PERTURBOT: BREAKING SHORTCUT PRIORS IN VISION-LANGUAGE-ACTION MODELS WITH PERTURBATIVE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

A vision–language–action (VLA) policy can complete complex tasks while ignoring the evidence that should determine its actions. An object held near the wrist camera can displace the instructed target. Language and action show the same pattern: a familiar noun can trigger the operation it was paired with in training even after the verb changes, and a gripper that closed on nothing may lift anyway. We call these dependencies *modality shortcuts*: regularities in successful demonstrations make visual, lexical, or motor cues sufficient to predict expert actions without the task evidence needed for the underlying decision. More demonstrations of the same kind can raise task success while leaving these shortcuts intact. We propose Perturbot, which makes task-relevant evidence easier to use and shortcuts insufficient on their own: it applies task-preserving wrist-view perturbations, enriches instructions with decision-relevant captions, and adds random and failed trajectory segments relabeled with the behavior they contain. It complements scaling by changing what is scaled, and leaves inference unchanged. Moreover, we propose GroundFscore, an offline score that diagnoses how severely a policy relies on modality shortcuts. Task success rate shows whether a policy improves, while GroundFscore reveals whether the policy scales healthily, relying on task evidence rather than shortcuts. Together, Perturbot and GroundFscore provide a training-and-evaluation framework for disentangling VLA decisions from shortcut priors while preserving responsiveness to task-relevant evidence.

1 INTRODUCTION

"Success is a lousy teacher."
— Bill Gates, *The Road Ahead* (1995)

Vision–language–action (VLA) policies can now perform complex manipulation tasks (Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024; Black et al., 2025b), yet a simple change in view can still divert them (Figure 1). An object held near the wrist camera can displace the instructed target; changing "pick up the bowl" to "push the bowl" may still elicit a grasp; and a gripper that closes on nothing may lift anyway. Task success cannot tell whether the policy uses the scene, instruction, and observed outcome to decide what to do, or merely replays an action when a familiar cue appears.

We call such dependencies *modality shortcuts*, a multimodal instance of shortcut learning (Geirhos et al., 2020) closely related to causal confusion in imitation learning (de Haan et al., 2019; Wen et al., 2020). A modality shortcut is not an unimportant input but an easily decoded cue that stands in for the evidence a decision requires. Such cues arise from the regularity of successful demonstrations: the target dominates the wrist view as the expert reaches for it, each object name may be paired with a single operation, and gripper closure is reliably followed by a lift. Behavior cloning rewards agreement with the demonstrated action, not the evidence used to predict it, so nothing in the objective stops the policy from relying on the cue.

This matters for scaling. Larger and more diverse demonstration corpora are central to generalist robot learning (Open X-Embodiment Collaboration et al., 2023; Khazatsky et al., 2024; Lin et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

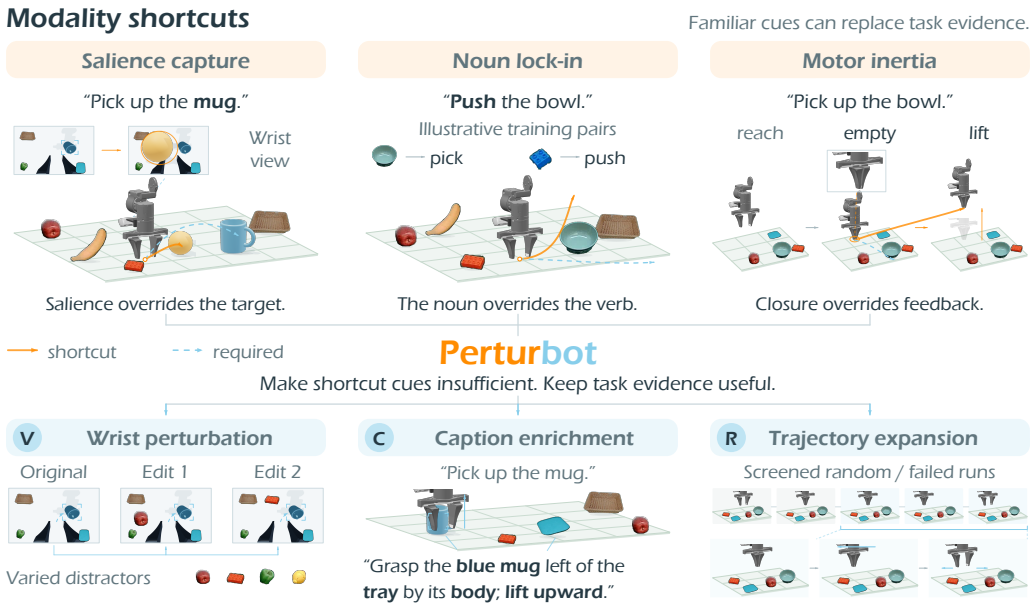


Figure 1: Modality shortcuts and Perturbot. Top: familiar visual, lexical, and motor cues can replace task evidence (solid: hypothetical shortcut choices; dashed: required alternatives). Bottom: task-preserving wrist-view perturbations (V), behavior-supported caption enrichment (C), and screened, relabeled trajectory segments (R; Section 3.2). Examples are conceptual; relabeled local behavior does not by itself establish recovery.

2025), but if new trajectories preserve the same correlations, the shortcut rule and the intended rule keep prescribing the same actions, and more data cannot distinguish them. Higher success on familiar tasks then need not mean more reliable target selection, instruction following, or outcome assessment: the policy grows more proficient where shortcut and evidence agree yet stays brittle where they disagree, which makes modality shortcuts a bottleneck for scaling.

Addressing this requires changing what the training data demand, not weakening an input modality: the evidence a correct decision needs should be easier to use, and shortcuts should be insufficient on their own (Section 3.1). Perturbot does so with three training-time data interventions and no extra model or forward pass at deployment: task-preserving wrist-view perturbations add plausible distractors without changing the target or invalidating the recorded action; decision-relevant captions make object attributes, spatial relations, and requested operations explicit, each supported by the recorded behavior; and caption-grounded trajectory expansion adds random-motion and failed-execution segments relabeled with the behavior they contain, which add behavioral support rather than recovery skills, since relabeling cannot create an unrecorded correction. Perturbot thus changes what is scaled: as data grow, it expands not only repetitions of familiar behavior but also the range of decisions that depend on the scene, instruction, and observed outcome.

Whether scaling achieves this cannot be read from task success: when shortcut and evidence agree, policies relying on either can succeed, and robustness to distractors can come from simply ignoring vision. We therefore propose GroundFscore, an offline score that diagnoses how severely a policy relies on modality shortcuts. It pairs each input with edits that should leave the action unchanged, such as an off-path distractor, and edits that should change it in a known way, such as naming a different object; a high score requires both, so ignoring an input does not help. Because it needs no robot rollouts, GroundFscore can be computed for every checkpoint as data grow: task success shows whether a policy improves, and GroundFscore whether it improves for the right reason.

With a $\pi_{0.5}$ backbone, we evaluate all eight component combinations on 50 real-robot General PnP cases and cover broader skills in RoboTwin 2.0 (Chen et al., 2025b); ablations test whether success and evidence use rise together as data grow and whether GroundFscore predicts online failures. Our contributions are threefold:

1. **We identify *modality shortcuts* in VLA learning**, unifying *salience capture*, *noun lock-in*, and *motor inertia*, and explain why scaling demonstrations that preserve these correlations need not remove them.
2. **We propose Perturbot**, three independently ablatable data interventions that keep action supervision valid, change what is scaled, and leave inference unchanged.
3. **We introduce GroundFscore**, an offline score that, unlike task success, indicates whether a policy scales healthily and cannot be raised by ignoring useful evidence.

2 Related work

Scaling VLA policies. Vision–language–action (VLA) policies map observations and instructions to actions; they are trained on large robot datasets and often initialized from pretrained vision–language models (Brohan et al., 2023; Zitkovich et al., 2023; Kim et al., 2025; Ghosh et al., 2024; Black et al., 2025b;a). Their progress has followed the growth of demonstration data across tasks and embodiments (Open X-Embodiment Collaboration et al., 2023; Khazatsky et al., 2024; Walke et al., 2023) and studies of how data diversity and composition shape transfer (Lin et al., 2025; Hejna et al., 2024; Zha et al., 2025), while evaluation has expanded from nominal simulation benchmarks (Liu et al., 2023; Mees et al., 2022; Li et al., 2024) to perturbed scenes and instructions (Fei et al., 2025; Zhou et al., 2025; Zhao et al., 2026). Success alone cannot show whether a policy improves by using task evidence or by using a shortcut more consistently. GroundFscore measures shortcut reliance offline; unlike action sensitivity to task-preserving perturbations, which has been used to select demonstrations (D'urso et al., 2026), it also requires the action to change when task-relevant evidence changes.

Shortcut learning. Neural networks often rely on features that are predictive in training but fail under distribution shift, as studied under shortcut learning, simplicity bias, and dataset bias (Geirhos et al., 2020; Shah et al., 2020; Torralba & Efros, 2011). Visual question answering models can answer from language priors without grounding the evidence in the image (Goyal et al., 2017; Agrawal et al., 2018; Cadène et al., 2019), and in imitation learning, causal confusion and the copycat problem show that matching expert actions need not identify the evidence that should control behavior (de Haan et al., 2019; Wen et al., 2020; Park et al., 2021). Recent VLA studies find that policies can underuse language or let vision override instructions (Xu et al., 2025; Lian et al., 2026; Fang et al., 2026), and diagnose or regularize how modalities are used (Shi et al., 2026; Xu et al., 2026; Jia et al., 2026; Lin et al., 2026). These accounts usually examine one modality at a time; we tie each shortcut to the decision it affects and trace all three to the regularity of successful demonstrations.

