LIBERO-MOR: A MULTIMODAL ROBUSTNESS BENCHMARK ACROSS THE VISION-LANGUAGE-ACTION PIPELINE

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models have achieved strong performance on robot manipulation benchmarks. Translating this progress into reliable deployment calls for systematic evaluation of how policies respond to variations throughout the robot interaction loop. We introduce **LIBERO-MOR**, a multimodal robustness benchmark across the vision-language-action pipeline. Built on standard LIBERO, it covers five deployment-facing domains: vision, language, proprioception, action execution, and environment. The benchmark comprises 22 deployment-inspired perturbations, each evaluated at three severity levels across 40 tasks under a paired clean–perturbed protocol, enabling controlled and reproducible comparisons. We evaluate seven representative VLA models spanning diverse action-generation mechanisms. Our experiments reveal substantial robustness gaps behind high clean-task success, together with shared vulnerabilities and distinct model-specific failure patterns across domains. Severity analysis further distinguishes progressive degradation, collapse at stronger interventions, and failures triggered at the mildest tested level. By connecting aggregate performance to pipeline domains, individual perturbations, and intervention strength, LIBERO-MOR provides a systematic framework for diagnosing VLA robustness and guiding the development of more reliable robot policies.

1 INTRODUCTION

Vision-language-action (VLA) models combine visual and linguistic representations with robot control, using action-generation mechanisms ranging from autoregressive decoding to regression, diffusion, and flow matching (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Kim et al., 2025). Benchmarks such as LIBERO make it possible to compare task completion across policies (Liu et al., 2023). Deployment-oriented evaluation additionally needs to characterize responses to corrupted observations, incomplete instructions, and delayed commands. These deviations enter different parts of the closed loop, making their location and structure central to understanding policy robustness.

Robustness benchmarks increasingly examine environmental, linguistic, and visual variation, as well as uncertainty across multiple interfaces (Pumacay et al., 2024; Zhou et al., 2025; Fei et al., 2026; Zhang et al., 2025a; Li et al., 2026; Guo et al., 2026). Their findings motivate a more specific question: *which intervention sites and perturbation types account for a model's loss of task success, and how does that loss change with intervention strength?* Answering this question requires both broad coverage and a common evaluation protocol. An overall score alone can obscure failures concentrated in one input or control channel.

We introduce **LIBERO-MOR**, a controlled stress-testing layer on four standard LIBERO suites. Its 22 perturbations cover vision, language, proprioception, action, and environment (Figure 1). A fixed manifest specifies three severity levels and five seeds for each of 40 tasks. Each perturbed episode is linked to a clean task–seed reference, allowing the same protocol to support overall comparisons, domain profiles, perturbation-specific fingerprints, and severity analysis.

We evaluate seven publicly available policies with heterogeneous action-generation designs. Similar clean performance can mask substantial differences under perturbation: GR00T-N1.7 and AVA-VLA

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.

both achieve 95.00% clean success, but their aggregate perturbed success differs by 7.71 percentage points. At a finer resolution, distractor instructions nearly eliminate success for two models, whereas motion blur is especially damaging to another. These patterns motivate a more detailed account of capability than nominal scores alone provide (Jiang et al., 2026).

Our contributions are threefold:

- **A pipeline-oriented protocol.** LIBERO-MOR evaluates five intervention domains and 22 perturbations with matched clean references, multiple severities, and a fixed task–seed manifest.
- **A comparative evaluation.** We report 93,800 rollouts across seven policies, measuring absolute success, the clean-to-perturbed change, and model-normalized relative robustness.
- **A diagnosis beyond aggregate scores.** Domain profiles, perturbation fingerprints, and severity responses reveal shared vulnerabilities and model-specific failure patterns.

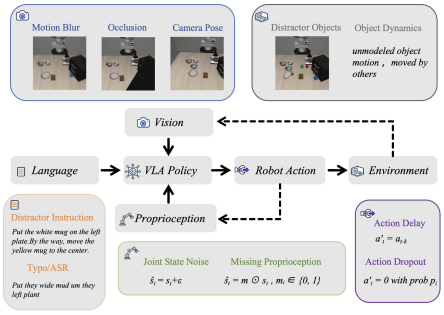

Figure 1: **LIBERO-MOR overview.** Interventions at five pipeline interfaces preserve the underlying task goal and success checker.

2 Related Work

2.1 Vision-Language-Action Models and Evaluation

VLA designs build on scalable transformer control and vision-language transfer (Brohan et al., 2022; 2023), extending to open generalist policies (Kim et al., 2024; Team et al., 2024), flow-matching action experts (Black et al., 2024; 2025; Bjorck et al., 2025), diffusion action modules (Li et al., 2024a), latent-action representations (Bu et al., 2025), efficient continuous action chunking (Kim et al., 2025), active visual attention (Xiao et al., 2026), and discrete temporal block diffusion (Lee et al., 2026). Our evaluation spans this diversity of policy families and action-generation mechanisms. CALVIN, LIBERO, SIMPLER, RoboCasa, RoboTwin, and VLABench address long-horizon control, transfer, simulated policy evaluation, and task diversity (Mees et al., 2022; Liu et al., 2023; Li et al., 2024b; Nasiriany et al., 2024; Chen et al., 2025; Zhang et al., 2025b). LIBERO-MOR complements these capability evaluations by measuring changes relative to each policy's clean performance under the same harness.

2.2 Robustness Benchmarks for Robot Policies

The Colosseum, RoboTwin, LIBERO-Pro, LIBERO-Plus, and VLA-Arena study environmental variation, domain randomization, and structured robustness tests (Pumacay et al., 2024; Chen et al., 2025; Zhou et al., 2025; Fei et al., 2026; Zhang et al., 2025a). LIBERO-VPro focuses on closed-loop visual evidence failures (Li et al., 2026). The closest multimodal precedent is Guo et al. (2026), who evaluate 17 perturbations across action, observation, environment, and instruction uncertainty and introduce a robustness-training method. RoVLA, ProGAL-VLA, and ACoT-VLA further explore training and reasoning strategies for robust control (Luo et al., 2026; Darabi & Trivedi, 2026; Zhong et al., 2026).

Building on prior multimodal stress tests, LIBERO-MOR explicitly separates proprioception from vision and evaluates 22 perturbations over the same task–seed grid at three intervention strengths. Its contribution lies in the shared diagnostic protocol and cross-model analysis: domain averages identify broad weaknesses, perturbation fingerprints reveal localized failures, and severity curves characterize how those failures develop.

Table 1: **LIBERO-MOR perturbation taxonomy.** Twenty-two perturbations are grouped by their injection site. The five domains include both policy inputs and downstream execution conditions.

Domain	#	Perturbations	Deployment factors represented
Vision	**10**	Gaussian noise; motion blur; defocus blur; compression; brightness; shadow; splash; occlusion; camera shake; camera pose shift	Sensor degradation, blur, bandwidth loss, illumination change, partial visibility, camera motion, and viewpoint/calibration shift
Language	**4**	Paraphrase; ambiguous instruction; distractor instruction; typo / ASR-like corruption	Alternative wording, underspecification, competing linguistic content, and transcription-like errors
Proprio.	**3**	Joint-state noise; end-effector pose noise; missing proprioception	Encoder/state-estimation noise, calibration error, and state-channel loss
Action	**3**	Action-execution noise; action delay; action dropout	Low-level execution error, control/communication latency, and dropped commands
Environment	**2**	Object dynamics; distractor object	Changes in object motion and task-irrelevant physical objects

3 LIBERO-MOR BENCHMARK

LIBERO-MOR is a controlled stress-testing layer on standard LIBERO: the underlying task goal and success criterion are preserved while perturbations are injected at specific points of the closed-loop VLA pipeline (Figure 1, Table 1).

3.1 BENCHMARK DESIGN PRINCIPLES

Deployment relevance. Perturbations represent deployment-facing failures in sensing, instruction, state estimation, control, and the physical scene. Their controlled formulation makes these factors reproducible and supports targeted analysis of policy responses.

Pipeline coverage. We organize failures by where they enter the closed loop: *vision*, *language*, *proprioception*, *action*, and *environment*. The first three modify policy inputs, action perturbations modify commands before execution, and environment perturbations modify the scene or dynamics.

Controlled single-factor interventions. Each perturbed rollout activates one perturbation at one severity while retaining the matched task, seed, goal, and original success checker. This design associates the measured response with a specific injected factor and captures its downstream consequences throughout the closed loop, providing a clear basis for comparing failure sources.

Graded severity. Each perturbation has three ordered intervention strengths $s \in \{1, 2, 3\}$. Each family has an ordered sequence of intervention settings. We measure the resulting success rates at every level, using the curves to characterize how closed-loop behavior changes with intervention strength.

