The Recovery Margin Is a Safety Margin: Auditing What Saturated Benchmarks Hide in Vision-Language-Action Policies

Socrates Osorio*
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, USA
Email: socratesj.osorio@berkeley.edu

Joy Zheyun Yang*
University of Oxford
Oxford, United Kingdom
Email: joy@robots.ox.ac.uk

***Abstract*—A vision-language-action (VLA) policy that scores near-perfect from benchmark initial states looks ready to deploy, but from-start success cannot see whether the policy can recover once the robot or scene drifts slightly off the nominal trajectory, the regime that determines whether a deployed system fails safely. We argue that this local recovery margin is a first-class, measurable safety property, auditable cheaply from the demonstrations a policy was already trained on. We present CREST (Controlled Reset Evaluation of Suffix Trajectories), a safety-audit layer that reconstructs intermediate states by replaying expert prefixes, retains only states whose expert suffix still solves the task, and reports a matched-denominator conditional completion curve R_σ under small controlled perturbations. On all 10 simulated LIBERO-10 tasks, OPENVLA-OFT completes 49/50 official starts and 178/185 clean intermediate states, yet drops from 82/85 clean to 63/85 under a 0.03-radian robot-joint perturbation, a $+22$ pp paired drop (task-clustered 95% CI $[8.9, 33.3]$), on states from the policy's own fine-tuning data. The gap survives seeds, phases, horizons, the other three LIBERO suites, object-XY perturbations, an IK oracle, a π_0 probe, and policy-reached held-out states. CREST serves as a deployment-time red-team and runtime-assurance audit, while its feasibility-verified state set feeds robust training as a reset and reward signal, giving one substrate that bridges evaluation-time and design-time safety.**

I. Introduction: A Saturated Benchmark Number Is Not a Safety Certificate

Foundation models for robotics are increasingly evaluated as if a single number from the benchmark initial-state distribution answers the central question of whether a policy works. This is convenient but unsafe to rely on, because a from-start rollout conflates at least three abilities that matter very differently for deployment: *reaching* useful intermediate states, *continuing* from those states once reached, and *recovering* when the robot or scene is nudged off the nominal trajectory. A policy can score highly on the first while remaining brittle on the third. Structured robot-evaluation work has argued the same point more broadly: binary success conceals important failures [31].

For safety the third ability is decisive. A deployed manipulator is continually displaced off its nominal path by contact, imperfect servoing, state-estimation drift, and re-registration after lost tracking. Whether the system fails safely or escalates a small disturbance into a damaging one is governed by how much of the local recovery basin the policy actually owns, not by how often it succeeds from a clean reset. This matters more as VLAs saturate nominal benchmarks: RT-style models, OPENVLA, OPENVLA-OFT, Octo, π_0, and action-tokenized variants reach increasingly strong performance [2, 3, 11, 12, 18, 1, 23], with OPENVLA-OFT near-saturating LIBERO [12]. At the ceiling the useful safety question shifts from "can the policy solve the benchmark?" to "where, along a successful trajectory, is the policy still able to recover?" We therefore treat the local recovery margin as a first-class safety property and ask how to *audit* it cheaply and reproducibly for already-trained VLA policies.

The hidden gap, made concrete. On all 10 simulated LIBERO-10 tasks, OPENVLA-OFT completes 49/50 official starts and 178/185 expert-verified intermediate states, but on a matched 85-state late-phase subset it drops from 82/85 clean continuation to 63/85 under a 0.03-radian robot-joint perturbation. The induced end-effector displacement is 0.028 MuJoCo units, and the robot-$qpos$ shift is 23% of the median same-task and same-phase nearest-neighbour distance among expert states. Because the audit states overlap OPENVLA-OFT's fine-tuning data, this is a *local* margin around the policy's own training states, the most favourable possible regime, and it is invisible to the from-start number a deployment decision would rely on.

Position and contribution. Trustworthy embodied foundation models need a safety property that nominal success cannot certify and a cheap protocol to measure it. We propose CREST (Controlled Reset Evaluation of Suffix Trajectories) as that protocol. CREST samples demonstration phases, reconstructs simulator states by replaying expert prefixes, retains only states whose expert suffix still solves the task (*expert-suffix verification*), and evaluates the policy from the clean state and from small controlled perturbations. The same substrate plays two complementary roles (Section IV): a deployment-time red-team that turns nominal demonstrations into an auditable recovery-margin layer, and a training-time reset and reward

*Equal contribution.

source for robust fine-tuning that targets recovery at the source. Our contributions are:

- A position: the local recovery margin is a measurable safety property that from-start success structurally cannot certify, and it should be reported alongside nominal success for any VLA safety case.
- CREST, a demonstration-anchored audit with explicit denominators, usable both as a deployment-time recovery red-team and as a feasibility-verified state distribution for robust training.
- A full LIBERO-10 OpenVLA-OFT package showing high official-start and clean completion alongside a quantified 22-point recovery-margin drop, replicated across seeds, phases, horizons, three additional suites, perturbation modalities, an IK oracle, a π_0 probe, and policy-reached held-out states.