Data interventions for robust policies. Data-centric methods improve robustness by varying appearance through augmentation and domain randomization (Tobin et al., 2017; Laskin et al., 2020; Bharadhwaj et al., 2023), synthesizing or counterfactually editing demonstrations (Mandlekar et al., 2023; Ameperosa et al., 2025), enriching or relabeling instructions (Kim et al., 2026; Glossop et al., 2025), and widening behavioral coverage with corrective, perturbed, or unstructured data (Ross et al., 2011; Laskey et al., 2017; Lynch et al., 2020; Hu et al., 2025). Perturbative training has likewise reduced language-prior hallucinations in vision–language models (Chen et al., 2025a). Each of these interventions is typically designed for one axis of variation, such as appearance, geometry, language, or coverage; Perturbot instead targets the cue that substitutes for evidence in a given decision, while keeping every action label valid and leaving inference unchanged.

3 Method

We first model modality shortcuts (Section 3.1), then describe the three data components of Perturbot (Section 3.2) and how GroundFscore measures the resulting evidence use (Section 3.3).

3.1 Modeling modality shortcuts

We study three failures, one per decision a manipulation policy makes. In *salience capture* (what to manipulate), an object near the wrist camera can divert a reach from the instructed target; in *noun lock-in* (which semantics apply), a policy that has pushed other objects but only picked up the bowl

may grasp it when asked to push it; in *motor inertia* (how to move), a gripper that closed on nothing may lift anyway. In each, a cue that is reliable in successful demonstrations stands in for the required evidence: wrist-view salience for the instruction, the noun for the verb, and the close–lift sequence for visual grasp feedback.

Consider a policy $\pi_\theta(a \mid v, \ell, q)$ mapping camera observations v (wrist views included), an instruction ℓ, and motor context q (proprioception, past actions) to an action or action chunk a. For a task decision d, a shortcut s, and task-relevant evidence e, Bayes' rule under the data distribution p separates a shortcut-based prediction from an evidence update, whose expectation splits again:

$$p(d \mid s, e) = p(d \mid s)\,\frac{p(e \mid d, s)}{p(e \mid s)}, \qquad \mathbb{E}\left[\log \frac{p(d \mid s, e)}{p(d \mid s)}\right] = \mathrm{I}(d; e \mid s) = \mathrm{I}(d; e) - \mathcal{R}, \tag{1}$$

where $\mathrm{I}(\cdot\,;\cdot)$ is mutual information and $\mathcal{R} = \mathrm{I}(d; s) + \mathrm{I}(d; e) - \mathrm{I}(d; s, e)$ is the redundancy, the information about d that s duplicates (nonnegative in the redundant regime we consider). A modality shortcut is the remainder $\mathrm{I}(d; e \mid s)$ vanishing on successful demonstrations, for one of two reasons. Under *collinearity*, $\mathcal{R} \approx \mathrm{I}(d; e)$: in *salience capture* the target fills the wrist view because the operator reaches for it, and in *noun lock-in* a noun always comes with one verb. Under *degeneracy*, the decision never varies, $\mathrm{H}(d) \approx 0$: in *motor inertia* nothing follows gripper closure but a lift. More demonstrations of the same kind preserve both, so scaling alone need not remove the shortcut.

Equation (1) thus suggests two moves with different costs. Lowering $\mathcal{R}$, as a decoy inserted off the grasp path does, makes the shortcut insufficient on its own without changing the expert action, so the labels come free. Raising $\mathrm{I}(d; e)$, as captions naming decision-relevant attributes do, makes the evidence easier to use, but under degeneracy $\mathrm{I}(d; e) \le \mathrm{H}(d)$ stays near zero until a missing branch, such as reopening after an empty grasp, is recorded; relabeling cannot supply it. These are goals for the training data, not a training loss or a guarantee of learned evidence use.

3.2 PERTURBOT

Perturbot trains on two sources: task demonstrations, which supervise real-robot General PnP and, separately, the 50 RoboTwin 2.0 tasks (Section 4), and recorded random-motion and failed-execution trajectories, screened and relabeled as auxiliary data rather than extra tasks. Its three components build training examples from both while keeping every action label valid (Figure 1): V targets the redundancy $\mathcal{R}$, C the evidence term $\mathrm{I}(d; e)$, and R adds recorded segments in which d varies.

The visual component V applies task-preserving wrist-view perturbations T_V: a plausible distractor of varying identity and placement is inserted off the grasp path, without hiding the target or changing contact geometry, so local salience changes while the task, motor context, and recorded action stay valid, and the edit itself cannot become a task cue. An edit that would change the target, collision constraints, or required grasp needs a new action and is rejected. The intended lesson is that a salient object need not be the instructed one, yet wrist evidence should still guide the motion; noise, masking, and camera removal are controls, not substitutes.

The caption component C applies decision-relevant caption enrichment T_C and leaves observations and actions unchanged. A vision–language model splits each recording into atomic skills (grasp, lift, move, put down) and describes each by the acting hand and whether it opens or closes, the operation, the object with just enough attributes to single it out, and its position, initial state, and grasp point (Section E.1): the grasp in *pick up the mug* becomes *the left hand closes and grasps the upright blue mug left of the tray by its body*. The aim is decision-relevant detail, not length; a different verb appears only when the data contain that action, since rewriting a grasp as a push does not create a push demonstration. Both sources mix brief and detailed instructions under the same rules, so caption style does not reveal the source, and the deployed policy takes ordinary instructions.

The trajectory component R applies caption-grounded trajectory expansion T_R to recorded random motion and failed executions. These are divided into segments with synchronized images, motor context, and actions; unusable segments are discarded, and each retained one gets a brief instruction for what it actually does, plus a detailed version from C. A failed attempt to pick up a bowl may contain a valid segment *move the empty gripper left and open it*, whose label describes this local behavior, not the original goal and not a failure that has yet to happen. R thus widens language-conditioned state–action coverage but does not by itself teach recovery, which requires recorded corrective continuations evaluated under the original instruction.

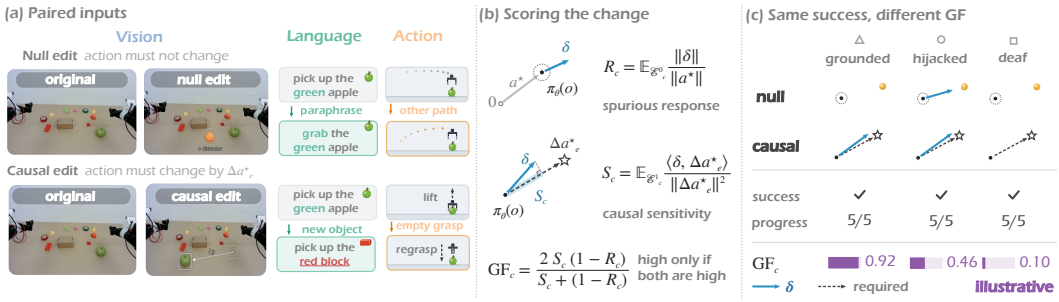


Figure 2: GroundFscore from paired inputs. (a) For each family, a null edit must leave the action unchanged (an off-path decoy, a paraphrase, another approach to the same pose), and a causal edit must change it by a known target $\Delta a^\star_e$ (the target moved by Δp, another object named, a grasp that closed on nothing). (b) R_c normalizes the predicted change δ; S_c projects it onto $\Delta a^\star_e$ and is clipped to $[0, 1]$. (c) Policies with equal success on the demonstration distribution can differ sharply in GroundFscore.

All three components train with the backbone's own action loss $\mathcal{L}_{\text{act}}$ (flow matching for $\pi_{0.5}$). With policy input $o = (v, \ell, q)$, empirical distributions $\mathcal{D}_{\text{demo}}$ of task demonstrations and $\mathcal{D}_{\text{extra}}$ of relabeled auxiliary segments, and an auxiliary fraction $\rho \in [0, 1]$, we train on

$$\begin{aligned} Q &= (1-\rho)\mathcal{D}_{\text{demo}} + \rho\mathcal{D}_{\text{extra}}, \\ \mathcal{L}(\theta) &= \mathbb{E}_{(v,\ell,q,a)\sim Q}\left[\mathcal{L}_{\text{act}}\left(\pi_\theta; \tilde{v}, \tilde{\ell}, q, a\right)\right], \end{aligned} \tag{2}$$

where $\tilde{v}$ is the original or perturbed visual input, $\tilde{\ell}$ is a brief or detailed instruction for the same behavior, and the expectation covers augmentation and caption sampling. Setting $\tilde{v} = v$, $\tilde{\ell} = \ell$, or $\rho = 0$ disables V, C, or R; R without C still uses accurate brief instructions. All rates are fixed within a run and selected on validation data. Because V, C, and R need not act only on the visual, lexical, or motor shortcut family (c_V, c_L, c_A), every configuration is scored on all three (Sections B and F).

3.3 GroundFscore

Task success cannot separate a policy that reads the shortcut from one that reads the evidence, since both succeed on the demonstration distribution. GroundFscore breaks this agreement with paired edits for each shortcut family $c \in \{c_V, c_L, c_A\}$ (Figure 2a): a *null edit* $e \in \mathcal{E}^0_c$ alters the input without changing the correct action, and a *causal edit* $e \in \mathcal{E}^1_c$ changes the correct action by a known amount $\Delta a^\star_e$. With $a^\star$ the reference action at the unedited observation o,

$$R_c = \mathop{\mathbb{E}}_{o,\, e\sim\mathcal{E}^0_c}\left[\frac{\|\pi_\theta(e(o)) - \pi_\theta(o)\|}{\|a^\star\|}\right], \qquad S_c = \mathop{\mathbb{E}}_{o,\, e\sim\mathcal{E}^1_c}\left[\frac{\langle \pi_\theta(e(o)) - \pi_\theta(o),\ \Delta a^\star_e\rangle}{\|\Delta a^\star_e\|^2}\right], \tag{3}$$

where the *spurious response* R_c measures how much the policy moves when it should not, and the *causal sensitivity* S_c is the fraction of the demanded change that its response realizes (1 for exact compliance, 0 for none). Clipping both to $[0, 1]$, we combine them as

$$\mathrm{GF}_c \;=\; \frac{2\,S_c\,(1-R_c)}{S_c + (1-R_c)}, \tag{4}$$

an F-score with precision $1 - R_c$ (not moving when nothing should move) and recall S_c (moving when something should). As in HalFscore (Chen et al., 2025a), the harmonic mean is high only when both terms are: a policy that ignores the input has $R_c = 0$ but also $S_c = 0$, and scores zero.