3.2 MULTIMODAL PERTURBATION TAXONOMY

The taxonomy is deliberately interface-centric: observation-side perturbations act before inference, action perturbations act between prediction and execution, and environment perturbations modify simulator state or scene. Domain labels follow the intervention site, which gives a consistent taxonomy even when effects propagate across interfaces. For example, an environment intervention changes the simulated scene and can subsequently alter visual observations. A vision intervention instead operates on the observation channel. Language under-specification changes the information presented to the policy while retaining the original benchmark goal and success checker. This separation between the task objective and the evidence available to the policy enables targeted evaluation of instruction robustness.

3.3 Evaluation Protocol and Metrics

We instantiate LIBERO-MOR over the 40 tasks in LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long (10 tasks per suite). Here, LIBERO-Long denotes the LIBERO-10 suite, completing the four-suite evaluation protocol. Let $\mathcal{T}$ denote the 40 tasks, $\mathcal{P}$ the 22 perturbations, $\mathcal{S} = \{1, 2, 3\}$ the three severity levels, and $\mathcal{R}$ the five evaluation seeds. The perturbed benchmark therefore contains

$$|\mathcal{T}|\,|\mathcal{P}|\,|\mathcal{S}|\,|\mathcal{R}| = 40 \times 22 \times 3 \times 5 = 13{,}200 \tag{1}$$

rollouts per model. Each perturbation contributes 600 episodes (200 per severity); adding one clean rollout per task–seed pair yields 13,400 episodes per model and 93,800 across seven VLAs. The 200 clean episodes are reused as references for all 66 perturbation–severity settings. This shared reference anchors all interventions to the same 200 task–seed conditions.

For model m, we report the clean success rate $\mathrm{SR}^m_{\mathrm{clean}}$ over the 200 clean episodes and the aggregate perturbed success rate $\mathrm{SR}^m_{\mathrm{pert}}$ over all 13,200 perturbed episodes. Because every perturbation contributes the same number of episodes, the aggregate rate equally weights the 22 perturbation types, the three severities, and the 40 tasks. Each domain contributes in proportion to its number of perturbation types: vision contains 10 types, whereas environment contains two. Domain-level rates average the perturbations within that domain. We summarize robustness degradation using

$$\Delta\mathrm{SR}^m = \mathrm{SR}^m_{\mathrm{pert}} - \mathrm{SR}^m_{\mathrm{clean}}, \qquad \mathrm{RR}^m = \frac{\mathrm{SR}^m_{\mathrm{pert}}}{\mathrm{SR}^m_{\mathrm{clean}}}, \tag{2}$$

where $\Delta\mathrm{SR}$ is reported in percentage points and RR is the ratio of two aggregate success rates. RR is defined when $\mathrm{SR}^m_{\mathrm{clean}} > 0$ and is reported as a percentage. RR summarizes performance over the full task–seed grid, including both success-to-failure and failure-to-success transitions relative to clean evaluation. Reporting it alongside both absolute rates separates the achieved level of task success from the proportional change under perturbation, giving complementary views of each evaluated policy system.

Domain-, perturbation-, and severity-level RR use the same model-specific clean denominator. Cross-model summaries give equal weight to the seven model-specific ratios. Min–max intervals in Figure 3 quantify the spread across evaluated policies. The matched task–seed design organizes the results into repeated interventions on common reference conditions, with task identity preserving the grouping of related outcomes. Our reported point estimates describe the measured differences across this shared evaluation grid.

4 Evaluation Setup

4.1 Evaluated VLA Models

We evaluate seven public LIBERO-capable policies—OpenVLA-OFT, $\pi_{0.5}$, UniVLA, GR00T-N1.7-LIBERO, AVA-VLA, TBD-VLA, and CogACT-Libero (Kim et al., 2025; Black et al., 2025; Bu et al., 2025; NVIDIA, 2026; Xiao et al., 2026; Lee et al., 2026; Li et al., 2024a)—chosen to span heterogeneous action-generation designs (Table 2).

We evaluate released or LIBERO-adapted checkpoints with fixed inference configurations. LIBERO-MOR perturbations are introduced exclusively during evaluation. Clean success measured by the same harness provides the reference in Eq. 2, aligning nominal and perturbed scores within a common execution protocol.

4.2 Implementation and Evaluation Protocol

All models share the standard LIBERO simulator (Liu et al., 2023), fixed manifest, 220-step episode cap, and action horizon of 10 environment steps. Model-specific adapters map the common interface to native preprocessing, state/action normalization, and output formats while preserving the shared tasks, perturbation operators, dynamics, and success criteria. Manifest audit confirms 200 clean and 13,200 perturbed episodes per model, 200 unique clean keys, 13,200 unique variant IDs, and zero invalid episodes. Execution validity verifies completion of the evaluation protocol, and the original LIBERO checker separately determines task success. The common horizon and episode cap

Table 2: **Evaluated VLA models.** Seven policy families cover diverse representations and action-generation mechanisms, enabling comparisons of robustness profiles across contemporary VLA systems.

Model	Representative design	Action generation
OpenVLA-OFT	Optimized fine-tuning of OpenVLA for efficient continuous robot control	Parallel continuous action-chunk regression
$\pi_{0.5}$	Pretrained vision-language backbone coupled to a specialized action expert	Flow-matched continuous action chunks
UniVLA	Cross-embodiment policy built around task-centric latent actions	Latent-action decoding to robot commands
GR00T-N1.7-LIBERO	Generalist robot policy with embodiment-conditioned state/action modules	Flow-matching transformer action head
AVA-VLA	Task-history modeling with active visual attention	Continuous action prediction
TBD-VLA	Discrete temporal block diffusion policy	Masked token diffusion within blocks; autoregression across blocks
CogACT-Libero	VLM-based cognition module paired with a specialized action module	Diffusion-transformer action-sequence generation

Table 3: **Overall LIBERO-MOR robustness.** Success rates and RR are in %; ΔSR is in percentage points. Color intensity encodes larger drop (red) or retention (blue).

Model	Clean	Pert.	ΔSR	RR
OpenVLA-OFT	92.50	48.64	-43.86	52.58
$\pi_{0.5}$	**97.00**	**62.68**	-34.32	**64.62**
UniVLA	79.00	39.68	-39.32	50.23
GR00T-N1.7	95.00	57.38	-37.62	60.40
AVA-VLA	95.00	49.67	-45.33	52.29
TBD-VLA	95.50	59.50	-36.00	62.30
CogACT	94.00	58.33	-35.67	62.06

establish a consistent execution budget for every model, making clean-to-perturbed changes directly interpretable within this operating regime.

This setup evaluates complete checkpoint–adapter systems under a common stress test. Native preprocessing and normalization connect simulator-side interventions to each policy's representation and action interface. The resulting system-level profiles identify targets for follow-up ablations: concentrated proprioceptive weaknesses motivate testing state inputs, while language-specific failures motivate examining instruction processing under matched task conditions.

5 Benchmark Results and Analysis

5.1 Robustness Degrades Substantially Beyond Clean Evaluation

Table 3 and Figure 2 show substantial degradation for every VLA: clean SR is 79.00–97.00%, perturbed SR falls to 39.68–62.68%, and ΔSR spans -45.33 to -34.32 points. RR ranges from 50.23% to 64.62%, quantifying a substantial robustness gap across the evaluated policies.

Perturbation evaluation reveals a different model ordering from clean evaluation. CogACT has slightly lower clean success than AVA-VLA (94.00% versus 95.00%) but higher perturbed success (58.33% versus 49.67%). Their RR values, 62.06% and 52.29%, show that this difference persists after normalization. Conversely, UniVLA's lowest perturbed success coincides with its lowest clean

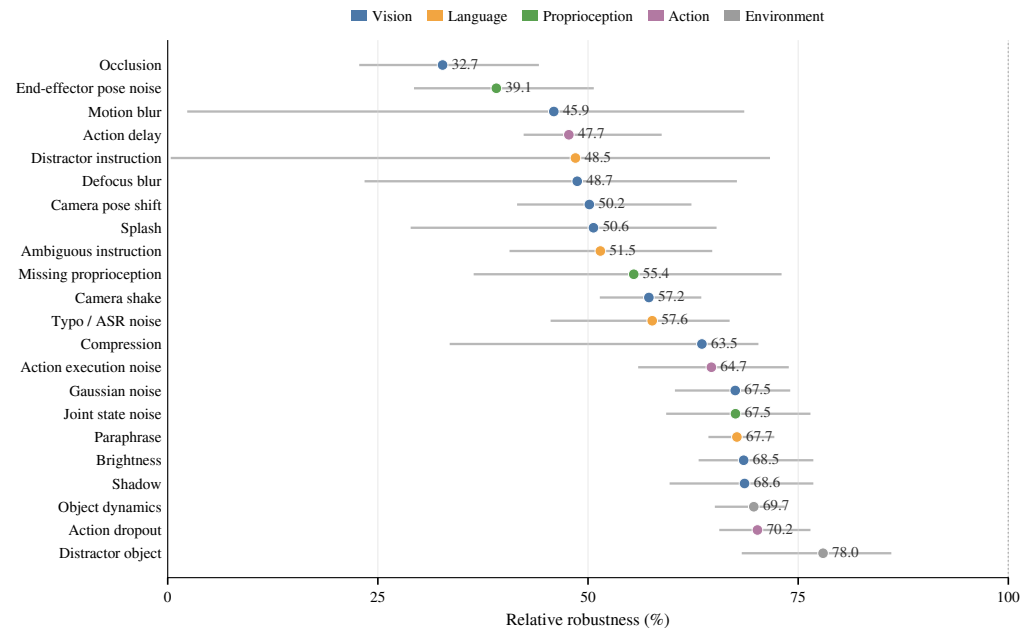

Figure 3: **Global perturbation vulnerability.** Dots show mean RR across seven VLAs; intervals denote model-wise minima and maxima, and colors indicate deployment domains.

success, illustrating why absolute performance and the relative change should be read together. Together, these scores characterize policy behavior under a shared perturbation mixture and execution budget.

