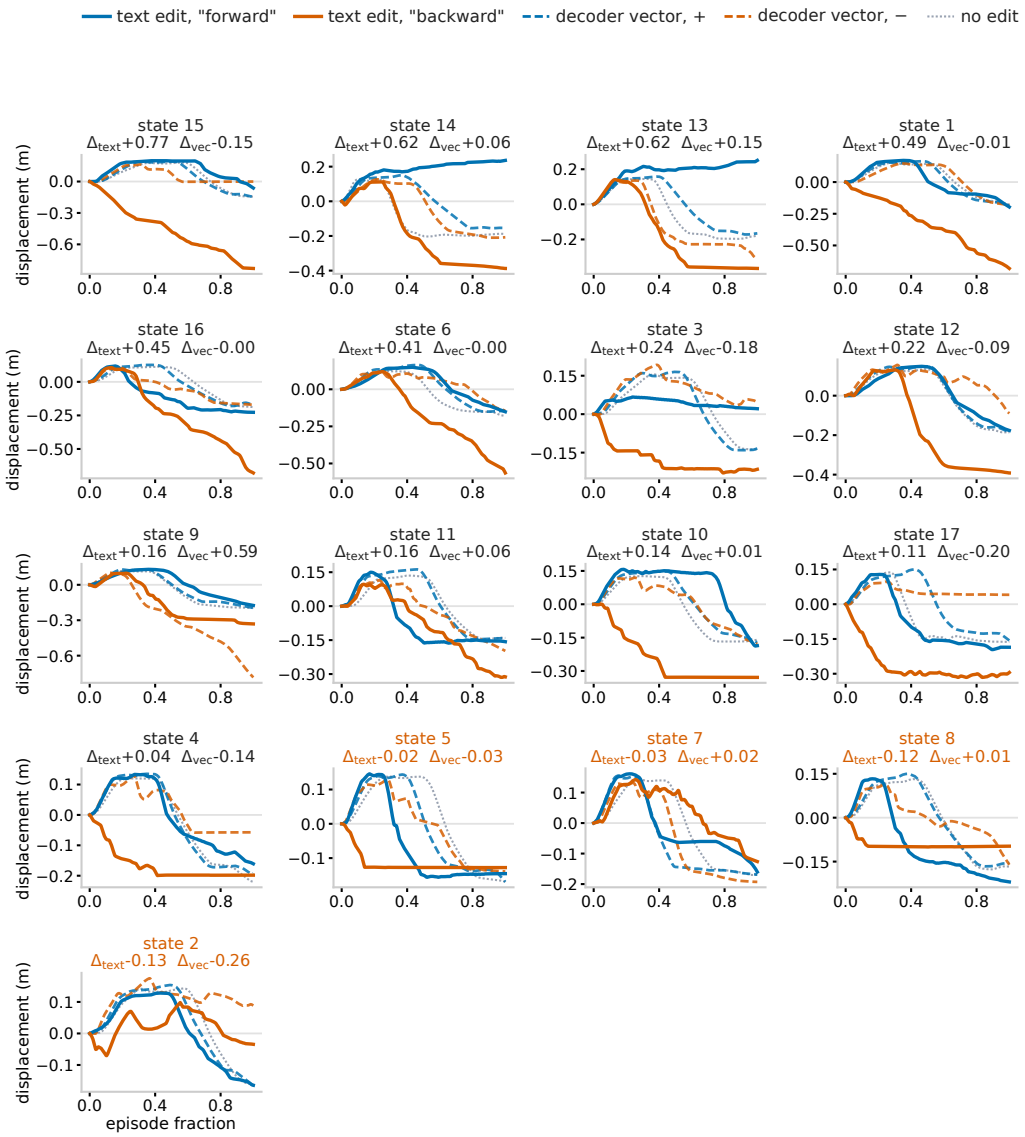


Figure 9: **The text edit has a consistent directional effect; the raw decoder-weight vector does not.** Each panel is a paired starting state from the same session as Figure 8, run under the same dose and the same per-call schedule. Solid lines are the text-edit direction forced to "forward" (blue) and "backward" (orange); dashed lines add the decoder's raw weight vector with either sign along the same axis; the dotted gray line is the unedited run. The solid pair separates while the dashed pair stays near the unedited run: the text-edit contrast (Δ_{text}) is in the commanded direction on 13 of 17 states, the decoder-weight contrast (Δ_{vec}) on 7 of 17, near half. All 17 states are numbered as in Figure 8. The encoding direction of §4, derived from the same weights through the activation covariance, also steers.

