ROBOSAFEBENCH: DIAGNOSING DYNAMIC CONTACT RISK IN VISION-LANGUAGE-ACTION POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent progress in Vision-Language-Action models has demonstrated strong generalization across diverse manipulation tasks, yet their safety behavior is still primarily evaluated in static, human-free environments. This poses a critical deployment gap: real environments are dynamic, and robots must react to moving obstacles and human intrusions while preserving both success rates and physical safety. We introduce RoboSafeBench, a non-invasive diagnostic meta-benchmark that upgrades existing robot manipulation benchmarks into dynamic contact testbeds while preserving each host task's instruction, observation–action interface, and success criterion, and adding only a scheduled intruder during evaluation. RoboSafeBench is a measurement tool, not a safety certificate: it records task success and robot–intruder contact jointly, separating contact-free from contact-involved success. We define four evaluation levels—static execution, sudden object intrusion, sliding object intrusion, and human-like reaching intrusion—and instantiate them on LIBERO, RoboTwin, and SimplerEnv across 39 task configurations. We find that: (i) high static success does not imply low dynamic contact; (ii) under human-like reaching, contact-involved success is the most common outcome for every evaluated policy, even where task success stays within 0.7 points of the static rate; (iii) a success-rate lead can consist almost entirely of contact-involved success: under sudden intrusion on SimplerEnv, XVLA exceeds StarVLA by 17.8 points of success rate but by only 0.4 points of contact-free success rate. A privileged yield-to-home reference for $\pi_{0.5}$ on LIBERO lowers reaching-level contact from 60.8% to 3.7% while retaining 89.8% success, showing that such an operating point exists on this host and is not reached by the evaluated policies.

1 INTRODUCTION

Vision-language-action (VLA) models are rapidly emerging as a promising foundation for general-purpose robotic manipulation (Physical Intelligence et al., 2026; Ye et al., 2026; Yu et al., 2026). By combining visual perception, language-conditioned reasoning, and continuous robot control, recent VLA policies have shown impressive capabilities across diverse tabletop manipulation tasks, including object picking, placing, rearrangement, tool use, and long-horizon instruction following.

However, adapting these embodied agents into real environments remains deeply constrained by safety (Zhang et al., 2025b). Most existing VLA policies are trained and evaluated in environments where the scene is largely static, the workspace is isolated from humans, and task success is the dominant evaluation objective. In such settings, a policy can achieve high success by executing a memorized or visually guided manipulation trajectory without needing to reason about unexpected changes or dangers (Tong et al., 2026). Real human environments are fundamentally different. Objects can fall into the robot's workspace, obstacles can slide across the table, and humans can unexpectedly reach near the robot during execution. In these scenarios, a robot must not only complete the commanded task, but also detect dynamic workspace changes, adapt its motion online, and avoid contact with objects or human-like intruders when task progress conflicts with safety.

This gap exposes a key limitation of current embodied evaluation. Existing manipulation benchmarks such as LIBERO (Liu et al., 2023), RoboTwin (Mu et al., 2025), and RoboArena (Atreya et al., 2025) primarily measure static task competence instead of real-world deployment safety. While LIBERO-PRO (Zhou et al., 2025), LIBERO-Plus (Fei et al., 2025), LIBERO-X (Wang et al., 2026),

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.

and RADAR (Chen et al., 2026) consider policies' robustness to visual variation or distribution shift, they perturb the conditions under which a rollout starts and do not capture physical risks, such as collisions induced by objects that move during execution. As a result, current evaluation protocols can overestimate the deployment readiness of robotic policies: a model can appear competent under standard success-rate metrics while behaving unsafely when the environment changes during execution.

Dynamic human-robot interaction further complicates this problem. A necessary precondition for human-safe behavior is avoiding contact with any dynamic intruder—whether or not the policy semantically recognizes it as human. We deliberately separate this necessary condition from the harder question of semantic human recognition: a policy that cannot avoid a human-like reaching motion is unsafe regardless of morphology recognition. RoboSafeBench's L3 level evaluates this necessary condition, establishing the floor that any human-safe policy must clear. Robustness benchmarks do not target this requirement, and dynamic safety suites (Zhang et al., 2025a; Cui et al., 2026) construct dedicated tasks rather than testing it on the host benchmarks where policies are already trained and compared. This is particularly concerning because robot foundation models are increasingly proposed as generalist controllers for open-ended environments, where unexpected human intervention is not an edge case but a normal operating condition.

To address this gap, we introduce RoboSafeBench, a non-invasive meta-benchmark layer that adds controlled dynamic intrusions to existing manipulation benchmarks. The host retains its observation and action interface and success predicate, while a backend adapter implements the disturbance schedule, contact events, and metric logging. The protocol covers static execution (L0), sudden object intrusion (L1), sliding object intrusion (L2), and human-like reaching intrusion (L3). Across 39 task configurations on LIBERO, RoboTwin, and SimplerEnv, the same protocol crosses task success with contact to report contact-free success (CFS), contact-involved success (CIS), contact-free failure (CFF), and contact-involved failure (CIF).

