Instruction Blindness in Vision–Language–Action Policies: Diagnosis and a Low-Rank Data Cure

Yuchen Liu*
Φ(fight) Research
The Hong Kong University of Science and Technology

Muchen Han*
Φ(fight) Research
The Hong Kong University of Science and Technology

Abstract

A vision–language–action (VLA) policy can pass a manipulation benchmark without reading its instruction: replace the words with the string "xxx" and it emits the same actions, moving to the object that usually occupies that position rather than resolving the name. Standard benchmarks cannot detect this, because they place the named object where the policy expects it, so a position shortcut and genuine language grounding earn the same score. We call this failure mode instruction blindness, and we measure it with a counterfactual that a position shortcut cannot pass: hold the pixels, the scene, and the robot pose fixed, change one word in the instruction, and record which object the arm reaches. Published evaluations put the field at 0–1% correct object selection on this axis, and four world-action models we ran reach 20–45%. We then locate the failure in the network. In behavior-cloning policies open enough to instrument, the backbone resolves the named object in its language stream—patching localizes the binding to a single head, and injecting the word direction flips the selected object—while the action head routes position and never reads that binding; on the policy we cure, no training-time regularizer on the confounded data reopens the route. Because the knowledge is present and only the routing is missing, the repair is a data recipe rather than an architecture: making the instruction the only predictor of the target, under a low-rank update that protects the pretrained resolution, takes object-identity grounding from 13.5% to 94.5%, where full fine-tuning on the identical demonstrations destroys it. The cured policy obeys a renamed target on 200 of 200 trials, grounds seven referent-type axes in one set of weights, and transfers to a second simulator. We release the policy, the deconfounded datasets and their generator, and the measurement code.

1 Introduction

Replace a manipulation instruction with the string "xxx" and a frontier vision–language–action (VLA) policy emits the same actions (Zhou et al., 2025). It still completes the task, because it moves to the object that usually occupies the target position instead of resolving the words. Standard benchmarks reward this: they place the named object where the policy expects it, so a position shortcut and genuine language grounding earn the same score. On the one axis that separates them—object identity, obtained by renaming the target and keeping the scene—the published field selects the correct object 0–1% of the time (Figure 1).

We call this failure mode instruction blindness, and we treat it as a concrete instance of a known phenomenon rather than a new mystery. Causal confusion in imitation learning (de Haan et al., 2019) predicts that a policy trained by behavior cloning will latch onto whichever feature most easily predicts the expert action; when the named object reliably sits in a fixed position, that feature is position, and the instruction becomes a spurious

*Equal contribution. Correspondence: yuchen@φ.monster. Project page, code, models, and data: https://xn--7xa.monster/Galahad/.

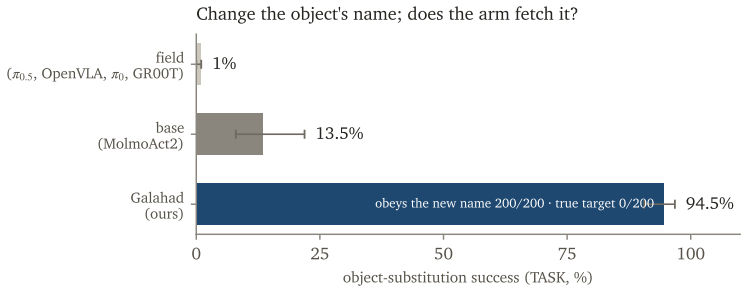

Figure 1: On the axis that separates language grounding from a position shortcut—renaming the target object while holding the scene fixed—the published field selects the correct object 0–1% of the time and the base model 13.5%. The cure of Section 6 reaches 94.5%, obeying the new name on 200 of 200 trials and never fetching the object it was trained to fetch. Intervals are Wilson 95%.

correlate the policy is free to ignore. Instruction blindness is what causal confusion looks like when the confounded variable is the language channel of a robot policy. Naming it this way sets the agenda of the paper: locate the mechanism, and remove the confound.

Measuring the phenomenon requires a test that a position shortcut cannot pass, and neither of the signals the field relies on qualifies. A policy that memorizes positions completes the task without reading the name, so task success does not separate the two. Blanking the cameras collapses a position-memorizer as surely as it collapses a grounded policy, so vision-dependence does not either. Our measurement changes one word. Holding the pixels, the scene, and the robot pose fixed, we rename the target and record where the arm goes; a grounded policy moves to the newly named object, a shortcut moves to the same place regardless of the word. This one comparison, which we call swap-obey, converts a hollow "it failed" into a positive control ("it obeyed the new name"), and it is the instrument every number in the paper is built on.

What is missing is a path, not a fact. On behavior-cloning policies open enough to instrument (SmolVLA (Shukor et al., 2025) and a Qwen2.5-VL policy (Bai et al., 2025)), the backbone resolves the named object in its language stream—activation patching localizes the binding to a single head, and injecting the word direction at an early layer flips the selected object 100% of the time—while the action head routes position and never reads that binding (Section 5). Because the knowledge is present and only the routing is missing, the repair is a data recipe, not an architecture. We make the instruction the only predictor of the target: every object appears as the target and as a distractor across randomized positions, so a position shortcut earns nothing and only reading the name pays. We protect the pretrained resolution with a low-rank update. On the identical data, full fine-tuning destroys grounding and returns 12.5%, while a rank-32 adapter installs it and returns 94.5%. No coupling loss, no detector, and no architecture change is added.

Contributions. This paper makes three contributions.

1. A diagnosis. We name instruction blindness and locate its mechanism three ways (imitation-learning theory, a black-box circuit, and a regularizer null): the object-name binding lives in the language stream, the action head routes position, and one word-direction injection flips the grounding.
2. A cure. Deconfounded data and low-rank protection form a causal pair—the same data under full fine-tuning destroys grounding, and no training-time regularizer on the confounded data substitutes for the data—and the recipe is cheap to extend: an object grounds by appearing in a training scene, with no demonstration of it required.
3. Galahad, the released artifact. The cured policy reaches 94.5% on object substitution where its base scores 13.5% and the field 0–1%, obeys a renamed target 200

of 200 times, grounds seven referent-type axes in one set of weights, and transfers to a second simulator. We release the policy, the deconfounded datasets and their generator, and the measurement code, so the diagnosis and the cure are both reproducible.