5.2 Which Deployment Perturbations Are Most Challenging?

Figure 3 ranks individual perturbations by mean model-normalized RR across seven VLAs; the min–max interval exposes cross-model dependence.

Several perturbations reveal **shared vulnerabilities**. Occlusion is the strongest aggregate failure, with 32.70% mean RR and a 22.78–44.16% range across models. End-effector pose noise follows at 39.10% (29.30–50.69%), showing that corrupted geometric state can be damaging without directly perturbing vision or language. Action delay is similarly broad: its mean RR is 47.72%, with all models lying between 42.34% and 58.76% across the tested severity mixture. Together, these perturbations provide common stress tests across sensing, state estimation, and execution.

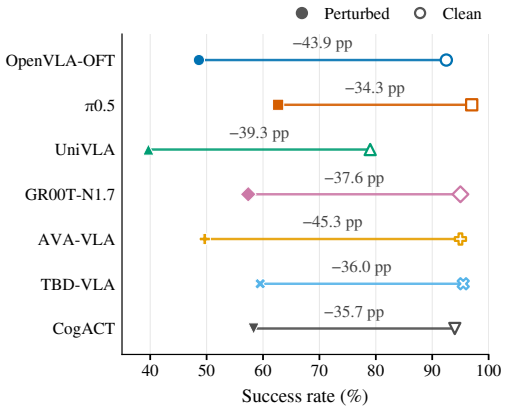

Figure 2: **Clean-to-perturbed degradation.** Hollow and filled markers denote clean and perturbed SR; labels show ΔSR (pp).

Other perturbations expose **strong cross-model disagreement**. Motion blur averages 45.93% RR but spans 2.32–68.59%; distractor instruction averages 48.50% but ranges from 0.36% to 71.65%; and defocus blur ranges from 23.42% to 67.71%. Thus, similar mean difficulty can conceal model-specific collapse. At the higher-RR end, distractor object reaches 77.96% mean RR, followed by action dropout (70.16%) and object dynamics (69.72%). The ranking characterizes the selected operators and severity ranges, providing an explicit map of challenging conditions within the benchmark.

5.3 Robustness Profiles Differ Across Pipeline Domains

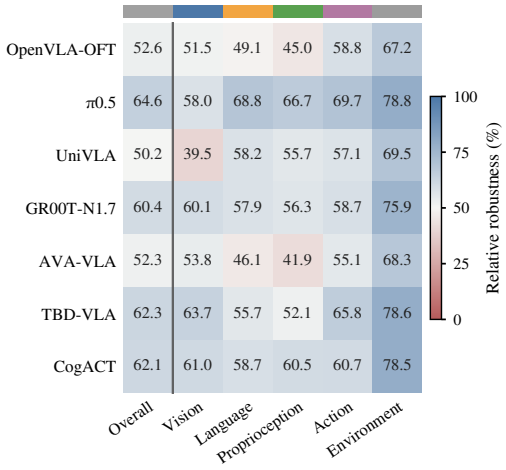

Figure 4: **Pipeline robustness profiles.** Cells report model-normalized RR (%) overall and across five deployment domains.

Aggregating perturbations by pipeline location reveals a complementary axis of robustness variation. Figure 4 reports model-normalized RR for each deployment domain, with overall RR as reference.

Environment has the highest RR for every model (67.2–78.8% RR). Across models, environment and action average 73.8% and 60.8%, whereas vision (55.4%), language (56.4%), and proprioception (54.0%) are more fragile; the domain profiles summarize these family-specific interventions and highlight the pipeline locations associated with the largest performance changes.

Profile shape differs across models. UniVLA is weakest in vision (39.5% RR), whereas AVA-VLA is weakest in proprioception (41.9%) and language (46.1%). $\pi_{0.5}$ retains 66.7–69.7% across language, proprioception, and action, while TBD-VLA is stronger in vision/action than language/proprioception. GR00T-N1.7 and CogACT are comparatively flat outside environment. Thus, similar aggregate RR can arise from different interface-specific strengths and weaknesses.

5.4 Fine-Grained Robustness Fingerprints Reveal Distinct Failure Modes

Domain averages can still mask sharply different failures within the same interface. Figure 5 therefore represents each model by a 22-dimensional perturbation-level RR fingerprint.

The clearest example occurs in the language domain. OpenVLA-OFT retains 64.3–66.8% RR under paraphrase, ambiguous instruction, and typo corruption, yet collapses to 0.4% under distractor instruction. AVA-VLA exhibits the same qualitative pattern: 56.7–65.3% RR for the other language perturbations but only 1.4% for distractor instruction. By contrast, several other models remain above 60% RR under the same distractor perturbation. The contrast identifies distractor-instruction processing as a specific diagnostic target. The strong separation from the other three language perturbations gives a concrete direction for examining instruction formatting, attention, and training coverage in subsequent controlled experiments.

Fine-grained visual and proprioceptive responses are similarly non-uniform. For $\pi_{0.5}$, brightness and shadow each retain 76.8% RR, whereas motion and defocus blur fall to 34.4% and 34.2%. UniVLA shows an even sharper visual fingerprint: Gaussian noise, brightness, and camera shake retain 57.6–64.8% RR, while motion blur falls to 2.3% and defocus blur to 23.4%. Within proprioception, CogACT is comparatively tolerant to joint-state noise and missing proprioception (69.1% and 70.9%) but substantially weaker to end-effector pose noise (41.3%); TBD-VLA instead retains 70.7% under joint-state noise but only 37.7% when proprioception is missing. These contrasts reveal distinct sensitivities within the same state interface and identify which proprioceptive operators are most informative for evaluating each policy.

Under the tested settings, action dropout is less damaging than action delay for every model despite sharing the control channel. The fingerprints separate these temporal disturbance patterns and support selecting mitigation targets at the level of individual control operators.

5.5 Severity Reveals Qualitatively Different Failure Dynamics

Severity analysis examines how vulnerabilities change as the intervention setting increases. Figure 6 traces model-normalized RR across three levels for four representative perturbations chosen to expose distinct response shapes.

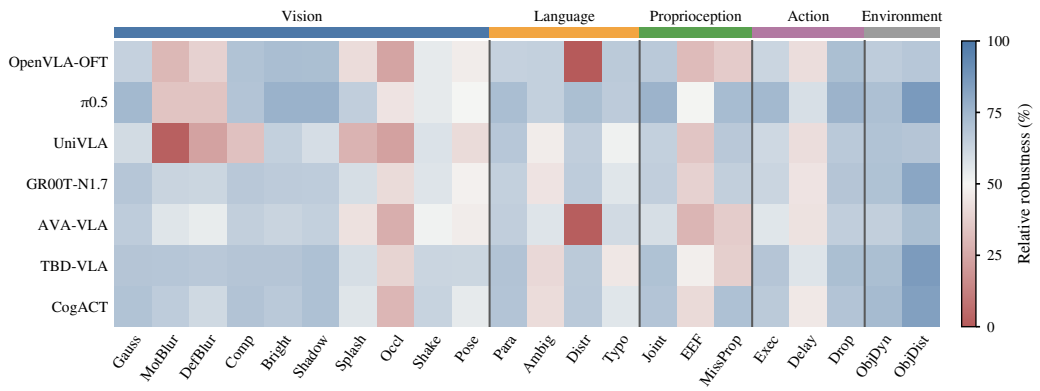

Figure 5: **Fine-grained robustness fingerprints.** Cell colors encode model-normalized RR (%) for all 22 perturbations, grouped by deployment domain.