II. Related Work

Robot foundation and VLA policies. Robot learning has moved toward generalist, language-conditioned policies trained on broad mixtures including Open X-Embodiment/RT-X and DROID [19, 10]. RT-1/RT-2 show transformer and vision-language-action training scale across tasks [2, 3], OpenVLA provides an open 7B VLA [11], and OpenVLA-OFT improves LIBERO success and inference speed via fine-tuning design [12]. Octo, π_0, FAST, and Diffusion Policy are complementary generalist, flow, and imitation families [18, 1, 23, 4].

Why imitation-learned policies are fragile off-trajectory. The recovery margin we audit is the safety-relevant face of a classic problem. Ross and Bagnell show imitation-as-supervised-learning is problematic because learner actions shape future states, and DAgger trains on learner-induced states [24, 25]. DART injects demonstration noise [14], backwards models and return-to-demo collection enlarge the recovery basin [22, 21], RecoveryChaining learns recovery controllers [28], and Rewind-IL respawns states for failure recovery [34]. These are *safety-by-design* interventions that grow the basin a policy can recover to. CREST is the *safety-by-practice* counterpart that measures how much of that basin a trained policy owns, and supplies the verified-state distribution those methods need.

Robustness benchmarks and runtime assurance. LIBERO-Plus, LIBERO-PRO, LIBERO-X, and STRONG-VLA perturb object layouts, viewpoints, robot initial states, language, lighting, backgrounds, and sensors, reporting large VLA drops [6, 35, 29, 33]. SIMPLER and AutoEval scale evaluation infrastructure, and PolaRiS provides real-to-sim evaluation [15, 36, 8]. These test whether a policy can *reach* useful states under shifted conditions. CREST is complementary on a distinct safety axis: it tests *continuation* and *recovery* after a verified expert prefix has already reached one. The diagnostics can disagree, and Section VI reports a same-checkpoint slice where the rankings are nearly uncorrelated. We use a fixed perturbation grid rather than adaptive adversaries [17, 30] so the red-team is reproducible and the per-state predicate stable enough to cite in a safety case, while noting adaptive attacks would give tighter worst-case coverage. The argument that binary success hides important failures [31] we sharpen into an auditable safety contract.

III. CREST: A Recovery-Margin Safety Audit

A. Conditional Completion From Verified States

Let π be a policy, T a language-conditioned task, and $d = (s_0, a_0, \ldots, s_H)$ an expert demonstration. For phase $\tau \in [0, 1]$, define $k = \lfloor \tau H \rfloor$. An intervention state is the simulator state reached after replaying the expert prefix $(a_0, \ldots, a_{k-1})$ from the benchmark reset. All reported policy results use this prefix-replay construction. Direct mid-trajectory state injection through the LIBERO/robosuite wrappers validated only 24/40 states in an initial test, vs. 185/240 under prefix replay.

We retain an intervention only if the expert suffix remains valid from it:

$$\mathrm{Valid}(d, \tau) = \mathbf{1}[\mathrm{Rollout}(a_k, \ldots, a_{H-1} \mid \hat{s}_{d,\tau}) \models \text{predicate}], \tag{1}$$

where $\hat{s}_{d,\tau}$ is the prefix-replayed state. The conditional completion rate over a retained set $\mathcal{I}$ is

$$\mathrm{CCR}(\pi; \mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{(d,\tau)\in\mathcal{I}} \mathbf{1}[\mathrm{Rollout}(\pi \mid \hat{s}_{d,\tau}) \models \text{predicate}]. \tag{2}$$

This conditioning is the core measurement contract: policies are scored only on states the demonstrator can still complete, so clean and perturbed trials share matched denominators. In the full LIBERO-10 package, 185 of 240 sampled candidate states pass this validity filter. The filter is deliberately one-sided: passing certifies the state is solvable by at least one action sequence, while failing does not certify infeasibility, so Section VI complements it with an IK-oracle recoverability probe and excluded-state bounds.

B. Perturbation Recovery

Clean continuation alone cannot diagnose recovery. We evaluate small perturbed variants of retained states,

$$\tilde{s}_{d,\tau,\sigma} = P_\sigma(\hat{s}_{d,\tau}), \tag{3}$$

where P_σ modifies simulator state and then refreshes the observation. Our primary perturbation is robot-joint Gaussian noise on the first seven MuJoCo *qpos* entries (the Panda arm joints), so σ is in radians. This is a controlled proxy for configuration error after imperfect servoing, contact, or state-estimation drift, not a claim that real robots receive instantaneous joint teleports. Our secondary perturbation is object-XY noise on free joints. We use small fixed grids rather than adaptive adversaries so that failures are reproducible. For each σ we report

$$R_\sigma(\pi; \mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{(d,\tau)\in\mathcal{I}} \mathbf{1}[\mathrm{Rollout}(\pi \mid P_\sigma(\hat{s}_{d,\tau})) \models \text{predicate}], \tag{4}$$

with R_0 the matched clean rate.

a) Recovery margin (formal definition).: The CREST *recovery margin* is the curve $\mathrm{RM}_P(\pi;\mathcal{I}) = \{R_\sigma(\pi;\mathcal{I}) : \sigma \in \Sigma\}$ plus three derived scalars: (i) the small-disturbance AUC AUC_{Σ_0}, normalized trapezoidal AUC of R_σ over $\Sigma_0 = [0, 0.03]$; (ii) the grid-estimated recovery radius $\sigma^\star_\alpha = \max\{\sigma \in \Sigma : R_\sigma \geq \alpha R_0\}$, default $\alpha = 0.8$; (iii) the clean-to-perturbed gap $R_0 - R_\sigma$ at named σ. "Recovery margin" refers to the curve and its scalars jointly: an empirical audit statistic, not a certified region of attraction in the sense of LQR-tree funnels [26], and a safety case must not over-read it as a guarantee.