2 Related work

What manipulation benchmarks actually measure. Open VLA policies map pixels and instructions to actions (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024). The suites that score them (Liu et al., 2023) share a design assumption that is rarely stated: the named object is placed where the training demonstrations put it. Under that assumption a policy that resolves the name and a policy that reaches a remembered position produce the same trajectory and earn the same number, so task success cannot separate them. Reported progress on these suites (Physical Intelligence, 2025; NVIDIA, 2025b; Fang et al., 2026a) is therefore evidence of motor competence and is silent on grounding. Our measurement breaks the assumption by moving the name and holding everything else fixed.

Instruction blindness as causal confusion. That a policy trained by behavior cloning will prefer an easy sufficient feature over a harder informative one is the standing prediction of the causal-confusion and shortcut-learning literature (de Haan et al., 2019; Geirhos et al., 2020). The mechanism usually invoked is gradient starvation: once one feature explains the target, the other stops receiving gradient (Pezeshki et al., 2021; Wen et al., 2020). Robot-policy evaluations report exactly the predicted signature. Renaming or displacing the target collapses the field to near zero on object identity (Zhou et al., 2025; Fei et al., 2025). Target selection sits at chance once grasp success is controlled for (Yu et al., 2026), and a thirty-policy sim-and-real benchmark puts the best open-vocabulary instruction-following score at 1.67% (Chen et al., 2026). What the literature has not supplied is where in the network the failure sits. We name the phenomenon, and we localize it: the binding is formed in the language stream and the action head never reads it, a result that extends activation-level steering of VLAs (Häon et al., 2025) from motion to object identity.

Supplying grounding through the architecture, or through the data. One line of work adds a grounding pathway to the policy: an explicit point or box routed into the action head (Tsai et al., 2026; Yu et al., 2025), or an attention-alignment objective (Jia et al., 2026). Each assumes the model does not have the grounding and must be given it. Our mechanism section shows the opposite: the pretrained backbone resolves the referent, and only the route to the action head is missing. That changes what the fix has to do: preserve a capability rather than install one. Counterfactual, role-balanced demonstrations (Pitis et al., 2020; Ameperosa et al., 2025) under a low-rank update (Hu et al., 2022) do exactly that, obtaining the gradient-isolation principle of Driess et al. (2025) without an architectural change. The same reasoning excludes reinforcement learning as the fix, since RL sharpens what a base already samples rather than adding an absent capability (Yue et al., 2025; Liu et al., 2025). The RL-over-SFT generalization gap reported by Chu et al. (2025) largely closes once referent diversity is matched (Lin et al., 2025). Installing instruction-following on a strong base with a data recipe follows a template established in language modeling (Taori et al., 2023; Wang et al., 2023).

World-action models. Models that predict a future alongside their actions are an established class (Cen et al., 2025b; Wang et al., 2026b; Ma et al., 2026). On our protocol the best of them obeys the instruction 20–45% of the time (Section 4), which the class's own structure explains: foresight conditioned on the action reaches the instruction only through the action head, and such models will report a successful future under a wrong action (Liu et al., 2026). Evaluation practice compounds this, since world-model leaderboards score perceptual quality rather than instruction-correctness (Shang et al., 2026; Li et al., 2025). We therefore adopt the functional, intervention-aware definition of a world model (Wang et al., 2026a) and measure the predicted future with the same counterfactual we apply to the action.

3 Measuring grounding: a counterfactual protocol

Every number in this paper is a behavioral counterfactual, so we state the protocol before any result.

Task success and vision-dependence do not measure grounding. A policy that memorizes where each object sits completes the task without reading the instruction, so task success certifies motor competence, not grounding. Blanking the cameras does not fix this: a position-memorizer collapses under blank cameras just as a grounded policy does, so an occlusion control (occ) rules out a state leak but not a position prior. We show this directly in Section 7: the base model passes task success on the spatial suite at 99% and on the goal suite at 98.8%, and both scores are shortcuts. A measurement of grounding must vary the one thing a shortcut ignores—the word.

swap-obey: rename the target and watch the arm. Our primary instrument holds the pixels, the scene, and the robot pose fixed, changes one word in the instruction, and records which object the end-effector reaches. The new name is drawn from an object actually present in the scene, so there is always a real target to go to, and we score two quantities: the success rate on the original target, which a grounded policy drives to zero, and obeyed-name, the rate at which the end-effector lands on the newly named object. obeyed-name is the load-bearing readout: it converts a hollow "the policy failed" into a positive control, "the policy obeyed the new name," and only that direction distinguishes a policy that reads the instruction from one that rides the scene. We rotate the renamed decoy across every in-scene non-target so the protocol cannot be passed by orienting toward one familiar object.

Two disciplines keep the readout honest. We verify the distance, not just the threshold: on the cured policy the end-effector lands a median 2 mm from the newly named object while sitting a median 9 cm from the originally trained one, and no trial falls in the 4–5 cm grazing band, so a 2 cm threshold gives the same verdict as a 5 cm one. And we report a reach-probe on every failure, splitting it into reached-target (the arm reached the right object and the grasp slipped, a motor failure) and went-distractor (the arm went to the wrong object, a grounding failure), so we never record a motor limit as a grounding limit.

Each control removes a different way to cheat. occ asks whether the policy uses vision (blank the cameras; success must collapse). swap-obey asks whether it follows the named object (rename the target; the arm must switch). nonsense asks whether it needs a name at all (replace the instruction with "xxx"; a policy that still succeeds is riding a scene prior). Success is lift-gated and init-false: the object must leave the table by a fixed margin, so a scene in which the target begins at the goal cannot score a do-nothing success. We measured the init value of every metric and required a do-nothing policy to score near zero before trusting any rate.

One protocol, every referent type. Object identity is the sharpest axis but not the only one. The same counterfactual instantiates for each way an instruction can name a target—color, category, ordinal position, spatial relation, negation, and composition—by varying the one word that selects the referent. This is the protocol behind the field-wide result of Section 4 and the seven-axis result of Section 7. It also extends to a policy that predicts a future: we change one word and ask whether the predicted region of motion moves to the newly named object, a control the world-model evaluation literature leaves open (Appendix B).