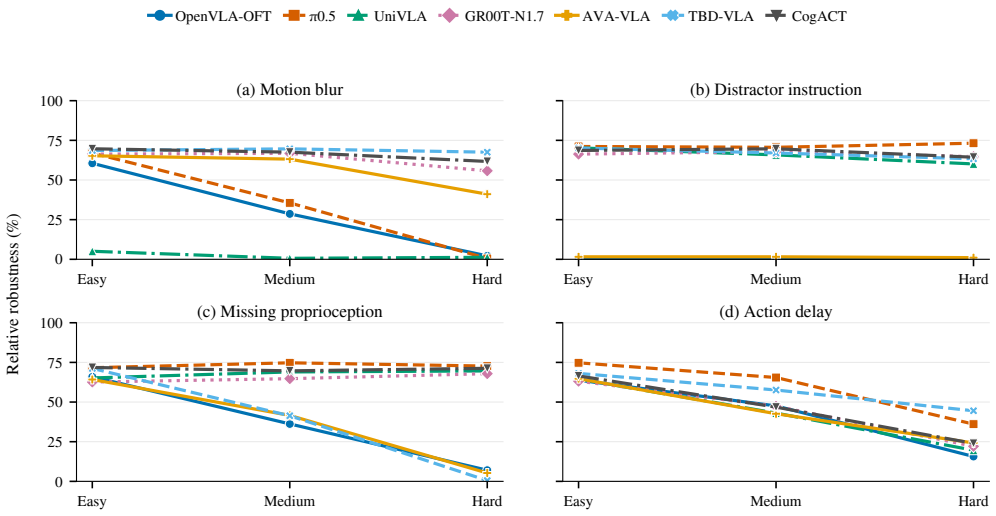

Figure 6: **Severity-dependent robustness.** Curves show RR at three intervention levels (Easy, Medium, and Hard denote mild, medium, and strong settings). Lines connect the measured response at each intervention level.

Progressive collapse versus a comparatively flat response. Motion blur produces sharply different trajectories. OpenVLA-OFT falls from 60.5% RR at mild severity to 28.6% at medium and 2.2% at strong severity; $\pi_{0.5}$ follows a similar 67.0% $\rightarrow$ 35.6% $\rightarrow$ 0.5% trajectory. TBD-VLA instead remains near 68% across the three levels, while CogACT and GR00T-N1.7 decline less steeply. The approximately flat TBD-VLA curve separates the initial clean-to-perturbed loss from the subsequent response to stronger blur. Read together with the clean reference, it shows both the impact of introducing blur and the stability of performance across the tested levels.

Missing proprioception yields a related split. OpenVLA-OFT, AVA-VLA, and TBD-VLA reach 7.0%, 5.3%, and 0.5% RR at the strongest level, whereas $\pi_{0.5}$, UniVLA, GR00T-N1.7, and CogACT remain approximately stable or show small increases. The three-level comparison distinguishes progressively increasing sensitivity from comparatively stable responses and identifies the strongest setting as a particularly informative stress test for the first three policies.

Failure at the mildest setting versus graded degradation. OpenVLA-OFT and AVA-VLA already have only 0.5% and 1.6% RR under the mildest distractor instruction and remain near zero at higher levels. The other five models stay around 60–73%. Thus, the tested distractor format is sufficient to trigger failure in the first two systems without requiring the strongest setting. This early-onset pattern identifies the introduction of competing instruction content as the decisive tested change, while the

near-zero values explain the flat response at subsequent levels. In contrast, all seven models lose RR as action delay increases, reaching 15.7–44.5% at strong severity. This shared downward trend identifies delayed execution as a broadly challenging control condition, with model-dependent rates of degradation.

Interpreting non-monotonic responses. Each severity point summarizes 200 episodes over the same 40 tasks and five seeds. At this sample size, one changed outcome corresponds to 0.5 percentage points of SR before clean-rate normalization. Severity indexes the intervention setting, while empirical success captures the policy's resulting closed-loop response. Small reversals can arise from finite-sample variation, task-specific interactions, or incidental benefits along a trajectory; stronger interventions can also change the distribution of visited states. Retaining the measured values preserves these diagnostic signals. Paired episode traces and operator-level checks provide a direct route to distinguishing these candidate explanations and locating the tasks or execution stages responsible for a reversal.

6 Discussion and Conclusion

6.1 Implications for Robustness Evaluation

The results support a sequence of reporting decisions. First, report clean and perturbed success under the same execution protocol, so that the absolute loss remains visible. Second, provide RR as a complementary normalization, while retaining its clean denominator. Third, inspect domain and perturbation profiles before selecting a mitigation target. Finally, examine severity curves to distinguish failures already present at the mildest setting from those that worsen across the tested range. This progression connects aggregate performance to specific failure conditions. The same hierarchy supports evaluating robustness interventions: an improvement can be traced to its affected perturbations and severity levels, while the remaining domains reveal whether the benefit transfers across the pipeline.

The choice of aggregation also matters for deployment-oriented interpretation. LIBERO-MOR uses uniform perturbation-type weights to establish a transparent and reproducible comparison distribution. A deployment with frequent communication delay may prioritize a different profile from one dominated by poor visual conditions. Accordingly, the overall benchmark score summarizes a specified test mixture, while the disaggregated results allow readers to examine the conditions relevant to their application. Aligning operators, severity ranges, checkpoints, and rollout budgets provides a common basis for extending these comparisons to other robustness benchmarks.

6.2 Scope and Future Directions

LIBERO-MOR establishes a controlled evaluation setting within the LIBERO embodiment, using family-specific severity scales, single-factor interventions, binary task success, and five seeds per task. The shared execution budget supports systematic comparisons and several directions for extending the benchmark. Recovery time, trajectory quality, and partial progress can enrich the behavioral characterization; expanded seed coverage and episode-level uncertainty estimates can sharpen task-specific comparisons. Controlled ablations of architecture, pretraining, fine-tuning, and adapters can further explain the observed system-level differences. Additional embodiments, compound interventions, policy-native execution settings, and real-robot experiments can test how these robustness profiles transfer across deployment conditions.

Conclusion. LIBERO-MOR evaluates perturbations across the VLA control loop with a matched, multi-severity protocol. Across seven policies, substantial aggregate losses coexist with distinct perturbation-specific and severity-dependent responses. These results make the case for reporting *which conditions cause failure and how the response changes with severity*, alongside clean task success.

AI USE STATEMENT

The research idea, methodological direction, and overall framework were conceived and determined by the authors. Generative AI tools were used to assist with searching for and summarizing relevant literature based on the authors' proposed research direction, to implement substantial portions of the code after the overall framework had been established by the authors, and to assist with manuscript drafting, language editing, and LaTeX formatting. All AI-assisted outputs, including literature references, code, experimental results, and manuscript content, were reviewed and verified by the authors, who take full responsibility for the final work.

REPRODUCIBILITY STATEMENT

Section 3 defines the benchmark composition, including the task–perturbation–severity–seed evaluation grid, matched clean references, and aggregation procedures. Section 4 reports the evaluated policy families, shared simulator and execution budget, the scope of model-specific adapters, and manifest audit statistics. Tables 1 and 2 summarize the perturbation taxonomy and the representative policy designs. The code, benchmark implementation, evaluation configurations, and associated datasets and artifacts developed for this project will be released publicly to support reproducibility and future research.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

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.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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.

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.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

Nastaran Darabi and Amit Ranjan Trivedi. Progal-vla: Grounded alignment through prospective reasoning in vision-language-action models. *arXiv preprint arXiv:2604.09824*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. Libero-plus: A progressive robustness benchmark for visual-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 38574–38583, 2026.

Jianing Guo, Zhenhong Wu, Chang Tu, Yiyao Ma, Xiangqi Kong, Zhiqian Liu, Jiaming Ji, Shuning Zhang, Yuanpei Chen, Kai Chen, et al. On robustness of vision-language-action model against

multi-modal perturbations. In *International Conference on Learning Representations*, volume 2026, pp. 70248–70272, 2026.

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, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model, 2024. *URL https://arxiv. org/abs/2406.09246*, 1(2):4, 2024.

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

Sung-Wook Lee, Xuhui Kang, and Yen-Ling Kuo. Tbd-vla: Temporal block diffusion vision language action model. *arXiv preprint arXiv:2606.07895*, 2026.

Huiqiong Li, Zhiting Mei, Anirudha Majumdar, Jingjing Chen, Yu-Gang Jiang, and Bin Zhu. Libero-vpro: Benchmarking closed-loop visual robustness of robotic foundation models, 2026. URL https://arxiv.org/abs/2609.24350.

Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024a.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024b.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jingzhou Luo, Yifan Wen, Yongjie Bai, Xinshuai Song, Yang Liu, and Liang Lin. Rovla: Multi-consistency constraints for robust vision-language-action models. *arXiv preprint arXiv:2605.19678*, 2026.