C. Controls Built Into the Audit

CREST includes four built-in controls. (i) Expert suffix replay estimates intervention validity. (ii) Zero-action rollouts verify that retained states are not trivially terminal. (iii) Official-start rollouts give the standard benchmark anchor. (iv) Full-task-horizon ablations test whether failures are simply budget-starved. We additionally evaluate two OpenVLA-OFT checkpoints and classic OpenVLA as policy controls. Algorithm 1 (Appendix A) states the protocol compactly.

IV. One Substrate, Two Safety Roles

CREST's core object, the verified-state set $\mathcal{I}$ paired with the conditional predicate R_σ, lets one protocol act on both sides of a safety pipeline. *On the deployment (practice) side*, the small-disturbance AUC and the clean-to-perturbed gap at named σ are runtime-assurance statistics: they quantify how far the system can be pushed before success collapses, on states it will actually pass through. The fixed grid keeps the audit reproducible and citable in a safety case, and the task-level AUC ranking (Fig. 1b) localizes the fragile behaviours a runtime monitor or shielding fallback should watch first. *On the design side*, each retained state is solvable by construction, so $\mathcal{I}$ is a feasibility-verified reset distribution, a seed for backwards-curriculum or DART-style robust training [14, 22, 21] and learned recovery controllers [28, 34]. The same predicate on a perturbed neighbourhood is a scoped per-state recovery reward needing no separate reward model. Because the same machinery filters states the policy reaches on its own (Section VI), the audit re-runs as the policy is hardened, closing the loop between measuring a margin and growing it.

V. Experimental Setup

Benchmark and policies. All 10 simulated LIBERO-10 tasks, public LIBERO demonstrations [16] on robosuite/MuJoCo [37, 27], and the benchmark success predicate. Main policy: task-specific OpenVLA-OFT LIBERO-10 checkpoint. We also evaluate a combined all-suite OpenVLA-OFT checkpoint, classic OpenVLA, and a zero-action sanity policy. Official-start baselines run five initial states per task (50 trials per checkpoint).

Intervention set. Phases $\tau \in \{0, 0.25, 0.5, 0.75\}$. After prefix replay and expert-suffix validation, the full clean set is 185/240 and the late-phase pool is 91/120. The deterministic matched 85-state subset (fixed before policy evaluation, `demos_per_task=5`) spans all 10 tasks with 6–13 late-phase states each.

Perturbation grids. Primary robot-joint grid $\sigma \in \{0.01, 0.03, 0.1, 0.2\}$ rad, where $\sigma = 0.01, 0.03$ are the primary small-disturbance range and $\sigma = 0.1, 0.2$ are stress endpoints. Each σ is repeated with three noise seeds. Object-XY grid $\sigma \in \{0.005, 0.01, 0.02\}$. On the matched 85-state late set, $\sigma = \{0.01, 0.03, 0.1, 0.2\}$ produces mean robot-*qpos* L2 shifts of $0.025, 0.075, 0.249, 0.498$ and mean EEF shifts of $0.009, 0.028, 0.093, 0.183$. The median same-task and same-phase nearest-neighbour distance among expert states is 0.322, placing $\sigma = 0.03$ at 23% of that manifold scale.

Data overlap. CREST states are built from the public LIBERO HDF5 demos, which after no-op filtering are the OpenVLA-OFT release's fine-tuning data for both checkpoints [12, 13, 20]. Clean CREST therefore audits continuation from in-distribution training states, and the perturbation curve quantifies a local recovery margin around those training states.

Metrics. Counts and percentages on matched sets, state-clustered bootstrap intervals for three-seed proportions, paired-bootstrap intervals for adjacent-σ differences, task-clustered bootstrap intervals for hierarchical uncertainty, and a normalized small-disturbance AUC over $\sigma \in [0, 0.03]$. Binomial intervals use Wilson [32] and bootstraps follow [5].

VI. Results: A 22-Point Recovery Gap Hidden Behind a Saturated Score

Result at a glance. OpenVLA-OFT scores 49/50 official-start and 178/185 clean CREST, yet drops from 82/85 clean to 63/85 at $\sigma = 0.03$ on the matched 85-state late-phase set. The rest of this section rules out reset artifacts, infeasibility, seed and phase variation, horizon, perturbation modality, and training-state memorization.