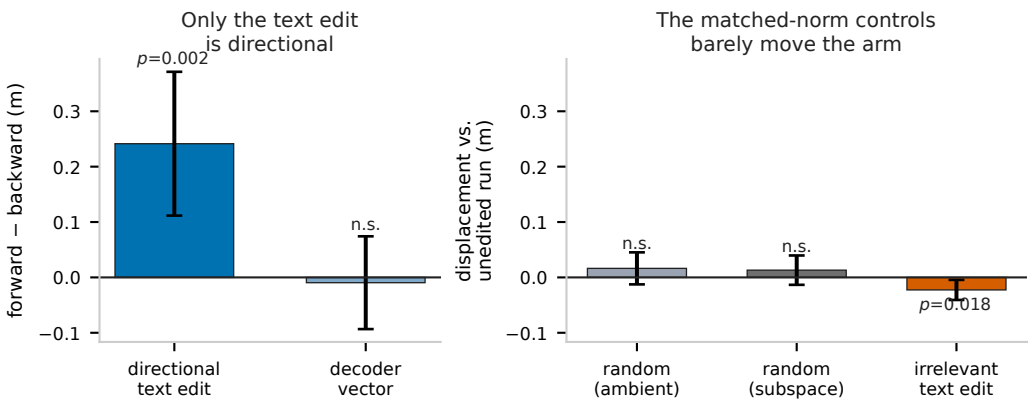


Figure 10: **Magnitude-matched controls produce much smaller effects, in the same session as Figures 8–9.** *(a)* The forward-minus-backward contrast for the directional text edit and for the raw decoder-weight vector injected along the same axis at the same norm. *(b)* The three single-arm controls relative to the unedited run: a matched-norm random direction in the full activation space, a matched-norm random direction inside the reconstructor's output subspace, and an irrelevant object-swap edit. Bars are mean paired contrasts over the shared starting states with 95% confidence intervals; two-sided Wilcoxon signed-rank p-values are printed on the bars.

F.4 A second axis

Commanded axes align with world axes: the corresponding-axis correlations are at least $+0.75$, and cross-axis correlations are at most 0.27. Natural movement ranges differ by a factor of 1.8–4.6 across tasks, so we report each axis separately.

The pre-specified control suite above steers the fore/aft axis, the one whose direction words dominate task 5's verbalizations. To check that the effect is not specific to that axis, we re-ran the closed-loop control suite on the same task, site and dose but forced the sentence's vertical motion words instead (*lower*/*raise*, *lift*/*drop*, *up*/*down*; axis 2), fitting the matched decoder-weight and subspace-random controls for that axis in the same way. Table 2 places the two axes side by side. The text edit steers the vertical axis on every one of the 18 starting states, with a paired up-minus-down contrast of $+0.140$ m (two-sided Wilcoxon signed-rank $p < 10^{-4}$), while the raw decoder-weight vector injected along the same axis at the same norm moves the arm less than a tenth as far ($+0.012$ m). On both tested axes, the text edit produces a larger directional effect than the raw decoder-weight vector.

F.5 Dose, axis, and the cost of the write path

§5 rests on the dose-response experiment in Figure 11, a sweep of the injection magnitude on task 5 with 18 paired starting states at each dose. The whole-episode displacement contrast between the forward and backward edits rises monotonically with the dose (r=0.932; Page's trend test [29] L=880, p=4.8×10^{-4}), so the effect is graded, not all-or-nothing (panel a). A re-run session at $\alpha \in \{5, 10, 15\}$ logged the gripper at every control call; over the first ten calls the contrast is significant at all three doses, so the effect begins early in the episode rather than accumulating slowly (panel b). Scoring each axis in units of its own standard deviation, so that axes with different natural ranges are comparable, shows that the effect is strongest on the named axis, with substantial vertical displacement: at α=20 it moves that axis by $+4.32$ standard deviations, the vertical axis by $+2.72$, and the lateral axis by -0.12 (panel c). Task success falls with dose in both edited arms, faster when the edit opposes the task (0.556 to 0.056) than when it is aligned with it (0.889 to 0.444), while the untouched baseline stays between 0.78 and 0.94 (panel d).