Across the three hosts, contact-involved success is the largest Level-3 outcome for every evaluated checkpoint, while task-success changes differ by host. Levels 1–3 produce different outcome orderings, so we treat them as distinct interaction conditions rather than one difficulty scale. Level-0 SR rankings need not match intruded CFSR rankings, and changing only the reaching speed nearly doubles $\pi_{0.5}$'s contact at unchanged success. A privileged $\pi_{0.5}$ reference shows that low contact with high success is attainable at Level 3 (CR from 60.8% to 3.7%, SR from 96.6% to 89.8%), while at Levels 1 and 2 none of the evaluated stop, yield, or visual-barrier overlays lowers contact without also lowering task success.

Our contributions are fourfold. (i) We introduce a common runtime interface that turns existing manipulation benchmarks into dynamic-contact testbeds without replacing their tasks or success criteria. (ii) We operationalize one robot–intruder contact event across three simulators and cross it with host task success under execution-time intrusions on unchanged host tasks, together with object toppling and within-backend contact severity. (iii) We instantiate diverse dynamic intrusions on three host benchmarks and evaluate 13 released VLA and manipulation checkpoints across 39 task configurations. (iv) We provide a controlled $\pi_{0.5}$ feasibility study that identifies both an attainable low-contact operating point and the task–contact trade-offs that future safety-aware policies must resolve.

2 Related Work

We organize prior work by which quantity an evaluation changes, when that change is applied, and whether task success and logged contact are reported together.

Manipulation competence and robustness. Language-conditioned and multi-task manipulation benchmarks, including CALVIN (Mees et al., 2022), VIMA (Jiang et al., 2023), LIBERO (Liu et al., 2023), RoboTwin (Mu et al., 2025; Chen et al., 2025a), and SimplerEnv (Li et al., 2025), compare policies by task success in the host scene. Robustness benchmarks (Zhou et al., 2025; Fei et al., 2025; Wang et al., 2026; Chen et al., 2026) perturb objects, layouts, initial states, instructions, cameras, illumination, or sensing and still score task success. RoboSafeBench keeps the host success predicate; Level 0 is the unmodified host task.

Execution-time safety evaluation. Other benchmarks score a contact or damage outcome during a rollout beside task completion. Concurrent work, SafeStage, builds 97 purpose-built risk scenarios with pre-existing hazards and scores native task success separately from stage-specific events, including execution-time contact, clearance, and region entry, in one RoboLab embodiment (Luo et al., 2026). OopsieVerse adds a mechanical, thermal, and fluid damage signal during simulation in OmniGibson and RoboCasa, and uses that signal both to evaluate policies and to train safer policies (Balaji et al., 2026). Contact force is an input to its mechanical-damage signal. RoboSafeBench keeps the host task's goal and success predicate, and adds one controlled intruder after execution has started. Logged contact with that intruder is then crossed with host success.

VLA safety and contact-aware evaluation. Several recent evaluations already report task completion separately from a contact-related outcome. SafeVLA-Bench is a post-hoc layer for existing simulator benchmarks (Fan et al., 2026). It leaves the host observations, actions, success predicate, and initial-state distribution unchanged, scores the native rollout with signal-temporal-logic specifications, and reports native success together with a succ-but-unsafe rate and a violation-severity index. Contact force is one of those specifications, together with displacement of non-target bystander objects, held-object stability, and self-contact. The published instantiations are LIBERO and RoboCasa-365. The layer applies across those hosts because it leaves the rollout unchanged. SafeManip monitors finite rollout traces with temporal-logic templates on 50 RoboCasa365 tasks (Huang et al., 2026). Contact is one of eight property categories, evaluated with predicates available from the execution. ForesightSafety-VLA partitions every episode into safe success, unsafe success, safe failure, and unsafe failure over 66 scenarios composed from RoboTwin assets and a 13-category taxonomy; its language and visual perturbations act on rendered observations while the physics scene stays unchanged (Lyu et al., 2026). SafeLIBERO places static obstacles near the target object or along the motion path of 16 LIBERO tasks at episode start and reports collision avoidance rate beside task success (Hu et al., 2025); RoboSafe-LIBERO reuses these obstacle locations but introduces the obstacles during execution. Training-time safety alignment, including constrained learning for VLA policies (Zhang et al., 2025b), changes the policy rather than the evaluation protocol. RoboSafeBench applies the established success–safety partition to a different event: it changes the rollout by introducing a controlled intruder during execution on unchanged host tasks, logs robot–intruder contact with one definition across three simulators, and reports CFSR, CIS, and the two failure cells for that contact.

Relationship to recent dynamic VLA safety benchmarks. Table 1 compares the closest VLA manipulation safety benchmarks rather than surveying every safety benchmark. VLA-Arena's DynamicObstacles suite progresses from a stationary obstacle to linear motion and then two obstacles with curved trajectories; its CBDDL supports simulator-space motion and contact- or force-threshold predicates (Zhang et al., 2025a). LIBERO-Safety defines dedicated physical and semantic safety tasks, including dynamic tabletop avoidance and dynamic avoidance of human hand–object configurations in 3D space (Cui et al., 2026). In its physical track, a safety-constraint violation terminates the episode, so success requires both task completion and constraint satisfaction. RoboSafeBench instead adds onset-conditioned intrusions to existing host tasks across three backends and retains host success and robot–intruder contact as separate bits, making success with contact observable.

3 The RoboSafeBench Evaluation Layer

RoboSafeBench is a portable evaluation layer placed around an existing benchmark. It defines a common runtime contract with four components: the host task, a disturbance controller, a contact monitor, and a metric logger. Porting the benchmark requires a backend adapter for simulator events and contact queries; the disturbance schedule and outcome schema remain unchanged.