A. The Audit Validates Its States Before It Reports a Failure

Table I summarizes the central measurements. Expert suffix replay validates 185/240 candidate interventions. On this verified set, OpenVLA-OFT completes 178/185 clean interventions and the combined all-suite OpenVLA-OFT completes 175/185. Zero-action solves 1/185, a sanity success after seven no-ops on task 5, and no retained state was terminal at reset. Retained states are nontrivial and clean continuation is high, so the failure the audit surfaces is not an artefact of weak or trivial states. Classic OpenVLA is much weaker (2/50 official, 37/185 clean) and serves only as a secondary policy control.

a) The gap is on training states, which makes it worse, not better.: Because the audited demos are part of OpenVLA-OFT's fine-tuning data, a 22-point drop on perturbations of *training* states means the policy does not recover from small deformations of its own demonstrations, the strongest possible setting for it. The task-specific vs. combined OpenVLA-OFT divergence (matched clean 82/85 for both, but 81/85 vs. 71/85 at $\sigma = 0.01$) is inconsistent with simple memorization. As a stronger probe, we harvested 49 states the policy itself reaches before *successful* official-start rollouts but that were not in fine-tuning data. Clean is 39/49 and $\sigma = 0.03$ is 32/49, a 14.3-pp

TABLE I: Core CREST audit on all 10 simulated LIBERO-10 tasks. Nominal competence is high, but the recovery margin is not.

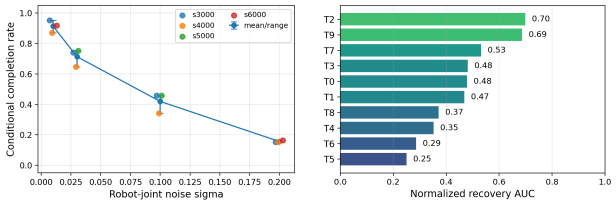
Lens	Measurement	Result
Official start	Task-specific	49/50 (98.0%)
Official start	Combined	48/50 (96.0%)
Validity	Expert suffix	185/240 (77.1%)
Clean CREST	Task-specific	178/185 (96.2%)
Clean CREST	Combined	175/185 (94.6%)
Sanity	Zero-action	1/185 (0.5%)
Matched clean late	Task-specific	82/85 (96.5%)
Joint $\sigma = 0.01$	late phases	81/85 (95.3%)
Joint $\sigma = 0.03$	late phases	63/85 (74.1%)
Joint $\sigma = 0.1$	late phases	39/85 (45.9%)
Joint $\sigma = 0.2$	late phases	13/85 (15.3%)

paired drop. The safety-relevant margin survives outside the training distribution.

B. Small Disturbances Quantify the Margin

The main result is the late-phase robot-joint curve in Fig. 1. On the matched 85-state set, clean OpenVLA-OFT is 82/85, and recovery is 81/85 at $\sigma = 0.01$ and 63/85 at $\sigma = 0.03$, with Wilson intervals $[88.5, 98.2]\%$ and $[63.9, 82.2]\%$. The paired-bootstrap difference between these matched levels is 21.2 pp with 95% CI $[12.9, 30.6]$. The normalized AUC over $\sigma \in [0, 0.03]$, including the matched clean point, is 0.884, and using measured EEF displacement rather than σ gives the same value to three decimals. A dense three-seed small-σ sweep places $\sigma = 0.03$ inside a smooth small-disturbance curve: recovery is $95.3, 84.7, 80.4, 54.9\%$ at $\sigma \in \{0.005, 0.015, 0.02, 0.05\}$. The task-clustered $\sigma = 0 \to 0.03$ paired drop (tasks resampled with replacement) is $+20.5$ pp with 95% CI $[+8.9, +33.3]$, strictly bounded above zero (state-weighted paired drop $+22.4$ pp).

a) Are the perturbed states even recoverable?: A safety audit must not flag infeasible states. Perturbed expert suffix replay drops from 72/85 at $\sigma = 0.01$ to $46/85, 13/85, 3/85$ at $\sigma = \{0.03, 0.1, 0.2\}$: open-loop expert actions cannot rescue all perturbations, so recovery requires closed-loop correction. A minimal IK proportional EEF-tracking oracle solves 45.5% of *the very states where* OpenVLA-OFT *fails* at $\sigma = 0.03$, with $56/85$ overall vs. OpenVLA-OFT's $63/85$. Conditioning on either oracle-recoverable subset (IK solves *or* perturbed expert suffix solves), the task-clustered paired drop is still $+16.9$ pp on $n = 65$ (95% CI $[+10.4, +23.6]$). The flagged failures are policy-specific, not perturbed-state infeasibility, so they are actionable.



(a) Per-σ seed ranges. (b) Task-level AUC.

Fig. 1: Robot-joint recovery on replay-verified late-phase states. **Left:** three 85-state seed groups per σ. The drop from small disturbances to stress endpoints is stable. **Right:** task-level recovery is structured, localizing the fragile behaviours a runtime monitor or robust-training budget should target first.