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

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

NVIDIA. GR00T-N1.7-LIBERO. https://huggingface.co/nvidia/GR00T-N1.7-LIBERO, 2026. Hugging Face model repository.

Wilbert Pumacay, Ishika Singh, Jiafei Duan, Ranjay Krishna, Jesse Thomason, and Dieter Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

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.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: improving vision-language-action models with active visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13453–13463, 2026.

Borong Zhang, Jiahao Li, Jiachen Shen, Yuhao Zhang, Yishuai Cai, Lu Liu, Hailu Ji, Yuanpei Chen, Juntao Dai, Jiaming Ji, et al. Vla-arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539*, 2025a.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, et al. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11142–11152. IEEE, 2025b.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Si Liu, and Guanghui Ren. Acot-vla: Action chain-of-thought for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8152–8162, 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 Reproducibility and Supplementary Protocol

This appendix documents the frozen LIBERO-MOR benchmark specification, evaluation adapters, complete aggregate results, and the severity audit. All numerical tablefig2s are generated from the episode-level JSONL files by `scripts/build_supplement.py`; the main paper and `benchmark_manifest.json` are not modified.

A.1 Benchmark specification

The benchmark contains 40 original LIBERO tasks (10 each from LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10), 22 perturbation operators, three severity levels, and five seed indices. This gives 13,200 perturbed episodes and 200 paired clean references per model. The manifest is identified by SHA-256 `2b1caa800b6fd9fd6420fad15b0eef1a000919dd365327b8313ac53ba4ef2d89`. Clean and perturbed rows are paired by suite, task id, seed index, and initial-state id; the original `env.check_success()` checker is used unchanged.

A.2 Perturbation contract

Table 4 lists the numerical severity contract recovered from the source implementation. The machine-readable CSV additionally records injection level, temporal scope, affected channels, and randomization rule.

Table 4: Exact perturbation severity parameters.

Domain	Perturbation	Injection	Severity 1	Severity 2	Severity 3
Vision	gaussian_noise	Sensor	mean_normalized=0.0; sigma_normalized=0.01	mean_normalized=0.0; sigma_normalized=0.03	mean_normalized=0.0; sigma_normalized=0.05
Vision	motion_blur	Sensor	direction_deg_range=[0.0, 180.0]; kernel_size_px=5	direction_deg_range=[0.0, 180.0]; kernel_size_px=9	direction_deg_range=[0.0, 180.0]; kernel_size_px=15
Vision	defocus_blur	Sensor	gaussian_sigma_px=0.5	gaussian_sigma_px=1.7	gaussian_sigma_px=4.0
Vision	compression	Sensor	codec=jpeg; quality=90	codec=jpeg; quality=50	codec=jpeg; quality=20
Vision	brightness	Scene	light_intensity_delta=0.1; sampling=seeded_dark_or_bright	light_intensity_delta=0.4; sampling=seeded_dark_or_bright	light_intensity_delta=0.75; sampling=seeded_dark_or_bright
Vision	shadow	Scene	ambient_intensity_factor=0.95; light_direction_jitter_deg=3.0; light_intensity_factor=0.9	ambient_intensity_factor=0.78; light_direction_jitter_deg=12.0; light_intensity_factor=0.64	ambient_intensity_factor=0.5; light_direction_jitter_deg=30.0; light_intensity_factor=0.34
Vision	splash	Sensor	material_pool=[water, milk, coffee, juice]; target_coverage_fraction=0.05	material_pool=[water, milk, coffee, juice]; target_coverage_fraction=0.17	material_pool=[water, milk, coffee, juice]; target_coverage_fraction=0.36
Vision	occlusion	Sensor	mask_type=seeded_polygon; target_coverage_fraction=0.1	mask_type=seeded_polygon; target_coverage_fraction=0.2	mask_type=seeded_polygon; target_coverage_fraction=0.4

Continued on next page

Table 4 (continued)

Domain	Perturbation	Injection	Severity 1	Severity 2	Severity 3
Vision	camera_shake	Scene	rotation_max_deg=0.7; translation_std_m=0.002	rotation_max_deg=2.8; translation_std_m=0.008	rotation_max_deg=8.4; translation_std_m=0.024
Vision	camera_pose_shift	Scene	rotation_max_deg=5.0; translation_max_m=0.01	rotation_max_deg=10.0; translation_max_m=0.03	rotation_max_deg=15.0; translation_max_m=0.05
Language	paraphrase	Task	semantic_constraint=preserve_goal; synonym_replacement_probability=0.2	semantic_constraint=preserve_goal; synonym_replacement_probability=0.5	semantic_constraint=preserve_goal; synonym_replacement_probability=0.85
Language	ambiguous_instruction	Task	attribute_removal_probability=0.25; relation_removal_probability=0.25	attribute_removal_probability=0.6; relation_removal_probability=0.6	attribute_removal_probability=0.92; relation_removal_probability=0.92
Language	distractor_instruction	Task	num_distractor_clauses=1; preserve_original_instruction=true	num_distractor_clauses=2; preserve_original_instruction=true	num_distractor_clauses=3; preserve_original_instruction=true
Language	typo	Task	character_corruption_probability=0.03; operations=[delete, insert, substitute]	character_corruption_probability=0.1; operations=[delete, insert, substitute]	character_corruption_probability=0.24; operations=[delete, insert, substitute]
Proprio.	joint_state_noise	State	joint_position_std_rad=0.01	joint_position_std_rad=0.03	joint_position_std_rad=0.05
Proprio.	eef_pose_noise	State	position_std_m=0.01; rotation_std_deg=5.0	position_std_m=0.03; rotation_std_deg=10.0	position_std_m=0.05; rotation_std_deg=15.0
Proprio.	missing_Proprio.	Sensor	drop_probability=0.1; replacement=previous_value	drop_probability=0.5; replacement=zero	drop_probability=1.0; replacement=zero
Action	action_execution_noise	Actuator	action_std_normalized=0.01	action_std_normalized=0.05	action_std_normalized=0.1
Action	action_delay	Actuator	delay_steps=1	delay_steps=3	delay_steps=5
Action	action_dropout	Actuator	drop_probability=0.05; replacement=zero_Action	drop_probability=0.1; replacement=zero_Action	drop_probability=0.2; replacement=zero_action
Env.	object_dynamics	Scene	friction_factor_range=[0.8, 1.25]; mass_factor_range=[0.8, 1.25]; sampling=log_uniform	friction_factor_range=[0.67, 1.5]; mass_factor_range=[0.67, 1.5]; sampling=log_uniform	friction_factor_range=[0.5, 2.0]; mass_factor_range=[0.5, 2.0]; sampling=log_uniform
Env.	distractor_object	Scene	num_objects=1; placement=seeded_collision_free	num_objects=3; placement=seeded_collision_free	num_objects=5; placement=seeded_collision_free

A.3 TASK INVENTORY, PAIRING, AND VALIDATION

The complete task inventory is given in Table 5. The manifest contains 13,200 unique variant ids and 200 clean pairing keys. The execution audit checks row counts, duplicate keys, manifest references, execution validity, and successful invocation of the unchanged success checker.

Table 5: The 40 original LIBERO tasks.

Suite	ID	Task	BDDL / initial-state files
libero_10	0	KITCHEN SCENE3 turn on the stove and put the moka pot on it	KITCHEN_SCENE3_turn_on_the_stove_and_put_the_moka_pot_on_it.bddl / KITCHEN_SCENE3_turn_on_the_stove_and_put_the_moka_pot_on_it.pruned_init
libero_10	1	KITCHEN SCENE4 put the black bowl in the bottom drawer of the cabinet and close it	KITCHEN_SCENE4_put_the_black_bowl_in_the_bottom_drawer_of_the_cabinet_and_close_it.bddl / KITCHEN_SCENE4_put_the_black_bowl_in_the_bottom_drawer_of_the_cabinet_and_close_it.pruned_init
libero_10	2	KITCHEN SCENE6 put the yellow and white mug in the microwave and close it	KITCHEN_SCENE6_put_the_yellow_and_white_mug_in_the_microwave_and_close_it.bddl / KITCHEN_SCENE6_put_the_yellow_and_white_mug_in_the_microwave_and_close_it.pruned_init
libero_10	3	KITCHEN SCENE8 put both moka pots on the stove	KITCHEN_SCENE8_put_both_moka_pots_on_the_stove.bddl / KITCHEN_SCENE8_put_both_moka_pots_on_the_stove.pruned_init
libero_10	4	LIVING ROOM SCENE1 put both the alphabet soup and the cream cheese box in the basket	LIVING_ROOM_SCENE1_put_both_the_alphabet_soup_and_the_cream_cheese_box_in_the_basket.bddl / LIVING_ROOM_SCENE1_put_both_the_alphabet_soup_and_the_cream_cheese_box_in_the_basket.pruned_init
libero_10	5	LIVING ROOM SCENE2 put both the alphabet soup and the tomato sauce in the basket	LIVING_ROOM_SCENE2_put_both_the_alphabet_soup_and_the_tomato_sauce_in_the_basket.bddl / LIVING_ROOM_SCENE2_put_both_the_alphabet_soup_and_the_tomato_sauce_in_the_basket.pruned_init
libero_10	6	LIVING ROOM SCENE2 put both the cream cheese box and the butter in the basket	LIVING_ROOM_SCENE2_put_both_the_cream_cheese_box_and_the_butter_in_the_basket.bddl / LIVING_ROOM_SCENE2_put_both_the_cream_cheese_box_and_the_butter_in_the_basket.pruned_init
libero_10	7	LIVING ROOM SCENE5 put the white mug on the left plate and put the yellow and white mug on the right plate	LIVING_ROOM_SCENE5_put_the_white_mug_on_the_left_plate_and_put_the_yellow_and_white_mug_on_the_right_plate.bddl / LIVING_ROOM_SCENE5_put_the_white_mug_on_the_left_plate_and_put_the_yellow_and_white_mug_on_the_right_plate.pruned_init