3.1 Host task

The host task is the original benchmark episode: its scene, instruction, observation and action interface, and success predicate. RoboSafeBench does not edit that predicate. Level 0 leaves the disturbance controller idle, so the episode is the host task. SR at Level 0 is therefore the host success rate under the original success criterion. Comparisons with Levels 1–3 change the intruder, not the definition of success.

Table 1: Closest VLA manipulation safety benchmarks. Columns: an intruder moves during execution; existing host tasks and success criteria are kept unchanged; a human-like intruder is included; task success and safety are reported as separate outcomes; more than one host benchmark is instrumented. †LIBERO-Safety's physical-track success requires no constraint violation. ‡Rigid reaching proxy. §Concurrent work (arXiv, September 2026).

Benchmark	Moving intruder	Unchanged host tasks	Human-like intruder	Separate success/safety	Multiple hosts
SafeVLA-Bench (Fan et al., 2026)	✗	✓	✗	✓	✓
SafeLIBERO (Hu et al., 2025)	✗	✓	✗	✓	✗
ForesightSafety-VLA (Lyu et al., 2026)	✗	✗	✗	✓	✗
SafeStage§ (Luo et al., 2026)	✗	✗	✗	✓	✗
VLA-Arena (Zhang et al., 2025a)	✓	✗	✗	✓	✗
LIBERO-Safety (Cui et al., 2026)	✓	✗	✓	✗†	✗
RoboSafeBench (ours)	✓	✓	✓‡	✓	✓

3.2 WRAPPER

Three modules implement the runtime contract around the host task.

The *disturbance controller* selects the regime and updates the intruder. At Level 0 it remains idle; at Levels 1–3 it introduces the object or reaching proxy defined below. A small calibration file maps the shared schedule to valid poses and timing in each simulator, while the host continues to provide observations, actions, and task success.

The *contact monitor* maps simulator events to a common episode-level contact bit and, when available, a severity scalar. Across all three hosts, the common event is contact between the robot body (arm or gripper) and the RoboSafeBench intruder; contact involving only a task or grasped object is not counted in CR. The adapter-level query and filtering rules are listed in Appendix B.

The *metric logger* stores the host success bit beside the monitor's contact bit and aggregates the common outcomes in Section 3.5. This separation is the portable part of RoboSafeBench: the host adapter handles physics queries, while the evaluation protocol consumes the same success–contact record.

3.3 INTERACTION REGIMES

Table 2 names the four regimes. Figure 1 shows them on RoboTwin; the LIBERO and SimplerEnv views are in Appendix H. The levels vary how an intruder appears and moves, allowing comparison across abrupt insertion, extended motion, and shared-space reaching. Across episodes, the schedule rotates obstacle instances and motion configurations instead of replaying one fixed path.

Level 1. The obstacle starts outside the camera frustum. When the planar distance from the end-effector to the calibrated target region falls below d_{trig}, the controller inserts it at a backend-specific pose near the manipulation goal and hands it to the host physics. This is a near-surface insertion or a short vertical offset, not a simulator-independent free fall. The offset is chosen so the object is visible



(a) L0: no added intrusion (b) L1: Sudden intrusion (c) L2: Sliding path (d) L3: Shared reaching

Figure 1: Robot-centric views of the four regimes on **RoboTwin**. LIBERO and SimplerEnv views are in Appendix H.

Table 2: Interaction regimes.

Level	Regime	What the controller does	Relation to the host task
L0	Static	No intruder	Host task and host success criterion
L1	Sudden intrusion	Near-surface insertion near the task region	Same success criterion; abrupt workspace change
L2	Sliding intrusion	Object traverses the workspace on a calibrated path	Same success criterion; extended moving intrusion
L3	Shared reaching	Proxy arm enters the target region	Same success criterion; human-like reaching proxy

and does not clip the scene geometry. Toppling is recorded when the object's tilt exceeds 30° from its upright reference.

Level 2. An obstacle moves from a calibrated $\mathbf{p}_{\text{start}}$ to a $\mathbf{p}_{\text{end}}$ near the goal, with start step t_{start} and end step t_{end}. The schedule rotates three entry points and two spatial path families, linear and parabolic; the latter uses arc height h_{arc}. The intruder is visible for multiple control steps before it reaches the task region.

Level 3. A reaching proxy passes through hidden, entry, approach, dwell, and withdrawal stages. Contact with the robot triggers its transition to withdrawal; if no contact occurs, it reaches the configured target and then withdraws. Its schedule likewise rotates three entry points and linear or parabolic spatial paths. Within each stage the selected path is time-parameterized with the Flash–Hogan minimum-jerk profile (Flash & Hogan, 1985) (Appendix L). The proxy is a rigid body calibrated per backend: a 4.5 kg MuJoCo hand and forearm with softened contact on RoboSafe-LIBERO, a 4.5 kg hand proxy on RoboSafe-RoboTwin, and a 4.5 kg SAPIEN hand actor at scale 0.6 on RoboSafe-SimplerEnv.

Residual obstacles. After the scheduled intrusion phase, the Level-1 and Level-2 objects remain in the workspace under host physics, whereas the Level-3 proxy withdraws.