C. Stability Across Seeds, Phases, Horizons, Suites, and Policies

Three-seed recovery means are $91.4, 71.4, 42.0, 15.7\%$ for $\sigma \in \{0.01, 0.03, 0.1, 0.2\}$ (state-clustered bootstraps in Appendix Table III). The degradation persists at earlier phases and under the full task horizon (small-disturbance AUC $0.884 \to 0.912$), and replicates across object-XY perturbations and across LIBERO-Spatial/Object/Goal (40-task pooled small-disturbance AUC 0.802, with $84/97, 92/100, 90/98$ valid candidates and clean $96.4, 100, 94.4\%$). A start-perturbation slice (5 starts per task, $\sigma \in \{0, 0.01, 0.03\}$) leaves OpenVLA-OFT at $49/50, 47/50, 42/50$ (start-AUC 0.913) with task rankings only weakly related to CREST (Pearson 0.18): reach and recovery are distinct safety signals. The combined all-suite OpenVLA-OFT drops to AUC 0.806, and a non-OpenVLA π_0 probe [7] shows the same clean$>$ σ=0.03 ordering on a 25-state subset (directional only, bounded by π_0's low base rate under direct injection).

VII. Discussion: Toward Recovery-Aware Safety Cases

A trustworthy embodied foundation model needs both faces of safety, and they come from one substrate: a deployment-time recovery red-team and a training-time reset/reward signal. The empirical evidence is a 22-point recovery-margin drop at $\sigma = 0.03$ on OpenVLA-OFT, replicated across seeds, phases, horizons, suites, perturbation modalities, oracles, and held-out states. The honest scope: all evidence is simulation-only, with controlled robot-joint and object-XY perturbations rather than calibrated physical disturbances or adaptive adversaries, and the strongest results are OpenVLA-OFT on LIBERO. The recovery margin is a red-team diagnostic, not a formal guarantee, and a safety case should treat it as a screening signal for where hardware testing, runtime monitoring, and robust training are most needed. The most informative next steps are a non-OpenVLA architecture sweep, learned perturbation and scenario generation to scale the audit to richer disturbance channels (contact, perception error, grasp slip), a hardware-correlation study (Appendix F), and a robust-training result that moves the small-disturbance AUC at fixed clean rate.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[5] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

[6] 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. *arXiv preprint arXiv:2510.13626*, 2025.

[7] Hugging Face LeRobot. π_0 LIBERO Finetuned (pi0_libero_finetuned_v044). Hugging Face model repository, commit b359bca, https://huggingface.co/lerobot/pi0_libero_finetuned_v044, October 2025.

[8] Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Zubair Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, Wei-Chiu Ma, Dhruv Shah, Abhishek Gupta, and Karl Pertsch. PolaRiS: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025.

[9] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[10] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *Robotics: Science and Systems (RSS); arXiv:2403.12945*, 2024.

[11] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[12] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[13] Moo Jin Kim, Chelsea Finn, and Percy Liang. OpenVLA-OFT code and LIBERO fine-tuning instructions. GitHub repository, https://github.com/moojink/openvla-oft, 2025.

[14] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. DART: Noise injection for robust imitation learning. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 143–156, 2017.

[15] 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 *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3705–3728. PMLR, 2025.

[16] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[17] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[18] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[19] Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

[20] OpenVLA Collaboration. Modified LIBERO RLDS datasets. Hugging Face dataset, https://huggingface.co/datasets/openvla/modified_libero_rlds, 2024.

[21] Georgios Papagiannis and Edward Johns. MILES: Making imitation learning easy with self-supervision. In *Conference on Robot Learning (CoRL)*, 2024.

[22] Jung Yeon Park and Lawson L. S. Wong. Robust imitation of a few demonstrations with a backwards model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[23] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[24] Stephane Ross and J. Andrew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 661–668, 2010.

[25] Stephane Ross, Geoffrey J. Gordon, and J. Andrew 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*, pages 627–635, 2011.

[26] Russ Tedrake, Ian R. Manchester, Mark Tobenkin, and John W. Roberts. LQR-Trees: Feedback motion planning via sums-of-squares verification. *International Journal of Robotics Research (IJRR)*, 29(8):1038–1052, 2010.

[27] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[28] Shivam Vats, Devesh K. Jha, Maxim Likhachev, Oliver Kroemer, and Diego Romeres. RecoveryChaining: Learning local recovery policies for robust manipulation. *arXiv preprint arXiv:2410.13979*, 2024.

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

[30] Taowen Wang, Cheng Han, James Chenhao Liang, Wenhao Yang, Dongfang Liu, Luna Xinyu Zhang, Qifan Wang, Jiebo Luo, and Ruixiang Tang. Exploring the adversarial vulnerabilities of vision-language-action models in robotics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6948–6958, 2025.

[31] Yi Ru Wang, Carter Ung, Christopher Tan, Grant Tannert, Jiafei Duan, Josephine Li, Anh Le, Rishabh Oswal, Markus Grotz, Wilbert Pumacay, Yuquan Deng, Ranjay Krishna, Dieter Fox, and Siddhartha Srinivasa. RoboEval: Where robotic manipulation meets structured and scalable evaluation. *arXiv preprint arXiv:2507.00435*, 2025.

[32] Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

[33] Yuhan Xie, Yuping Yan, Yunqi Zhao, Handing Wang, and Yaochu Jin. STRONG-VLA: Decoupled robustness learning for vision-language-action models under multimodal perturbations. *arXiv preprint arXiv:2604.10055*, 2026.

[34] Gehan Zheng, Shrey Seenivasan, Matthew Johnson-Roberson, and Weiming Zhi. Rewind-IL: Online failure detection and state respawning for imitation learning. *arXiv preprint arXiv:2604.16683*, 2026.

[35] 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.