Continued on next page

Table 5 (continued)

Suite	ID	Task	BDDL / initial-state files
libero_10	8	LIVING ROOM SCENE6 put the white mug on the plate and put the chocolate pudding to the right of the plate	LIVING_ROOM_SCENE6_put_the_white_mug_on_the_plate_and_put_the_chocolate_pudding_to_the_right_of_the_plate.bddl / LIVING_ROOM_SCENE6_put_the_white_mug_on_the_plate_and_put_the_chocolate_pudding_to_the_right_of_the_plate.pruned_init
libero_10	9	STUDY SCENE1 pick up the book and place it in the back compartment of the caddy	STUDY_SCENE1_pick_up_the_book_and_place_it_in_the_back_compartment_of_the_caddy.bddl / STUDY_SCENE1_pick_up_the_book_and_place_it_in_the_back_compartment_of_the_caddy.pruned_init
libero_goal	0	open the middle drawer of the cabinet	open_the_middle_drawer_of_the_cabinet.bddl / open_the_middle_drawer_of_the_cabinet.pruned_init
libero_goal	1	open the top drawer and put the bowl inside	open_the_top_drawer_and_put_the_bowl_inside.bddl / open_the_top_drawer_and_put_the_bowl_inside.pruned_init
libero_goal	2	push the plate to the front of the stove	push_the_plate_to_the_front_of_the_stove.bddl / push_the_plate_to_the_front_of_the_stove.pruned_init
libero_goal	3	put the bowl on the plate	put_the_bowl_on_the_plate.bddl / put_the_bowl_on_the_plate.pruned_init
libero_goal	4	put the bowl on the stove	put_the_bowl_on_the_stove.bddl / put_the_bowl_on_the_stove.pruned_init
libero_goal	5	put the bowl on top of the cabinet	put_the_bowl_on_top_of_the_cabinet.bddl / put_the_bowl_on_top_of_the_cabinet.pruned_init
libero_goal	6	put the cream cheese in the bowl	put_the_cream_cheese_in_the_bowl.bddl / put_the_cream_cheese_in_the_bowl.pruned_init
libero_goal	7	put the wine bottle on the rack	put_the_wine_bottle_on_the_rack.bddl / put_the_wine_bottle_on_the_rack.pruned_init
libero_goal	8	put the wine bottle on top of the cabinet	put_the_wine_bottle_on_top_of_the_cabinet.bddl / put_the_wine_bottle_on_top_of_the_cabinet.pruned_init
libero_goal	9	turn on the stove	turn_on_the_stove.bddl / turn_on_the_stove.pruned_init
libero_object	0	pick up the alphabet soup and place it in the basket	pick_up_the_alphabet_soup_and_place_it_in_the_basket.bddl / pick_up_the_alphabet_soup_and_place_it_in_the_basket.pruned_init
libero_object	1	pick up the bbq sauce and place it in the basket	pick_up_the_bbq_sauce_and_place_it_in_the_basket.bddl / pick_up_the_bbq_sauce_and_place_it_in_the_basket.pruned_init
libero_object	2	pick up the butter and place it in the basket	pick_up_the_butter_and_place_it_in_the_basket.bddl / pick_up_the_butter_and_place_it_in_the_basket.pruned_init
libero_object	3	pick up the chocolate pudding and place it in the basket	pick_up_the_chocolate_pudding_and_place_it_in_the_basket.bddl / pick_up_the_chocolate_pudding_and_place_it_in_the_basket.pruned_init

Continued on next page

Table 5 (continued)

Suite	ID	Task	BDDL / initial-state files
libero_object	4	pick up the cream cheese and place it in the basket	pick_up_the_cream_cheese_and_place_it_in_the_basket.bddl / pick_up_the_cream_cheese_and_place_it_in_the_basket.pruned_init
libero_object	5	pick up the ketchup and place it in the basket	pick_up_the_ketchup_and_place_it_in_the_basket.bddl / pick_up_the_ketchup_and_place_it_in_the_basket.pruned_init
libero_object	6	pick up the milk and place it in the basket	pick_up_the_milk_and_place_it_in_the_basket.bddl / pick_up_the_milk_and_place_it_in_the_basket.pruned_init
libero_object	7	pick up the orange juice and place it in the basket	pick_up_the_orange_juice_and_place_it_in_the_basket.bddl / pick_up_the_orange_juice_and_place_it_in_the_basket.pruned_init
libero_object	8	pick up the salad dressing and place it in the basket	pick_up_the_salad_dressing_and_place_it_in_the_basket.bddl / pick_up_the_salad_dressing_and_place_it_in_the_basket.pruned_init
libero_object	9	pick up the tomato sauce and place it in the basket	pick_up_the_tomato_sauce_and_place_it_in_the_basket.bddl / pick_up_the_tomato_sauce_and_place_it_in_the_basket.pruned_init
libero_spatial	0	pick up the black bowl between the plate and the ramekin and place it on the plate	pick_up_the_black_bowl_between_the_plate_and_the_ramekin_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_between_the_plate_and_the_ramekin_and_place_it_on_the_plate.pruned_init
libero_spatial	1	pick up the black bowl from table center and place it on the plate	pick_up_the_black_bowl_from_table_center_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_from_table_center_and_place_it_on_the_plate.pruned_init
libero_spatial	2	pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate	pick_up_the_black_bowl_in_the_top_drawer_of_the_wooden_cabinet_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_in_the_top_drawer_of_the_wooden_cabinet_and_place_it_on_the_plate.pruned_init
libero_spatial	3	pick up the black bowl next to the cookie box and place it on the plate	pick_up_the_black_bowl_next_to_the_cookie_box_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_next_to_the_cookie_box_and_place_it_on_the_plate.pruned_init
libero_spatial	4	pick up the black bowl next to the plate and place it on the plate	pick_up_the_black_bowl_next_to_the_plate_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_next_to_the_plate_and_place_it_on_the_plate.pruned_init
libero_spatial	5	pick up the black bowl next to the ramekin and place it on the plate	pick_up_the_black_bowl_next_to_the_ramekin_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_next_to_the_ramekin_and_place_it_on_the_plate.pruned_init

Continued on next page

Table 5 (continued)

Suite	ID	Task	BDDL / initial-state files
libero_spatial	6	pick up the black bowl on the cookie box and place it on the plate	pick_up_the_black_bowl_on_the_cookie_box_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_on_the_cookie_box_and_place_it_on_the_plate.pruned_init
libero_spatial	7	pick up the black bowl on the ramekin and place it on the plate	pick_up_the_black_bowl_on_the_ramekin_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_on_the_ramekin_and_place_it_on_the_plate.pruned_init
libero_spatial	8	pick up the black bowl on the stove and place it on the plate	pick_up_the_black_bowl_on_the_stove_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_on_the_stove_and_place_it_on_the_plate.pruned_init
libero_spatial	9	pick up the black bowl on the wooden cabinet and place it on the plate	pick_up_the_black_bowl_on_the_wooden_cabinet_and_place_it_on_the_plate.bddl / pick_up_the_black_bowl_on_the_wooden_cabinet_and_place_it_on_the_plate.pruned_init

A.4 MODELS AND ADAPTERS

Table 6 records the checkpoint identifiers and adapter entry points available in the frozen evaluation metadata. Metadata that were not captured by the frozen rollout records are omitted rather than inferred.

A.5 MODELS AND ADAPTERS

Table 6 records the checkpoint identifiers and adapter entry points available in the frozen evaluation metadata. Metadata that were not captured by the frozen rollout records are omitted rather than inferred.

Table 6: Resolved model checkpoints and adapter contracts.