4 Instruction blindness is field-wide

We run the protocol of Section 3 across the field. The result is not a uniform wall but a graded one: policies fail to different depths, and the depth is set by how the instruction reaches the output.

The base has full motor and near-chance grounding. The base model, MolmoAct2 (Fang et al., 2026a), executes standard pick-and-place at 100% and, under object substitution,

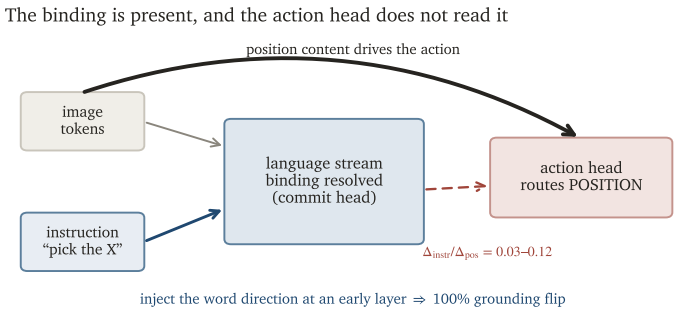

Figure 2: The binding is present and the action head does not read it. The instruction forms an object binding in the language stream (solid), but the action head is driven by position content routed from the image (heavy), and the path from the binding to the action head is weak (dashed, $\Delta_{\text{instr}}/\Delta_{\text{pos}}$=0.03–0.12). Injecting the word direction at an early layer flips the selected object 100% of the time.

selects the correct object 13.5% of the time (measured cleanly at n=96, $[8.0, 21.9]$). The gap between 100% motor and 13.5% grounding is instruction blindness in one model: the motor works and the instruction does not drive it. This before-number also settles how much grounding the cure starts from—13.5%, not zero.

The imitation field collapses on object identity. On the object-substitution axis the published imitation field scores 0–1%: $\pi_{0.5}$ at 1%, and OpenVLA, π_0, and GR00T at 0% (Zhou et al., 2025). We do not rest the claim on one source. Yu et al. (2026) report frontier VLAs selecting the correct object at near-chance rates after controlling for grasp success; Chen et al. (2026) run thirty policies on an independent sim-and-real benchmark and find the best open-vocabulary instruction-following score is 1.67%. The newest releases repeat the pattern: an open VLA reporting 98.7% on LIBERO reproduces its published positive control on our harness (50/50), then drops to 10.0% [4.3, 21.4] under object substitution, and completes the canonical task 52% of the time with the instruction replaced by "xxx" (Unitree Robotics, 2026). The scene decides; the words are optional.

World-action models are a leaderboard, not a wall. We ran the battery on four open world-action models under a state-leak-clean protocol (blank cameras and blank proprioception both collapse success, so no model succeeds from a leaked state). Obey rates: UniVLA (Wang et al., 2026b) 20%, InternVLA-A1.5 (Ma et al., 2026) 23.3%, WorldVLA (Cen et al., 2025b) 25%, and RynnVLA-002 (Cen et al., 2025a) 45%. The best open world-action model reaches roughly half the grounding of our cured policy, and the rest sit near chance. Their prediction heads are action-conditioned and instruction-blind at the frame level, so the instruction reaches the predicted frame only through the same weak action head. The graded leaderboard is one disease read at different depths: the closer the instruction sits to the output objective, the more of it survives, and Appendix A measures that spectrum across the generative world-model zoo.

5 Why imitation is instruction-blind

The knowledge a grounded policy needs is already inside the network. The pretrained backbone resolves the named object; the action head routes position and never consults that resolution. Three lines of evidence converge on this—the field's own ablations, a black-box circuit, and a null—and removing the confound then closes the gap at the level of the circuit.

The field's ablations already show language is decorative. When the instruction is masked, published policies barely move: OpenVLA-OFT falls from 97% to 0.9%, π_0 from 94% to

0.6%, and $\pi_{0.5}$ from 97% to 0.0% (Fang et al., 2026b), and a vision-only controller matches a language-conditioned one (44.6% vs 47.8%) (Lian et al., 2026). A capability that can be removed with no change in behavior was never driving the policy.

A black-box circuit localizes the binding. On the behavior-cloning substrates open enough to instrument (SmolVLA (Shukor et al., 2025) and a Qwen2.5-VL policy (Bai et al., 2025); we state this scope rather than claim it transfers untested), the action output moves with target position and barely with the instruction: across eight models the per-model perturbation ratio $\Delta_{\text{instr}}/\Delta_{\text{pos}}$ is 0.03–0.12 (n=40 each). The object binding is nonetheless present and localizable: a single commit head (L19H17) carries it, and it is causal in one direction only. Injecting the word direction at an early layer flips the selected object 100% of the time, while steering the action read-out does nothing (0%). The binding exists upstream and the action head simply does not consult it.

No training-time surrogate on the confounded data reopens the route. If the knowledge were merely under-weighted, a penalty or a bottleneck on the original data would surface it. None does: a spectral-decoupling penalty, an action-head loss reweighting, and a one-sided visual information bottleneck all sit at the disease level regardless of hyperparameter (Section 6). The route cannot be re-opened by regularizing the old data; the confound has to be removed from the data itself.

Fixing the data moves the circuit. The same measurement that localizes the failure also tracks the repair. In the base, the action head's read-out is weakly aligned to the object role (cosine +0.83 on the canonical object, +0.47 on the distractor, with 22% of distractor reads anti-aligned); after the cure the same measurement reads +0.92 and +0.94 with no anti-aligned reads. The deconfounded data does not teach the backbone what the words mean—it already knew—it teaches the action head to route what the backbone resolves. This is why the cure is a data recipe and not an architecture, and it is the subject of the next section.

6 Deconfounded data installs grounding, and low-rank protection preserves it

The diagnosis of Section 5 determines what the repair has to do. The backbone already resolves the referent, so the training must open the route from that resolution to the action head without overwriting the resolution itself. Two ingredients accomplish this, and a falsification lattice isolates them as the active pair.

