SUCCESS RATES ARE NOT MODEL PROPERTIES: CONTROLLED DIAGNOSTICS FOR VLA EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Success rates on standard manipulation benchmarks are the primary evidence for comparing vision-language-action (VLA) models. But a success rate is not a model property: it is measured under conditions fixed by the benchmark or by the configuration distributed with the model. We show that three of these conditions, changed with the model fixed, move success by amounts that are large for some models and small for others. The three are (1) which initial states are tested, (2) what instruction the model receives, and (3) how many predicted actions it executes between observations, the execution horizon. We test multiple VLA models on LIBERO and LIBERO-Plus, and execution also in SimplerEnv. (1) For initial states, X-VLA and $\pi_{0.5}$ both succeed on about 98% of LIBERO-Object's standard initial states at the same execution horizon, but on 22.1% and 89.5% of placements outside these states on which a separate VLA model succeeds at a high rate. After task-specific fine-tuning of $\pi_{0.5}$, success on the same placements falls by more than the standard score changes, for every recipe we tested. (2) For the instruction, removing the parameters that LIBERO-Plus appends to most instructions raises the success of the two SmolVLA models by up to 51.7 percentage points and changes the other eight models by at most 8.2 points. For these two models, the delivered text, not just the scene, drives success. Appending two ordinary words to standard instructions also lowers their success, even though both models perform another task when given that task's instruction. (3) For execution, a horizon of ten instead of each model's default raises the standard score of five of the six models on LIBERO and also changes success on the placements from (1), while in SimplerEnv each tested configuration does worst at its shortest horizon. Because models respond differently, these conditions move the margins between models and sometimes reverse which model scores higher, so a comparison made under one setting does not necessarily hold under another. We give an evaluation procedure that reports each model's response to these conditions and states when a comparison holds, asking not only how often a model succeeds, but under which conditions it does.

1 INTRODUCTION

Success rates on standard manipulation benchmarks are the primary evidence for progress in vision-language-action (VLA) models (Liu et al., 2023; Black et al., 2025b;a; Kim et al., 2025). A high standard score, the success rate on the benchmark's own initial states and instructions, establishes performance on the evaluated episodes. However, it does not show whether the model succeeds from other initial states, tolerates text that does not change the task, or maintains its success under a different execution horizon. VLA models predict several actions at each observation, and the execution horizon h is the number of them executed before the next observation. A controlled diagnosis measures how each condition affects the same model. This paper asks one question:

How do evaluation choices change the conclusions drawn from VLA success rates?

The LIBERO-Object suite illustrates this problem. In 6 of its 10 tasks, the 50 standard initial states never move the target object. We translate this object to placements absent from the standard initial states, retaining only those where a separate VLA model, used as a reference, succeeds at a high rate (termed reference-selected conditions). At the same execution horizon, X-VLA and $\pi_{0.5}$ both

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.

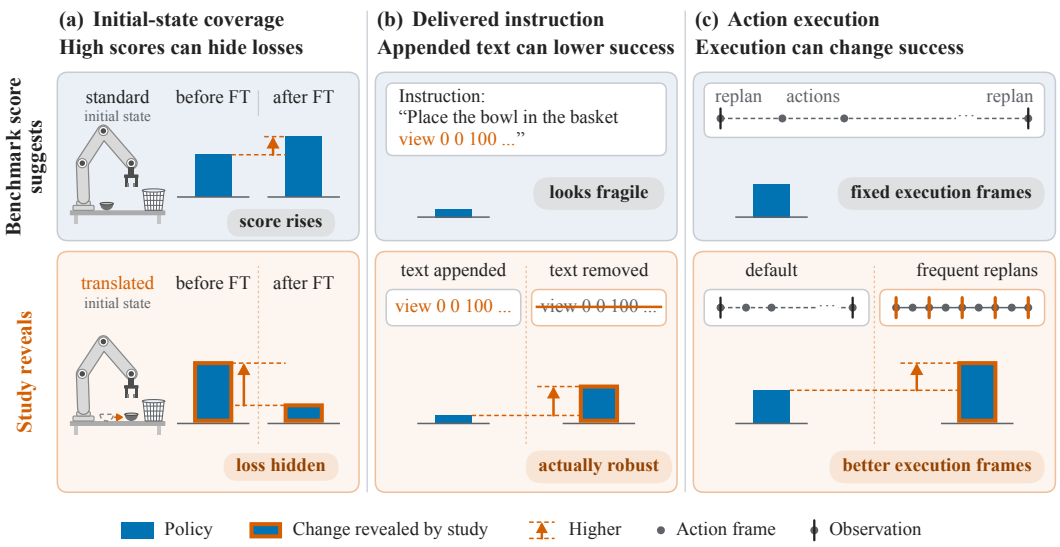

Figure 1: **Three evaluation choices, and what a controlled change of each reveals.** Schematic: (a) compares a model before and after fine-tuning; (b) and (c) change one condition of a fixed model.

succeed on about 98% of the standard initial states, but on 22.1% and 89.5% of the reference-selected conditions. Two models with nearly equal standard scores thus differ by 67.3 points at other placements of the same tasks.

We organize our diagnosis around three conditions fixed by the benchmark or the configuration distributed with the model, which can be modified at evaluation time without retraining (Figure 1). (1) *Initial-state coverage* asks whether a high standard score extends to other initial states. (2) The *delivered instruction*, the text the model actually receives, asks whether text that does not change the task alters success. Instructions assembled by software or written by people can carry such text. (3) *Action execution* asks how the execution horizon changes the standard score and success elsewhere. We isolate each factor as a controlled intervention on a fixed model, and refer to the three diagnostics as the state, instruction and execution studies. For the state study, we also compare a model before and after task-specific fine-tuning (e.g., Kim et al., 2025) on the same initial states, to evaluate how fine-tuning alters success at other placements.

Applied to multiple VLA models from 8 families on LIBERO (Liu et al., 2023), its perturbed extension LIBERO-Plus (Fei et al., 2026), and SimplerEnv (Li et al., 2025) (Table 1), the diagnostics reveal that each condition changes success by model-dependent amounts, and each exposes something the standard score does not show. (1) For initial-state coverage, the standard score measures success from start positions that the training demonstrations also use. Beyond them, models with similar standard scores differ widely, and fine-tuning can lose success there while the standard score stays high. (2) For the delivered instruction, a model that follows the instruction can still fail when task-irrelevant words are added. A robustness benchmark that alters the text alongside the scene therefore partly measures sensitivity to that text. (3) For action execution, the execution horizon changes both the standard score and the gap to other placements, and the favorable horizon differs across models and simulators. Because responses differ across models, these conditions move the margins between them and sometimes reverse which model scores higher. Which model is better therefore depends on the conditions, whereas a benchmark fixes conditions that vary in deployment.

Practitioners know that these conditions matter. VLA models can collapse when object positions change (Zhou et al., 2025), and shortcuts and overfitting limit what high scores establish (Jiang et al., 2026). Users have reported the text LIBERO-Plus appends to its instructions and measured its cost for one model[1], and the number of executed actions is set per model. The public configuration of $\pi_{0.5}$ executes fifty actions, while an issue report finds that executing ten recovers much of its LIBERO

[1] https://github.com/sylvestf/LIBERO-plus/issues, issues 48, 64 and 65.

performance[2]. Knowing that they matter does not tell how large each effect is when the model is held fixed and the altered condition is one on which another model succeeds, whether the effect is the same across models, and therefore whether a comparison made at one setting holds at another. We measure these quantities for three conditions that the benchmark or the model's configuration fixes. Our position is methodological: a success rate is a property of a model together with these conditions, not of the model alone, and it should be reported with them. Our contributions are as follows:

- **Controlled measurements of three evaluation conditions.** For fixed models, we measure how initial-state coverage, the delivered instruction and the execution horizon change success, including before and after fine-tuning. The effects are large and model-dependent: nearly equal standard scores hide a 67.3-point difference at other placements, fine-tuning can cut success there from 93.3% to 34.7% while the standard score stays at 100.0%, and two task-irrelevant words lower one model's success by 76.0 points (Section 4).
- **An evaluation procedure for comparisons that hold.** We evaluate additional initial states selected by another model's success and check this choice by repeating it with each other model as the reference, pair conditions on the same initial states with intervals, and report each model's response to a condition instead of one rate. This separates comparisons that survive a change of condition from those that do not (Sections 4.4 and 6).

2 Evaluation Questions

We treat success as a property of a model evaluated under specified conditions. For a model m and evaluation conditions c,

$$S(m, c) = \mathbb{E}[\, Y \mid m, c \,], \qquad Y \in \{0, 1\}, \tag{1}$$

where Y is the success of one episode, and the expectation is over the initial states and rollout randomness specified by the study. The conditions c encompass the initial states, the delivered instruction, and the execution horizon. A model and the evaluation conditions together determine the policy that acts in closed loop, and we hold the model fixed and vary the conditions. We define the *standard score* as the success rate on the initial states and instructions that the benchmark provides at a stated execution horizon, which is the metric that benchmarks report. Other interface choices, such as the observation format, the action parameterization, the episode time limit, and the success check, remain fixed within each comparison.

We ask one question for each evaluation choice, focusing on how a fixed model responds when exactly one condition changes. For (1) *initial-state coverage*, we ask what performance variations the standard initial states conceal. Section 4.1 translates the target object while holding the instruction and execution fixed. To select placements with evidence that they can be solved, it retains those on which a separate VLA model, used as a reference, succeeds at a high rate (termed reference-selected conditions). The task is unchanged at these placements, and the reference's success is empirical evidence that they can be solved (Appendix B.3). Appendix F gives a minimal construction in which success on the evaluated initial states does not determine success at such placements. For (2) the *delivered instruction*, we ask how much task-irrelevant text alters measured success. Section 4.2 removes or appends text while holding the scene and execution fixed. For (3) *action execution*, we ask how the number of actions executed between observations changes both the standard score and reference-selected success. Section 4.3 varies this horizon while holding the instruction and initial states fixed. When two settings of a condition use the same initial states, such as two execution horizons, episodes are paired by initial state.

Beyond these three diagnostics, we run two further designs. First, the adaptation case study applies the state study to fine-tuning: it compares a base model with its fine-tuned models under identical initial states and execution horizons, so the model changes while the conditions remain fixed. Second, an instruction-following control on LIBERO-Goal tests whether a model follows the instruction it receives: each episode is given the instruction of a different task, and we record completion of both the requested and the original tasks. Throughout our analysis, a *response* denotes the change in a single model's success when a condition shifts, whereas a *comparison* denotes the performance margin between two models. We first quantify each model's response, then evaluate in Section 4.4 how these responses alter reported margins between models.

[2] https://github.com/huggingface/lerobot/issues/2114

3 EXPERIMENTAL SETUP

Benchmarks and their roles. We employ three benchmarks to isolate specific evaluation conditions. From LIBERO (Liu et al., 2023) we use four suites of ten tasks, Spatial, Object, Goal and Long (LIBERO-10), where each task is evaluated from a fixed set of 50 initial states. We call LIBERO with the initial states and instructions it provides *standard LIBERO*, to distinguish it from the perturbed variants below. Within standard LIBERO, we assign specific suites to distinct studies. We use LIBERO-Object, whose tasks require picking a named object, to translate the target object and to run the suffix test (which appends neutral words, ordinary English words that name no object, direction or action). We use LIBERO-Goal, whose ten tasks share a single scene, for the instruction-following control, and all four suites to sweep over execution horizons. LIBERO-Plus (Fei et al., 2026) provides 10,030 perturbed variants of the LIBERO tasks, such as camera and lighting changes. We use it across all four suites to remove the perturbation parameters appended to most instructions (the repair). Finally, SimplerEnv (Li et al., 2025) evaluates released models in another simulator with the WidowX and Google Robot embodiments, which we use to repeat the execution sweep.