[36] Zhiyuan Zhou, Pranav Atreya, You Liang Tan, Karl Pertsch, and Sergey Levine. AutoEval: Autonomous evaluation of generalist robot manipulation policies in the real world. *arXiv preprint arXiv:2503.24278*, 2025.

[37] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martin-Martin, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

APPENDIX A
PROTOCOL SUMMARY

Algorithm 1 states the full protocol. The appendix below reproduces the quantitative detail behind the main-text claims: uncertainty, scale, conditioning sensitivity, fidelity audits, cross-suite breadth, and the minimal hardware protocol.

Algorithm 1 CREST

Input: task T with reset s_0 and predicate; demos $\mathcal{D}$; phases Φ; policy π; perturbation family P and grid Σ.
Output: retained $\mathcal{I}$, R_0, $\{R_\sigma\}$.

1) **Sample.** For each demo $d^{(i)}$ and $\tau \in \Phi$, replay the expert prefix to obtain $\hat{s}_{i,\tau}$.
2) **Validity filter.** Roll out the expert suffix from $\hat{s}_{i,\tau}$ and keep $(i,\tau) \in \mathcal{I}$ iff the predicate fires.
3) **Zero-action sanity.** Roll out zero actions and drop already-terminal states.
4) **Clean rollout.** Roll out π from each $\hat{s}_{i,\tau}$ and report R_0.
5) **Perturbed rollout.** For each $\sigma \in \Sigma$, set $\tilde{s}_{i,\tau,\sigma} = P_\sigma(\hat{s}_{i,\tau})$, refresh obs, roll out π, and report R_σ.
6) **Scale audit.** Measure induced $qpos$, EEF, and same-task/phase nearest-neighbour distances on $\mathcal{I}$.
7) **Excluded-state bounds.** Report pessimistic/optimistic variants over the full candidate denominator.

APPENDIX B
ADDITIONAL QUANTITATIVE DETAILS

TABLE II: CREST diagnostic-panel interpretation. Each row answers a specific safety-audit question rather than producing another leaderboard scalar.

Measurement	What it estimates	What it rules out or localizes
Official-start success	Standard benchmark-start competence	Policy is simply weak on nominal starts
Expert suffix validity	Solvability of replay-reconstructed states	States the demonstrator itself cannot complete
Clean CREST completion	Continuation from verified expert progress	Trivial mid-trajectory state failures
Zero-action control	Terminal-state triviality	Retained states already satisfying the predicate
Small-disturbance recovery AUC	Near-expert-state recovery margin	Loss not visible from official-start or clean
Full-horizon control	Sensitivity to remaining rollout budget	Failures caused only by too little suffix time
Excluded-state bounds	Dependence on the expert-validity filter	How much conditioning can change the apparent rate
Task/state diagnostics	Where failures concentrate	Whether degradation is uniform or task-tied

TABLE III: State-clustered bootstrap intervals for the three-seed late-phase robot-joint recovery and matched-state paired deltas.

Quantity	Estimate	95% interval
Three-seed $\sigma = 0.01$ recovery	233/255 (91.4%)	[86.7, 95.3]%
Three-seed $\sigma = 0.03$ recovery	182/255 (71.4%)	[64.3, 78.4]%
Three-seed $\sigma = 0.1$ recovery	107/255 (42.0%)	[34.9, 49.0]%
Three-seed $\sigma = 0.2$ recovery	40/255 (15.7%)	[10.6, 21.2]%
Paired delta $\sigma = 0.01 \to 0.03$	21.2 points	[12.9, 30.6]
Paired delta $\sigma = 0.03 \to 0.1$	28.2 points	[16.5, 40.0]
Paired delta $\sigma = 0.1 \to 0.2$	30.6 points	[21.2, 40.0]

TABLE IV: Measured perturbation scale and conditioning sensitivity for the matched 85-state late-phase robot-joint set. The conditional rate is the reported estimand. The bounds score the 35 candidates outside the matched denominator as failures or successes, bracketing how much the validity filter could change the apparent rate.

σ	qpos L2	EEF	Conditional	Excl. fail	Excl. succ	Manifold %
0.01	0.025	0.009	81/85 (95.3%)	81/120	116/120	8%
0.03	0.075	0.028	63/85 (74.1%)	63/120	98/120	23%
0.1	0.249	0.093	39/85 (45.9%)	39/120	74/120	77%
0.2	0.498	0.183	13/85 (15.3%)	13/120	48/120	154%

TABLE V: Three-seed recovery summaries (recovery percentages over matched verified sets). Robot-joint $\sigma_{1..4} = 0.01, 0.03, 0.1, 0.2$ and object-XY $\sigma_{1..3} = 0.005, 0.01, 0.02$.

Setting	σ_1	σ_2	σ_3	σ_4
Robot joint, phases 0.5/0.75	91.4	71.4	42.0	15.7
Robot joint, phase 0.25	88.9	66.7	24.6	5.6
Robot joint, phase 0.0	93.9	87.9	41.7	15.2

TABLE VI: Small-disturbance AUC under the suffix-horizon and full-horizon controls.