Make the instruction the only predictor of the target. We build demonstrations in which no variable other than the instruction predicts which object to fetch. Every object appears as the target and as a distractor across randomized positions, and the distractors are always in frame. A position prior earns nothing because the target is not where it was last time; a scene prior earns nothing because every scene contains every object; only reading the name pays. The principle is one sentence, and applying it turns the base's 13.5% into a grounded policy.

Protect the pretrained resolution with a low-rank update. The protection is structural, a property of the rank rather than of the step size. On identical data, full fine-tuning returns 12.5% and a rank-32 adapter returns 94.5%: the same demonstrations either destroy grounding or install it, decided by how much of the pretrained model the update is allowed to move. Doubling the learning rate barely moves the result (68.75% to 66.7% on the development data) while releasing the rank breaks it, and the protection-capacity curve rises and then plateaus—56.7 / 65.0 / 68.75 / 68.3 at ranks 8 / 16 / 32 / 64—so rank-32 is a measured choice with no headroom above it. On clean data the curve is flatter and higher (85 at rank 8, 93 at rank 32). Low rank preserves what the backbone knows; the deconfounded data teaches the action head to read it.

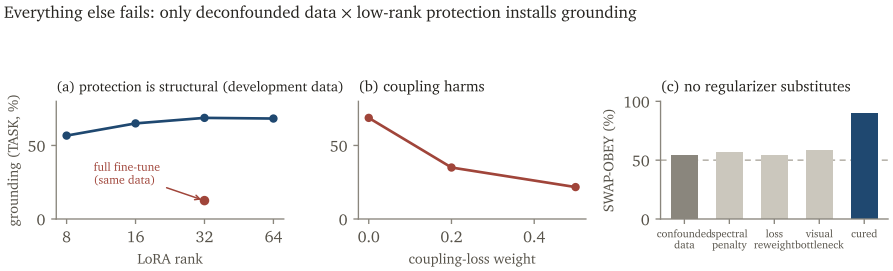

Figure 3: The deconfounded data and the low-rank update are the active pair. (a) Protection is a property of the rank: the capacity curve rises and plateaus, and full fine-tuning on the same data returns 12.5%. (b) Coupling supervision degrades grounding in a dose-response. (c) Three training-time regularizers on confounded data all land at the confounded-data level.

The data is the active ingredient. The interventions below rule out the substitutes a reader reaches for first (Figure 3).

A training-time objective does not stand in for the data. We trained three regularizers on the original confounded data—a spectral-decoupling penalty on the action, an action-head loss reweighting, and a one-sided visual information bottleneck. Every arm at every hyperparameter lands at the confounded-data level (task 2–25%, obey 48–62%) against the deconfounded recipe's task 60% and obey 90%; the one arm that raises a metric does so by damping the motor. Grounding is not a quantity the objective can surface from confounded data.

Supervising the attention is worse than leaving it alone. Forcing the action expert's attention to peak on the named object (Jia et al., 2026)—the intervention a reader expects to help—degrades grounding in a clean dose-response: 68.75% with no coupling, 35% at weight 0.2, 21.7% at weight 0.5. The result follows from the diagnosis: the backbone already resolves where to look, so constraining the read-out over-determines it.

More data of the same kind adds nothing: on the minimal testbed, doubling deconfounded demonstrations moves object selection from 52% to 50.5%, raising the blind floor and leaving the grounding where it was.

One principle, one generator per referent type. The principle is fixed and the design is per type: color must be made orthogonal to position and shape, spatial to identity, and negation, relation, ordinal, and composition each break a different confound (Appendix G). All share one scripted-oracle generator, so a new referent type is a small design rather than a new pipeline, and the generator produces the counterfactual evaluation and the training set from the same code.

7 Experiments

We evaluate the cured policy—which we release as Galahad—on the protocol of Section 3. It grounds object identity where the published field scores 0–1%, its released weight grounds seven referent-type axes at once, and its remaining failures are motor rather than grounding.

Two weights carry the results. The object-family weight is trained on the object axis alone and holds the 94.5% of Table 1 and the prediction-face result of Appendix B. The release weight folds seven referent axes into one set of weights (Table 2), with grounding that matches the specialist's.

Setup. The base is MolmoAct2-Think-LIBERO; the cure is a rank-32 low-rank adapter over both the VLM and the action expert, trained for 3,000 steps (8-way data parallelism on 24 GB consumer GPUs) on 900 deconfounded episodes—90 per object, every object

Table 1: Main battery, object substitution, all ten slots, n=200, on the object-family weight. The cured policy passes every face; the base has the motor and lacks the grounding.

face	question	cured	base
task	fetches the named object?	94.5% [90.6,96.8]	13.5% [8.0,21.9]
swap-obey	goes to the newly named object?	200/200	n/a
swap-obey true-tgt	avoids the trained object?	0/200	n/a
occ	needs vision?	0/20	n/a
nonsense	needs a real name?	4.5%	n/a
motor	vanilla pick-place intact?	n/a	100%

appearing as target and as distractor across randomized positions. Evaluation covers all ten scene slots over that object set: at test time the position shortcut the base rides is fully available and earns nothing, which is what the counterfactual isolates. The merged model runs inference on a single 24 GB card.

Main battery. Table 1 is the object-substitution result, all ten slots, n=200. The policy succeeds 94.5% of the time where its base scores 13.5% and the field 0–1%. It obeys a renamed target on 200 of 200 trials and never completes the originally trained pick (0/200); it needs the pixels (blank cameras collapse it) and it needs a real name ("xxx" drops it to 4.5%). What moved is the reading of the instruction: the advance runs from 13.5% to 94.5% while the pick-and-place motor is untouched. The result is not seed-fragile—retrained at seeds 2000 and 3000 the grounding spine replicates, obeying the renamed target 60/60 at seed 2000 with the trained pick never completed.

The position-prior reading is wrong, and the one cliff is a motor wall. The benchmark's own perturbation suite refutes "it memorized where things sit." Displacing the objects 21 cm costs nothing (96.7%), so the policy tracks objects visually. At a 35 cm displacement the score craters to 6.7%, but the reach-probe shows the arm still reaches the correct object every time (reached-target 3/3) and doubling the step budget does not rescue it: the grounding survives the displacement and the low-level grasp fails at the edge of the workspace. The cliff is a motor limit of the base, not a grounding limit of the recipe.