Models and study coverage. We evaluate 10 VLA models spanning 8 families (Black et al., 2025b;a; Shukor et al., 2025; Sun et al., 2026; Fang et al., 2026; Lin et al., 2026; Zheng et al., 2026; Kim et al., 2025). Table 1 lists the models included in each study, grouped by condition. The state study evaluates five models, with a sixth, VLA-JEPA (○), as the reference. Five models (π_0, $\pi_{0.5}$, SmolVLA, SmolVLA-LP, and X-VLA) enter all three fixed-model studies, enabling cross-study comparisons. The suffix "-LP" marks a model fine-tuned on LIBERO-Plus data, while the *Actions* column specifies whether a model predicts delta or absolute end-effector targets. Unless explicitly varied, each model uses its distributed configuration, including its default execution horizon h (Table 1), which ranges from 7 to 50.

Table 1: **The ten models and the studies each enters.**

Model	Actions	h	(1)		(2)		(3)
			States	Adapt.	Repair	Suffix	Horizon
π_0	delta	50	✓		✓	✓	✓
$\pi_{0.5}$	delta	50	✓	✓	✓	✓	✓
SmolVLA	delta	50	✓		✓	✓	✓
SmolVLA-LP	delta	50	✓		✓	✓	✓
EVO-1	delta	50			✓	✓	✓
X-VLA	absolute	30	✓		✓	✓	✓
MolmoAct2	delta	10			✓	✓	
OpenVLA-OFT	delta	8			✓		
OpenVLA-OFT-LP	delta	8			✓		
VLA-JEPA	delta	7	○		✓	✓	

Outcomes and uncertainty. We report success rates in percent and performance changes in percentage points (pp). The state study evaluates each LIBERO-Object task with the target object at its original position and at five translated positions. Each position is evaluated from 20 initial states under three different evaluation seeds, and all tasks are weighted equally. Standard-LIBERO evaluations run each task's 50 initial states once, so that compared settings share their initial states (Appendix A.1). The instruction study runs the exact same LIBERO-Plus variants under both original and repaired instructions. Confidence intervals resample the primary unit of variation for each design (Owen & Eckles, 2012). Appendices A.1 and A.2 give the sample sizes, estimands, and resampling units.

4 FINDINGS

4.1 INITIAL-STATE COVERAGE

The standard initial states cover a narrow range of target placements. In 6 of the 10 LIBERO-Object tasks, the target never moves across the 50 initial states, and in the others it varies by at most 1.9 cm. These positions also lie within 4.8 mm of the starts of the training demonstrations distributed with LIBERO, and within 0.7 mm where the target never moves (Appendix B.1). We test what performance differences this evaluation leaves unmeasured.

Evaluation protocol. We translate the target by 4 to 12 cm, outside the standard initial states and the benchmark's specified initialization region, and leave the rest of each standard initial state unchanged. Each task–translation condition uses the same 20 initial states and three evaluation seeds. We keep as reference-selected conditions the 27 of 50 on which VLA-JEPA, the reference, succeeds at least 75% of the time. The reference's success rate on a condition is evidence that the condition can be solved,

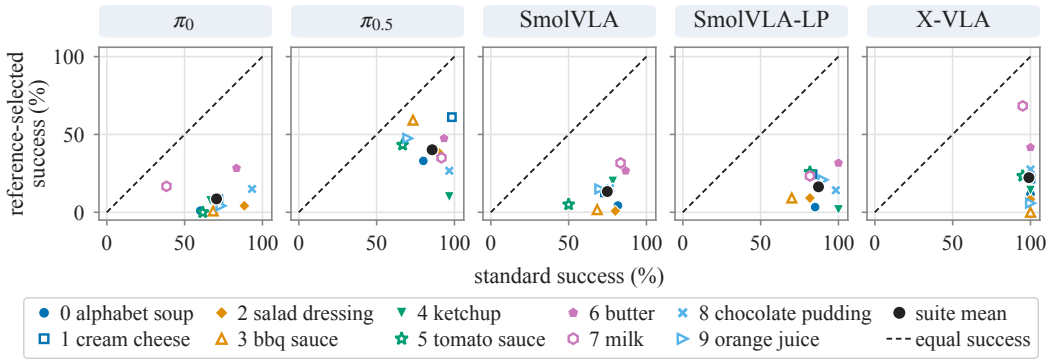

Figure 2: **At default settings, every model scores lower on reference-selected conditions.** Each point is one LIBERO-Object task and black points are suite means, at each model's default horizon.

not that every initial state in it can. For each of the five evaluated models, we report the standard score B, the reference-selected success F_{ref} (averaged over each task's retained conditions, then equally over tasks), and the gap $B - F_{\text{ref}}$. We evaluate B and F_{ref} at $h = 10$ for all five models, using one fixed set of conditions selected by the reference at $h = 7$, and report each model's default horizon on the same conditions (Table 2). With each other evaluated model as the reference in turn, excluded from its own comparison, or with thresholds of 50% and 90%, no model–task pair succeeds more often on the reference-selected conditions than on the standard initial states (Appendix B.3).

Standard scores mask differences across placements. At $h = 10$, X-VLA succeeds on 97.8% of the standard initial states and 22.1% of the reference-selected conditions, whereas $\pi_{0.5}$ succeeds on 98.2% and 89.5%. Similar standard scores thus accompany a 67.3-point difference between the two models on the reference-selected conditions. Reference-selected success is lower in 49 of the 50 model–task pairs and equal in one, so the gap is not confined to a few tasks. At each model's default horizon, it is lower in all 50 pairs (Figure 2). Figure 3 shows one task in which moving the target by 8 cm changes X-VLA's outcome from success on all evaluated initial states to failure on all of them.



Figure 3: **One task, two initial states.**

Fine-tuning can preserve standard success while losing success elsewhere. We next apply these evaluations before and after task-specific fine-tuning of $\pi_{0.5}$, whose public checkpoint was already fine-tuned on LIBERO. On tasks 0, 5 and 8, seven fine-tuning recipes, which we call regimes, vary the scripted demonstration set, the trained parameters and the update budget, with three training runs per demonstration set (Appendix B.4). Base and fine-tuned models use $h = 5$ on the same standard and reference-selected conditions. The evaluated standard starts also occur in the training demonstrations, and no demonstration begins at a translated placement. On task 5, standard success stays at 100.0%, while reference-selected success falls from 93.3% to 34.7% (Figure 4). All 21 task–regime settings reduce reference-selected success, with simultaneous one-sided 95% upper bounds below zero. In 15 of them, the 90% interval of the change in the standard score lies within ±10 pp (Table 8).

Table 2: **State study at the default h and at $h = 10$.**

Model	Default	At default h			At $h = 10$		
	h	B	F_{ref}	Gap	B	F_{ref}	Gap
π_0	50	70.5	8.6	61.9	87.3	12.9	74.4 (+12.5)
$\pi_{0.5}$	50	85.7	40.2	45.5	98.2	89.5	8.7 (−36.8)
SmolVLA	50	74.8	13.3	61.5	94.3	36.9	57.4 (−4.1)
SmolVLA-LP	50	87.2	16.4	70.8	98.5	61.0	37.5 (−33.3)
X-VLA	30	99.0	22.3	76.7	97.8	22.1	75.7 (−1.1)

Losses persist with fixed normalization. The fine-tuning procedure used here recomputes the normalization statistics of states and actions on the fine-tuning data, so it changes these statistics

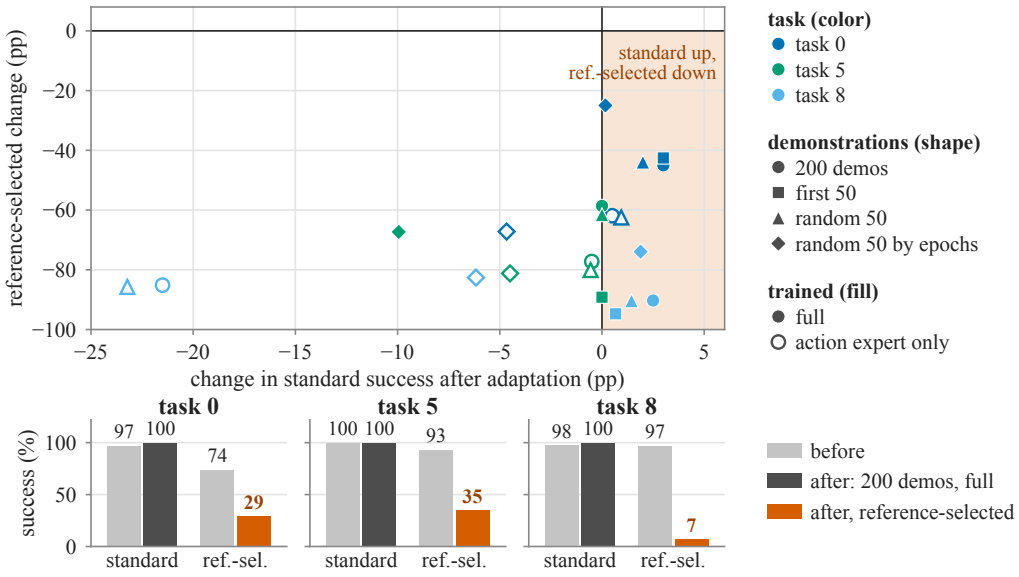

Figure 4: **Adapting $\pi_{0.5}$: every regime loses reference-selected success.** Top: one point per task and regime, averaged over training runs. Bottom: full-model fine-tuning on 200 demonstrations. All at $h = 5$.

along with the weights. To test whether this change explains the loss, we repeat full-model fine-tuning with the base statistics fixed during training and evaluation. Reference-selected success still falls by 46.8 to 92.2 pp at $h = 5$, and the loss remains at $h = 10$. Tasks 0 and 5 keep 100.0% standard success, and task 8's falls to 74.3%. A separate control without any update shows that changing the statistics alone can also reduce success (Tables 9 and 10). Normalization changes can therefore contribute to the loss, but they do not explain it alone.

Takeaway: (1) Initial-state coverage

Standard success can stay high while success on reference-selected conditions is low or falls after fine-tuning. Evaluating both reveals differences and losses that the standard score can miss.

4.2 Delivered instruction

We next hold the scene fixed and test whether task-irrelevant text changes success, first in instructions that a benchmark generates and then in controlled text variations.

Evaluation protocol. LIBERO-Plus appends perturbation parameters to the instructions of 8,493 of its 10,030 variants (84.7%). The model therefore receives, for example, *pick up the alphabet soup and place it in the basket* followed by `view 0 0 100 0 0 initstate 381`. This construction and its cost for one model were reported earlier in the benchmark's issue tracker[1]. We compare the original instructions with repaired ones that remove these parameters, keeping the scene, the model and its execution unchanged. Variants that rewrite the language are left unchanged (Appendix C.1).