Model	Checkpoint	Adapter	Action contract
OpenVLA-OFT	openvla-7b-oft-finetuned-libero-spatial-object-goal-10	libero/libero/benchmark/models/openvla_oft_adapter.py	continuous 7D, clipped to [-1,1] by adapter/controller
pi0.5	pi05_libero_pytorch_retry2	libero/libero/benchmark/models/pi0_adapter.py	flow-matching 7D action; benchmark executes 10 steps before replanning
UniVLA	univla-7b-224-sft-libero	libero/libero/benchmark/models/univla_adapter.py	decoded 7D action, gripper conversion, clip [-1,1]
GR00T-N1.7-LIBERO	GR00T-N1.7-LIBERO	benchmark_plugins/rpc.py + GR00T backend configuration	continuous 7D delta action
AVA-VLA	avavla-libero-4in1	benchmark_plugins/avavla.py	8-action chunk; OXE gripper conversion to LIBERO 7D
TBD-VLA	tbdvla_libero	benchmark_plugins/tbdvla.py	7D output with saved unnormalizer and LIBERO gripper convention
CogACT-Libero	CogACT-Libero	libero/libero/benchmark/models/cogact_adapter.py	OXE action converted to LIBERO 7D; native signed gripper

A.6 COMPLETE RESULTS

The overall results in Table 7 exactly reproduce the frozen main-paper values after filtering the shared OpenVLA-OFT/pi0.5 result files by policy name. Domain and suite breakdowns are provided in machine-readable CSV files and compact appendix tables; the complete model–perturbation and model–severity matrices are in `artifacts/results_by_perturbation.csv` and `artifacts/results_by_severity.csv`.

Table 7: Overall clean and perturbed success rates.

Model	Clean SR	Perturbed SR	ΔSR	RR
OpenVLA-OFT	92.50	48.64	-43.86	52.58
pi0.5	97.00	62.68	-34.32	64.62
UniVLA	79.00	39.68	-39.32	50.23
GR00T-N1.7-LIBERO	95.00	57.38	-37.62	60.40
AVA-VLA	95.00	49.67	-45.33	52.29
TBD-VLA	95.50	59.50	-36.00	62.30
CogACT-Libero	94.00	58.33	-35.67	62.06

Table 8: Perturbed success rates by modality.

Model	Domain	Perturbed SR	RR
OpenVLA-OFT	vision	47.60	51.46
OpenVLA-OFT	language	45.38	49.05
OpenVLA-OFT	proprioception	41.67	45.05
OpenVLA-OFT	action	54.39	58.80
OpenVLA-OFT	environment	62.17	67.21
pi0.5	vision	56.23	57.97
pi0.5	language	66.71	68.77
pi0.5	proprioception	64.72	66.72
pi0.5	action	67.61	69.70
pi0.5	environment	76.42	78.78
UniVLA	vision	31.20	39.49
UniVLA	language	45.96	58.18
UniVLA	proprioception	44.00	55.70
UniVLA	action	45.11	57.10
UniVLA	environment	54.92	69.51
GR00T-N1.7-LIBERO	vision	57.05	60.05
GR00T-N1.7-LIBERO	language	54.96	57.85
GR00T-N1.7-LIBERO	proprioception	53.50	56.32
GR00T-N1.7-LIBERO	action	55.78	58.71
GR00T-N1.7-LIBERO	environment	72.08	75.88
AVA-VLA	vision	51.15	53.84
AVA-VLA	language	43.75	46.05
AVA-VLA	proprioception	39.78	41.87
AVA-VLA	action	52.39	55.15
AVA-VLA	environment	64.92	68.33
TBD-VLA	vision	60.85	63.72
TBD-VLA	language	53.17	55.67
TBD-VLA	proprioception	49.72	52.07
TBD-VLA	action	62.83	65.79
TBD-VLA	environment	75.08	78.62
CogACT-Libero	vision	57.32	60.98
CogACT-Libero	language	55.21	58.73
CogACT-Libero	proprioception	56.83	60.46
CogACT-Libero	action	57.06	60.70
CogACT-Libero	environment	73.83	78.55

A.7 Severity and non-monotonicity audit

For every model–perturbation pair, we report success rates at severity levels 1–3 and paired adjacent-level transitions. A curve is flagged when the point estimate increases from one severity to the next. This is an empirical property of policy/environment trajectories and is not treated as evidence that the ordered operator contract is invalid. The full transition table identifies affected task–seed keys, task names, and seed indices.

Table 9: Severity curves and flagged reversals.

Model	Perturbation	S1	S2	S3	Reversals
AVA-VLA	action_delay	61.50	40.50	23.00	none
AVA-VLA	action_dropout	62.50	64.00	60.50	1-¿2

Continued on next page

Table 9 (continued)

Model	Perturbation	S1	S2	S3	Reversals
AVA-VLA	action_execution_noise	57.00	54.00	48.50	none
AVA-VLA	ambiguous_instruction	52.50	52.50	56.50	2-¿3
AVA-VLA	brightness	58.00	63.00	59.00	1-¿2
AVA-VLA	camera_pose_shift	61.00	48.00	25.00	none
AVA-VLA	camera_shake	63.00	56.50	27.00	none
AVA-VLA	compression	61.00	63.50	61.00	1-¿2
AVA-VLA	defocus_blur	64.50	61.00	27.50	none
AVA-VLA	distractor_instruction	1.50	1.50	1.00	none
AVA-VLA	distractor_object	67.50	66.00	70.50	2-¿3
AVA-VLA	eef_pose_noise	49.50	22.00	12.00	none
AVA-VLA	gaussian_noise	64.50	64.50	60.00	none
AVA-VLA	joint_state_noise	63.50	54.50	51.00	none
AVA-VLA	missing_proprioception	61.00	39.50	5.00	none
AVA-VLA	motion_blur	62.00	60.00	39.00	none
AVA-VLA	object_dynamics	62.50	63.50	59.50	1-¿2
AVA-VLA	occlusion	45.00	20.50	11.50	none
AVA-VLA	paraphrase	61.00	64.00	61.00	1-¿2
AVA-VLA	shadow	61.50	61.00	61.50	2-¿3
AVA-VLA	splash	52.00	42.50	30.00	none
AVA-VLA	typo	61.50	59.50	52.50	none
CogACT-Libero	action_delay	62.50	44.00	22.50	none
CogACT-Libero	action_dropout	68.50	64.00	63.50	none
CogACT-Libero	action_execution_noise	64.00	66.00	58.50	1-¿2
CogACT-Libero	ambiguous_instruction	40.50	37.50	40.00	2-¿3
CogACT-Libero	brightness	64.00	64.00	62.00	none
CogACT-Libero	camera_pose_shift	63.50	50.50	39.50	none
CogACT-Libero	camera_shake	62.50	66.50	50.00	1-¿2
CogACT-Libero	compression	67.00	63.50	66.50	2-¿3
CogACT-Libero	defocus_blur	64.50	67.50	40.50	1-¿2
CogACT-Libero	distractor_instruction	64.50	65.50	60.50	1-¿2
CogACT-Libero	distractor_object	72.50	76.00	87.50	1-¿2,2-¿3
CogACT-Libero	eef_pose_noise	58.50	38.50	19.50	none
CogACT-Libero	gaussian_noise	67.50	65.00	65.00	none
CogACT-Libero	joint_state_noise	67.50	66.50	61.00	none
CogACT-Libero	missing_proprioception	67.50	65.50	67.00	2-¿3
CogACT-Libero	motion_blur	65.50	63.50	58.00	none
CogACT-Libero	object_dynamics	68.50	69.00	69.50	1-¿2,2-¿3
CogACT-Libero	occlusion	45.00	23.50	15.50	none
CogACT-Libero	paraphrase	63.50	64.00	68.50	1-¿2,2-¿3
CogACT-Libero	shadow	68.00	66.00	65.50	none
CogACT-Libero	splash	58.50	55.00	46.00	none
CogACT-Libero	typo	64.50	56.50	37.00	none
GR00T-N1.7-LIBERO	action_delay	60.00	45.00	21.00	none
GR00T-N1.7-LIBERO	action_dropout	68.50	65.50	63.00	none
GR00T-N1.7-LIBERO	action_execution_noise	62.50	61.00	55.50	none
GR00T-N1.7-LIBERO	ambiguous_instruction	44.00	42.00	40.50	none

Continued on next page

Table 9 (continued)