Object identity is not the only shortcut the base hides. The trial exposes three, and the recipe removes each. Task success plus occlusion pass on all three while swap-obey fails on all three—the point of Section 3. On the spatial suite the base scores 99% on task and 50% on swap-obey (chance for two bowls, so the 99% is a canonical-bowl shortcut); the cure raises swap-obey to 86% at a motor cost (task 51%, occlusion-clean). On the goal suite the base scores 98.8% on the canonical scene but 31.2% on held-out scenes with the objects displaced and the instruction fixed, reaching for the remembered position; the cure raises the held-out rate to 79.2% [65.7, 88.3] with occlusion 0/48 and the reach-probe going to the moved object 12/12.

The recipe transfers to a second simulator. On RoboCasa (Nasiriany et al., 2024), a different simulator with a different renderer and object set, the same recipe grounds the object name at swap-obey 44–60% across two seeds against a 20% chance rate, with occlusion 0. The task rate is motor-limited at 14–28%, the same pattern the spatial and goal suites show: the grounding transfers and the motor sets the ceiling.

One adapter grounds six referent types. A single adapter trained across six axes at once—including two (ordinal, relation) beyond the release set—grounds all six (Figure 4; full table in Appendix D): every axis clears swap-obey 77–95% with occlusion 0, so one model reads color, category, ordinal position, spatial relation, negation, and composition. The task rates (26–51%) are the motor ceiling and swap-obey is the pure grounding metric.

Trained types also compose without further training: on a never-trained composition of two trained types—"the second red one from the left"—the adapter gains +55.6 points over its adapter-ablated base on paired scenes (McNemar $p \approx 10^{-5}$).

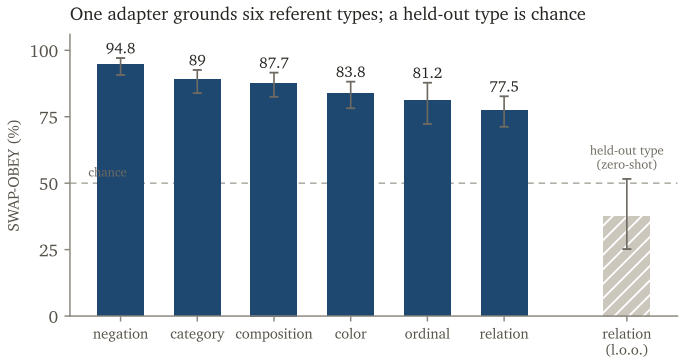

Figure 4: One adapter grounds six referent types (swap-obey 77–95%, occlusion 0 on every axis), and generality stops at its edge: a type held out of training grounds at chance. Intervals are Wilson 95%; the dashed line is chance.

Table 2: The release: seven axes on the shipped merged checkpoint. swap-obey is grounding; task is grounding times motor and is bench-dependent. Spatial's obey count is backed by a 6.2× distance separation (d_{named} median 20 mm vs d_{true} 124 mm), since its two referents are identical bowls.

axis	swap-obey	occ	task
object	99.0% (198/200; true-target 0/200)	0%	80.0%
spatial	obeyed 187 / went-true 0	0%	55.0%
goal (displaced referent)	83.3% (base 31.2%)	0%	n/a
color	88.9%	0%	22.2%
negation	89.6%	0%	16.7%
category	78.0%	2%	20.0%
composition	75.0% (color 87.5 / size 62.5)	0%	2.1%

The release grounds seven axes in one weight. The released artifact folds seven axes into one merged checkpoint, and Table 2 is measured on that shipped weight. Every axis grounds (swap-obey 75–99%) and every axis is vision-gated (occlusion ≤2%). On object identity the release obeys the renamed target 198 of 200 times and never fetches the trained object, matching its adapter, while completion reads 80.0% against the specialist's 94.5%: the gap is motor: the arm reaches the named object on 90.5% of trials and the grasp is what fails, the measured price of carrying seven axes in one set of weights, and grounding pays none of it.

The same deconfounding principle applies to a second output: on the object family, one set of weights carries an action and a prediction of the task-relevant future that both obey the instruction (Appendix B). Appendix F shows the cured and uncured policies side by side in pixels.

Throughout, swap-obey is the grounding measurement and task additionally carries the motor and the difficulty of the bench it was run on, which is why the two columns move independently. The spatial axis shows why the distinction matters: the uncured base scores 99% on task and exactly chance on swap-obey, so reading task alone reverses the conclusion about which model is following the spatial phrase. Composition decomposes cleanly as well—the color attribute follows at 87.5% and size at 62.5%—which localizes the short leg where a pooled number would not.

8 The recipe scales by coverage, and acts through the identified route

Three properties of the result follow from the diagnosis rather than from tuning.

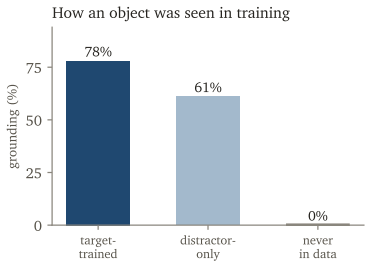

Figure 5: Grounding by how an object was seen in training (n=18 per tier): trained as a target 78%, seen only as a distractor 61%, absent from the data 0%. Appearing in a scene is worth most of what a grasp demonstration buys.

The data is what carries grounding, and that is why the recipe is cheap. No training-time objective on the confounded data reaches the grounding that the deconfounded data reaches (Section 6). The capability is therefore bought with data that a generator produces automatically, at a fixed cost per referent type, on commodity hardware. Nothing in the recipe requires a new architecture, an auxiliary detector, a coupling loss, or a larger pretrain.

An object's grounding costs its appearance more than a demonstration of it. Exposure comes in tiers (Figure 5): an object trained as a target grounds at 78%, an object that appears only as a distractor—never grasped, never named as the goal—grounds at 61%, and an object absent from the data grounds at 0% (n=18 per tier). Merely appearing in a scene is therefore worth most of what a grasp demonstration buys, which matters because appearing in a scene is the cheapest operation the generator performs. Per-object rates inside the distractor-only tier range from 6/6 to 2/6, so we read the gradient as an ordering rather than as a rate.