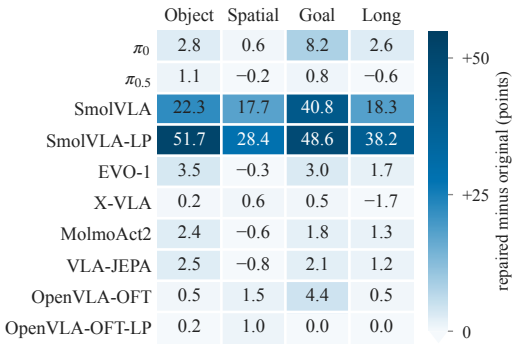

Figure 5: **The largest repair effects occur in the SmolVLA pair.**

Repair effects differ across models. Removing the appended parameters improves SmolVLA by 17.7 to 40.8 pp and SmolVLA-LP by 28.4 to 51.7 pp across the four suites (Figure 5). The other eight models change by at most 8.2 pp. Across all Object variants, SmolVLA-LP, which was fine-tuned on LIBERO-Plus, improves from 7.3% to 59.0%. Table 16 reports all rates and intervals. A separate four-model confirmation reproduces all 9 repair effects larger than 5 pp in the full study that it tests, and whether the effect depends on h remains unresolved (Appendix C.6).

Filler text reproduces large losses. To test whether the appended parameters themselves are required, we pad repaired instructions to their original word count with neutral filler, on a fixed 100-variant subset of each suite. On Object, SmolVLA-LP succeeds on 10.0% of the subset with the original text, 62.3% after the repair and 11.3% after the padding (Table 12). The filler lowers success in all 8 SmolVLA model–suite pairs, so the appended parameters are not required for a large loss.

Sensitivity also appears on standard LIBERO-Object. Appending two neutral words lowers SmolVLA-LP by 76.0 pp and SmolVLA by 31.8 pp. At 32 words, their success falls to 0.0% and 0.4%, $\pi_{0.5}$ loses 16.4 pp, and the other five models stay within 5.8 pp at every length (Figure 6, Appendix C.4).

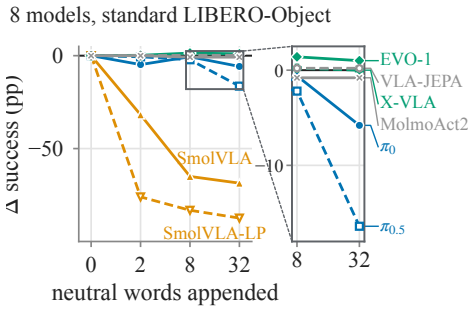

Figure 6: **Neutral words after the benchmark's own instruction.**

Following instructions does not ensure robustness to task-irrelevant text. LIBERO-Goal places its ten tasks in one shared scene. We give each episode the instruction of another task and record completion of both the requested and the original task. Every tested model completes the requested task in 75.8 to 98.8% of episodes and the original task in none. Yet two neutral words appended to a task's own instruction change SmolVLA and SmolVLA-LP by -61.6 and -51.2 pp (Table 14).

Takeaway: (2) Delivered instruction

Task-irrelevant text can change success even when a model follows task-changing instructions. Robustness evaluations should therefore control the text delivered alongside the scene.

4.3 Action Execution

We finally vary h, the number of predicted actions executed before the next observation, and measure the response on both standard and reference-selected conditions.

Evaluation protocol. The six models of the execution study are evaluated at each $h \in \{5, 10, 25\}$ that does not exceed their predicted action sequence and at their default. The models, instructions, action units and episode budgets stay fixed. We report the full grid and use $h = 10$ for the common comparison (Appendix D).

Execution effects differ across models. Relative to each model's default, $h = 10$ improves five of the six models on all four LIBERO suites, by $+3.4$ to $+23.8$ pp, while X-VLA changes by -4.4 to $+2.2$ pp (Figure 7a). Reducing h from 50 to 5 or 25 also raises the point estimate in all 20 model–suite pairs of the models whose default is 50. Table 17 gives the full grid.

Execution also changes the gap. On the fixed reference-selected conditions, $h = 10$ raises success for 4 of the 5 state-study models, and every gap keeps its sign. The gap narrows by 36.8 pp for $\pi_{0.5}$ but widens by 12.5 pp for π_0, whose standard success rises more than its reference-selected success, and X-VLA's changes little (Figure 7b, Table 2). The difference between standard and reference-selected success therefore depends on execution as well as on placement.

Execution responses differ across robot configurations. We repeat the sweep in SimplerEnv (Li et al., 2025) with the X-VLA checkpoints trained separately for WidowX and for Google Robot (Zheng et al., 2026) and with DB-CogACT, the CogACT model of the Dexbotic toolbox, on WidowX (Li et al., 2024; Xie et al., 2025). All three configurations do worst at their shortest tested horizon. X-VLA on WidowX rises from 0% at $h = 5$ to 94% at $h = 30$, and DB-CogACT from 8% at $h = 1$

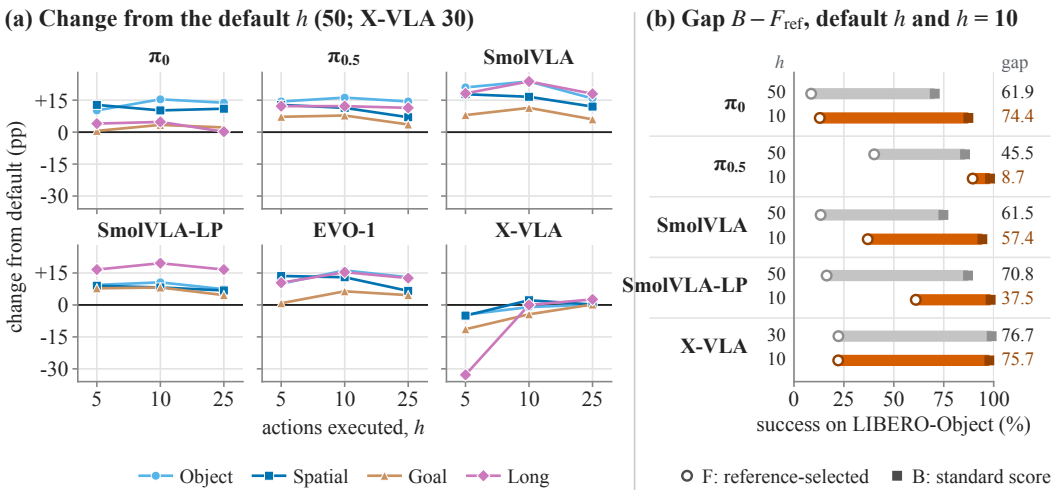

Figure 7: **The execution horizon changes each model differently, and changes the gap.** (a) Six models on four LIBERO suites. (b) Five models on LIBERO-Object, both sets of initial states.

to 46% at $h = 10$. X-VLA on Google Robot peaks at $h = 10$, not at $h = 30$ (Table 18). Neither shorter nor longer execution is uniformly better across these configurations.

Takeaway: (3) Action execution

The execution horizon affects both standard success and the gap to reference-selected success. Responses should be measured for the evaluated model and robot configuration.

4.4 Consequences for model comparisons

Model-dependent responses change the margins between models, even when their order is preserved. For models i and j under condition c, let $D_{ij}(c) = S_i(c) - S_j(c)$ be their margin, and let $\delta_i = S_i(c_1) - S_i(c_0)$ be the response of model i to a change from c_0 to c_1. Then

$$K_{ij} = D_{ij}(c_1) - D_{ij}(c_0) = \delta_i - \delta_j. \tag{2}$$

We call an ordering reversed when the unadjusted 95% intervals of the two margins exclude zero with opposite signs. The intervals resample LIBERO-Plus variants, and because variants derived from the same LIBERO task, their base task, share its difficulty, we also resample base tasks. On LIBERO-Plus Object at $h = 50$, the margin of π_0 over SmolVLA-LP is +36.5 pp before the instruction repair and -12.4 pp after it. This pair also reverses at $h = 10$, and both reversals hold when base tasks are resampled (Table 19). In contrast, the same repair reverses $\pi_{0.5}$ and SmolVLA-LP at $h = 50$ only when variants are resampled, and at $h = 10$, $\pi_{0.5}$ still leads after the repair.

Most changes affect margins without reversing orders (Appendix Figure 8). The gaps of the state study reverse one suite-average ordering (Appendix E.3), and switching from $h = 50$ to $h = 10$ reverses 0 of the 40 pair–suite combinations among the five models whose default is fifty (Appendix E). No common shift reproduces any of the three changes: after subtracting the shift that fits all models best, the largest remaining deviation on Object is 12.4 to 33.5 pp across the three, with the state gaps at $h = 10$ (Appendix E.4). Evaluation conditions therefore change the size of an advantage even when its direction holds.

5 Related Work

Benchmark validity and coverage. Relative performance depends on the benchmark (Dehghani et al., 2021) and, for language models, on the prompt format (Sclar et al., 2024; Mizrahi et al., 2024). For VLAs, Jiang et al. (2026) show that shortcuts and overfitting limit what high scores establish, LIBERO-PRO reports near collapse when object positions change (Zhou et al., 2025),

and LIBERO-Plus finds models largely insensitive to language variations (Fei et al., 2026). Other benchmarks test manipulation beyond one clean setting (Wang et al., 2025; Pumacay et al., 2024; Morgan et al., 2026). We add what the standard initial states cover, reference-selected conditions, and a before-and-after adaptation comparison. Our repair of the appended parameters complements the language perturbations of LIBERO-Plus.

Action execution. How many predicted actions to execute before replanning is an established design choice, studied through prediction and execution horizons (Chi et al., 2023), chunk size (Zhao et al., 2023), and methods that overlap or adapt replanning (Black et al., 2025c; Feng et al., 2026; Wang et al., 2026; Liang et al., 2026; Liu et al., 2025). SmolVLA ablates the number of executed steps on LIBERO (Shukor et al., 2025), and OpenVLA-OFT specifies the number its evaluation executes (Kim et al., 2025). We measure the response across models, simulators and initial states.

Evaluation practice and adaptation. VLA-Eval standardizes inference across code bases (Choi et al., 2026), and implementation choices and finite samples affect reported results (Henderson et al., 2018; Snyder et al., 2026; Agarwal et al., 2021). Fine-tuning can trade out-of-distribution accuracy for in-distribution gains (Kumar et al., 2022), and in VLAs, training a new action expert can degrade knowledge transferred from the vision-language model (Driess et al., 2025). Factor-wise studies separate generalization to object position from other factors (Xie et al., 2024). Imitation learning collects corrections away from the demonstrations (Ross et al., 2011; Laskey et al., 2017), and reachability analysis and safe reinforcement learning characterize states from which a target can be reached or a violation avoided (Bansal et al., 2017; Thananjeyan et al., 2021). Our reference-selected conditions give empirical evidence that a task can be completed there, not a guarantee.

6 Evaluating VLA Models Under Stated Conditions

What the diagnostics suggest. The controls narrow the explanations without identifying mechanisms. Losses after fine-tuning persist when the normalization statistics are kept fixed. Padding the repaired instruction back to its length brings most of the instruction loss back, and the tested models follow task-changing instructions yet remain sensitive to task-irrelevant additions. Changing h substantially reduces $\pi_{0.5}$'s gap, whereas X-VLA's changes little between the tested settings.