3.4 Hosts and calibration

The layer is instantiated on LIBERO (Liu et al., 2023), RoboTwin (Mu et al., 2025; Chen et al., 2025a), and SimplerEnv (Li et al., 2025). Appendix Table 6 gives the per-host adapter parameters used to keep the shared interaction patterns visible and physically valid in each simulator. The reported LIBERO Level 3 peak speed is 0.7 m/s. Appendix Table 7 states the task coverage used in the main comparison. SimplerEnv is restricted to the Bridge tasks in that table.

3.5 Outcome metrics

Task success and contact are separate bits. Let $\mathcal{E}$ be the valid episodes, $N = |\mathcal{E}|$, $s_i \in \{0, 1\}$ the host success bit, and $c_i \in \{0, 1\}$ the monitor's logged-contact bit, using the geometry in Section 3. The four outcome cells are contact-free success (CFS: $s_i = 1, c_i = 0$), contact-involved success (CIS: $s_i = 1, c_i = 1$), contact-free failure (CFF: $s_i = 0, c_i = 0$), and contact-involved failure (CIF: $s_i = 0, c_i = 1$). They partition $\mathcal{E}$:

$$N = n_{\text{CFS}} + n_{\text{CIS}} + n_{\text{CFF}} + n_{\text{CIF}}. \tag{1}$$

The reported rates are

$$\text{SR} = \frac{1}{N}\sum_{i\in\mathcal{E}} s_i, \qquad \text{CR} = \frac{1}{N}\sum_{i\in\mathcal{E}} c_i, \qquad \text{CFSR} = \frac{1}{N}\sum_{i\in\mathcal{E}} s_i(1 - c_i). \tag{2}$$

CFSR is the CFS rate, n_{CFS}/N. The CIS rate is n_{CIS}/N. Because every successful episode is either contact-free or contact-involved, and every contact episode is either a success or a failure,

$$\text{SR} = \text{CFSR} + \frac{n_{\text{CIS}}}{N}, \qquad \text{CR} = \frac{n_{\text{CIS}}}{N} + \frac{n_{\text{CIF}}}{N}. \tag{3}$$

The first identity writes the SR−CFSR gap as the CIS rate. The second identity is why a lower CR need not raise CFSR: additional contact-free episodes can still fall into CFF rather than CFS. CFSR counts only successes without contact, so $\mathrm{CFSR} \leq 1 - \mathrm{CR}$, and the same SR and CR can occupy different cells. All four cells are outcome labels under a fixed intrusion protocol.

Topple rate uses the backend `toppled` field and is defined for the Level 1 and Level 2 objects only.

Level 3 contact severity is summarized by $F_{\mathrm{peak,mean}}$. Within each seed, every valid episode contributes its maximum robot–proxy contact scalar, with zero assigned to non-contact episodes; we average these episode peaks and report the mean and sample standard deviation of the three seed-level averages over $\{1, 7, 2007\}$. This full-denominator statistic combines contact incidence with episode-peak magnitude rather than describing only contact episodes. Appendix F specifies the force extraction and aggregation for each backend.

4 Experiments

4.1 Setup

We evaluate released checkpoints without RoboSafeBench-specific safety adaptation on RoboSafe-LIBERO, RoboSafe-RoboTwin, and RoboSafe-SimplerEnv. The benchmark covers 39 task configurations: 16 LIBERO tasks, 20 RoboTwin tasks, and three SimplerEnv Bridge tasks. RoboSafe-LIBERO reuses the obstacle locations introduced by SafeLIBERO (Hu et al., 2025), but applies the RoboSafeBench runtime schedule and outcome schema. Level 0 is the static host task. Levels 1–3 are sudden object intrusion, sliding intrusion, and human-like reaching. Each model–task–level–seed cell has 50 episodes on RoboSafe-LIBERO, 20 on RoboSafe-RoboTwin, and 50 on RoboSafe-SimplerEnv, using seeds $\{1, 7, 2007\}$. Episodes marked `setup_invalid` are excluded.

RoboSafe-LIBERO uses π_0 (Black et al., 2024), $\pi_{0.5}$ (Physical Intelligence et al., 2025), XVLA (Zheng et al., 2025), StarVLA (StarVLA Community, 2026), and TurboVLA (Xie et al., 2026). RoboSafe-RoboTwin uses LingBot-VA (Li et al., 2026), InternVLA-A1 (Cai et al., 2026), StarVLA, XVLA, and FastWAM (Yuan et al., 2026). RoboSafe-SimplerEnv uses InternVLA-M1 (Chen et al., 2025b), StarVLA, and XVLA. StarVLA and XVLA appear on more than one host. Each name is the checkpoint released for that host, not one set of weights shared across backends. A separate RoboSafe-LIBERO study applies privileged interventions to the same $\pi_{0.5}$ checkpoint. Those arms serve as a privileged dynamic-avoidance reference (Section 4.3).

We report SR, CR, and CFSR, together with CIS, CFF, and CIF, as defined in Section 3.5. Full means and sample standard deviations are in Appendix Tables 8–10. L1/L2 toppling and within-backend L3 contact severity are reported in Appendix Tables 11 and 14. Paired end-effector deviation from successful Level-0 twins is reported in Appendix E.