GroundFscore needs only paired forward passes, not rollouts. We compute it on 100 held-out General PnP observations, each given null and causal edits of all three families: language edits rewrite the instruction, visual edits are made by image editing, and motor edits substitute recorded alternatives. For each causal edit, $\Delta a^\star_e$ is the difference between the action chunks of two expert recordings from the same state, one under the original and one under the edited condition. Actions are full predicted chunks in the normalized action space of $\pi_{0.5}$, and the edited and unedited passes share the flow-matching noise, so sampling noise does not count as a response; $\|a^\star\|$ is floored at a tenth of its median, and causal edits with $\|\Delta a^\star_e\|$ below a fifth of the median are discarded. The score is a diagnostic, not a certificate of task competence; Section 5.4 tests its association with online failures.

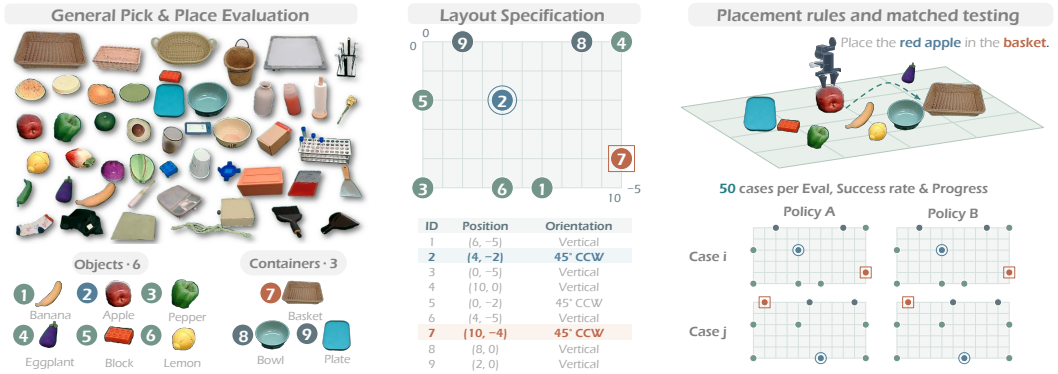


Figure 3: Real-robot General PnP setup. Left: the material pool and the six objects and three containers used in the task. Middle: an example stored layout, specified by each item's position and orientation. Right: an instruction-conditioned scene; every policy is tested on the same 50 stored cases.

Table 1: Main results on real-robot General PnP with a $\pi_{0.5}$ backbone. G1–G5 are success rates (%) on five blocks of ten cases, SR is success over all 50 cases, and Prog. is the mean 0–5 progress score. GF_V, GF_L, and GF_A are GroundFscore readouts on 100 held-out observations (Section 3.3); SC and MI are failure rates (%) on salience-capture and motor-inertia probes of 50 trials each, separate from the 50 cases. $\pm$ is the half-width of the 95% bootstrap interval over the evaluation units: the ten cases of a block, the 50 cases for SR and Prog., the 100 observations for GroundFscore, and the 50 probe trials for SC and MI. V, C, and R denote wrist-view perturbation, detailed captions, and trajectory expansion; the upper block covers all eight combinations and the lower block the controls. Bold marks the best value in each column; the shaded row is the full method.

	Task success ↑							GroundFscore ↑			Failure ↓	
Method	G1	G2	G3	G4	G5	SR	Prog.	GF_V	GF_L	GF_A	SC	MI
$\pi_{0.5}$ fine-tune	40 ±30	40 ±30	50 ±30	40 ±30	40 ±30	42 ±14	2.71 ±0.55	0.28 ±0.04	0.24 ±0.03	0.19 ±0.03	68 ±13	82 ±11
+ V	50 ±30	50 ±30	60 ±30	50 ±30	50 ±30	52 ±14	3.12 ±0.54	0.51 ±0.04	0.26 ±0.03	0.21 ±0.03	36 ±13	80 ±11
+ C	60 ±30	50 ±30	50 ±30	50 ±30	60 ±30	54 ±14	3.20 ±0.53	0.30 ±0.04	0.47 ±0.04	0.22 ±0.03	62 ±14	78 ±11
+ R	50 ±30	60 ±30	50 ±30	60 ±30	60 ±30	56 ±14	3.34 ±0.52	0.29 ±0.04	0.27 ±0.03	0.44 ±0.04	64 ±13	44 ±14
+ V + C	70 ±30	60 ±30	60 ±30	60 ±30	70 ±30	64 ±13	3.62 ±0.50	0.55 ±0.04	0.50 ±0.04	0.24 ±0.03	30 ±13	76 ±12
+ V + R	60 ±30	70 ±30	70 ±30	60 ±30	70 ±30	66 ±13	3.73 ±0.48	0.54 ±0.04	0.28 ±0.04	0.47 ±0.04	34 ±13	40 ±14
+ C + R	70 ±30	70 ±30	60 ±30	70 ±30	70 ±30	68 ±13	3.82 ±0.47	0.31 ±0.04	0.51 ±0.04	0.48 ±0.04	58 ±14	38 ±14
Perturbot (V + C + R)	**90** ±15	**80** ±25	**80** ±25	**80** ±25	**90** ±15	**84** ±10	**4.36** ±0.37	**0.62** ±0.04	**0.57** ±0.04	**0.53** ±0.04	20 ±11	**24** ±12
Noise / dropout	40 ±30	50 ±30	40 ±30	40 ±30	50 ±30	44 ±14	2.80 ±0.55	0.33 ±0.04	0.25 ±0.03	0.20 ±0.03	60 ±14	82 ±11
2× demonstrations	50 ±30	40 ±30	50 ±30	50 ±30	50 ±30	48 ±14	2.95 ±0.55	0.29 ±0.04	0.25 ±0.03	0.20 ±0.03	66 ±13	80 ±11
Negative guidance	50 ±30	40 ±30	50 ±30	40 ±30	50 ±30	46 ±14	2.86 ±0.55	0.41 ±0.04	0.36 ±0.04	0.28 ±0.04	48 ±14	62 ±14
Wrist camera removed	30 ±30	20 ±25	30 ±30	20 ±25	30 ±30	26 ±12	1.94 ±0.54	0.16 ±0.03	0.30 ±0.04	0.18 ±0.03	**12** ±9	88 ±9

4 Experiments

4.1 Protocol

In real-robot General PnP, the robot places the instructed object in the instructed container. Pick-and-place underlies most longer tasks yet is far from trivial here: each layout draws six objects and three containers from 100 items, so combinations rarely repeat and evaluation layouts are unseen in training. Distractors in every scene invite salience capture, instructions require grounding the named object and container, and any grasp can close on nothing, the setting of motor inertia. Short, uniform demonstrations also make data scaling affordable (Section 5.1); RoboTwin 2.0 covers other skills (Section A.3).

The 50 evaluation layouts, generated once by rule-based random placement, are restored before each rollout, so every policy runs the same cases once (Figure 3, Section A.2).

All experiments use $\pi_{0.5}$ (Black et al., 2025a); conditions share initialization, optimizer, training steps, and batch size, so each comparison varies only the training data, with auxiliary examples replacing a fixed fraction of each batch. Recordings and scene groups are split before segmentation, captioning, or augmentation, validation selects the fixed rates, and the primary result uses brief instructions (Section A).

Table 2: RoboTwin 2.0 results (task-averaged success, %). Avg. is the mean of clean and randomized scenes. Published results are as reported by Cai et al. (2026b), rounded to one decimal, and – marks an unreported setting; Table 7 lists every published result, including systems with higher reported scores. The lower block uses our $\pi_{0.5}$ fine-tuning recipe. Bold marks the best value in each column; the shaded row is the full method.

	Full			Clean2Random		
Method	Clean	Rand.	Avg.	Clean	Rand.	Avg.
π_0 (Black et al., 2025b)	65.9	58.4	62.2	–	–	–
$\pi_{0.5}$ (Black et al., 2025a)	82.7	76.8	79.8	70.7	46.0	58.4
ABot-M0 (Yang et al., 2026b)	81.2	80.4	80.8	–	–	–
X-VLA (Zheng et al., 2026)	–	–	–	68.0	20.9	44.5
X-WAM (Guo et al., 2026)	–	–	–	70.0	25.8	47.9
Spatial Forcing (Li et al., 2026a)	–	–	–	77.2	26.7	52.0
4D-WAM (Yang et al., 2026a)	–	–	–	**81.5**	41.8	61.6
Perturbot w/o V	84.9	79.3	82.1	72.6	49.8	61.2
Perturbot w/o C	84.1	80.6	82.4	71.8	55.4	63.6
Perturbot w/o R	84.4	80.9	82.7	72.0	56.2	64.1
Perturbot	**85.6**	**81.9**	**83.8**	73.4	**58.2**	**65.8**

4.2 General Pick and Place

Table 1 covers all eight combinations of V, C, and R, so each component appears alone and as a leave-one-out. An off-path wrist distractor probes salience capture (SC) and a grasp scripted to close on nothing probes motor inertia (MI); verb-change probes for noun lock-in run in RoboTwin 2.0 (Section F.2). The controls test generic regularization (noise and dropout at the V budget), data quantity (twice the demonstrations), discarding evidence (camera removal), and inference-time correction (negative guidance; Section A).