Model	Perturbation	S1	S2	S3	Reversals
GR00T-N1.7-LIBERO	brightness	62.00	66.50	61.00	1-¿2
GR00T-N1.7-LIBERO	camera_pose_shift	63.00	46.00	29.00	none
GR00T-N1.7-LIBERO	camera_shake	66.00	64.00	31.50	none
GR00T-N1.7-LIBERO	compression	61.50	67.00	65.00	1-¿2
GR00T-N1.7-LIBERO	defocus_blur	67.00	64.00	47.00	none
GR00T-N1.7-LIBERO	distractor_instruction	63.00	64.50	61.50	1-¿2
GR00T-N1.7-LIBERO	distractor_object	72.00	78.50	81.50	1-¿2,2-¿3
GR00T-N1.7-LIBERO	eef_pose_noise	59.50	33.50	16.50	none
GR00T-N1.7-LIBERO	gaussian_noise	66.00	66.00	63.50	none
GR00T-N1.7-LIBERO	joint_state_noise	66.00	60.00	60.50	2-¿3
GR00T-N1.7-LIBERO	missing_proprioception	59.50	61.50	64.50	1-¿2,2-¿3
GR00T-N1.7-LIBERO	motion_blur	63.00	63.50	53.00	1-¿2
GR00T-N1.7-LIBERO	object_dynamics	67.50	66.50	66.50	none
GR00T-N1.7-LIBERO	occlusion	57.00	41.50	20.50	none
GR00T-N1.7-LIBERO	paraphrase	61.50	62.00	61.00	1-¿2
GR00T-N1.7-LIBERO	shadow	63.00	63.00	62.50	none
GR00T-N1.7-LIBERO	splash	58.50	55.00	55.00	none
GR00T-N1.7-LIBERO	typo	61.00	56.00	42.50	none
OpenVLA-OFT	action_delay	59.00	44.00	14.50	none
OpenVLA-OFT	action_dropout	64.50	70.00	64.00	1-¿2
OpenVLA-OFT	action_execution_noise	63.50	59.00	51.00	none
OpenVLA-OFT	ambiguous_instruction	59.50	59.00	61.00	2-¿3
OpenVLA-OFT	brightness	68.50	67.00	65.00	none
OpenVLA-OFT	camera_pose_shift	58.50	43.00	29.50	none
OpenVLA-OFT	camera_shake	66.50	59.50	25.00	none
OpenVLA-OFT	compression	70.00	62.00	63.00	2-¿3
OpenVLA-OFT	defocus_blur	68.00	37.00	1.50	none
OpenVLA-OFT	distractor_instruction	0.50	0.50	0.00	none
OpenVLA-OFT	distractor_object	64.50	61.00	64.00	2-¿3
OpenVLA-OFT	eef_pose_noise	52.00	24.00	11.50	none
OpenVLA-OFT	gaussian_noise	64.50	57.50	56.00	none
OpenVLA-OFT	joint_state_noise	67.50	62.00	57.00	none
OpenVLA-OFT	missing_proprioception	61.00	33.50	6.50	none
OpenVLA-OFT	motion_blur	56.00	26.50	2.00	none

Continued on next page

Table 9 (continued)

Model	Perturbation	S1	S2	S3	Reversals
OpenVLA-OFT	object_dynamics	60.00	60.50	63.00	1-¿2,2-¿3
OpenVLA-OFT	occlusion	41.50	17.50	7.50	none
OpenVLA-OFT	paraphrase	61.00	61.00	56.50	none
OpenVLA-OFT	shadow	67.50	66.00	65.00	none
OpenVLA-OFT	splash	52.00	38.50	26.00	none
OpenVLA-OFT	typo	66.50	61.00	58.00	none
TBD-VLA	action_delay	65.00	55.00	42.50	none
TBD-VLA	action_dropout	70.00	69.00	66.00	none
TBD-VLA	action_execution_noise	66.00	68.00	64.00	1-¿2
TBD-VLA	ambiguous_instruction	38.00	40.00	38.50	1-¿2
TBD-VLA	brightness	62.50	68.00	66.50	1-¿2
TBD-VLA	camera_pose_shift	66.50	62.50	49.50	none
TBD-VLA	camera_shake	67.00	70.50	42.00	1-¿2
TBD-VLA	compression	64.50	67.50	65.00	1-¿2
TBD-VLA	defocus_blur	69.00	69.50	55.50	1-¿2
TBD-VLA	distractor_instruction	67.00	64.00	60.50	none
TBD-VLA	distractor_object	78.50	79.00	88.00	1-¿2,2-¿3
TBD-VLA	eef_pose_noise	65.00	39.50	32.50	none
TBD-VLA	gaussian_noise	65.50	66.50	66.00	1-¿2
TBD-VLA	joint_state_noise	71.50	69.00	62.00	none
TBD-VLA	missing_proprioception	68.00	39.50	0.50	none
TBD-VLA	motion_blur	65.50	66.50	64.50	1-¿2
TBD-VLA	object_dynamics	69.50	68.00	67.50	none
TBD-VLA	occlusion	52.00	42.00	19.00	none
TBD-VLA	paraphrase	70.00	65.00	64.50	none
TBD-VLA	shadow	65.00	71.50	66.50	1-¿2
TBD-VLA	splash	63.50	56.00	49.50	none
TBD-VLA	typo	62.00	44.00	24.50	none
UniVLA	action_delay	51.00	34.00	15.50	none
UniVLA	action_dropout	54.00	56.00	49.50	1-¿2
UniVLA	action_execution_noise	51.50	50.00	44.50	none
UniVLA	ambiguous_instruction	37.00	37.50	37.50	1-¿2
UniVLA	brightness	56.00	52.50	45.00	none
UniVLA	camera_pose_shift	46.00	28.00	24.50	none
UniVLA	camera_shake	56.00	53.00	27.50	none
UniVLA	compression	50.50	24.00	5.00	none
UniVLA	defocus_blur	52.50	2.00	1.00	none
UniVLA	distractor_instruction	55.50	52.00	47.50	none
UniVLA	distractor_object	54.00	50.00	59.50	2-¿3
UniVLA	eef_pose_noise	47.00	24.00	11.00	none
UniVLA	gaussian_noise	57.00	51.00	35.00	none
UniVLA	joint_state_noise	55.00	50.50	47.50	none
UniVLA	missing_proprioception	51.50	54.50	55.00	1-¿2,2-¿3
UniVLA	motion_blur	4.00	0.50	1.00	2-¿3
UniVLA	object_dynamics	54.00	57.00	55.00	1-¿2
UniVLA	occlusion	28.00	18.00	8.00	none
UniVLA	paraphrase	57.00	51.00	54.00	2-¿3
UniVLA	shadow	52.50	52.00	37.00	none
UniVLA	splash	39.00	20.00	9.50	none
UniVLA	typo	52.50	37.50	32.50	none
pi0.5	action_delay	72.50	63.50	35.00	none

Continued on next page

Table 9 (continued)

Model	Perturbation	S1	S2	S3	Reversals
pi0.5	action_dropout	73.50	76.50	72.50	1-¿2
pi0.5	action_execution_noise	73.00	73.50	68.50	1-¿2
pi0.5	ambiguous_instruction	60.50	64.00	64.00	1-¿2
pi0.5	brightness	74.00	75.00	74.50	1-¿2
pi0.5	camera_pose_shift	68.50	46.00	31.50	none
pi0.5	camera_shake	72.00	63.00	23.50	none
pi0.5	compression	74.50	68.00	59.50	none
pi0.5	defocus_blur	72.50	27.00	0.00	none
pi0.5	distractor_instruction	69.00	68.50	71.00	2-¿3
pi0.5	distractor_object	75.50	82.00	93.00	1-¿2,2-¿3
pi0.5	eef_pose_noise	68.50	48.00	31.00	none
pi0.5	gaussian_noise	74.00	71.50	70.00	none
pi0.5	joint_state_noise	76.00	74.00	72.50	none
pi0.5	missing_proprioception	69.50	72.50	70.50	1-¿2
pi0.5	motion_blur	65.00	34.50	0.50	none
pi0.5	object_dynamics	66.50	70.50	71.00	1-¿2,2-¿3
pi0.5	occlusion	62.00	45.00	21.50	none
pi0.5	paraphrase	71.00	70.50	68.50	none
pi0.5	shadow	74.00	75.50	74.00	1-¿2
pi0.5	splash	70.50	66.00	53.50	none
pi0.5	typo	71.50	64.50	57.50	none

A.8 Provenance note

The shared OpenVLA-OFT and pi0.5 merged JSONL files contain 1,404 perturbed rows per model carrying an older manifest hash (`9f29e3019bcfd01cbb8b60883251d6b0a8ed7dfa28c394e3aa24fb876c305f03`). These rows map to current manifest variant IDs, are execution-valid, invoke the original success checker, and preserve the expected 13,200-row count and published aggregate values. They are retained and explicitly reported in `artifacts/manifest_audit.json`; no row was silently discarded.

A.9 Reproducibility package

The accompanying artifact directory contains the manifest JSONL, perturbation parameter CSV, frozen model registry, adapter metadata, episode-level results, aggregate CSV tables, non-monotonicity transitions, hashes, and environment inventory. Existing root-cause audit plots are copied into `supplement/appendix_figures/` without alteration.