An evaluation procedure. (a) Evaluate the standard initial states and additional initial states on which another model succeeds, and repeat the selection with each other model as the reference, excluded from its own comparison (Section 4.1). (b) Compare conditions within a model on the same initial states, with intervals over the study's sampling unit (Section 3). (c) Report each model's response to the execution horizon and to task-irrelevant text, not one rate (Sections 4.2 and 4.3). (d) Report the margin between two models under each stated condition, and call an advantage consistent across the tested conditions only when the intervals support its direction throughout (Section 4.4).

Reporting card. To support interpretation and reproduction of a VLA success rate, report (1) the evaluated initial states, with identifiers and task weighting; (2) the instruction the model actually received; (3) the execution horizon, and the control rate (actions per second); (4) the action interface and the episode budget; (5) the unit of randomness and the rule that pairs episodes; and (6) the rule used to select checkpoints or settings.

Scope and limitations. Three boundaries delimit our claims, each with a next step. First, the evidence comes from simulation. In SimplerEnv, reference-selected success and the standard score rank models against real-robot results with the same Kendall's τ (Appendix D.2), so these conditions serve as a diagnostic until real-robot tests are available. Second, the state and instruction studies cover the LIBERO family, the adaptation study one model on three tasks, and the translation one object, selected mainly by one reference; more models and references would show how far the effects extend. Third, the text controls use fixed word sequences, and a common h spans different times when control frequencies differ. We report the response to h rather than a recommended value, and leave trial allocation across conditions to future work.

The standard score remains a useful measure under a specified evaluation, and the state results are instances of underspecification (D'Amour et al., 2022), in which models that agree on the evaluated conditions diverge elsewhere. Success rates are not model properties: a rate describes a model under the conditions that produced it.

AI USE STATEMENT

Large language models were used as writing assistants to improve clarity and readability, to assist with literature coverage checks, and for routine coding support. All core research questions, experimental designs, theoretical formulations, and empirical analyses were developed by the authors. The authors have verified all citations and results, and take full responsibility for the contents of this paper.

REPRODUCIBILITY STATEMENT

Code for all experiments is available at https://anonymous.4open.science/r/success-rates-are-not-model-properties-63D5 and in the supplementary material. It contains the evaluation code for LIBERO, LIBERO-Plus and SimplerEnv, the demonstration and fine-tuning code of the adaptation case study, the environment setup, and one analysis script per experiment that computes the reported estimates and intervals from the per-episode records. Appendix A lists the models and evaluation protocols, and Appendix A.2 defines the estimands and resampling procedures.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J. Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *IEEE Conference on Decision and Control (CDC)*, pp. 2242–2253, 2017.

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 *Conference on Robot Learning (CoRL)*, volume 305, pp. 17–40, 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 (RSS)*, 2025b.

Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025c.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

Suhwan Choi, Yunsung Lee, Yubeen Park, Chris Dongjoo Kim, Ranjay Krishna, Dieter Fox, and Youngjae Yu. vla-eval: A unified evaluation harness for vision-language-action models. *arXiv preprint arXiv:2603.13966*, 2026.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley.

Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Mostafa Dehghani, Yi Tay, Alexey A. Gritsenko, Zhe Zhao, Neil Houlsby, Fernando Diaz, Donald Metzler, and Oriol Vinyals. The benchmark lottery. *arXiv preprint arXiv:2107.07002*, 2021.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhar, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action reasoning models for real-world deployment. *arXiv preprint arXiv:2605.02881*, 2026.

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: A progressive robustness benchmark for visual-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 38574–38583, 2026.

Xiangdong Feng, Yuxuan Cheng, Chen Shi, Boyao Han, Yuxuan Yan, Yitong Hong, Zhuotao Tian, and Li Jiang. Denoising tells when to replan: Denoising-variance adaptive chunking for flow-based robot policies. *arXiv preprint arXiv:2606.03847*, 2026.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, pp. 3207–3214, 2018.

Tianchong Jiang, Xiangshan Tan, Samuel Wheeler, Luzhe Sun, Tewodros W. Ayalew, and Matthew Walter. What are we actually benchmarking in robot manipulation? *arXiv preprint arXiv:2606.04233*, 2026.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems (RSS)*, 2025.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations (ICLR)*, 2022.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Conference on Robot Learning (CoRL)*, volume 78, pp. 143–156, 2017.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, Xiaofan Wang, Bei Liu, Jianlong Fu, Jianmin Bao, Dong Chen, Yuanchun Shi, Jiaolong Yang, and Baining Guo. CogACT: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, 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 *Conference on Robot Learning (CoRL)*, volume 270, pp. 3705–3728, 2025.

Yuanchang Liang, Xiaobo Wang, Kai Wang, Shuo Wang, Xiaojiang Peng, Haoyu Chen, David Kim Huat Chua, and Prahlad Vadakkepat. Adaptive action chunking at inference-time for vision-language-action models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20802–20811, 2026.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, Lixing Zou, Zhaoye Zhou, Gen Li, and Bo Zhao. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13397–13406, 2026.

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.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Maximilian Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations (ICLR)*, 2025.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

Jeremy Morgan, Hyeonho Oh, Prajwal Vijay, Jincen Song, Ashvin Arora, Hojung Lim, Alina Du, Jesse Thomason, Gaurav S. Sukhatme, and Ishika Singh. Colosseum V2: Benchmarking generalization for vision-language-action models. *arXiv preprint arXiv:2605.27759*, 2026.

Art B. Owen and Dean Eckles. Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics*, 6(3):895–927, 2012.

Physical Intelligence. openpi: Open-source models and packages for robotics. https://github.com/Physical-Intelligence/openpi, 2025. `examples/libero/main.py` at commit `231a1cf`, `replan_steps = 5`.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. THE COLOSSEUM: A benchmark for evaluating generalization for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15, pp. 627–635, 2011.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations (ICLR)*, 2024.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

David Snyder, Apurva Badithela, Nikolai Matni, George Pappas, Anirudha Majumdar, Masha Itkina, and Haruki Nishimura. Beyond binary success: Sample-efficient and statistically rigorous robot policy comparison. In *Robotics: Science and Systems (RSS)*, 2026.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. In *European Conference on Computer Vision (ECCV)*, pp. 478–497, 2026.

Brijen Thananjeyan, Ashwin Balakrishna, Suraj Nair, Michael Luo, Krishnan Srinivasan, Minho Hwang, Joseph E. Gonzalez, Julian Ibarz, Chelsea Finn, and Ken Goldberg. Recovery RL: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6(3): 4915–4922, 2021.

Haoxuan Wang, Gengyu Zhang, Yan Yan, Ramana Rao Kompella, and Gaowen Liu. VLA knows its limits: Adaptive execution horizons for robot policies. In *European Conference on Computer Vision (ECCV)*, pp. 607–623, 2026.

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. VLATest: Testing and evaluating vision-language-action models for robotic manipulation. *Proceedings of the ACM on Software Engineering*, 2(FSE):1615–1638, 2025.

Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3153–3160, 2024.

Bin Xie, Erjin Zhou, Fan Jia, Hao Shi, Haoqiang Fan, Haowei Zhang, Hebei Li, Jianjian Sun, Jie Bin, Junwen Huang, Kai Liu, Kaixin Liu, Kefan Gu, Lin Sun, Meng Zhang, Peilong Han, Ruitao Hao, Ruitao Zhang, Saike Huang, Songhan Xie, Tiancai Wang, Tianle Liu, Wenbin Tang, Wenqi Zhu, Yang Chen, Yingfei Liu, Yizhuang Zhou, Yu Liu, Yucheng Zhao, Yunchao Ma, Yunfei Wei, Yuxiang Chen, Ze Chen, Zeming Li, Zhao Wu, Ziheng Zhang, Ziming Liu, Ziwei Yan, and Ziyu Zhang. Dexbotic: Open-source vision-language-action toolbox. *arXiv preprint arXiv:2510.23511*, 2025.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *International Conference on Learning Representations (ICLR)*, 2026.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

A Experimental Details

In this section, we describe the models we evaluate, the evaluation protocol of each study, and the statistics we report.

Models. We evaluate the public checkpoints in Table 3, each with the inference code and configuration it is distributed with. The four suite-specific OpenVLA-OFT checkpoints, with the suffixes `spatial`, `object`, `goal` and `10`, are each evaluated on their own suite. X-VLA predicts absolute end-effector targets, and the other models predict changes of the end-effector pose. In both cases, the executed horizon counts the actions applied before the next observation. Table 1 lists the studies each model enters. The code that moves objects for the state study does not support EVO-1 and MolmoAct2. The execution study leaves out VLA-JEPA, which predicts seven actions per call, and MolmoAct2, whose default already equals ten; we did not run the other settings for MolmoAct2. The two OpenVLA-OFT variants run only through their own evaluation code, so they enter the instruction study but not the state or execution studies.

Table 3: **Evaluated checkpoints.**

Model	Checkpoint
π_0	`lerobot/pi0_libero_finetuned_v044`
$\pi_{0.5}$	`lerobot/pi05_libero_finetuned_v044`
SmolVLA	`lerobot/smolvla_libero`
SmolVLA-LP	`lerobot/smolvla_libero_plus`
VLA-JEPA	`lerobot/VLA-JEPA-LIBERO`
MolmoAct2	`lerobot/MolmoAct2-LIBERO-LeRobot`
EVO-1	`zuoxingdong/evo1_libero`
X-VLA	`lerobot/xvla-libero`
OpenVLA-OFT	`moojink/openvla-7b-oft-finetuned-libero-`*`suite`*
OpenVLA-OFT-LP	`sii-research/openvla-7b-oft-finetuned-libero-plus`

A.1 Evaluation protocol

Standard LIBERO. Each evaluated LIBERO suite has ten tasks, and each task has 50 standard initial states. We run every initial state once, in a fixed order, so all conditions of a model are evaluated on the same 500 initial states per suite. An episode ends when the task succeeds or when the suite's step limit is reached: 280 steps on Spatial and Object, 300 on Goal and 520 on Long, at LIBERO's default control frequency of 20 Hz, which our code does not change. The execution study evaluates every model at 5, 10 and 25 executed actions and at its default, with 500 episodes per suite and setting.

Initial-state study. Each LIBERO-Object task is evaluated under six conditions: the standard initial states and five translations of the target object. Each condition runs standard initial states 0 to 19, each under three evaluation seeds. That is, each condition is measured from 60 episodes (20 initial states $\times$ 3 seeds), and each model from 3,600 episodes over the ten tasks. The adaptation study evaluates the base model and every fine-tuned model in the same way on tasks 0, 5 and 8, except that the standard initial states run under ten evaluation seeds, 200 episodes per model and task, to measure changes in the standard score more precisely.

Instruction study. Each of the 10,030 LIBERO-Plus variants is run once with its original instruction and once with the repaired one. The two runs of a variant share its scene and initial state, so 20,060 episodes are planned per model. Appendix C.2 gives the paired outcomes each model actually has.

SimplerEnv. Each configuration runs 24 episodes per task at every executed horizon it supports. The WidowX configurations use the four WidowX tasks (carrot on plate, eggplant in basket, spoon on towel, stack cube), and the Google Robot configuration the six Google Robot tasks (close drawer, move near, open drawer, pick coke can, pick object, place in closed drawer). Success rates weight the tasks of a configuration equally.

A.2 Statistics

We report success rates in percent and differences in percentage points. Unless stated otherwise, intervals are 95% bootstrap percentile intervals (Owen & Eckles, 2012), and every difference is paired: conditions that share initial states or variants are resampled together.