Each component raises success and mainly the GroundFscore of its target family: V lifts GF_V from 0.28 to 0.51 and cuts SC from 68% to 36%, C lifts GF_L from 0.24 to 0.47, and R lifts GF_A from 0.19 to 0.44 and cuts MI from 82% to 44%. The gains compound: every pair beats both of its members, and the full method doubles baseline success to 84% with the best readout in all three families, while removing any one component costs 16–20 points. The other controls stay within 6 points of the baseline, and camera removal attains the lowest SC only by losing 16 points of success and lowering GF_V to 0.16.

4.3 Generality across skills

RoboTwin 2.0 (Chen et al., 2025b) tests whether Perturbot generalizes beyond General PnP. Its 50 bimanual tasks span skills such as handing over a block, opening a microwave, hammering, and hanging a mug, and its domain randomization varies textures, lighting, clutter, tabletop height, and instruction wording. We follow the two settings of Cai et al. (2026b): Full trains on clean and randomized demonstrations, whereas Clean2Random trains only on clean ones, so randomized scenes are unseen; both evaluate on clean and randomized scenes. Clean2Random is the more direct test of modality shortcuts, since randomized clutter adds salient cues that clean training never contradicted. R uses simulator rollouts relabeled as in the real-robot study (Section A.3), and all variants share one budget; published results, from different architectures and recipes, give context only.

Perturbot improves the $\pi_{0.5}$ baseline in every column of Table 2, most where shortcuts should matter: on randomized scenes in Clean2Random it gains 12.2 points, against 2.7 on clean scenes, the best randomized-scene result among the listed systems. Removing V costs most on randomized scenes in both settings (8.4 and 2.6 points), consistent with its role against salient distractors.

5 Ablation Study

We test whether Perturbot's gains require new rather than more data, are mechanism-specific, and preserve control, and whether GroundFscore predicts online failures. Unless noted, ablations use real-robot General PnP with $\pi_{0.5}$ (Figure 4; full tables in Section F).

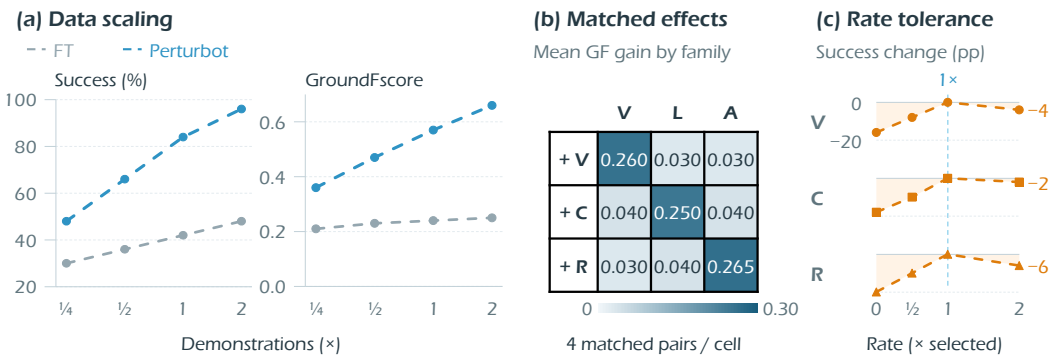


Figure 4: Scaling and ablation effects with a $\pi_{0.5}$ backbone. (a) Success (left) and mean GroundFscore (right) on a log-scaled demonstration budget; Table 3 compares their slopes. (b) Mean GroundFscore gain per family from adding V, C, or R, averaged over four contrasts with the other components fixed (Table 1). (c) Change in success from the selected rate as one sampling rate varies (Table 10); shading shows the loss, not uncertainty.

Table 3: Data scaling at fixed optimizer steps. $\overline{\text{GF}}$: mean GroundFscore; slope: change per doubling; ρ: SR–$\overline{\text{GF}}$ correlation over 4 budgets × 3 seeds.

	FT				Perturbot			
Demos	SR↑	$\overline{\text{GF}}$↑	SC↓	MI↓	SR↑	$\overline{\text{GF}}$↑	SC↓	MI↓
1/4×	30	0.21	74	86	48	0.36	38	46
1/2×	36	0.23	70	84	66	0.47	28	34
1×	42	0.24	68	82	84	0.57	20	24
2×	48	0.25	66	80	96	0.66	14	18
Slope	+6.0	+0.013	−2.6	−2.0	**+16.2**	**+0.100**	**−8.0**	**−9.4**
ρ		0.41				**0.93**		

Table 4: Budget-matched controls. Each control matches the budget of the component below it but removes its mechanism.

Condition	SR↑	GF_V↑	GF_L↑	GF_A↑	SC↓	MI↓
FT	42	0.28	0.24	0.19	68	82
Generic aug.	44	0.33	0.25	0.20	60	82
Random decoys	40	0.38	0.24	0.19	50	82
+ V	52	**0.51**	0.26	0.21	**36**	80
Paraphrases	44	0.28	0.28	0.19	66	82
Length-matched	44	0.29	0.30	0.20	66	80
+ C	54	0.30	**0.47**	0.22	62	78
Success demos	46	0.29	0.25	0.20	66	80
Random only	50	0.29	0.26	0.34	66	60
Failed only	52	0.29	0.26	0.39	64	52
+ R	56	0.29	0.27	**0.44**	64	**44**

5.1 Data Scaling

The default remedy for a brittle policy is more demonstrations, yet Section 3.1 predicts that same-kind ones preserve the collinearity and degeneracy behind shortcuts, so scaling tests whether the data must change. We train the $\pi_{0.5}$ baseline and Perturbot on nested subsets of 125, 250, 500 (1×), and all 1,000 General PnP demonstrations at fixed optimizer steps, three seeds each, and report each curve's least-squares slope per doubling (Table 3).

Perturbot scales faster, gaining 16.2 points of success per doubling against 6.0 for fine-tuning; at 1/4× it already matches fine-tuning at 2×, with eight times fewer demonstrations. The gains also differ in kind: from 1/4× to 2×, fine-tuning adds 18 points while $\overline{\text{GF}}$ moves by only 0.04 and both failure rates stay high; extra demonstrations add proficiency but not reliance on task evidence, the unhealthy regime GroundFscore is meant to expose. Under Perturbot, success and $\overline{\text{GF}}$ rise together, by 48 points and 0.30, and both failure rates fall at every budget. Success and $\overline{\text{GF}}$ correlate at Spearman $\rho = 0.93$ for Perturbot but 0.41 for fine-tuning, whose $\overline{\text{GF}}$ barely varies, so seed noise dominates its ranking.

5.2 Component Specificity

Every component adds training data, so its gain could come from more samples or generic regularization rather than from the shortcut it targets (Equation (1)). Each is therefore compared with controls that match its budget but remove its mechanism (Table 4): for V, noise, cropping, and dropout, or the same distractors placed anywhere in the wrist view, including on the grasp path; for C, paraphrases or length-matched text that does not bear on the decision; for R, equal-duration successful demonstrations or either auxiliary source alone. Only the mechanism moves the targeted readout. V reaches a GF_V of 0.51 against at most 0.38, and random placement drops success below

Table 5: Controllability on real-robot General PnP (%). Target-selection probes change which object is instructed; grasp-outcome probes compare a secured object with an empty grasp. Each probe has 50 trials per policy, separate from the 50 cases.

	Target selected		Lift executed	
Condition	instructed ↑	distractor ↓	secured ↑	empty ↓
$\pi_{0.5}$ fine-tune	88	68	94	82
No wrist camera	48	12	90	88
Perturbot	**90**	20	92	**24**

Table 6: Spearman ρ with the negative online failure rate over 36 checkpoints; intervals in Table 13. †Ranks the no-wrist policy above the baseline.

Predictor	ρ ↑
Task success	0.31
Stability $(1 - R)^{\dagger}$	0.44
Responsiveness (S)	0.38
GroundFscore	**0.72**

the baseline, as on-path decoys invalidate the recorded action. Paraphrases and length-matched text raise GF_L by at most 0.06, against 0.23 for C. Extra successful demonstrations leave MI near the baseline, whereas random and failed segments each lower it and together reach 44%.

5.3 Controllability

A policy can pass shortcut probes by ignoring an input, as removing the wrist camera does for salience capture (Table 1), but such robustness discards evidence the task needs. Table 5 therefore pairs changes that should not alter the action with changes that should: a distractor should not redirect a reach, yet the same object should be selected when it is instructed, and lifting should depend on the observed grasp outcome. Perturbot picks the distractor in 20% against 68% for fine-tuning, yet still selects the instructed object in 90%; without a wrist camera, a policy avoids the distractor more often but finds the instructed object in only 48%. Perturbot also lifts secured objects nearly as often as fine-tuning (92% vs. 94%) but lifts after empty grasps in 24% rather than 82%.

5.4 Validity of GroundFscore

GroundFscore is meant to replace costly shortcut-targeted rollouts with offline forward passes, which is only useful if it predicts what those rollouts reveal. Across the twelve conditions in Table 1 and three seeds, we rank-correlate GroundFscore on the 100 paired observations of Section 3.3 with the negative online failure rate on shortcut-targeted trials, and ablate the score itself: stability alone $(1 - R)$ rewards ignoring an input, and responsiveness alone (S) rewards moving under any change. GroundFscore predicts online failures best ($\rho = 0.72$), and its advantage of 0.41 over task success has a 95% cluster-bootstrap interval of [0.12, 0.68] (Table 13). Neither term suffices alone: stability reaches 0.44 but ranks the no-wrist policy above the baseline, and responsiveness reaches only 0.38.