Horizon mode	$\sigma = 0.01$	$\sigma = 0.03$	AUC over $[0, 0.03]$
Remaining horizon plus buffer	81/85 (95.3%)	63/85 (74.1%)	0.884
Full task horizon	81/85 (95.3%)	70/85 (82.4%)	0.912

TABLE VII: Task-clustered bootstrap intervals on the main 85-state matched set (10000 draws). The $\sigma = 0 \to 0.03$ task-clustered drop has 95% CI [+8.9, +33.3] points, strictly above zero.

σ	Rate	Task-clustered 95% CI
0.00	82/85 (96.5%)	[91.2, 100.0]%
0.01	81/85 (95.3%)	[87.9, 100.0]%
0.03	63/85 (74.1%)	[69.0, 79.3]%
0.1	39/85 (45.9%)	[31.6, 59.1]%
0.2	13/85 (15.3%)	[6.2, 26.4]%
Task-equal delta $\sigma = 0 \to 0.03$	+20.5 pts	[+8.9, +33.3]
Task-equal delta $\sigma = 0.01 \to 0.03$	+18.2 pts	[+5.6, +31.9]

TABLE VIII: Object-XY perturbation evidence (three-seed means where shown).

Setting	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.02$
Late phase, main seed	73/85 (85.9%)	66/85 (77.6%)	42/85 (49.4%)
Late phase, three-seed mean	91.0%	79.6%	52.5%
Phase 0.25, three-seed mean	91.8%	81.4%	64.2%
Phase 0.0, three-seed mean	92.1%	87.1%	82.0%

APPENDIX C
TASK DIAGNOSTICS

TABLE IX: Most fragile and most robust tasks under the main late-phase robot-joint sweep, ranked by recovery AUC across $\sigma \in \{0.01, 0.03, 0.1, 0.2\}$. The fragile tasks are the natural targets for runtime monitoring and robust-training budget.

Rank	Task	AUC
Fragile 1	Task 5: pick up the book and place it in the caddy	0.250
Fragile 2	Task 6: put the white mug on the plate and pudding to the right	0.286
Fragile 3	Task 4: place two mugs on two plates	0.353
Robust 1	Task 2: turn on the stove and put the moka pot on it	0.699
Robust 2	Task 9: put the mug in the microwave and close it	0.687
Robust 3	Task 7: put alphabet soup and cream cheese in the basket	0.532

TABLE X: Retention sensitivity for the main late-phase robot-joint sweep. Task-equal and leave-one-task-out rates show denominator imbalance is not driving the curve.

σ	State-weighted	Task-equal	LOTO range	Validity/recovery r
0.01	95.3%	95.7%	94.7–98.7%	-0.15
0.03	74.1%	73.8%	72.4–75.6%	-0.06
0.1	45.9%	44.1%	42.1–49.4%	-0.13
0.2	15.3%	14.5%	10.7–17.3%	-0.52

TABLE XI: Policy recovery conditioned on whether open-loop expert suffix replay still succeeds after the same perturbation.

σ	Policy recovery if expert succeeds	if expert fails
0.01	70/72 (97.2%)	11/13 (84.6%)
0.03	38/46 (82.6%)	25/39 (64.1%)
0.1	8/13 (61.5%)	31/72 (43.1%)
0.2	1/3 (33.3%)	12/82 (14.6%)

APPENDIX D
CROSS-SUITE EXTENSIONS

TABLE XII: Suite-specific official-start anchors for the LIBERO-Spatial/Object/Goal extensions, showing recovery drops are measured for nominally competent suite checkpoints.

Suite	Official-start success	Tasks
LIBERO-Spatial	47/50 (94.0%)	10
LIBERO-Object	50/50 (100.0%)	10
LIBERO-Goal	49/50 (98.0%)	10

TABLE XIII: All-task extensions on the three remaining LIBERO suites.

Suite	Statistic	Valid cand.	Clean	σ=0.01	σ=0.03	σ=0.1
Spatial	Main seed	84/97	81/84	75/84	45/84	16/84
Spatial	3-seed mean	–	–	84.5	54.4	26.2
Object	Main seed	92/100	92/92	73/92	52/92	23/92
Object	3-seed mean	–	–	82.2	54.0	29.0
Goal	Main seed	90/98	85/90	80/90	68/90	27/90
Goal	3-seed mean	–	–	90.0	67.8	34.4

TABLE XIV: Pooled four-suite robot-joint recovery lens. Counts combine the main LIBERO-10 late matched set with the all-task Spatial/Object/Goal extensions. A 40-task simulator breadth summary, not hardware evidence.

Summary	Matched clean	σ=0.01	σ=0.03	AUC	σ=0.1
Main seed	340/351	309/351	228/351	0.818	105/351
3-seed pool	–	916/1053	651/1053	0.802	346/1053

APPENDIX E
PERTURBATION IMPLEMENTATION AND FIDELITY AUDIT

The audit-critical question is whether the recovery drop at small σ reflects a meaningful local-recovery limit or simulator/controller inconsistencies introduced by direct $qpos$ injection.

a) What is modified and preserved.: For the robot-joint family, `apply_perturbation` reads MuJoCo `sim.data.qpos`, adds independent $\mathcal{N}(0, \sigma^2)$ offsets to the first seven entries (the arm joints), writes `qpos` back, calls `sim.forward()`, and returns a fresh observation via the deployment `get_observation` path. It does not touch `qvel`, `qacc`, controller targets `ctrl`, actuator state `act`, mocap pose, gripper $qpos$, or observation history.