State study. A condition's success on a task is the mean over its 20 initial states and three evaluation seeds, F_{ref} averages the task's kept conditions, and suite means weight the ten tasks equally. Intervals resample initial states within each task, so the three seeds of a state move together, and apply one draw to the standard condition, every translated condition and both horizons (2,000 draws). The kept conditions are fixed by the reference and are not resampled.

Adaptation. Each draw applies one vector of initial states to every condition and evaluation seed, and resamples the three training runs, and the random 50-demonstration subsets, once for all regimes and the base model. The 21 contrasts form one family with studentized max-T bounds (Appendix B.4).

LIBERO-Plus. Intervals resample variants with one weight vector shared by every condition, so both runs of a variant move together. Appendix E.2 resamples base tasks instead. The variation between repeated runs of one variant is not a term in these intervals, and Appendix C.6 measures it.

Standard LIBERO and model comparisons. Intervals resample initial states within each task, shared by every model and setting (10,000 draws for the LIBERO-Goal control). A margin between two models uses the same draw for both models and both conditions (2,000 draws).

B Initial States

This section gives what the standard initial states vary, the per-task results of Section 4.1, the choice of threshold and reference, and the adaptation study with its normalization controls.

B.1 What the standard initial states vary

Each LIBERO-Object task has 50 standard initial states that place seven objects on a table. Table 4 gives, for each task, how many of the seven objects move across the 50 states and how far the target

object moves. The target is the object that the task instruction names. In 6 of the 10 tasks the target never moves, and no object varies in height.

Each task file declares a 5 cm by 5 cm region for the target, and the stored states cover at most 39% of its width. The original LIBERO demonstrations start the target in the same place: every start in either set lies within 4.8 mm of the nearest start in the other. The five translations move the target by $+4$, $+8$ and $+12$ cm and by -8 cm along the world x axis, and by -8 cm along y (written $+4x$, $+8x$, $+12x$, $-8x$ and $-8y$), and leave the rest of the stored state unchanged. Every translated position lies outside the declared region and 2.1 to 12.0 cm from the nearest demonstration start.

In tasks 1, 2 and 5, the translation of -8 cm along the y axis places the target against a neighboring object, and one of the two is pushed away when the simulator settles. At each model's default horizon, removing these conditions leaves the comparison unchanged: 50 of the 50 model–task pairs still succeed less often on the reference-selected conditions, and no model's mean moves by more than 1.1 points.

Table 4: **Variation across the 50 standard initial states of each LIBERO-Object task.** The target range is the larger of the target's ranges along the world x and y axes in the stored states.

Task	Objects that move	Target range (cm)
0	4	0.0
1	4	1.0
2	3	0.0
3	3	1.9
4	4	0.0
5	4	0.0
6	3	1.0
7	4	1.0
8	2	0.0
9	3	0.0

B.2 Per-task results

The reference keeps 27 of the 50 conditions: all five on task 0; $+4x$, $-8x$ and $-8y$ on task 1; $+4x$, $+8x$, $-8x$ and $-8y$ on tasks 2 and 8; $+4x$ and $-8x$ on tasks 3, 5 and 9; $+4x$ and $-8y$ on tasks 4 and 6; and $+4x$ on task 7. Table 5 gives the per-task results of the initial-state study at each model's default horizon and at $h = 10$. The ten-action rows of Table 5 keep the reference-selected conditions defined at the default horizon, so the two row groups differ only in the executed horizon. Its last rows give the change in the gap from each model's default to ten executed actions. The 95% interval resamples initial states within tasks and applies each draw to both horizons and all conditions. It excludes zero for π_0, $\pi_{0.5}$ and SmolVLA-LP, and it includes zero for SmolVLA and X-VLA. At five executed actions, the setting of the LIBERO example in its developers' code, $\pi_{0.5}$ reaches a standard score of 97.8% and reference-selected success of 85.5% on the same conditions, close to its values at ten.

B.3 Choice of threshold and reference

Section 4.1 keeps a translated condition when a reference model, VLA-JEPA, succeeds in at least 75% of its 60 episodes. A natural question is whether the comparison depends on this threshold or on the choice of reference. We first check that translated conditions can be solved at all: on task 0, a scripted controller with access to the simulator state solves each of 21 translations of the target in all of its 15 episodes. We then repeat the comparison with thresholds of 50% and 90%, at each model's default horizon and at ten executed actions (Table 6). At the default horizons, the standard score exceeds reference-selected success in every model–task pair that keeps a condition, at every threshold. At ten actions, the remaining pairs are ties in which $\pi_{0.5}$ succeeds in every episode on both sets. At 90%, tasks 3 and 4 keep no condition.

We also let each of the five evaluated models act as the reference at the same three thresholds, leaving it out of its own comparison, at the default horizons and at ten executed actions. A weaker reference succeeds on fewer translations, so it keeps fewer conditions, and a task without a kept condition is

Table 5: **Per-task success at the standard initial states and on the reference-selected conditions.**

Task	π_0		$\pi_{0.5}$		SmolVLA		SmolVLA-LP		X-VLA	
	Standard	Ref.	Standard	Ref.	Standard	Ref.	Standard	Ref.	Standard	Ref.
Default executed actions										
0	60.0	1.0	80.0	33.0	81.7	4.3	85.0	3.3	100.0	11.7
1	71.7	8.3	98.3	61.1	73.3	12.8	83.3	24.4	100.0	21.7
2	88.3	4.2	90.0	38.3	80.0	0.8	81.7	9.2	100.0	8.3
3	68.3	0.8	73.3	59.2	68.3	1.7	70.0	9.2	100.0	0.0
4	66.7	7.5	96.7	10.0	78.3	20.0	100.0	1.7	100.0	14.2
5	61.7	0.0	66.7	43.3	50.0	5.0	81.7	25.8	95.0	23.3
6	83.3	28.3	93.3	47.5	86.7	26.7	100.0	31.7	100.0	41.7
7	38.3	16.7	91.7	35.0	83.3	31.7	81.7	23.3	95.0	68.3
8	93.3	15.0	96.7	26.7	76.7	15.4	98.3	14.2	100.0	27.5
9	73.3	4.2	70.0	47.5	70.0	15.0	90.0	20.8	100.0	5.8
Mean	70.5	8.6	85.7	40.2	74.8	13.3	87.2	16.4	99.0	22.3
Ten executed actions										
0	80.0	2.7	100.0	82.3	91.7	21.7	96.7	44.3	100.0	14.7
1	95.0	14.4	100.0	83.3	96.7	43.9	100.0	72.8	100.0	18.9
2	98.3	11.3	95.0	82.9	95.0	11.7	100.0	45.0	100.0	10.8
3	85.0	4.2	93.3	90.0	95.0	17.5	98.3	48.3	100.0	0.8
4	88.3	5.0	98.3	75.0	93.3	30.0	98.3	25.0	100.0	11.7
5	76.7	7.5	98.3	97.5	90.0	32.5	96.7	83.3	100.0	15.0
6	93.3	35.8	100.0	100.0	98.3	76.7	98.3	88.3	100.0	45.0
7	63.3	18.3	96.7	93.3	95.0	41.7	98.3	70.0	83.3	73.3
8	100.0	16.7	100.0	99.6	90.0	52.1	100.0	73.8	100.0	26.3
9	93.3	13.3	100.0	90.8	98.3	41.7	98.3	59.2	95.0	5.0
Mean	87.3	12.9	98.2	89.5	94.3	36.9	98.5	61.0	97.8	22.1
Change in the gap from the default to ten executed actions, with its 95% interval										
	$+12.5$ $[+8.1, +17.0]$		-36.8 $[-41.3, -32.4]$		-4.1 $[-10.1, +2.3]$		-33.3 $[-37.3, -29.2]$		-1.1 $[-4.4, +2.0]$	

left undefined, not scored as zero. Where conditions remain, the standard score is the higher of the two in 108 of the 120 model–task pairs at the default horizons and in 266 of the 284 at ten actions. In all others, the model succeeds in every episode on both sets. Overall, the threshold and the reference change how many conditions are kept, but no pair has reference-selected success above its standard score.

Table 6: **Success on the reference-selected conditions at three thresholds.** Standard success averages the ten LIBERO-Object tasks; each threshold column averages the tasks that keep a condition (tasks 3 and 4 keep none at 90%). At $h = 10$ the kept conditions are those selected at the default horizon, as in Table 2.

Model	Default h				$h = 10$			
	Standard	Threshold			Standard	Threshold		
		50%	75%	90%		50%	75%	90%
π_0	70.5	6.4	8.6	12.3	87.3	10.3	12.9	19.0
$\pi_{0.5}$	85.7	37.1	40.2	48.4	98.2	86.7	89.5	96.0
SmolVLA	74.8	10.5	13.3	17.7	94.3	31.3	36.9	49.6
SmolVLA-LP	87.2	14.8	16.4	21.8	98.5	56.3	61.0	74.4
X-VLA	99.0	16.0	22.3	35.0	97.8	15.7	22.1	36.1
Tasks with a kept condition	–	10	10	8	–	10	10	8
Pairs with standard above	–	50/50	50/50	40/40	–	50/50	49/50	37/40

B.4 Fine-tuning in the adaptation case study

The base model is the public $\pi_{0.5}$ LIBERO checkpoint, which was fine-tuned on all four LIBERO suites before this study. For each of tasks 0, 5 and 8, a scripted controller with access to the simulator state records 200 demonstrations, four from each stored initial state. The controller moves the end effector and the gripper and never rotates, and every demonstration uses the task's own instruction. Table 7 lists the fine-tuning hyperparameters. The regimes share all of them except the demonstrations, the trained parameters and, in two regimes, the number of optimizer steps.

The seven regimes differ in the demonstrations they see and in the parameters they train. They use all 200 demonstrations, the first 50, or a random 50, and they train either the whole model or only the action expert with the vision-language backbone frozen. The first 50 demonstrations start once from every stored state, and the three random subsets cover 35, 38 and 35 of the 50 states. Two regimes scale the number of updates with the number of demonstrations. Each regime is trained three times per demonstration set, so a random-subset regime has nine fine-tuned models.

Every regime recomputes the normalization statistics of actions and states on the task's 200 demonstrations. Because the demonstrations never rotate, the recomputed standard deviation of the rotation actions is zero, which fixes the fine-tuned model's rotation output at zero. Each fine-tuned model is evaluated at its last checkpoint with five executed actions, following Appendix A.1. The standard initial states 0 to 19 used in evaluation are also starts of training demonstrations, and no demonstration starts from a translated condition.

The 21 task–regime contrasts form one family. Each contrast is the change in reference-selected success from the base model, and its one-sided 95% upper bound is the estimate plus c times the contrast's bootstrap standard error (20,000 resamples of initial states and training runs). The critical value $c = 2.62$ is the $\lceil 0.95\,(B+1) \rceil$-th order statistic, over the B resamples, of the largest one-sided deviation across the family, with each contrast's deviation divided by its own bootstrap standard error. No contrast has a zero standard error. Every upper bound lies below zero, the largest at -13.5 pp. The 90% interval of the change in the standard score lies within ± 10 points in 15 settings and within ± 5 points in 7 (Table 8).

Table 8 gives every regime on the three tasks.