6 Limitations

Perturbot needs detailed captions and relabeled segments annotated to match the recorded behavior, and random and failed trajectories in addition to demonstrations; relabeling cannot create behaviors that were never recorded, such as recovery. Real-robot rollouts offer a ready source of such trajectories: successful trajectories tend to resemble one another, while each failure goes wrong in its own way.

7 Conclusion

VLA policies can succeed on familiar tasks while relying on modality shortcuts: visual, lexical, or motor cues that successful demonstrations make sufficient to predict the expert's action, and that more demonstrations of the same kind keep predictive. Perturbot changes what the training data demand through task-preserving wrist-view perturbations, decision-relevant captions, and relabeled random and failed segments, without changing inference, and GroundFscore complements task success by diagnosing, offline, whether a policy improves through task evidence or through shortcuts. Together, they point to a different aim for scaling robot data: expanding not only repetitions of familiar behavior but also the range of decisions that depend on the scene, instruction, and observed outcome.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing for readability, including rephrasing sentences and reorganizing sentence order and logical flow. All AI-assisted text was reviewed and verified by the authors, who take full responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

We will open-source the data, the trained checkpoints, and the models upon publication. The main modeling relation is Equation (1), and the GroundFscore equations and probes are specified in Section 3.3. Section A describes the real-robot platform and evaluation protocols, Section B the training data and its collection, Section C the $\pi_{0.5}$ configuration, Section D the wrist-view perturbation, and Section E.1 the prompt used to generate detailed captions; the common action-training objective is Equation (2).

REFERENCES

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4971–4980, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Agrawal_Dont_Just_Assume_CVPR_2018_paper.html.

Ezra Ameperosa, Jeremy A. Collins, Mrinal Jain, and Animesh Garg. RoCoDA: Counterfactual Data Augmentation for Data-Efficient Robot Learning from Demonstrations. In *IEEE International Conference on Robotics and Automation*, 2025. URL https://arxiv.org/abs/2411.16959.

Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. RoboAgent: Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking, 2023. URL https://arxiv.org/abs/2309.01918.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, Hongyan Zhao, Hanyu Liu, Zhizhong Su, Lei Ma, Hang Su, and Jun Zhu. Motus: A Unified Latent Action World Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Bi_Motus_A_Unified_Latent_Action_World_Model_CVPR_2026_paper.html.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. In *Proceedings of the 9th Conference on Robot Learning (CoRL)*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 2025a.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems XXI*, volume 21, 2025b. doi: 10.15607/RSS.2025.XXI.010. URL https://www.roboticsproceedings.org/rss21/p010.html.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S Ryoo, Grecia Salazar, Pannag R Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan H Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: Robotics Transformer for Real-World Control at Scale. In *Robotics: Science and Systems XIX*, volume 19, 2023. doi: 10.15607/RSS.2023.XIX.025. URL https://www.roboticsproceedings.org/rss19/p025.html.

Rémi Cadène, Corentin Dancette, Hédi Ben-Younes, Matthieu Cord, and Devi Parikh. RUBi: Reducing unimodal biases for visual question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 839–850, 2019.

Jisong Cai, Long Ling, Shiwei Chu, Zhongshan Liu, Jiayue Kang, Zhixuan Liang, Wenjie Xu, Yinan Mao, Weinan Zhang, Xiaokang Yang, Ru Ying, Ran Zheng, and Yao Mu. AHA-WAM:Asynchronous Horizon-Adaptive World-Action Modeling with Observation-Guided Context Routing, 2026a. URL https://arxiv.org/abs/2606.09811.

Jisong Cai, Yao Mu, Ganlin Yang, Zhe Cao, Zhangzheng Tu, Xing Gao, Kailin Li, Xinyu Zhan, Lixin Yang, Yangkun Zhu, Haoxiang Ma, Ming Zhou, Qiaojun Yu, Yufei Xue, Liqun He, Yifei Yao, Yifan Zhu, Long Ling, Bingqi Jiang, Haoyu Guo, Xueyue Zhu, Bowen Zhou, Bin Zhao, Tianfan Xue, Chunhua Shen, and Weinan Zhang. InternW0: A Foundational Physical World Model for Efficient Real-World Interactions, 2026b. URL https://arxiv.org/abs/2609.27656.

Cong Chen, Mingyu Liu, Chenchen Jing, Yizhou Zhou, Fengyun Rao, Hao Chen, Bo Zhang, and Chunhua Shen. PerturboLLaVA: Reducing Multimodal Hallucinations with Perturbative Visual Training. In *International Conference on Learning Representations*, 2025a. URL https://iclr.cc/virtual/2025/poster/28657.

Ronghan Chen, Yandan Yang, Zuojin Tang, Dongjie Huo, Tong Lin, Haoning Wu, Haoyun Liu, Yuzhi Chen, Lulu Zheng, Botai Yuan, Tianlun Li, Mingxin Wang, Dekang Qi, Bin Hu, Wei Mei, Yuze Xuan, Haolong Yang, Yanqing Zhu, Mu Xu, Zhiheng Ma, and Xinyuan Chang. ABot-M0.5: Unified Mobility-and-Manipulation World Action Model, 2026. URL https://arxiv.org/abs/2607.00678.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, Huan-ang Gao, Kaixuan Wang, Zhixuan Liang, Yusen Qin, Xiaokang Yang, Ping Luo, and Yao Mu. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b. URL https://arxiv.org/abs/2506.18088.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal Confusion in Imitation Learning. In *Advances in Neural Information Processing Systems*, pp. 11693–11704, 2019. URL https://arxiv.org/abs/1905.11979.

Giovanni D'urso, Kaushik Roy, Nicholas Lawrance, and Brendan Tidd. It's Not Just More Demos: Counterfactual Action Sensitivity Coverage for Data-Efficient Robust Robot Imitation. In *RSS 2026 Workshop on Data-Centric Robotics: What Data Do Robots Really Need?*, 2026. URL https://arxiv.org/abs/2607.27261.

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, 2026. URL https://arxiv.org/abs/2602.17659.

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. URL https://arxiv.org/abs/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. doi: 10.1038/s42256-020-00257-z. URL https://www.nature.com/articles/s42256-020-00257-z.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Robotics: Science and Systems XX*, volume 20, 2024. doi: 10.15607/RSS.2024.XX.090. URL https://www.roboticsproceedings.org/rss20/p090.html.

GigaBrain Team, Angen Ye, Axiang Sun, Can Jin, Chenxi Cheng, Chong Shi, Dengke Shang, Dingqian Zhang, Guan Huang, Guangqiang Wang, Guangqing Ding, Guo Li, Hangcong Li, Hengyu Zhong, Hongtao Lu, Jianbo Qin, Jiming Mao, Jing Zhu, Jindi Lv, Jingzhi Cui, Junjie Xie, Junyi Bao, Kai Liu, Lei Yuan, Limin Long, Lv Feng, Mingming Yu, Peng Li, Pengfei Yi, Qi Li, Qianli Zhang, Qingfang Li, Qitang Hu, Rui Zhang, Shaoyan Sun, Shibo Sun, Shiying Duan, Tenghui Chen, Tianze Liu, Weijie Ke, Wenyao Xue, Xiaofeng Wang, Xiaoyu Tian, Xinyu Liu, Xinze Chen, Yang Wang, Yankai Wang, Yejun Zeng, Yifan Li, Yifei Nie, Yilong Li, Yilong Liu, Yongchao Feng, Yumeng Wang, Yun Ye, Zhichao Liu, Ziheng He, Zonghai Yang, and Zheng

Zhu. GigaBrain-0.7: Scaling Embodied Foundation Models to Emergent Capabilities with a Three-System Architecture, 2026. URL https://arxiv.org/abs/2608.15875.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. CAST: Counterfactual Labels Improve Instruction Following in Vision-Language-Action Models, 2025. URL https://arxiv.org/abs/2508.13446.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6325–6334, 2017. doi: 10.1109/CVPR.2017.670.

Jun Guo, Qiwei Li, Peiyan Li, Zilong Chen, Nan Sun, Yifei Su, Heyun Wang, Yuan Zhang, Xinghang Li, and Huaping Liu. Unified 4D World Action Modeling from Video Priors with Asynchronous Denoising, 2026. URL https://arxiv.org/abs/2604.26694.

Joey Hejna, Chethan Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Re-Mix: Optimizing Data Mixtures for Large Scale Imitation Learning, 2024. URL https://arxiv.org/abs/2408.14037.

Zheyuan Hu, Robyn Wu, Naveen Enock, Jasmine Li, Riya Kadakia, Zackory Erickson, and Aviral Kumar. RaC: Robot Learning for Long-Horizon Tasks by Scaling Recovery and Correction, 2025. URL https://arxiv.org/abs/2509.07953.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, Qingwen Bu, Haidong Cao, Chao Wu, Qifeng Li, Zhenjie Yang, Chenhe Zhang, Hongyang Li, Zuxuan Wu, Junchi Yan, and Yu-Gang Jiang. GuidedVLA: Specifying Task-Relevant Factors via Plug-and-Play Action Attention Specialization. In *Robotics: Science and Systems XXII*, volume 22, 2026. URL https://www.roboticsproceedings.org/rss22/p084.html.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.120.

Bosung Kim, Ruiyi Wang, David Acuna, Jaehun Jung, Alexander Trevithick, Brandon Cui, Yejin Choi, and Prithviraj Ammanabrolu. How to Instruct Your Robot: Dense Language Annotations Power Robot Policy Learning, 2026. URL https://arxiv.org/abs/2605.17077.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R 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 *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise Injection for Robust Imitation Learning. In *Conference on Robot Learning*, pp. 143–156, 2017. URL https://proceedings.mlr.press/v78/laskey17a.html.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial Forcing: Implicit Spatial Representation Alignment for Vision-language-action Model. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=euMVC1DO4k.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control, 2026b. URL https://arxiv.org/abs/2601.21998.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, pp. 3705–3728. PMLR, 2024.