The cure acts through the route the diagnosis identifies. If instruction blindness is a severed route rather than absent knowledge, the repair should take hold where a route exists to re-open, and that is what we observe across bases. MolmoAct2 interleaves the backbone's tokens with the action head at every layer, exposing the object binding to the action computation, and the recipe reaches 94.5% on it. Two bases that instead condition their action expert on a pooled summary of the backbone—$\pi_{0.5}$ under both adapter placements (n=60 each) and Cosmos 3 Nano's action head (n=12, pilot)—never present that binding to the action head, and hold swap-obey at their base rate under the same recipe. Curability tracks architectural exposure of the binding, which restates the diagnosis: what was missing was the route. The account is post hoc, and a base that exposes the binding per layer yet resists the recipe would falsify it.

9 Limitations

Coverage bounds the vocabulary and the referent types. The recipe grounds the words it has seen. An object withheld from training entirely—absent as target and as distractor—grounds at 0 of 16 while trained-object controls in the same runs hold at 75%, and a referent type withheld from training grounds at 37.5% against 75% when trained, with disjoint intervals. Compositions of trained types transfer (Section 7); an unseen type does not. Open-vocabulary grounding is a different problem, and these results do not address it.

Architecture scope. The recipe is demonstrated on one base, and the architectural account of which bases it applies to (Section 8) is drawn from three observations rather than registered in advance. The recipe trains on a fixed camera and a fixed initial pose, so it buys semantic robustness and not perceptual or initialization robustness.

Evidence is primarily in simulation. The counterfactual results come from two simulators. On hardware we verified the load-bearing control at n=21 reach trials on a low-cost arm

(Appendix C): the vision gating transfers, and the quantitative grounding claims rest on the simulator batteries.

10 Conclusion

A vision–language–action policy can pass an instruction benchmark without reading the instruction. We named that failure mode instruction blindness, showed with a change-one-word counterfactual that it holds across the published field on object identity, and traced it to a specific place in the network: the pretrained backbone resolves the named object, and the action head routes position and never reads that binding. Because the knowledge is already there, the repair is a data recipe rather than an architecture. Making the instruction the only predictor of the target, under a low-rank update that protects what the backbone knows, takes object-identity grounding from 13.5% to 94.5% where full fine-tuning on the same demonstrations destroys it, and one set of weights grounds seven referent-type axes at once. The policy, the deconfounded datasets, the generator that produces them, and the measurement code are released. Grounding, on this axis, was never a matter of scale—it was a severed route, and the data reconnects it.

Acknowledgments

We thank Manifund and Gavin Leech for supporting this work.

References

Ezra Ameperosa, Jeremy A. Collins, Mrinal Jain, and Animesh Garg. RoCoDA: Counterfactual data augmentation for data-efficient robot learning from demonstrations. In ICRA, 2025. arXiv:2411.16959.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report, 2025. arXiv:2502.13923.

Kevin Black, Noah Brown, Danny Driess, et al. π_0: A vision-language-action flow model for general robot control. arXiv preprint arXiv:2410.24164, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In CoRL, 2023. arXiv:2307.15818.

Jun Cen, Siteng Huang, Yuqian Yuan, et al. RynnVLA-002: A unified vision-language-action and world model. arXiv preprint arXiv:2511.17502, 2025a.

Jun Cen, Chaohui Yu, Hangjie Yuan, et al. WorldVLA: Towards autoregressive action world model. arXiv preprint arXiv:2506.21539, 2025b.

Tianxing Chen, Yue Chen, Zixuan Li, Junyuan Tang, Kailun Su, Haoran Lu, et al. Robodojo: A unified sim-and-real benchmark for comprehensive evaluation of generalist robot manipulation policies. arXiv preprint arXiv:2607.04434, 2026.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In ICML, 2025. arXiv:2501.17161.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In NeurIPS, 2019.

Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z. Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, and Sergey Levine. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. arXiv preprint arXiv:2505.23705, 2025.

Haoquan Fang, Jiafei Duan, Donovan Clay, et al. MolmoAct2: Action reasoning models for real-world deployment. arXiv preprint arXiv:2605.02881, 2026a.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in vlas. arXiv preprint arXiv:2602.17659, 2026b.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models, 2025. arXiv:2510.13626.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2(11):665–673, 2020. arXiv:2004.07780.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In ICLR, 2022. arXiv:2106.09685.

Xiaosong Jia, Bowen Yang, Zuhao Ge, 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, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In CoRL, 2024. arXiv:2406.09246.

Dacheng Li, Yunhao Fang, Yukang Chen, et al. WorldModelBench: Judging video generation models as world models. arXiv preprint arXiv:2502.20694, 2025.

Shijie Lian, Bin Yu, Xiaopeng Lin, Laurence T. Yang, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Cong Huang, and Kai Chen. LangForce: Bayesian decomposition of vision language action models via latent action queries. In ICML, 2026. arXiv:2601.15197.

Xiaofeng Lin, Hejian Sang, Zhipeng Wang, and Xuezhou Zhang. Debunk the myth of SFT generalization. arXiv preprint arXiv:2510.00237, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In NeurIPS Datasets and Benchmarks, 2023. arXiv:2306.03310.

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. In NeurIPS, 2025. arXiv:2505.19789.

Xiaokang Liu, Zechen Bai, Hai Ci, Kevin Yuchen Ma, and Mike Zheng Shou. World-VLA-Loop: Closed-loop learning of video world model and vla policy. arXiv preprint arXiv:2602.06508, 2026.

Haoxiang Ma, Junhao Cai, Xiaoxu Xu, Hao Li, Yuyin Yang, Yang Tian, Jiafei Cao, Hongrui Zhu, Zherui Qiu, Zhaxizhuoma, Yuqiang Yang, Jiaqi Peng, Xueyuan Wei, Yangkun Zhu, Jiahao Jiang, Xing Gao, Hanqing Wang, Feng Yuan, Kailin Li, Xueyue Zhu, Tai Wang, Yan Ding, Jiangmiao Pang, Jia Zeng, Jingjing Zhang, Bowen Zhou, Yao Mu, Chunhua Shen, and Weinan Zhang. InternVLA-A1.5: Unifying understanding, latent foresight, and action for compositional generalization. arXiv preprint arXiv:2607.04988, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, et al. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In RSS, 2024. arXiv:2406.02523.