TABLE XV: Perturbation fidelity audit on the 85-state matched set ($n = 85$ per σ). Only the seven robot-arm $qpos$ entries change. `qvel`, `ctrl`, gripper $qpos$, and remaining $qpos$ have zero pre/post L2.

Quantity	$\sigma = 0.01$	$\sigma = 0.03$	$\sigma = 0.1$
$qpos$ first 7 (arm) L2 mean	0.0258	0.0775	0.2583
$qpos$ remaining L2 mean	0.000	0.000	0.000
$qvel$ L2 mean	0.000	0.000	0.000
Controller targets (`ctrl`) L2 mean	0.000	0.000	0.000
Contact count, pre / post	27.5/28.9	27.8/29.1	27.7/28.3

b) Joint limits and stale goals.: An analytic audit finds 85/85 pre-states inside Panda joint ranges (mean margin 0.866 rad), and deterministic perturbations produce $0/85$ violations at all σ. The OSC controller [9] recomputes its goal each step from current EEF position with a null interpolator (`robosuite` [37]), so no stale goal can produce a snap-back across a *qpos* jump.

c) Physically realized cross-check.: Driving the arm to the perturbed-target configuration via $K = 10$ OSC delta commands (consistent *qvel/ctrl* at handoff) on a 28-state subset gives clean 28/28, direct-injection $\sigma = 0.03$ 20/28, and controller-realized $\sigma_{\text{target}} = 0.03$ 15/28: the recovery gap *persists and grows* under physically realized displacement, inverting the worry that direct injection is artifactually strict. A calibrated single-step controller-action perturbation (action = expert$+\mathcal{N}(0, \sigma_a^2)$) gives 82/85, 81/85, 78/85 at $\sigma_a = 0, 0.3, 0.6$, so OpenVLA-OFT is robust to random controller-action noise of comparable EEF scale. The configuration-displacement gap is specific.

TABLE XVI: IK proportional EEF-tracking recovery oracle at $\sigma = 0.03$ (matched 85-state set), $K = 15$ steps before the expert suffix. On states where OpenVLA-OFT *fails*, the oracle recovers 45.5%: perturbed states are not infeasible.

Recovery method at $\sigma = 0.03$	Success	Rate
OpenVLA-OFT (closed-loop policy, full 85)	63/85	74.1%
Expert suffix open-loop (full 85)	46/85	54.1%
IK proportional + expert suffix (full 85)	56/85	65.9%
IK on states where OpenVLA-OFT *fails*	10/22	**45.5%**
IK on states where OpenVLA-OFT succeeds	46/63	73.0%

TABLE XVII: Oracle-conditioned recovery margin. OpenVLA-OFT clean-to-perturbed paired drop ($\sigma = 0$ vs 0.03) on subsets where each oracle solves the task. Here "oracle solves" is a partial recoverability indicator, not a feasibility certificate. The margin survives.

Oracle probe	n	Clean	σ=0.03	Drop (pp, 95% CI)
Full set (no oracle)	85	96.5%	74.1%	$+22.4$ $[+8.9, +33.3]$
IK oracle solves	56	98.2%	82.1%	$+16.1$ $[+8.9, +22.9]$
Expert replay solves	46	95.7%	82.6%	$+13.0$ $[+2.2, +25.5]$
Either oracle solves	65	96.9%	80.0%	$+16.9$ $[+10.4, +23.6]$

d) Cross-policy probe: π_0.: The LeRobot LIBERO-finetuned π_0 [7] on a 25-state subset gives clean 8/25 and $\sigma = 0.03$ 7/25, while OpenVLA-OFT on the same states gives 25/25 then 19/25. Both show the clean$>$ σ=0.03 ordering, but π_0's low base rate under direct injection bounds the resolvable signal, so it is a directional probe, not a peer baseline.

Appendix F
Minimal Hardware Correlation Protocol

The smallest informative hardware study is a correlation study, not exact sim-to-real transfer: pick one physical task, identify 10–20 successful nominal executions, reset each to a measured mid-trajectory state via the robot's normal controller, apply small measured offsets (1–3 cm), resume closed-loop control, and run the same policy/phases/offsets in simulation. Primary criterion: positive rank correlation (or clean $>$ small-offset $>$ larger-offset ordering) between sim and hardware. This tests whether the simulator recovery margin is useful for prioritizing where a safety case needs hardware scrutiny, not whether the absolute numbers transfer.

a) Reproducibility.: Released under MIT/Apache-2.0: `crest/`, `scripts/`, `configs/crest/`, `tests/`; the 185-state verified cache and cross-suite caches; per-run `raw_rows.jsonl`/metrics/bootstrap/failure CSVs; `commands_reproduce.md`; `audit_paper_claims.py`. Sampling, filter, and perturbation grids are deterministic given the seed base (per row: rollout seed, σ, git hash), with main late-phase seed bases $\{1000, 4000, 6000\}$. LIBERO demonstrations and OpenVLA-OFT checkpoints follow their original licenses.