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 *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65457.

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data Scaling Laws in Imitation Learning for Robotic Manipulation. In *International Conference on Learning Representations*, 2025. URL https://iclr.cc/virtual/2025/poster/28305.

Jianman Lin, Shailesh Shailesh, Zhongyi Luo, and Jiafei Duan. Breaking the Vision-Action Shortcut: Latent Interface Training for Generalizable Robotics Foundation Models, 2026. URL https://arxiv.org/abs/2609.12641.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning Latent Plans from Play. In *Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 2020. URL https://proceedings.mlr.press/v100/lynch20a.html.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. In *Conference on Robot Learning*, pp. 1820–1864, 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.html.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Open X-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne

Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic Learning Datasets and RT-X Models, 2023. URL https://arxiv.org/abs/2310.08864.

Jongjin Park, Younggyo Seo, Chang Liu, Li Zhao, Tao Qin, Jinwoo Shin, and Tie-Yan Liu. Object-aware regularization for addressing causal confusion in imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3029–3042, 2021.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Haoyuan Shi, Xiancong Ren, Yingji Zhang, Qinfan Zhang, Jiayu Hu, Haozhe Shan, Han Dong, Jinpeng Lu, Yinda Chen, Yi Zhang, Yong Dai, and Xiaozhu Ju. VLA-Trace: Diagnosing Vision-Language-Action Models through Representation and Behavior Tracing, 2026. URL https://arxiv.org/abs/2605.30117.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133. URL https://arxiv.org/abs/1703.06907.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1528, 2011. doi: 10.1109/CVPR.2011.5995347.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale. In *Proceedings of the 7th Conference on Robot Learning (CoRL)*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1723–1736. PMLR, 2023.

Yuran Wang, Siqiao Huang, Mingleyang Li, Chenhao Zhang, Jiaqi Liang, Weiyang Jin, Yue Chen, Xuemin Chi, Donghao Zhou, Qize Yu, Yu-Kai Wang, Yuhan Rui, Shenzhe Yao, Zhen Yuan, Zhenhao Shen, Kefei Zhu, Zijie Zhu, Ning Gao, Xiaowei Chi, Guanqi He, Shanghang Zhang, Hao Dong, Lin Shao, and Hang Zhao. OpenWAM: An Open, Modular Exploration Towards Systematic World-Action Model Pretraining, 2026. URL https://arxiv.org/abs/2609.07398.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting Copycat Agents in Behavioral Cloning from Observation Histories. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2010.14876.

Haiweng Xu, Sipeng Zheng, Hao Luo, Wanpeng Zhang, Ziheng Xi, and Zongqing Lu. Unmasking the Illusion of Embodied Reasoning in Vision-Language-Action Models, 2026. URL https://arxiv.org/abs/2604.18000.

Kechun Xu, Zhenjie Zhu, Anzhe Chen, Shuqi Zhao, Qing Huang, Yifei Yang, Haojian Lu, Rong Xiong, Masayoshi Tomizuka, and Yue Wang. Seeing to Act, Prompting to Specify: A Bayesian Factorization of Vision Language Action Policy, 2025. URL https://arxiv.org/abs/2512.11218.

Lishan Yang, Wenxuan Song, Xi Wang, Pingyue Sheng, Zheng Fang, Ziyang Zhou, Junjie He, Haodong Yan, Jiayi Chen, Nan Sun, Qiao Sun, Pengwei Wang, Lingqiao Liu, Yan Wang, Yuxiang Gao, Feras Dayoub, and Haoang Li. 4D-WAM: Infusing Spatiotemporal Awareness into World Action Models through Trajectory Fields, 2026a. URL https://arxiv.org/abs/2608.08023.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. ABot-M0: VLA Foundation Model for Robotic Manipulation with Action Manifold Learning, 2026b. URL https://arxiv.org/abs/2602.11236.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do World Action Models Need Test-time Future Imagination?, 2026. URL https://arxiv.org/abs/2603.16666.

Lihan Zha, Apurva Badithela, Michael Zhang, Justin Lidard, Jeremy Bao, Emily Zhou, David Snyder, Allen Z. Ren, Dhruv Shah, and Anirudha Majumdar. Guiding Data Collection via Factored Scaling Curves, 2025. URL https://arxiv.org/abs/2505.07728.

Mengao Zhao, Ziang Li, Chaodong Huang, Mengchen Ma, Haoyi Jiang, Yiwei Jin, Xinjie Wang, Yun Du, Xuewu Lin, Taojun Ding, Hongyu Xie, Jackson Jiang, Chunlei Yu, Kaihua Zhang, Lichao Huang, Liu Liu, Tianwei Lin, and Zhizhong Su. InstructMove: A Text-Indispensable Benchmark for Instruction-Following Manipulation, 2026. URL https://arxiv.org/abs/2608.22990.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Tai Wang, Ya-Qin Zhang, Jingjing Liu, and Xianyuan Zhan. X-VLA: Soft-Prompted Transformer as Scalable Cross-Embodiment Vision-Language-Action Model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

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.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

APPENDIX

This appendix describes the experimental setup and evaluation protocols (Section A), the training data (Section B), the $\pi_{0.5}$ training configuration (Section C), the wrist-view perturbation (Section D), the detailed-caption annotation (Section E), and additional ablation and probe results (Section F).

A EXPERIMENTAL SETUP

A.1 REAL-ROBOT PLATFORM

All real-robot data (task demonstrations, random-motion and failed-execution recordings, and evaluation rollouts) are collected on the dual-arm AgileX PiPER X platform in Figure 5. Two 6-DoF arms, each with a two-finger parallel gripper, are mounted side by side facing the tabletop. A USB head camera on a vertical post behind the arms has a 110° field of view, and a USB wrist camera on each arm has a 160° field of view; all three record at 30 FPS and 640×360 pixels. The head camera supplies the third-person view v^{third} and the wrist cameras supply v^{wrist}; the three views are resized to the policy input resolution (Table 9) and form the visual input v. The motor context q is the proprioceptive state of both arms, and the action a commands both arms and their grippers. Demonstrations and auxiliary recordings are collected by teleoperation.

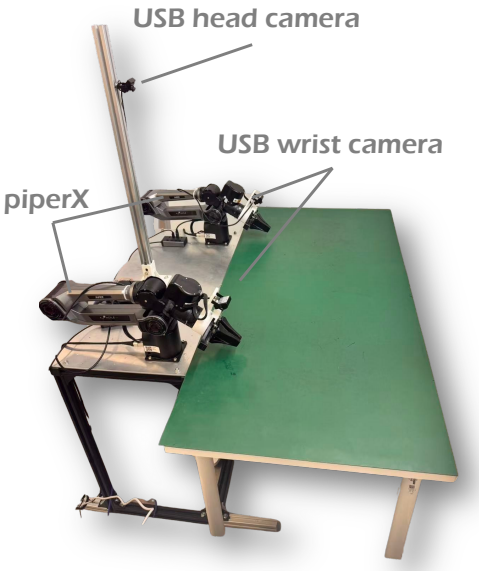


Figure 5: Real-robot platform. Two AgileX PiPER X arms are mounted side by side and face the tabletop workspace. A USB head camera on a vertical post behind the arms provides the third-person view, and a USB wrist camera beside each gripper provides a wrist view.

A.2 GENERAL PNP EVALUATION

The real-robot study evaluates only General PnP: six objects and three containers on the tabletop, with instructions such as *place the red apple in the basket* (Figure 3). Placement rules are fixed before evaluation. Each of the 50 cases specifies the object identities, target, destination, and instruction; object and container positions are drawn at random in a common tabletop frame, and each object's orientation follows its placement rule. Conflicts are resolved by fixed rules (a distractor overlapping the target is moved rather than the target, and an object–container conflict moves the container), and the adjusted coordinates are stored. The 50 layouts are generated once and restored before every rollout, so all policies and training seeds face the same cases.

Each checkpoint executes every case once. We report success over the 50 cases, the success of the five blocks G1–G5 (cases 1–10, ..., 41–50), and the mean 0–5 progress score. The primary evaluation uses brief instructions; a second pass replays the same 50 cases with detailed instructions (Section F.3). The salience-capture, motor-inertia, target-selection, and grasp-outcome probes use separate trial lists and add no physical-robot task; each list has 50 trials per policy. The intervals in

Table 7: All published RoboTwin 2.0 results reported by Cai et al. (2026b) (task-averaged success, %), rounded to one decimal; – marks an unreported setting. They come from different architectures and training recipes; Table 2 gives the matched $\pi_{0.5}$ comparison.

	Full			Clean2Random		
Method	Clean	Rand.	Avg.	Clean	Rand.	Avg.
π_0 (Black et al., 2025b)	65.9	58.4	62.2	–	–	–
$\pi_{0.5}$ (Black et al., 2025a)	82.7	76.8	79.8	70.7	46.0	58.4
ABot-M0 (Yang et al., 2026b)	81.2	80.4	80.8	–	–	–
Motus (Bi et al., 2026)	88.7	87.0	87.8	–	–	–
Fast-WAM (Yuan et al., 2026)	91.9	91.8	91.8	77.8	1.9	39.9
LingBot-VA (Li et al., 2026b)	92.9	91.5	92.2	–	–	–
AHA-WAM (Cai et al., 2026a)	93.4	92.2	92.8	–	–	–
OpenWAM-α (Wang et al., 2026)	93.7	93.5	93.6	–	–	–
ABot-M0.5 (Chen et al., 2026)	94.0	94.2	94.1	–	–	–
X-VLA (Zheng et al., 2026)	–	–	–	68.0	20.9	44.5
X-WAM (Guo et al., 2026)	–	–	–	70.0	25.8	47.9
Spatial Forcing (Li et al., 2026a)	–	–	–	77.2	26.7	52.0
4D-WAM (Yang et al., 2026a)	–	–	–	81.5	41.8	61.6
GigaBrain-0.7 (GigaBrain Team et al., 2026)	–	–	–	66.8	67.9	67.3
InternW0 (Cai et al., 2026b)	93.2	93.0	93.1	83.2	68.0	75.6

Table 1 are 95% percentile bootstrap intervals from 10,000 resamples of the evaluation units of one checkpoint (cases, probe trials, or GroundFscore observations), reported as half-widths.