4.2 Task Success and Contact across Interaction Regimes

Figure 2 separates task success from contact-free success and shows the Level-3 outcome partition, and Table 3 summarizes each host by level. SR and CFSR coincide at Level 0, where no intruder is added. At Level 3, CIS is the largest outcome cell for every evaluated model; because contact triggers proxy withdrawal, CIS does not measure recovery under continued intrusion. On RoboSafe-LIBERO, every model's SR remains within 0.7 percentage points of its Level-0 rate despite contact in most episodes; on RoboSafe-RoboTwin, CR exceeds 93% and CFSR is near the floor. On RoboSafe-SimplerEnv, StarVLA and XVLA each lose more than ten points of SR at Level 3.

At Level 1, the three hosts diverge. On RoboSafe-LIBERO, sudden intrusion produces the largest loss of task completion, with contact still common; at Level 1, $\pi_{0.5}$ has more CIS than CIF, while TurboVLA's largest cell is CIF. RoboSafe-RoboTwin has lower contact at Level 1 than at Levels 2 and 3, where CFSR approaches the floor. On RoboSafe-SimplerEnv at Level 1, XVLA's SR exceeds StarVLA's by 17.78 points, but their CFSR differs by only 0.44 points: most of the SR gap is CIS. Level 2 has higher CR than Level 3 for every SimplerEnv checkpoint.

Static rank need not transfer. Level-0 SR rankings do not consistently match CFSR rankings under intrusion in these evaluations. On RoboSafe-SimplerEnv, XVLA leads Level-0 SR by 15.1–

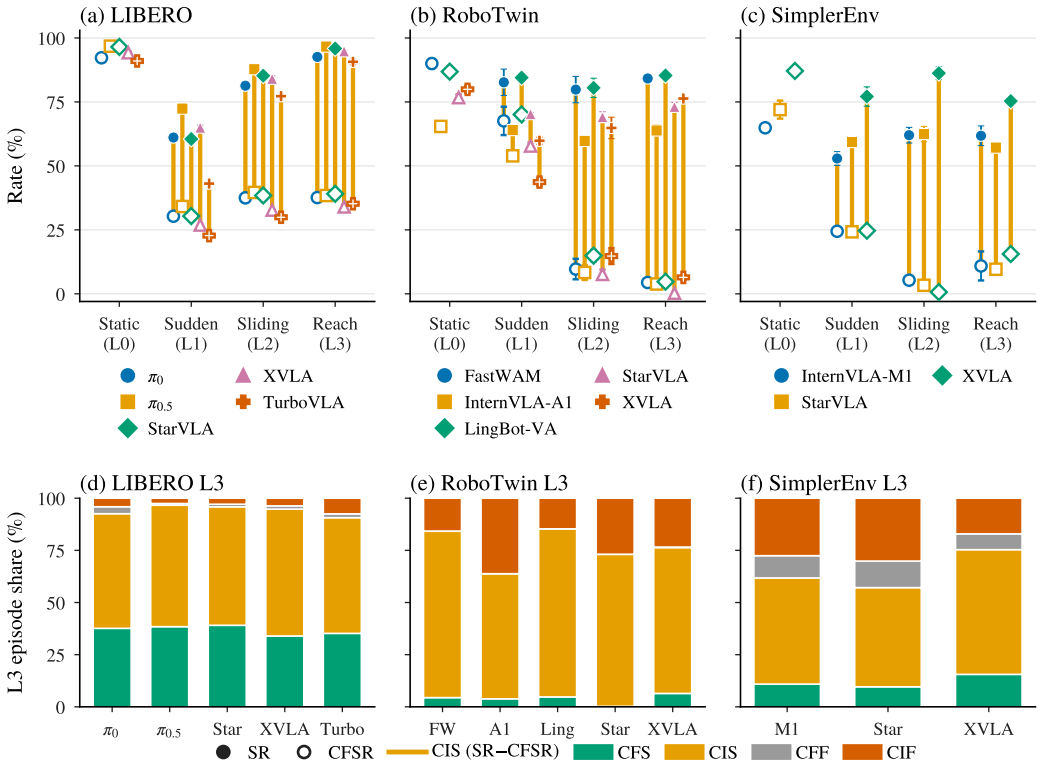


Figure 2: **Task success and contact under L0–L3, one host per panel.** (a–c) Each marker shape and color is one checkpoint released for that host. Filled markers are model-level mean SR and open markers are CFSR; the orange segment is CIS. Error bars are the sample standard deviation across seeds $\{1, 7, 2007\}$. (d–f) The same Level-3 rates as CFS, CIS, CFF, and CIF; the orange segment in (a–c) and the orange block in (d–f) are the same quantity. Tasks are pooled within each seed; points are means over seeds. Numerical values are in Appendix Tables 8–10.

Table 3: **Host-level summary across checkpoints.** Each cell is the [min, max] over the checkpoints evaluated on that host. ΔSR is the change from each checkpoint's own Level-0 SR, in percentage points; the other columns are rates in percent. The last column counts checkpoints by their largest outcome cell. Per-checkpoint values are in Appendix Tables 8–10.