NVIDIA. Cosmos-Predict2. Model and code release, https://github.com/nvidia-cosmos/cosmos-predict2, 2025a.

NVIDIA. GR00T N1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734, 2025b.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. arXiv preprint arXiv:2606.02800, 2026.

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In NeurIPS, 2021.

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. arXiv:2504.16054.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. In NeurIPS, 2020. arXiv:2007.02863.

Yu Shang, Zhuohang Li, Yiding Ma, et al. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. arXiv preprint arXiv:2602.08971, 2026.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics, 2025. arXiv:2506.01844.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Shiang-Feng Tsai, Jin-Cheng Jhang, Yen-Ling Tai, Jia-Hong Lai, Shih-Yun Wong, Tung-Hsu Kang, and Yi-Ting Chen. Direct action-head injection of a grounded 3d point unlocks spatial and task generalization. arXiv preprint arXiv:2606.27663, 2026.

Unitree Robotics. Unifolm-vla-0: A vision-language-action framework under the unifolm family. https://github.com/unitreerobotics/unifolm-vla, 2026.

Fangyuan Wang, Ziyuan Wang, Guorui Pei, et al. World models for robotic manipulation: A survey. arXiv preprint arXiv:2606.00113, 2026a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In ACL, 2023. arXiv:2212.10560.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In ICLR, 2026b. arXiv:2506.19850.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In NeurIPS, 2020. arXiv:2010.14876.

Bin Yu, Yao Zhang, Haishan Liu, Shijie Lian, Yuliang Wei, Xiaopeng Lin, Zhaolong Shen, Changti Wu, Ruina Hu, Bailing Wang, Cong Huang, and Kai Chen. RoboSemanticBench: Diagnosing semantic grounding in action prediction for vla models, 2026. arXiv:2606.02277.

Hang Yu, Juntu Zhao, Yufeng Liu, et al. Point what you mean: Visually grounded instruction policy. arXiv preprint arXiv:2512.18933, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? arXiv preprint arXiv:2504.13837, 2025.

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, 2025. arXiv:2510.03827.

A The conditioning gradient of the predicted future

Grounding of a predicted future tracks how the instruction reaches the prediction, not model scale (Figure 6). A world model whose future is conditioned on the action cannot read the instruction at the frame level—the name reaches the pixels only through a weak action head, and such models hallucinate a successful future even under a wrong action (Liu et al., 2026). A world model whose future is conditioned on the text reads the name directly. We ran the switch counterfactual on a text-conditioned video world model (NVIDIA, 2025a) over held-out frames, hand-scored in pixels under a continuity protocol that rejects composited outcomes: it obeys the swapped object name 64.7% of the time [41.3, 82.7] against a 25% chance rate. The tier replicates at the frontier: Cosmos 3's Nano model (NVIDIA, 2026), two generations newer, lands at 77.8% [61.9, 88.3] on the same protocol. The gradient is one disease along one spectrum: text-conditioned generation grounds (64.7–77.8%), action-conditioned world models do not (20–45%), and imitation policies fail (0–1%).

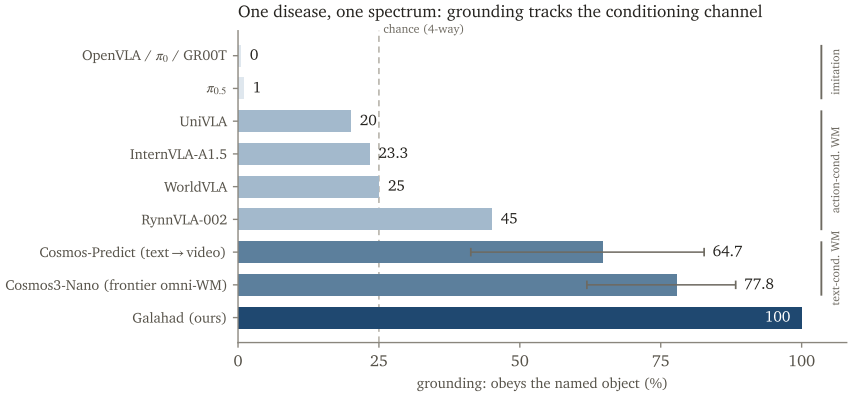

Figure 6: One disease, one spectrum. Grounding of the output tracks how the instruction reaches it: text-conditioned generation reads the name, action-conditioned world models bottleneck it through a weak action head, and imitation policies discard it. Obey rate against a 25% chance rate.

B The principle extends to a predicted future

The deconfounding principle is not specific to the action output. Applying it to a foresight branch that predicts the task-relevant future region yields one set of weights whose action and whose prediction obey the instruction: holding the pixels fixed and changing one word, the predicted region of motion moves to the newly named object 58.3% of the time against a 25% chance rate ($p < 10^{-6}$, n=48), and blanking the cameras removes the effect entirely (0/48), so the prediction reads the scene rather than a name-to-position table. Across the world-action models of Section 4 the predicted future does not follow the instruction at all. The two faces are grounded to different degrees and each is reported on its own measurement (action 100%, prediction 58.3%), and carrying the second face costs the first nothing: trained jointly, the two-branch model ties the action-only control at 93.3% with fully overlapping intervals on a non-saturated bench.

Figure 7 illustrates the two grounded outputs: changing one word moves both the action endpoint and the predicted region of motion to the newly named object.