A.3 RoboTwin 2.0

RoboTwin 2.0 is evaluated on its 50-task suite in the two settings of Cai et al. (2026b). Full trains on the official `demo_clean` and `demo_randomized` demonstrations, whereas Clean2Random trains only on `demo_clean`; both evaluate on held-out clean and randomized trials, so Clean2Random tests unseen randomized scenes. Within a setting, all methods share the simulator randomization, demonstration pool, initial-state seeds, and trial budget, and the simulator's domain randomization is not counted as V. Verb-change probes for noun lock-in use only task pairs that admit both operations from matched states with valid demonstrations (Section F.2). Table 7 lists all published results reported by Cai et al. (2026b).

Auxiliary trajectories. In simulation, R uses rollouts instead of teleoperated recordings. Random-motion trajectories start from a task's initial state and execute scripted, randomly sampled arm motions that sweep the workspace, as in the real-robot protocol. Failed trajectories come from the simulator's scripted expert with injected perturbations that make a grasp or placement go wrong, such as offset grasp poses, early gripper release, or shifted placement targets; only rollouts that the success check marks as failed are kept, so they contain the near misses, empty grasps, and misplacements that successful demonstrations lack. Because the simulator records exact states and actions, both kinds are segmented, captioned, and relabeled with the same rules as the real recordings (Section B). In Clean2Random, rollouts use clean scenes only, so no randomized scene enters training.

A.4 Baselines and controls

All controls in Table 1 use the $\pi_{0.5}$ recipe of Section C. Noise/dropout applies generic visual augmentation at the rate of V, $2\times$ demonstrations trains on all 1,000 demonstrations instead of the 500 of $1\times$ at matched optimizer steps, and wrist-camera removal drops both wrist views. Negative guidance combines the fine-tuned policy with three crippled experts trained with the same recipe: wrist-only (both wrist views and proprioception), proprioception-only, and instruction-blind (all three views with a constant instruction token). At inference, $\log \pi_{\text{guided}}(a \mid o) = \log \pi_\theta(a \mid o) - \sum_k \alpha_k \log \pi^k_{\text{bias}}(a \mid o)$ is applied to the predicted velocity field of $\pi_{0.5}$, with each α_k tuned on validation data over $\{0.1, 0.2, \ldots, 1.0\}$; this costs three extra forward passes per control step.

Table 8: Real-robot training data, all collected on the General PnP setup. Main experiments use a fixed subset of 500 demonstrations ($1\times$) and all auxiliary recordings; the $2\times$ budget uses all 1,000 demonstrations.

Source	Collection	Recordings
General PnP demonstrations	teleoperated task executions	1,000
Random motion	free arm motion among the objects	200
Failed execution	deliberate contact attempts and failures	200
Total		1,400

B TRAINING DATA

Table 8 summarizes the real-robot training data. All of it is collected on the General PnP setup: task demonstrations supervise the task, and the auxiliary recordings supply R, never complete the task, and are not evaluation cases.

B.1 GENERAL PNP DEMONSTRATIONS

We collect 1,000 teleoperated demonstrations of the task, each with its brief instruction. All main conditions train on a fixed subset of 500 ($1\times$); the $1/4\times$ and $1/2\times$ budgets are nested subsets of it, the $2\times$ budget and the $2\times$ demonstrations control use all 1,000, and the equal-duration successful demonstrations of the R control come from the 500 outside the $1\times$ subset. Items come from a pool of 100 objects and containers (Figure 3); each layout places nine of them, six objects and three containers as in evaluation, at random positions, and the 50 evaluation layouts are never used for training. Recordings and scene groups are split into training, validation, and evaluation before any segmentation, captioning, or augmentation, and all derivatives of a recording stay in its split.

B.2 RANDOM-MOTION AND FAILED-EXECUTION RECORDINGS

The auxiliary recordings are collected to produce varied contacts and contact failures rather than completed tasks: the arm approaches objects, touches them, and occasionally touches the wrong one, pushes in the wrong direction, or grasps nothing. Each recording lasts 5–10 s and contains three to six distinct motions, without long pauses and without completing a task. Each uses a layout of nine items from the same pool as the demonstrations, with a randomly varied tablecloth color.

Random motion (200 recordings). With the items of a layout randomly placed, the operator moves the arm through three to six consecutive motions (left–right, forward–backward, up–down, wrist rotation, and gripper opening or closing), covering the workspace without idling, reaching its edge, or repeating one motion, and then resets. Each recording must show clear motion with changes of direction and height. These recordings expose how commanded actions move the robot.

Failed execution (200 recordings). Each recording brings the gripper within 2–5 cm above or 1–3 cm beside an object, makes one contact attempt (pressing down, pushing horizontally, or closing on part of the object), follows with one variation (pushing again, changing direction, lifting slightly and setting down, or rotating the wrist), and leaves. Operators deliberately make small errors: near misses, off-center or wrong-direction pushes, empty grasps, grasps that slip or are released after a short lift, and pushes that barely move the object. They act quickly, as in play rather than task execution. The recordings are balanced across three outcomes: a deliberate miss, a light touch, and a clear interaction, for example tipping a bowl. They expose the consequences of contact, including the empty grasps and failed pushes that successful demonstrations never show.

B.3 SEGMENTATION, RELABELING, AND MIXTURE

Each auxiliary recording is assigned to a split and then divided into atomic-skill segments with synchronized images, motor context, and actions (Section E); unusable segments are discarded. Each retained segment receives a correct brief instruction for its local behavior and a detailed caption. The original goal of a failed attempt is never kept as its label, an action chunk must stay inside the segment its instruction describes, and measured states and recorded actions are never overwritten.

Table 9: Fine-tuning configuration for $\pi_{0.5}$, shared by all conditions.

Setting	Value
Adaptation	full fine-tuning
Optimizer	AdamW, $(\beta_1, \beta_2, \epsilon) = (0.9,\ 0.999,\ 10^{-8})$
Learning rate	2×10^{-5}, constant, no warmup
Weight decay	0
Gradient clipping	1.0
Batch size	256 (32 per GPU, no accumulation)
Precision	BF16 mixed precision
Image input	224×224 per view
Chunk length H	16
Flow matching	4 noise samples per example; 10 Euler steps at inference
Hardware	8× NVIDIA A100 80 GB
Training time per run	about 4 days

Because the recordings contain no corrective continuations, R widens state–action coverage but does not demonstrate recovery under the original goal.

Training samples task demonstrations with probability $1 - \rho$ and auxiliary segments with probability ρ (Equation (2)); the wrist-perturbation probability p_V and the detailed-caption probability p_C apply to both sources, so caption style does not reveal the source. The three rates are fixed within a run and selected on validation data. Disabling V, C, or R sets p_V, p_C, or ρ to zero, and R without C still uses correct brief instructions. Auxiliary segments replace a fraction of each batch rather than adding optimizer updates.

C $\pi_{0.5}$ TRAINING

We fine-tune $\pi_{0.5}$ (Black et al., 2025a) with the configuration in Table 9, shared by all conditions; only the data construction varies. The policy receives the head and both wrist views, the instruction, and the dual-arm proprioceptive state, and is trained with its flow-matching action loss to predict chunks of $H = 16$ actions, about one second at the 15 Hz control rate. All modules are fine-tuned. Every condition and data-scaling budget uses the same optimizer steps, each run takes about four days on eight GPUs, and the enabled values of p_V, p_C, and ρ are shared across the factorial comparisons.

D WRIST-VIEW PERTURBATION

V edits wrist-view images with GPT-Image-2.5 (`gpt-image-2.5-flare`),[1] an image generation and editing model, to add plausible distractor objects. An edit is kept only if the instructed target, the relevant spatial relations, the grasp path, and the contact evidence remain readable and no new collision is implied; otherwise it is rejected rather than given the original label. Only the wrist image changes: the head view, motor state, instruction, and action label stay as recorded. Distractor identity, appearance, and placement vary independently of the task, and the training distractors are disjoint from the decoys used by the GroundFscore probes (Section 3.3). Generic corruption (noise, cropping, and dropout at the same rate) and wrist-camera removal are separate controls, and the salience-capture probes use physical distractors, so a policy cannot pass them by detecting editing artifacts.

E DETAILED-CAPTION ANNOTATION

Each behavior has a brief instruction and a detailed caption. Detailed captions are generated offline by a vision–language model with the prompt in Section E.1: the model receives the timestamped frames of an entire recording, splits it into contiguous atomic-skill segments, and describes each segment by the acting hand and whether it opens or closes, the operation, the position, the object's state, its distinguishing attributes, the object, and the grasp point. Attributes such as color or position are added only when needed to single out the object, and duplicates are distinguished by counting

[1] OpenAI, https://openai.com/index/introducing-chatgpt-images-2-5/.

from one side, so captions carry decision-relevant detail rather than length. Each description uses only features visible in the frames and is checked against the images and action record; uncertain object identities, motion directions, or contact descriptions are corrected or excluded. A different verb is used only when the recorded action matches it, and the same rules apply to demonstrations and auxiliary recordings.

E.1 Annotation prompt

The prompt is written in Chinese and requests Chinese descriptions; we reproduce an English translation that keeps its structure, rule numbering, and JSON field names.