Host	Level	ΔSR	CR	CFSR	CIS	Largest cell
LIBERO	L1	[−47.9, −24.4]	[64.6, 73.0]	[22.8, 34.1]	[20.3, 38.3]	CIF 3/5, CIS 2/5
(5 models)	L2	[−13.8, −9.0]	[58.8, 66.4]	[29.9, 39.6]	[43.8, 51.3]	CIS 5/5
	L3	[−0.6, +0.6]	[59.0, 64.5]	[34.0, 39.1]	[55.0, 60.8]	CIS 5/5
RoboTwin	L1	[−20.1, −1.4]	[20.6, 41.3]	[43.7, 70.1]	[10.1, 16.2]	CFS 5/5
(5 models)	L2	[−15.1, −5.7]	[83.9, 91.5]	[7.6, 14.9]	[50.1, 70.2]	CIS 5/5
	L3	[−5.8, −1.5]	[93.3, 99.7]	[0.3, 6.4]	[60.0, 80.6]	CIS 5/5
SimplerEnv	L1	[−12.7, −10.0]	[63.8, 72.9]	[24.2, 24.7]	[28.4, 52.4]	CIS 2/3, CIF 1/3
(3 models)	L2	[−9.6, −0.9]	[93.3, 99.3]	[0.7, 5.3]	[56.7, 85.6]	CIS 3/3
	L3	[−14.9, −3.1]	[76.9, 78.4]	[9.6, 15.6]	[47.5, 59.8]	CIS 3/3

22.2 points, yet all three checkpoints reach the same Level-1 CFSR (24.2–24.7%), and XVLA has the highest Level-1 CR (72.9%). On RoboSafe-RoboTwin, InternVLA-A1 has the lowest Level-0 SR (65.4% versus 79.9% for XVLA) but the higher Level-1 CFSR (53.9% versus 43.7%), because XVLA's Level-1 CR (41.3%) is 16–21 points above every other checkpoint.

Toppling and severity. Contact is not the only physical consequence. The intruding object topples in 54.1–63.0% of Level-1 episodes on RoboSafe-LIBERO and in 62.0–72.2% of Level-1/2 episodes on RoboSafe-SimplerEnv; on RoboSafe-RoboTwin, toppling rises from 9.7–25.7% at Level 1 to 33.3–37.9% at Level 2, tracking the rise in CR (Appendix Table 11). Severity separates checkpoints that CR does not: LIBERO Level-3 CR spans only 59.0–64.5%, yet TurboVLA's $F_{\text{peak,mean}}$ (19.4 ± 1.2 N) exceeds those of π_0, $\pi_{0.5}$, and StarVLA (17.0 ± 0.2, 14.3 ± 2.5, and 15.3 ± 1.4 N; Appendix Table 14).

Static-placement control. A static-placement control shows that Level-2 contact is not produced by intruder motion alone. With the Level-2 obstacle placed at its end pose from the first step ($\pi_{0.5}$, seed 7, same 800 initial states), CR rises to 69.6% from 59.1% and CFSR falls to 29.0% from 39.0% (Appendix K). The policy contacts a resting, fully visible obstacle at least as often as a sliding one, so the logged contact reflects how the policy handles an obstacle in its workspace.

Intrusion speed changes contact, not success. Rerunning the LIBERO Level-3 path at 0.3 and 1.2 m/s instead of 0.7 m/s, $\pi_{0.5}$'s SR stays within 96.1–96.7% while CR rises by 30.6 points, from 34.8% to 65.4%, and CFSR falls from 63.7% to 33.5% (Table 4). The layer thus exposes speed as a per-host difficulty knob; where Level-3 CR saturates, as on RoboSafe-RoboTwin (93.3–99.7%), the CIS/CIF split still separates checkpoints (CIF 14.7–36.2%).

4.3 Privileged Dynamic-Avoidance Reference

We use the released $\pi_{0.5}$ checkpoint to test whether the benchmark admits operating points with substantially less contact without forcing task failure. All arms run on the same 16 RoboSafe-LIBERO tasks (Table 5). Stop and yield use privileged timing or control. AEGIS is the plug-and-play vision-based control-barrier filter introduced by VLSA (Hu et al., 2025); here it follows the privileged stop or yield, with visual fallback to simulator geometry in 4.2–4.3% of episodes (Appendix Table 15). Yield-to-home is implemented as continuous physical control rather than a simulator-state reset. Its retreat, waiting, and task-resumption phases all remain inside the episode contact log and time budget. Level 3 reports yield-to-home alone, because the reaching proxy leaves no residual obstacle after withdrawal.

At Level 1, Stop+AEGIS halves CR (64.6% to 30.5%) and raises CFSR from 34.1% to 39.4%, while SR falls from 72.4% to 49.6% as contact-free failures grow (CFF 1.3% to 30.1%). At Level 2, Yield+AEGIS likewise lowers CR to 31.0% with CFSR close to the base rate (38.0% versus 39.6%). Yield-to-home alone raises Level-2 CR to 68.9%: once its path ends or it is contacted, the Level-2 obstacle falls under host physics and comes to rest in the workspace, so resuming the task meets a resting obstacle, consistent with the 69.6% CR of the static-placement control. At Level 3, privileged yield-to-home lowers CR from 60.8% to 3.7% and raises CFSR from 38.4% to 86.3%, while SR is 89.8%, compared with 96.6% for the base checkpoint; its CR stays within 3.7–4.0% at reaching speeds from 0.3 to 1.2 m/s (Table 4). Full-denominator $F_{\text{peak,mean}}$ falls from 14.30 ± 2.52 N for the base checkpoint to 0.20 ± 0.02 N under Level-3 yield-to-home.

At Levels 1 and 2, a well-timed yield is therefore not enough: high contact-free success also requires replanning around an obstacle that remains in the workspace.