Table 7: **Fine-tuning hyperparameters of the adaptation case study.**

Hyperparameter	Value
Optimizer	AdamW
Learning rate	2.5×10^{-5}, 1,000 warmup steps, then decay
Batch size	16
Precision	bfloat16
Optimizer steps	12,000 (12,000 $\times$ $n/200$ in the two scaled regimes)
Trained parameters	whole model (4.14B) or action expert (0.69B)
Demonstrations	200 per task, or a subset of 50
Training runs	3 per demonstration set

Table 8: **Adaptation regimes on tasks 0, 5 and 8.** The change is in reference-selected success relative to the base model, and "Within" gives the smaller band, ± 5 or ± 10 points, that contains the 90% interval of the change in the standard score.

Demonstrations	Trained part	Updates	Standard	Ref.-selected	Change	Within
Task 0						
Base model			97.0	74.0	–	
All 200	Whole model	Fixed	100.0	29.0	−45.0 [−55.7, −34.1]	10 pp
All 200	Action expert	Fixed	97.5	12.1	−61.9 [−70.3, −53.2]	5 pp
First 50	Whole model	Fixed	100.0	31.4	−42.6 [−54.0, −30.1]	10 pp
Random 50	Whole model	Fixed	99.0	30.0	−44.0 [−57.1, −31.0]	10 pp
Random 50	Action expert	Fixed	97.9	11.5	−62.5 [−70.4, −53.9]	10 pp
Random 50	Whole model	Scaled	97.2	49.0	−25.0 [−33.6, −16.2]	10 pp
Random 50	Action expert	Scaled	92.3	6.8	−67.2 [−74.7, −59.1]	No
Task 5						
Base model			100.0	93.3	–	
All 200	Whole model	Fixed	100.0	34.7	−58.6 [−77.8, −42.5]	5 pp
All 200	Action expert	Fixed	99.5	16.1	−77.2 [−89.2, −63.6]	5 pp
First 50	Whole model	Fixed	100.0	4.2	−89.2 [−96.7, −79.2]	5 pp
Random 50	Whole model	Fixed	100.0	31.8	−61.6 [−76.1, −48.1]	5 pp
Random 50	Action expert	Fixed	99.4	13.2	−80.1 [−90.8, −67.5]	5 pp
Random 50	Whole model	Scaled	90.1	26.0	−67.3 [−83.5, −51.4]	No
Random 50	Action expert	Scaled	95.5	12.1	−81.2 [−90.7, −69.2]	No
Task 8						
Base model			97.5	97.1	–	
All 200	Whole model	Fixed	100.0	6.8	−90.3 [−95.8, −82.9]	10 pp
All 200	Action expert	Fixed	76.0	11.9	−85.1 [−91.2, −78.2]	No
First 50	Whole model	Fixed	98.2	2.4	−94.7 [−98.8, −88.7]	5 pp
Random 50	Whole model	Fixed	98.9	6.7	−90.4 [−96.0, −83.6]	10 pp
Random 50	Action expert	Fixed	74.3	11.3	−85.7 [−91.7, −78.8]	No
Random 50	Whole model	Scaled	99.4	23.1	−73.9 [−82.2, −65.6]	10 pp
Random 50	Action expert	Scaled	91.3	14.5	−82.6 [−88.5, −75.5]	No

B.5 Normalization during fine-tuning

Section 4.1 notes that the loss after fine-tuning persists when the normalization statistics are held fixed. Recomputing the statistics changes what the model receives and emits before any update, so the loss could come from the statistics rather than from the weight updates. We separate the two with two controls.

First, we repeat the regime with all 200 demonstrations and the whole model, but keep the base model's statistics during fine-tuning and evaluation (Tables 9 and 10). With five executed actions, success on the reference-selected conditions still falls by 46.8 to 92.2 points, with every interval below zero, within 5.3 points of the loss with recomputed statistics. With ten executed actions the loss is 60.2 to 92.8 points, and every interval lies below zero. On task 8 the standard score also falls, to 74.3%, so there the standard initial states show part of the loss.

Second, we evaluate the base model without any update, with the recomputed statistics swapped in (Table 10). Swapping the input statistics alone changes success by amounts whose intervals include zero on all three tasks, and swapping both costs 17.3 to 47.5 points on all three tasks, mostly through the output statistics. The last row sets only the rotation output to zero, as the recomputed output statistics do. The statistics alone therefore cost up to 47.5 points, and the loss that remains with fixed statistics shows that the weight updates also reduce success on the reference-selected conditions.

Table 9: **Fine-tuning with the base model's normalization statistics.** Five executed actions; "Recomputed" is the procedure of Table 8.

Task	Standard			Reference-selected		
	Base	Recomputed	Fixed	Base	Recomputed	Fixed
0	97.0	100.0	100.0	74.0	29.0	27.2
5	100.0	100.0	100.0	93.3	34.7	29.4
8	97.5	100.0	74.3	97.1	6.8	4.9

Table 10: **Normalization controls as differences from the base model.** Change in reference-selected success, in points, with 95% intervals, at five executed actions.

Control	Task 0	Task 5	Task 8
Fine-tuned, fixed statistics	−46.8 [−55.3, −38.4]	−63.9 [−77.2, −50.0]	−92.2 [−98.5, −83.3]
Input statistics, no update	+2.0 [−4.0, +8.3]	+5.8 [+0.0, +14.2]	−0.4 [−5.0, +5.0]
Output statistics, no update	−9.7 [−18.7, +0.0]	−47.5 [−57.5, −37.5]	−43.3 [−55.0, −31.2]
Both, no update	−17.3 [−26.0, −8.7]	−47.5 [−57.5, −37.5]	−42.5 [−52.5, −31.2]
Rotation set to zero, no update	+2.0 [−6.3, +9.7]	−69.2 [−80.0, −57.5]	−28.7 [−40.8, −17.5]

C Instruction

In this section, we describe how LIBERO-Plus builds its instructions and give the per-model results, the text controls and the second evaluation of Section 4.2.

C.1 The appended text and its removal

LIBERO-Plus builds most variant instructions by appending the parameters that generated the variant to the task sentence, for example the camera, the initial state, the lighting, the layout and the noise level. The variants that rewrite the language carry no such text. We classify each variant by its task metadata, and Table 11 counts the variants whose instruction differs from the task sentence. The repair removes the appended parameters from these variants and leaves every rewritten instruction unchanged.

Rewritten-language variants. Because the repair leaves their instructions unchanged, these variants show how much the repaired run differs from the original run for reasons other than the text. Their

success changes by at most 1.0 point for every model and suite. This bounds the difference between the two runs on these variants only, and Appendix C.6 measures it over repeated runs.

Table 11: **LIBERO-Plus variants whose instruction carries appended parameters.**

Suite	Variants	With appended text	Rewritten language
Object	2,518	2,164	354
Spatial	2,402	2,012	390
Goal	2,591	2,181	410
Long	2,519	2,136	383
Total	10,030	8,493	1,537

C.2 Results for all models

Table 16 gives the success of every model with the original and the repaired instruction on every suite, together with the change and its interval. Each interval resamples variants and keeps the two runs of a variant together (10,000 resamples). The resampling unit is the variant, so the interval does not cover the variation between repeated runs of a variant, which Appendix C.6 measures. Appendix E.2 resamples base tasks. The eight models other than the SmolVLA pair change by at most 8.2 points, and 14 of their 32 model–suite intervals exclude zero.

OpenVLA-OFT. The two OpenVLA-OFT variants run through their own evaluation code, which records no outcome when a run aborts. They are therefore paired on 2,396 to 2,590 of the 2,402 to 2,591 variants of a suite. Counting every missing outcome as a success or as a failure moves an OpenVLA-OFT change by at most 2.6 points. On the variants that both OpenVLA-OFT models complete under both instructions, 94.7 to 99.8% of each suite, no model's change moves by more than 0.4 points.

C.3 Length-matched filler

The repair shortens the instruction, so we test whether a large loss needs the parameter values themselves. We append neutral filler to the repaired instruction, as many words as the repair removed, drawn from the twenty neutral words of Appendix C.4. We compare the original, repaired and padded instructions for the two SmolVLA models on the 100 variants per suite of Appendix C.6, with its three seeded repetitions. Table 12 gives the results.

The filler lowers success in all 8 of the 8 model–suite pairs, by 11.7 to 51.0 points, most of what the parameters cost, so the parameter text itself is not needed for a large loss. Replacing the parameters with the filler moves success by at most 3.7 points. This change excludes zero in 2 of the 8 pairs, and in the others the intervals are too wide to tell the two apart. The word count is matched but the token count is not: the padded instruction has fewer tokens than the original in 8,493 of the 10,002 distinct instruction strings among the 10,030 variants, and it still reproduces most of the loss. Both SmolVLA models truncate the instruction at 48 tokens, and only one of the 10,002 original instructions is longer.

Table 12: **Original, repaired and padded instructions for the SmolVLA models.** The padding cost is repaired minus padded success, and the replacement effect is padded minus original success.

Model	Suite	Original	Repaired	Padded	Padding cost	Replacement
SmolVLA	Object	14.3	31.0	13.3	+17.7 [+14.8, +20.5]	−1.0 [−3.5, +1.5]
	Spatial	17.0	39.0	13.3	+25.7 [+15.3, +36.0]	−3.7 [−9.4, +2.1]
	Goal	1.7	34.0	5.0	+29.0 [+26.5, +31.5]	+3.3 [+1.9, +4.8]
	Long	9.3	24.0	12.3	+11.7 [+4.1, +19.3]	+3.0 [−4.5, +10.5]
SmolVLA-LP	Object	10.0	62.3	11.3	+51.0 [+46.7, +55.3]	+1.3 [−4.4, +7.1]
	Spatial	18.0	46.7	19.3	+27.3 [+17.3, +37.4]	+1.3 [−12.8, +15.5]
	Goal	7.7	34.3	7.7	+26.7 [+22.9, +30.5]	+0.0 [−4.3, +4.3]
	Long	13.3	51.3	16.3	+35.0 [+28.4, +41.6]	+3.0 [+0.5, +5.5]

C.4 NEUTRAL WORDS AFTER THE BENCHMARK'S INSTRUCTION

Section 4.2 appends k ordinary English words to the instructions of standard LIBERO-Object. The words name no object, direction or action. They come from one fixed cycle of twenty adverbs: *perhaps, otherwise, certainly, usually, rather, somewhat, eventually, briefly, apparently, indeed, thereby, likewise, moreover, namely, hence, besides, altogether, instead, presumably, meanwhile*. Each instruction has a fixed start position in the cycle, set by a hash of the instruction, and its suffix of k words is the next k words from there, so a longer suffix extends every shorter one. For example, *pick up the alphabet soup and place it in the basket* becomes *pick up the alphabet soup and place it in the basket instead presumably* at $k = 2$. The LIBERO-Goal control of Appendix C.5 uses the same rule. Table 13 gives success for every length. Each length, including $k = 0$, runs the 500 standard episodes at each model's default horizon, and the run with $k = 0$ reproduces the default of Table 17 exactly. With the longest suffix, the instructions have 42 to 44 tokens, below the length at which the SmolVLA models truncate.