C Real-robot verification

The battery's load-bearing control transfers to hardware (Figure 8). We ran the same recipe on a low-cost SO-101 arm (commodity servos, one third-person and one wrist camera) and verified the vision-gating claim directly: with the cameras blanked the policy collapses to

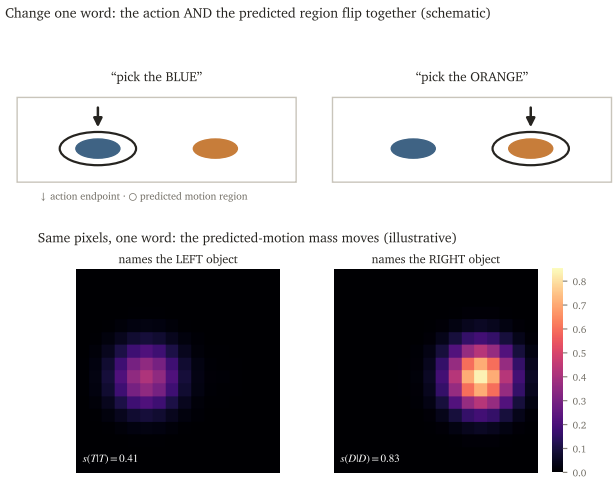

Figure 7: Two faces of one weight. Top: changing one word moves both the action target and the predicted region of motion to the newly named object (schematic). Bottom: the predicted motion mass over two objects, same pixels, one word changed. The panels are illustrative of the measured effect rather than a rendering of measured pixels.

Table 3: The prediction face, object-family weight, n=48. The predicted future obeys the instruction and needs the pixels; the action face is more grounded than the prediction face.

counterfactual	result	chance
switch (one word, predicted region moves to named object)	58.3% ($p < 10^{-6}$)	25%
blind switch (same, cameras blank)	0/48	25%
action face swap-obey (Section 7)	100%	n/a

a single instruction-independent attractor, and with vision restored the reach varies with the named referent. On the physical arm, reach-obey is 10/21 (48% [28, 68] against a 33% chance rate) and grasp-obey is 5/15; the reach-to-grasp gap is attributed by the reach-probe to actuation, the SO-101 being mechanically marginal for the grasp. Adjudication is by pixel co-location in the third-person view, never by joint angles, and runs are gated to those postdating the checkpoint. The hardware evidence establishes the vision gating; the quantitative grounding claims of this paper rest on the simulator batteries (Section 9).

D The referent-type axes in full, and the verb axis

Table 4 is the full per-axis table for the single six-axis adapter of Section 7 (visualized in Figure 4).

Table 4: One adapter, six referent-type axes, $n \geq 96$ per axis. swap-obey is grounding; task is grounding times motor. Leave-one-out: a type held out of training grounds at chance (37.5% vs 75% trained, disjoint intervals).

axis	category	color	ordinal	relation	negation	composition
swap-obey	89.0	83.8	81.2	77.5	94.8	87.7
occ	0	0	0	0	0	0

An eighth referent type grounds the verb. On a deconfounded lift-versus-slide design (the same object under both verbs), the cured policy's verb selectivity is +44.3 points: told to lift, it lifts 46.3% of the time; told to slide, it lifts 2.0% (disjoint intervals); the base's selectivity is +0.0. The lift program collapses to zero with the cameras blanked, and the

Figure 8: The hardware setup: an SO-101 arm with commodity servos over a flat work surface, with three candies as the referent set. Raw phone footage; no fixture, no calibrated lighting, no motion capture.

render shows a real grasp-then-raise against a real push-along-surface. The magnitude is motor-limited (pooled obey 37%, a grasp-stability wall), and the claim is the selectivity: the word chooses which motor program runs. Joint training dilutes this axis (+6 points in the unified weight), so the released checkpoint does not carry it and we report the single-axis result.

E Measurement discipline

Three rules govern every number in this paper. First, a rate is trusted only after its init value is measured: success is lift-gated (the object must leave the table by a fixed margin), so a scene in which the target begins at the goal cannot score a do-nothing success, and a do-nothing policy scores near zero on every metric by construction. Second, a saturated or empty cell (a 100% or a 0%) is treated as a suspect until a disconfirming control rules out the instrument rather than the policy—which is why every grounding rate in this paper is accompanied by an occlusion control and a reach-probe. Third, we verify in the modality the result lives in: trained-policy rollouts are rendered and inspected in pixels before a scalar is believed, because a scalar can describe a broken world.

F A qualitative demonstration

Figure 9 shows the difference the cure makes in pixels. Both panels are the same scene, the same seed, and the same initial state; the instruction is replaced while the arm is already moving, from “pick the milk” to “pick the orange juice and place it in the basket.” The uncured base carries the originally named object to the basket and never revisits its choice. The cured weight abandons its committed approach and completes the newly named task. This is a demonstration rather than a measurement: a mid-execution replacement is not the protocol behind any rate in this paper, all of which are measured from the initial state (Section 3), and the rollout shown is one successful episode rather than a sampled average.

G Referent-type generator designs

One principle—the instruction is the only predictor of the target—instantiates as one small design per type, sharing a scripted-oracle generator and the battery (Table 5).

Figure 9: Same scene, same seed, same initial state, one instruction replaced mid-execution ("milk" → "orange juice"), shown at the same step. Left: the base continues to the originally named object. Right: the cured weight switches to the newly named one and places it in the basket. Labels A and B mark the two referents. A demonstration, not a measurement.

Table 5: One generator, one design per referent type: the confound each breaks and the counterfactual that verifies it.

type	confound broken	swap counterfactual
object identity	position, scene	rename the target object
color	color ⊥ position ⊥ shape	change the color word
category	category ⊥ instance ⊥ co-appearance	change the category word
ordinal	ordinal ⊥ absolute position ⊥ identity	change the ordinal word
spatial relation	side ⊥ identity	name the other referent
negation	keyword-match	flip the negated attribute
composition	single-attribute lookup	change one attribute in the pair
goal	remembered destination	displace objects, hold the goal word

H Compute and release

All models train and run on commodity GPUs. Each Galahad variant trains with 8-way data parallelism on NVIDIA RTX 3090s (24 GB each), and the merged model runs inference on a single 24 GB card; the one exception is the 14B text-to-video world model of Appendix A, which we ran on an A100. We release four artifacts: the policy (the seven-axis merged weight and the object-family dual-output weight, rebuilt from the released recipe and battery-verified), the deconfounded referent-type datasets in a standard format, the generator that produces both the exam and the training set from one principle, and the one-command measurement battery that reports swap-obey, occ, nonsense, and the reach-probe for any policy. Per-slot and per-object tables with Wilson intervals, seed ranges, the protection-capacity and coupling-dose curves, the regularizer-null sweep, and the LIBERO-Plus per-factor table ship with the code alongside the frozen per-slot result artifacts.