4.4 Key Findings and Research Implications

F1: Task success alone obscures contact involvement. On RoboSafe-LIBERO at Level 3, SR stays near its Level-0 rate while contact remains common; on SimplerEnv at Level 1, XVLA's SR advantage over StarVLA is almost entirely CIS. Changing only the reaching speed nearly doubles $\pi_{0.5}$'s CR at unchanged SR (Table 4). Comparisons should pair SR with CFSR and the four outcome cells, asking whether a success gain also raises contact-free completion.

F2: Dynamic safety is regime-specific. The largest outcome cell changes across regimes (Table 3), and SimplerEnv has more contact at Level 2 than Level 3 for every checkpoint, so the levels probe distinct interactions rather than one severity scale. A resting obstacle at the Level-2 end pose draws even more contact than the sliding one (Appendix K), so handling an obstacle that enters and stays in the workspace is itself unsolved.

Table 4: **LIBERO Level-3 reaching speed on a fixed path.** Mean with sample standard deviation across seeds as subscript, in percent, after pooling 16 tasks within each seed; N=800 valid episodes per seed. Yield-home is the privileged arm of Section 4.3; the 0.7 m/s rows repeat Table 5.

Arm	Peak	SR	CR	CFSR	CIS
$\pi_{0.5}$	0.3 m/s	$96.1_{\pm 0.4}$	$34.8_{\pm 0.4}$	$63.7_{\pm 0.4}$	$32.4_{\pm 0.7}$
	0.7 m/s	$96.6_{\pm 0.3}$	$60.8_{\pm 2.2}$	$38.4_{\pm 2.0}$	$58.2_{\pm 2.4}$
	1.2 m/s	$96.7_{\pm 0.2}$	$65.4_{\pm 0.2}$	$33.5_{\pm 0.3}$	$63.2_{\pm 0.4}$
Yield-home	0.3 m/s	$89.8_{\pm 0.4}$	$4.0_{\pm 0.3}$	$86.0_{\pm 0.7}$	$3.8_{\pm 0.2}$
	0.7 m/s	$89.8_{\pm 0.2}$	$3.7_{\pm 0.1}$	$86.3_{\pm 0.4}$	$3.5_{\pm 0.3}$
	1.2 m/s	$90.3_{\pm 0.5}$	$3.8_{\pm 0.2}$	$86.7_{\pm 0.7}$	$3.7_{\pm 0.3}$

Table 5: **Dynamic-avoidance reference for $\pi_{0.5}$ on RoboSafe-LIBERO.** Mean with sample standard deviation across seeds as subscript, in percent, after pooling 16 tasks within each seed. Each row has $N = 2{,}400$ valid episodes.

Level	Arm	CR	SR	CFSR	CIS	CFF	CIF
L1	$\pi_{0.5}$	$64.6_{\pm 0.7}$	$72.4_{\pm 1.3}$	$34.1_{\pm 0.9}$	$38.3_{\pm 0.9}$	$1.3_{\pm 0.4}$	$26.3_{\pm 1.3}$
	Stop+AEGIS	$30.5_{\pm 1.1}$	$49.6_{\pm 0.6}$	$39.4_{\pm 0.3}$	$10.2_{\pm 0.4}$	$30.1_{\pm 0.9}$	$20.3_{\pm 1.4}$
L2	$\pi_{0.5}$	$59.0_{\pm 0.8}$	$87.8_{\pm 1.7}$	$39.6_{\pm 0.6}$	$48.2_{\pm 1.9}$	$1.4_{\pm 0.6}$	$10.8_{\pm 1.2}$
	Yield-home	$68.9_{\pm 0.8}$	$65.9_{\pm 0.4}$	$26.4_{\pm 0.7}$	$39.5_{\pm 1.0}$	$4.7_{\pm 0.3}$	$29.4_{\pm 0.5}$
	Yield+AEGIS	$31.0_{\pm 2.1}$	$47.2_{\pm 1.3}$	$38.0_{\pm 2.3}$	$9.1_{\pm 1.0}$	$30.9_{\pm 0.3}$	$21.9_{\pm 1.0}$
L3	$\pi_{0.5}$	$60.8_{\pm 2.2}$	$96.6_{\pm 0.3}$	$38.4_{\pm 2.0}$	$58.2_{\pm 2.4}$	$0.8_{\pm 0.2}$	$2.5_{\pm 0.2}$
	Yield-home	$3.7_{\pm 0.1}$	$89.8_{\pm 0.2}$	$86.3_{\pm 0.4}$	$3.5_{\pm 0.3}$	$10.0_{\pm 0.3}$	$0.2_{\pm 0.1}$

F3: Contact rate alone is not the whole picture. Static SR rankings do not consistently match intruded CFSR rankings, toppling records a scene consequence that CR does not, and contact severity separates checkpoints whose CR is similar. Safety comparisons should report the regime, CFSR, and these physical side effects together.

F4: An attainable target and an open control problem. Privileged Level-3 yield-to-home attains low contact with substantial task completion, whereas the Level-1/2 overlays trade contact for contact-free failure; this motivates unprivileged yield–replan–resume, judged by CFSR, CFF, and severity rather than CR alone.