Table 13: **Success on LIBERO-Object with neutral words appended.**

Model	Words appended			
	0	2	8	32
π_0	65.4	60.6	64.8	59.6
$\pi_{0.5}$	83.2	82.6	81.0	66.8
SmolVLA	69.0	37.2	4.0	0.4
SmolVLA-LP	87.4	11.4	4.2	0.0
VLA-JEPA	99.8	99.8	100.0	100.0
MolmoAct2	99.8	99.8	99.0	99.0
EVO-1	83.6	83.8	85.0	84.6
X-VLA	99.2	99.2	99.4	99.2

C.5 INSTRUCTIONS ON LIBERO-GOAL

LIBERO-Goal places all ten tasks in one shared scene. We run the 500 standard episodes of each model under five instructions: the task's own, the same with two or eight neutral words appended, the instruction of another task, and an empty string. The other task is fixed, and task i receives the instruction of task $i + 1$ modulo ten. Success counts completion of the evaluated task, and under another task's instruction we also record whether the named task is completed. Both goal predicates are checked after every action of the same episode, and a task counts as completed when its predicate is false in the initial state and becomes true within the step budget, which is the same for all five instructions. Episodes in which a predicate already holds at the start stay in the denominator. Table 14 gives the results. Given another task's instruction, every model completes the named task in 75.8 to 98.8% of episodes and the evaluated task in none. Two neutral words change the success of SmolVLA and SmolVLA-LP by -61.6 and -51.2 points, and eight words change that of π_0 by -12.4 points. Eight words change the success of the other five models by at most 7.2 points. With an empty instruction, every model succeeds in fewer than one in ten episodes.

Table 14: **Success on LIBERO-Goal under five instructions.** "Named task" is the fraction of episodes that complete the task another instruction names.

Model	Own	Neutral words		Another task's instruction		Empty
		2	8	Own task	Named task	
π_0	81.2	78.6	68.8	0.0	75.8	0.0
$\pi_{0.5}$	87.4	87.8	84.2	0.0	86.4	4.8
SmolVLA	77.6	16.0	2.4	0.0	82.8	0.0
SmolVLA-LP	87.2	36.0	1.8	0.0	86.0	0.0
VLA-JEPA	96.8	97.2	97.6	0.0	97.4	4.6
MolmoAct2	97.6	97.8	95.6	0.0	97.8	5.8
EVO-1	89.0	84.4	81.8	0.0	89.6	3.6
X-VLA	97.6	98.8	98.2	0.0	98.8	9.2

C.6 REPLICATION WITH SEEDED EPISODES

Section 4.2 reports that the large instruction effects replicate in a second evaluation. It reruns π_0, $\pi_{0.5}$, SmolVLA and SmolVLA-LP on 100 variants per suite under both instructions, at 50 and at 10 executed actions. The variants were chosen before any result, in equal numbers from each task group of the suite at evenly spaced positions in the benchmark's listing. Each of three repetitions has its own seed, and at every reset the random number generators are reseeded from the repetition, the suite, the variant and the episode. Within a repetition, the two instructions and the two horizons of a variant are therefore paired. Its intervals are 95% t-intervals over the three repetitions. Table 15 gives the change at each horizon and the difference between them.

We count an effect of more than 5 points in the full study as reproduced when the change at 50 actions has the same sign and its interval excludes zero. All 9 such effects are reproduced. No difference between the two horizons has an interval that excludes zero, so the second evaluation does not resolve whether the instruction effect depends on the horizon. On Goal, half of the subset consists of severity variants, which make up 7% of the suite.

Table 15: **Instruction repair in the second evaluation.** An asterisk marks an effect of more than 5 points in the full study.

Model	Suite	Change, 50 actions	Change, 10 actions	Difference
π_0	Object	+3.7 [−2.6, +9.9]	+8.0 [+3.7, +12.3]	+4.3 [−2.8, +11.5]
	Spatial	+3.7 [+0.8, +6.5]	−0.3 [−5.5, +4.8]	−4.0 [−11.5, +3.5]
	Goal	+8.0 [+3.0, +13.0]*	+6.7 [+0.9, +12.4]	−1.3 [−8.9, +6.3]
	Long	+5.3 [−3.4, +14.1]	+7.7 [−4.6, +19.9]	+2.3 [−12.8, +17.5]
$\pi_{0.5}$	Object	+1.3 [−10.1, +12.8]	−1.7 [−3.1, −0.2]	−3.0 [−13.8, +7.8]
	Spatial	−1.0 [−5.3, +3.3]	−1.0 [−5.3, +3.3]	+0.0 [−4.3, +4.3]
	Goal	+2.0 [−7.0, +11.0]	+0.0 [−4.3, +4.3]	−2.0 [−12.8, +8.8]
	Long	+6.3 [−5.9, +18.6]	+0.0 [−4.3, +4.3]	−6.3 [−14.3, +1.7]
SmolVLA	Object	+16.7 [+15.2, +18.1]*	+18.3 [+5.8, +30.8]	+1.7 [−9.5, +12.9]
	Spatial	+22.0 [+7.1, +36.9]*	+12.3 [−3.4, +28.1]	−9.7 [−39.3, +20.0]
	Goal	+32.3 [+30.9, +33.8]*	+32.3 [+28.5, +36.1]	+0.0 [−4.3, +4.3]
	Long	+14.7 [+13.2, +16.1]*	+20.3 [+12.3, +28.3]	+5.7 [−1.5, +12.8]
SmolVLA-LP	Object	+52.3 [+47.2, +57.5]*	+49.3 [+47.9, +50.8]	−3.0 [−8.0, +2.0]
	Spatial	+28.7 [+9.4, +48.0]*	+19.3 [+16.5, +22.2]	−9.3 [−29.6, +10.9]
	Goal	+26.7 [+25.2, +28.1]*	+31.3 [+25.1, +37.6]	+4.7 [−1.6, +10.9]
	Long	+38.0 [+33.7, +42.3]*	+42.7 [+23.4, +62.0]	+4.7 [−16.2, +25.5]

Table 16: **Success with the original and the repaired instruction on LIBERO-Plus.**

Model	Suite	Original	Repaired	Change	Paired variants
π_0	Object	43.9	46.7	+2.8 [+0.9, +4.6]	2,518
	Spatial	49.0	49.6	+0.6 [−1.5, +2.8]	2,402
	Goal	48.1	56.3	+8.2 [+6.2, +10.2]	2,591
	Long	27.1	29.7	+2.6 [+0.5, +4.7]	2,519
$\pi_{0.5}$	Object	54.2	55.3	+1.1 [−0.5, +2.7]	2,518
	Spatial	66.9	66.7	−0.2 [−1.8, +1.2]	2,402
	Goal	64.2	65.0	+0.8 [−0.7, +2.4]	2,591
	Long	65.3	64.7	−0.6 [−2.1, +0.9]	2,519
SmolVLA	Object	14.9	37.2	+22.3 [+20.5, +24.1]	2,518
	Spatial	17.7	35.4	+17.7 [+15.5, +19.9]	2,402
	Goal	3.8	44.7	+40.8 [+38.9, +42.8]	2,591
	Long	11.0	29.2	+18.3 [+16.4, +20.0]	2,519
SmolVLA-LP	Object	7.3	59.0	+51.7 [+49.7, +53.7]	2,518
	Spatial	21.8	50.2	+28.4 [+26.0, +30.8]	2,402
	Goal	8.0	56.6	+48.6 [+46.6, +50.6]	2,591
	Long	13.0	51.2	+38.2 [+35.9, +40.3]	2,519
VLA-JEPA	Object	80.4	83.0	+2.5 [+1.5, +3.5]	2,518
	Spatial	87.7	86.8	−0.8 [−2.0, +0.3]	2,402
	Goal	76.3	78.4	+2.1 [+1.0, +3.2]	2,591
	Long	66.5	67.6	+1.2 [−0.2, +2.5]	2,519
MolmoAct2	Object	87.4	89.8	+2.4 [+1.4, +3.5]	2,518
	Spatial	91.6	91.0	−0.6 [−1.5, +0.4]	2,402
	Goal	80.0	81.8	+1.8 [+0.7, +2.9]	2,591
	Long	80.3	81.6	+1.3 [−0.1, +2.7]	2,519
EVO-1	Object	42.2	45.7	+3.5 [+2.2, +4.8]	2,518
	Spatial	51.9	51.5	−0.3 [−1.7, +1.0]	2,402
	Goal	47.3	50.3	+3.0 [+1.6, +4.5]	2,591
	Long	45.4	47.1	+1.7 [+0.2, +3.3]	2,519
X-VLA	Object	70.3	70.5	+0.2 [−0.5, +1.0]	2,518
	Spatial	73.7	74.3	+0.6 [−0.3, +1.5]	2,402
	Goal	68.3	68.8	+0.5 [−0.6, +1.7]	2,591
	Long	69.4	67.7	−1.7 [−2.8, −0.6]	2,519
OpenVLA-OFT	Object	68.9	69.4	+0.5 [−0.7, +1.6]	2,513
	Spatial	85.5	87.0	+1.5 [+0.6, +2.5]	2,402
	Goal	64.0	68.4	+4.4 [+2.9, +5.8]	2,590
	Long	68.0	68.5	+0.5 [−0.9, +1.9]	2,423
OpenVLA-OFT-LP	Object	83.2	83.4	+0.2 [−0.6, +1.0]	2,517
	Spatial	86.4	87.4	+1.0 [+0.2, +1.8]	2,396
	Goal	70.0	70.0	+0.0 [−1.0, +1.0]	2,587
	Long	82.5	82.5	+0.0 [−1.2, +1.2]	2,482

D EXECUTED ACTIONS

This section gives the full execution grid of Section 4.3 on standard LIBERO and in SimplerEnv.

Standard LIBERO. Table 17 gives the success of every model at 5, 10 and 25 executed actions and at its default. We use ten as the common setting because every model in the study can execute it and it lies within the range used by the models' own evaluations, 5 to 30 actions. The public π_0 and $\pi_{0.5}$ configurations execute 50 actions, while the LIBERO example in the models' own code base executes five (Physical Intelligence, 2025). The two settings differ by 7.2 to 14.4 points on $\pi_{0.5}$ and by 0.6 to 12.8 points on π_0 across the four suites.

Published results. The LeRobot LIBERO documentation[3] reports results of $\pi_{0.5}$ at ten executed actions for `lerobot/pi05_libero_finetuned`, obtained with LeRobot's evaluation loop. In the LeRobot version we use (0.5.2), this loop skips an initial state after each success, so the states it evaluates depend on the outcomes. For this comparison we evaluate the checkpoint of Table 3 with that loop. With ten executed actions, $\pi_{0.5}$ reaches 99.6, 96.6, 97.8 and 95.8 on Object, Spatial, Goal and Long, against 99.0, 97.0, 98.0 and 96.0 in the documentation. Table 17 gives the same setting on the fixed order of Appendix A.1. With its own evaluation code, OpenVLA-OFT reaches 97.0, 99.0, 97.6 and 94.8, against the reported 98.4, 97.6, 97.9 and 94.5.

Table 17: **Success on standard LIBERO at each number of executed actions.** The default is 50 for every model except X-VLA, whose default is 30.