Prompt Used to Annotate Detailed Captions

You are an expert in action segmentation and description for robot manipulation videos.
-Input Description-
- Input footage: a sequence of timestamped frames sampled from the whole video at a fixed frame rate; each `[x.xx second]` is followed by the corresponding frame.
- Timestamps are time coordinates of the original video, in seconds.
-Task-
Given the timestamped frame sequence of an entire video, first segment it by atomic skill into consecutive action segments, then output an unambiguous Chinese action description for each segment.
-Segmentation Rules (Segment by Atomic Skill, V2)-
1. Core principle: segment by atomic skill. Every atomic-skill action performed to accomplish the task goal must form its own segment. For example, moving an object from the left side of the tabletop to the right side should be split into four segments: "grasp – lift – move to – put down".
2. Each single small action, such as grasping, lifting, moving to, pressing against, or pushing, must be segmented separately.
3. Two different atomic skills may be merged into the same segment only when they occur within 1 second and cannot be separated.
4. Do not lump a chain of atomic actions that serve the same subgoal (pick up $\rightarrow$ move $\rightarrow$ pour $\rightarrow$ put down) into a single segment.
5. Auxiliary actions, such as fine-tuning an object's angle or slightly adjusting the hand pose, are not segmented separately.
6. Temporarily steadying an auxiliary object with a hand (e.g., holding the side of the bowl while pouring water) is not segmented separately as long as the main manipulated object does not change.
7. Remove all stationary pauses and pauses between narration, keeping only footage of effective manipulation.
8. Times are in seconds, in the original video's time coordinates; do not reset them to start from 0.0.
9. The `start` of the first segment equals the start time of the input video, the `end` of the last segment equals the end time of the input video, and adjacent segments join seamlessly (each segment's `start` equals the previous segment's `end`).
10. Give all timestamps with two decimal places.
-Description Rules (Unambiguous, Caption V4)-
1. Describe each segment unambiguously in natural language; decide whether to add modifiers such as color or position according to whether the reference is already unique.
2. Follow this reference formula for the sentence structure: [acting hand] + [hand open/closed state] + [verb] + [absolute & relative position] + [state] + [color + material + shape + other attributes] + [main object] + [grasp point].
3. Refer to the acting hand only as "left hand", "right hand", or "both hands". Words such as "gripper", "clamp", "robotic arm", "robotic hand", and "claw" are strictly forbidden. For example, "the left gripper closes" should be written as "the left hand closes", and "the left and right grippers" as "both hands".
4. Always include the hand's open/closed state (opens / closes / stays closed / stays open).
5. Add the object's initial configuration (e.g., cup opening facing up; a pen lying front to back with its cap toward the back).

Table 10: Sensitivity to the sampling rates on real-robot General PnP with a $\pi_{0.5}$ backbone. One rate varies at a time, relative to its validation-selected value ($1\times$, the full method; $2\times$ is capped at one), with the other two held at $1\times$. Level 0 disables the component and matches the corresponding leave-one-out row of Table 1. Columns follow Table 3.

	Wrist perturbation p_V				Detailed captions p_C				Auxiliary fraction ρ			
Level	SR↑	$\overline{\text{GF}}$↑	SC↓	MI↓	SR↑	$\overline{\text{GF}}$↑	SC↓	MI↓	SR↑	$\overline{\text{GF}}$↑	SC↓	MI↓
0	68	0.43	58	38	66	0.43	34	40	64	0.43	30	76
$1/2\times$	76	0.50	34	30	74	0.50	26	32	74	0.50	24	46
$1\times$	84	0.57	20	24	84	0.57	20	24	84	0.57	20	24
$2\times$	80	0.55	16	26	82	0.58	20	26	78	0.56	22	20

Prompt Used to Annotate Detailed Captions (Continued)

6. When duplicate or similar objects are present, distinguish them as "the n-th, counting from the top / bottom / left / right toward x".
7. Movement directions must match the frames. If a rotation or flip is already completed during a movement, do not label the rotation again for the final put-down.
8. Label a failed operation as "attempts to XX and fails".
9. Name objects only by features visible in the frames; do not invent unknown information.
10. Directions follow the front view: the side away from the operator is the front (far side), and the side close to the operator is the back (near side).

-Common Atomic Skills (Verbs) for Reference-

clamp / pick / grab / grasp / hold; push; pull; flip; move / put a lid back / put back / place / put down / toss into / release / stack; rotate / turn; unfold / flip open / lift open; insert / plug into / stuff into / tuck into; fold; wipe; unplug / pull out; press / hold down; shake / jiggle; stir; knock; pour out / pour into; straighten / set upright / stand up; tighten; squeeze / pinch; unscrew; tear; separate; bring together; hand over; pass through; scoop; rub.

-Output Format-

Output only a plain JSON array; do not output Markdown or explanatory text:

```
[{"start": float timestamp, "end": float timestamp,
  "action_zh":
    "complete, unambiguous Chinese action description"}]
```

F Additional ablation results

The factorial design in Table 1 covers every single-component and leave-one-out comparison; R-enabled conditions keep correct brief instructions even without C, so caption supervision is not counted twice. The mechanism controls in Table 4 compare V with generic corruption and random-position decoys, C with paraphrases and length-matched captions, and R with equal-duration successful demonstrations and with each auxiliary source alone. The data-scaling results are in the main text (Table 3). This appendix adds rate sensitivity, verb-change probes for noun lock-in, a detailed-instruction evaluation, and intervals for the validity analysis of Section 5.4.

F.1 Sensitivity to the sampling rates

We vary the wrist-perturbation probability p_V, the detailed-caption probability p_C, and the auxiliary fraction ρ one at a time, holding the other two at their validation-selected values (Table 10). Level 0 disables the component and coincides with the corresponding leave-one-out row of Table 1, and $1\times$ is the full method. Reporting success together with GroundFscore and the two failure rates tests whether stronger perturbation buys resistance to shortcuts at the cost of useful conditioning; Figure 4c plots the change in success from the selected setting against each rate.

F.2 Verb-change probes for noun lock-in

Real-robot General PnP has a single operation, so it cannot test whether a noun overrides a changed verb. We therefore probe noun lock-in in RoboTwin 2.0 under the Full setting, using 5 task pairs

Table 11: Verb-change probes for noun lock-in in RoboTwin 2.0 (Full setting, clean scenes, 100 trials per policy). Followed: the instructed operation is completed; lock-in: the operation of the restored task is performed; other: any remaining failure. Rows sum to 100%.

Method	Followed (%) ↑	Lock-in (%) ↓	Other (%)
$\pi_{0.5}$ fine-tune	36	52	12
Perturbot w/o V	64	26	10
Perturbot w/o C	45	43	12
Perturbot w/o R	66	24	10
Perturbot	**71**	**19**	10

Table 12: Success (%) over the 50 real-robot General PnP cases with brief and detailed instructions. Brief values repeat Table 1; Δ is detailed minus brief.

Method	Brief	Detailed	Δ
$\pi_{0.5}$ fine-tune	42	34	−8
+ V	52	46	−6
+ C	54	60	+6
+ R	56	46	−10
+ V + C	64	68	+4
+ V + R	66	58	−8
+ C + R	68	74	+6
Perturbot (V + C + R)	**84**	**86**	+2

whose shared object admits both operations from the same initial state and whose scripted expert produces a valid demonstration of each, such as pressing versus moving the stapler. A probe restores the initial state of one task in a pair on a clean scene and issues the brief instruction of the other, so the scene and the noun favor the familiar operation while only the verb specifies the correct one. Each pair is probed in both directions with 10 trials per direction, giving 100 trials per policy. A trial counts as followed if the policy completes the instructed operation, as lock-in if it performs the operation of the restored task, and as other otherwise.

Fine-tuning performs the familiar operation in 52% of probes and the instructed one in only 36% (Table 11). Perturbot reverses this, and removing C restores most of the lock-in (43%), whereas removing V or R raises it by at most 7 points, consistent with C making the requested operation explicit. Because C never introduces a verb absent from the data, the gain reflects grounding verbs the demonstrations already contain.

F.3 Detailed-instruction evaluation

The primary evaluation uses brief instructions, as a deployed policy would receive. A second pass replays the same 50 cases with a detailed instruction for each, written under the rules of Section E from the stored case specification and checked against the initial scene; the rest of the protocol is unchanged (Table 12). Without C, detailed instructions are out of distribution and cost 6–10 points; with C, they add 2–6 points. Perturbot therefore does not depend on detailed instructions at test time: its brief-instruction success exceeds every other configuration under either instruction style.

F.4 Confidence intervals for the validity of GroundFscore

The 36 checkpoints of Table 6 (twelve conditions, three seeds) share evaluation cases and training recipes, so they are not independent. We therefore report 95% percentile intervals from 10,000 two-level cluster bootstrap resamples: each resample draws conditions and seeds with replacement and, within each checkpoint, resamples its evaluation units (cases, probe trials, and the 100 GroundFscore observations). All predictors share the same draws, so their differences are paired (Table 13). The lower end of the GroundFscore interval lies above the point estimate of every other predictor, and its paired advantage excludes zero against each of them; the interval for task success includes zero.

Table 13: Spearman ρ with the negative online failure rate over 36 checkpoints, with 95% cluster-bootstrap intervals. The fourth column is the paired difference from GroundFscore; the last records whether the predictor ranks the no-wrist policy below the fine-tuned baseline.

Predictor	ρ	95% interval	GroundFscore − predictor	No-wrist below baseline
Task success	0.31	[−0.06, 0.60]	0.41 [0.12, 0.68]	yes
Stability $(1 - R)$	0.44	[0.09, 0.69]	0.28 [0.04, 0.52]	no
Responsiveness (S)	0.38	[0.02, 0.65]	0.34 [0.08, 0.60]	yes
GroundFscore	**0.72**	[0.49, 0.86]	–	yes