Table 10 lists the same sweep numerically, so the steering that remains at each level of task success can be read directly. The re-run session at α=5 gives a whole-episode contrast of $+0.026$ m (uncorrected p=0.024) with success 0.889 untouched, 0.944 aligned and 0.722 opposed; the sweep's own α=5 contrast is -0.002 m. Success at n=18 per arm has a standard error near 0.1, so differences in success of that size between neighboring doses are not resolved.

Table 10: **Steering and task success across the dose sweep on task 5.** Whole-episode forward-minus-backward contrast over 18 paired starting states per dose, with two-sided Wilcoxon signed-rank p (uncorrected) and the smallest effect detectable at 80% power (MDE); task success for the untouched run and for the edit aligned with or opposed to the task.

	steering			task success		
α	Δ (m)	p	MDE (m)	untouched	aligned	opposed
5	−0.002	0.39	0.083	0.833	0.889	0.556
10	+0.075	0.12	0.119	0.944	0.667	0.278
15	+0.154	0.0056	0.149	0.833	0.556	0.056
20	+0.185	0.00084	0.144	0.778	0.556	0.000
30	+0.215	0.00053	0.178	0.944	0.444	0.056

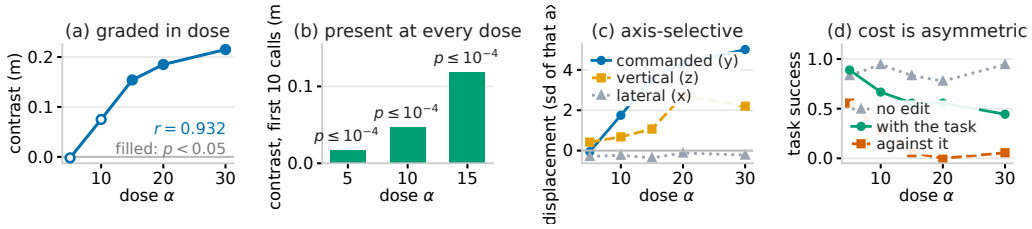


Figure 11: **Dose response on task 5.** *(a)* Whole-episode contrast against dose (r=0.932; Page's L=880, p=4.8×10^{-4}). *(b)* Contrast over the first ten control calls at each dose. *(c)* Displacement per axis in units of each axis's own variability. *(d)* Task success at each dose for the aligned edit, the opposed edit, and the untouched baseline.

G The second policy

GR00T-N1.7 is a later checkpoint of the N1 family, which has no separate report, so we cite the N1 paper [4]. Table 11 gives the eight GR00T conditions of the confirmatory family in §4: the four tasks of Table 4 under the SFT and GRPO reconstructors, editing the sentence's vertical words and measuring the paired up-minus-down contrast along axis 2 at α=100 over shared starting states. One condition, task 8 under the GRPO reconstructor, survives Holm correction within the 16-test family, at +0.028 m with 33 of 48 states in the commanded direction; the SFT task-8 session was cut short by a simulator crash and has 9 pairs. The other seven conditions, including that truncated one, are within 0.017 m of zero.

Figure 12 is the diagnostic from an earlier GR00T session. That session returned a null whose magnitude-matched random control also had no effect, so it could not distinguish "language cannot steer this policy" from "nothing we injected perturbed it". The three panels resolve this: the deployed site has the highest leverage we measured, the dose moves the emitted action by 10% of its scale, and the edit is applied on nearly every control call; in that session the text-edit direction still did not steer (+0.004 m over 31 paired states, p=0.52, against a smallest detectable effect of 0.016 m). The steered arms lose success (0.781 and 0.781 against 0.875 untouched) while the magnitude-matched random arm does not (0.875), so the edit changed behavior without directing it. The eight-condition battery above, run later with 48 states per condition, recovers one significant condition at +0.028 m;

Table 11: **The eight GR00T conditions of the confirmatory battery.** Paired up-minus-down displacement contrasts in meters along axis 2 at α=100, with the number of shared starting states and the two-sided Wilcoxon signed-rank p. **Bold** marks the one condition significant at the 0.05 level after Holm–Bonferroni correction within the 16-test family of Table 1.