Model	Executed actions h			
	$h = 5$	$h = 10$	$h = 25$	Default
Object				
π_0	75.6	80.8	79.2	65.4
$\pi_{0.5}$	97.6	99.4	97.6	83.2
SmolVLA	90.0	92.8	84.8	69.0
SmolVLA-LP	96.8	98.0	94.8	87.4
EVO-1	93.6	99.8	96.6	83.6
X-VLA	94.6	98.2	99.4	99.2
Spatial				
π_0	71.6	69.0	69.8	58.8
$\pi_{0.5}$	96.8	95.4	91.0	84.0
SmolVLA	76.6	75.4	70.8	58.8
SmolVLA-LP	92.4	91.6	90.2	83.4
EVO-1	92.2	91.6	85.2	78.6
X-VLA	87.2	94.4	92.4	92.2
Goal				
π_0	81.8	84.6	83.4	81.2
$\pi_{0.5}$	94.6	95.2	91.0	87.4
SmolVLA	85.6	89.0	83.6	77.6
SmolVLA-LP	95.0	95.4	91.8	87.2
EVO-1	89.8	95.4	93.6	89.0
X-VLA	86.2	93.2	97.8	97.6
Long				
π_0	42.6	43.4	38.8	38.6
$\pi_{0.5}$	94.2	94.2	93.4	82.0
SmolVLA	65.0	70.6	64.8	46.8
SmolVLA-LP	82.8	85.8	82.8	66.2
EVO-1	86.8	91.8	89.0	76.4
X-VLA	58.2	91.0	93.6	91.0

[3] https://huggingface.co/docs/lerobot/libero, section "Reproducing published results".

D.1 Execution responses in SimplerEnv

The SimplerEnv evaluation code (Li et al., 2025) sets the number of actions a model executes between observations, so the intervention of Section 4.3 applies without change. We evaluate the X-VLA checkpoints for WidowX and for Google Robot (Zheng et al., 2026) and DB-CogACT on WidowX (Li et al., 2024; Xie et al., 2025), which predicts 16 actions per call. Table 18 gives the mean success over each configuration's tasks. Each of the three configurations does worst at its shortest setting, and one of them peaks in the middle of its range. As a reproduction check, X-VLA succeeds in 92 of 96 episodes on the WidowX visual-matching set, against 91 of 96 in the outcomes distributed with the evaluation code.

Table 18: **Success on SimplerEnv at each number of executed actions.**

Model	Robot	Executed actions h			
		1	5	10	30
DB-CogACT	WidowX	8.3	42.7	45.8	–
X-VLA	Google Robot	–	44.4	51.4	48.6
X-VLA	WidowX	–	0.0	30.2	93.7

D.2 Predictive validity for real-robot rankings

A separate study, with its own checkpoints, asks whether success on reference-selected conditions predicts real-robot rankings. It compares, for five checkpoints over 3,876 episodes, how well success on reference-selected translated conditions in SimplerEnv and the standard score predict published real-robot orderings. The simulator reproduces the published simulator results, and the reference succeeds on the kept conditions. Against the real-robot success rates, both orderings have a Kendall's τ of 0.60. The mean maximum rank violation (Li et al., 2025) is 0.064 for the standard score and 0.056 on the reference-selected conditions, and the 95% interval of the difference is $[-0.09, +0.10]$.

E Model Comparisons

This section gives the margins behind the reversals of Section 4.4, their resampling by base task, the state comparisons at the default horizon, and the residuals of the common offset.

E.1 Reversals under the instruction repair

Table 19 gives the two model pairs that reverse on LIBERO-Plus Object, at 50 and at 10 executed actions. Figure 8 places every pair of models by its margin before and after each change, and pairs in the shaded quadrants change order. Over the 45 pairs of the ten models, the repair moves 17, 17, 16 and 17 pairs by at least 10 points on Object, Spatial, Goal and Long. Each of these pairs contains SmolVLA or SmolVLA-LP, and none of the 28 pairs without them moves that far or reverses. The repair reverses 0 to 4 pairs per suite. The execution change reverses 0 of the 40 pair–suite combinations of the five models whose default is fifty, and 2 of the 60 once X-VLA is included. Here a reversal requires the margins at the two settings to have opposite signs with both 95% intervals excluding zero, and the point estimates change sign in 4 and 12 of these combinations.

E.2 Resampling base tasks

The 10,030 variants are generated from forty base tasks, so variants of one base task share its difficulty. Table 20 repeats the counts with the base task as the resampling unit: each draw takes base tasks with replacement and keeps all of their variants. The intervals widen two- to fourfold. Every pair whose margin the repair moves by at least 10 points still has an interval that excludes zero on Object, Goal and Long, and 14 of 17 do on Spatial. Five of the seven reversals remain.

Table 19: **Margins before and after the instruction repair on LIBERO-Plus Object.** The margin is the first model's success minus that of SmolVLA-LP, over all 2,518 Object variants with every outcome at both horizons (not the subset of Appendix C.6); 95% intervals resample variants or base tasks ("tasks"; 10,000 draws each), and a pair is reversed under a unit when both of its intervals exclude zero with opposite signs.

Pair		Original margin		Repaired margin		Reversed
	h	variants	tasks	variants	tasks	variant / task
π_0 − SmolVLA-LP	50	+36.5 [34.4, 38.6]	[29.0, 43.4]	−12.4 [−14.3, −10.3]	[−21.2, −4.2]	yes / yes
	10	+23.4 [20.9, 25.8]	[12.1, 32.1]	−12.6 [−14.4, −10.8]	[−17.3, −7.9]	yes / yes
$\pi_{0.5}$ − SmolVLA-LP	50	+46.9 [44.8, 49.0]	[40.2, 54.0]	−3.7 [−6.1, −1.3]	[−11.2, 4.7]	yes / no
	10	+58.9 [56.9, 60.8]	[53.8, 63.4]	+13.3 [11.4, 15.3]	[9.5, 16.2]	no / no

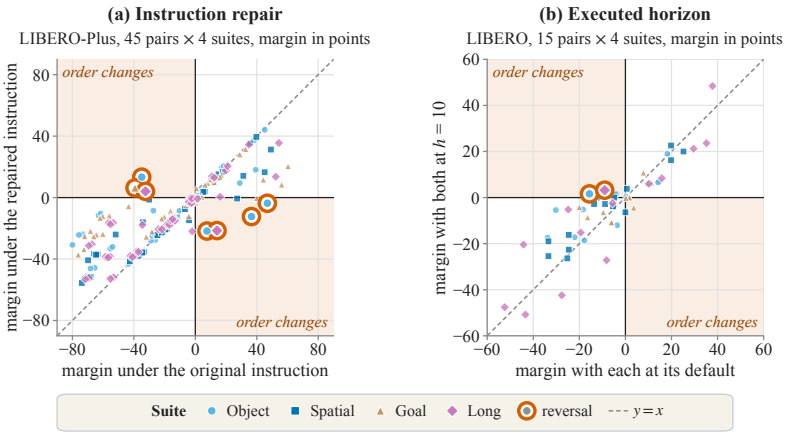

Figure 8: **Margins between models before and after each change.** (a) The instruction repair on LIBERO-Plus. (b) Changing each model's default to ten executed actions on standard LIBERO.

Table 20: **Pairs of models whose change in margin has an interval excluding zero, and reversals.** Intervals resample variants or base tasks.

Suite	Change in margin, interval excludes zero		Reversals	
	Variants	Base tasks	Variants	Base tasks
Object	30	24	4	3
Spatial	23	15	0	0
Goal	33	21	1	1
Long	25	22	2	1

E.3 State comparisons at the default horizon

Section 4.4 reports that the initial-state gaps reverse one ordering of two models averaged over LIBERO-Object. Table 21 gives the difference between every pair of the five ranked models at their default settings, averaged over the ten tasks, at the standard initial states and on the reference-selected conditions. Each of 20,000 draws resamples initial states within each task and applies to every model and condition, and the table prints the percentile interval of each difference. The asterisk uses simultaneous intervals, computed separately for the standard initial states and the reference-selected conditions: two-sided 95% max-$|T|$ intervals over the ten pairs, with each difference studentized by its bootstrap standard deviation, together with Holm-adjusted tests at the 5% level.

Table 21: **Suite-average differences between the ranked models.** The first model minus the second, in points; an asterisk marks a pair whose differences have opposite signs with both simultaneous intervals excluding zero after correction over the ten pairs.

Pair	Standard	Reference-selected
$\pi_{0.5}$ − SmolVLA	+10.8 [+4.0, +17.7]	+26.8 [+22.1, +31.5]
$\pi_{0.5}$ − SmolVLA-LP	−1.5 [−7.0, +4.0]	+23.8 [+19.3, +28.2]
$\pi_{0.5}$ − π_0	+15.2 [+8.3, +22.0]	+31.6 [+27.3, +35.8]
$\pi_{0.5}$ − X-VLA*	−13.3 [−18.0, −9.0]	+17.9 [+13.4, +22.4]
SmolVLA − SmolVLA-LP	−12.3 [−18.8, −5.7]	−3.0 [−7.2, +1.2]
SmolVLA − π_0	+4.3 [−3.3, +12.0]	+4.7 [+1.2, +8.4]
SmolVLA − X-VLA	−24.2 [−30.0, −18.5]	−8.9 [−13.2, −4.7]
SmolVLA-LP − π_0	+16.7 [+10.3, +23.2]	+7.8 [+4.5, +11.1]
SmolVLA-LP − X-VLA	−11.8 [−15.8, −8.0]	−5.9 [−10.1, −1.7]
π_0 − X-VLA	−28.5 [−34.2, −23.2]	−13.7 [−17.3, −10.0]

E.4 Common offset

Section 4.4 summarizes each change by the largest deviation that remains after subtracting the one offset that fits all models best. This residual is half the range of the models' changes, and it is zero when every model responds equally. Table 22 gives it for each change and suite, with the state gap at each model's default horizon and at ten executed actions. Without the two SmolVLA models, the instruction residual falls to at most 4.1 points from 25.7 points, and the execution residual falls from 7.2–12.4 to 5.4–8.6 points, so the other models still respond by different amounts. A change shared by all models would still leave a residual from sampling at the same sample sizes.

Table 22: **Residual of the best common offset, in points.**

Change	Models	Object	Spatial	Goal	Long
Initial state (default actions)	5	15.6	–	–	–
Initial state (ten actions)	5	33.5	–	–	–
Instruction	10	25.7	14.6	24.3	19.9
Instruction, without the SmolVLA pair	8	1.6	1.2	4.1	2.1
Executed actions, LIBERO	6	12.4	7.2	7.9	11.9

F A Toy Non-identifiability Construction

The state comparison has a simple limiting case in which even perfect measurement on the evaluated initial states does not necessarily determine performance outside them.

Proposition 1. *Let an evaluation observe only trajectories initialized in a set C. There exist deterministic dynamics and two policies whose trajectory distributions agree for all initial states in C, but whose success probabilities differ on states outside C that another policy solves.*

Proof. Consider $x_{t+1} = 1.1x_t + a_t$ with absorbing failure at $|x| \geq 1$, let an episode succeed when the state does not reach the failure set within a horizon of $T \geq 8$ steps, and let $C = (-0.5, 0.5)$.

Define

$$\pi_g(x) = -0.6x, \qquad \pi_b(x) = \begin{cases} -0.6x, & |x| < 0.5, \\ 0, & |x| \geq 0.5. \end{cases}$$

For every initial state in C, both policies induce $x_{t+1} = 0.5x_t$ and remain in C. Their actions, states, and success outcomes therefore agree. The same holds for every metric computed from those trajectories. Let the wider evaluation distribution be uniform on $[-0.95, 0.95]$. Policy π_g contracts to the origin from every such state and succeeds with probability one. Under π_b, states with $|x_0| < 0.5$ contract, while the remainder move outward and fail within eight steps. Its survival probability is $1/1.9 \simeq 0.526$. All states in the wider distribution are recoverable because π_g solves them. Equality of the trajectory distributions from C therefore does not identify success on the wider distribution. □