	SFT reconstructor			after GRPO		
task	Δ (m)	n	p	Δ (m)	n	p
0	−0.010	48	0.50	+0.017	48	0.27
3	+0.009	48	0.031	0.000	48	0.91
5	+0.003	48	0.31	+0.005	48	0.0074
8	−0.007	9	0.91	**+0.028**	48	**0.0012**

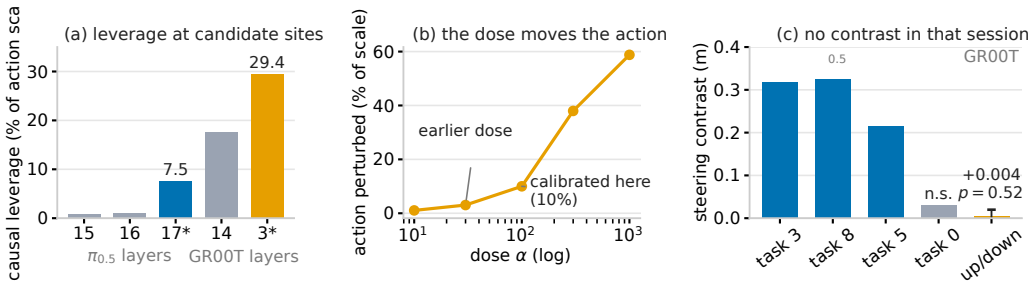


Figure 12: **Earlier GR00T session: high leverage and a calibrated dose, with no steering detectable at n=31.** *(a)* Causal leverage at the candidate layers of both policies, the fraction of the action scale by which the emitted action moves when another frame's activation is pinned there (n=40; deployed sites starred). *(b)* At the dose used for the closed-loop test the injection moves GR00T's emitted action by 10% of its scale, against 3.1% at α=30. *(c)* Closed-loop contrasts from that session: $\pi_{0.5}$ steers on three tasks, while GR00T's vertical axis gives +0.004 m over 128 episodes on 31 paired states (Wilcoxon signed-rank, p=0.52); the error bar is the smallest effect this design could detect, 0.016 m, under a tenth of $\pi_{0.5}$'s smallest significant contrast.

both sessions are consistent with a GR00T effect of a few centimeters, near this design's detection floor.

H Statistical protocol

Pairing. Every contrast is computed on starting states shared by both arms within one session, with the pairing key (seed, initial state), and tested with a two-sided Wilcoxon signed-rank test [39]. Truncated rollouts are dropped before pairing. Sessions are never pooled, because measured success drifts by ±0.12–0.15 between sessions with nothing changed.

Families. Holm–Bonferroni correction [20] is applied within each declared family:

- **Confirmatory battery** (Tables 1 and 11): 16 per-task contrasts, the four tasks under {SFT, GRPO} × {$\pi_{0.5}$, GR00T}; 7 are significant after correction at the 0.05 level, and the largest surviving raw p is 1.2×10^{-3}.
- **Earlier closed-loop family**: 27 tests covering the per-task steering contrasts, the four control-suite arms on both tasks, the plain-label ablation, the axis controls and the dose sweep; 14 are significant after correction, and every arm predicted to be null fails to reach significance.
- **Pre-specified replication family**: the control-suite replication on task 5, the instruction comparison on both tasks, and the early-window dose analysis; every predicted positive remains significant after correction (largest uncorrected signed-rank p=0.0020) and both predicted nulls are null.
- **Scene-fidelity family**: the two read-out-versus-baseline contrasts, one per policy (Figure 4b).

Everything else is exploratory and is reported with uncorrected p-values.

Path normalization. Dividing each episode's axial displacement by the total path traveled leaves all three per-task steering contrasts significant, and the steered arms have shorter paths than the unedited ones on all four tasks, so increased travel distance does not explain the directional effect.

Power. For the earlier GR00T experiment and the dose sweep, we report the smallest effect detectable at 80% power from the observed paired standard deviation: 0.016 m for the earlier GR00T closed-loop contrast at n=31, under a tenth of $\pi_{0.5}$'s smallest significant contrast, and 0.083–0.178 m per dose at n=18 in the dose sweep. Success rates are much noisier than displacements at these sample sizes, which is why every primary contrast is a displacement.