5 Conclusion

RoboSafeBench exposes contact during execution while keeping host tasks and success criteria unchanged. Across 13 released checkpoints on three hosts, contact-involved success is the most common outcome under human-like reaching, static SR rankings do not consistently match intruded CFSR rankings; and, for $\pi_{0.5}$ on RoboSafe-LIBERO, intrusion speed changes contact without changing success. Privileged yield-to-home shows that low contact with high Level-3 success is attainable on LIBERO; reaching it without privileged event timing, and replanning around obstacles that remain in the workspace, are open problems the benchmark makes measurable.

As VLA policies move toward homes and other shared spaces, we encourage reporting contact alongside success whenever models are compared, so that gains in manipulation capability are matched by measurable progress in scene and human safety.

AI Use Statement

In this work, we used large language models and other generative-AI tools to aid and polish writing, including language editing and manuscript organization, and for retrieval and discovery of potentially relevant literature. The authors independently reviewed the AI-assisted text, verified citations against the corresponding sources, and take responsibility for the final content of the paper.

Reproducibility Statement

Our code is available at https://anonymous.4open.science/r/dynamicEmbodySafty-Bench-jklfjals/README.md.

References

Pranav Atreya, Karl Pertsch, Tony Lee, Moo Jin Kim, Arhan Jain, Artur Kuramshin, Clemens Eppner, Cyrus Neary, Edward Hu, Fabio Ramos, et al. RoboArena: Distributed real-world evaluation of generalist robot policies. *arXiv preprint arXiv:2506.18123*, 2025.

Arnav Balaji, Arpit Bahety, Sriniket Ambatipudi, Daniel Lam, Junhong Xu, and Roberto Martín-Martín. OopsieVerse: A safety benchmark with damage-aware simulation for robot manipulation. In *Robotics: Science and Systems*, 2026. URL https://arxiv.org/abs/2606.31993. arXiv:2606.31993.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. URL https://arxiv.org/abs/2410.24164.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. InternVLA-A1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

Tianxing Chen, Kaixuan Wang, Zhaohui Yang, Yuhao Zhang, Zanxin Chen, Baijun Chen, Wanxi Dong, Ziyuan Liu, Dong Chen, Tianshuo Yang, Haibao Yu, Xiaokang Yang, Yusen Qin, Zhiqiang Xie, Yao Mu, Ping Luo, et al. Benchmarking generalizable bimanual manipulation: RoboTwin dual-arm collaboration challenge at CVPR 2025 MEIS workshop. *arXiv preprint arXiv:2506.23351*, 2025a. URL https://arxiv.org/abs/2506.23351.

Xinyi Chen, Yilun Chen, Yanwei Fu, Ning Gao, Jiaya Jia, et al. InternVLA-M1: A spatially guided vision-language-action framework for generalist robot policy. *arXiv preprint arXiv:2510.13778*, 2025b.

Yuhao Chen, Zhihao Zhan, Xiaoxin Lin, Zijian Song, Hao Liu, Qinhan Lyu, Yubo Zu, Xiao Chen, Zhiyuan Liu, Tao Pu, et al. RADAR: Benchmarking vision-language-action generalization via real-world dynamics, spatial-physical intelligence, and autonomous evaluation. *arXiv preprint arXiv:2602.10980*, 2026.

Rongxu Cui, Zongzheng Zhang, Jingrui Pang, Haohan Chi, Jinbang Guo, Saining Zhang, Shaoxuan Xie, Xin Jin, Yao Mu, Jiaolong Yang, Guocai Yao, Xianyuan Zhan, Ya-Qin Zhang, and Hao Zhao. LIBERO-Safety: A comprehensive benchmark for physical and semantic safety in vision-language-action models. *arXiv preprint arXiv:2606.23686*, 2026. URL https://arxiv.org/abs/2606.23686.

Jialiang Fan, Weizhe Xu, Oleg Sokolsky, Insup Lee, and Fanxin Kong. SafeVLA-Bench: A benchmark for the success–safety gap in vision-language-action models. *arXiv preprint arXiv:2606.00773*, 2026. URL https://arxiv.org/abs/2606.00773.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Tamar Flash and Neville Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985. doi: 10.1523/JNEUROSCI.05-07-01688.1985.

Songqiao Hu, Zeyi Liu, Shuang Liu, Jun Cen, Zihan Meng, Shihefeng Wang, Xiang Li, and Xiao He. VLSA: Vision-language-action models with plug-and-play safety constraint layer. *arXiv preprint arXiv:2512.11891*, 2025. URL https://arxiv.org/abs/2512.11891.

Chengyue Huang, Khang Vo Huynh, Sebastian Elbaum, Zsolt Kira, and Lu Feng. SafeManip: A property-driven benchmark for temporal safety evaluation in robotic manipulation. *arXiv preprint arXiv:2605.12386*, 2026. URL https://arxiv.org/abs/2605.12386.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: General robot manipulation with multimodal prompts. In *International Conference on Machine Learning (ICML)*, 2023.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In *Proceedings of the 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3705–3728, 2025. URL https://arxiv.org/abs/2405.05941. arXiv:2405.05941.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.03310.

Jinzhu Luo, Qi Zhang, Wei Wang, and Wei Jiang. SafeStage: Evaluating safety before, during, and after vision-language-conditioned robot manipulation. *arXiv preprint arXiv:2609.21223*, 2026. URL https://arxiv.org/abs/2609.21223.

Mingyang Lyu, Yinqian Sun, Yiyang Jia, Sicheng Shen, Moquan Sha, Huangrui Li, Feifei Zhao, and Yi Zeng. ForesightSafety-VLA: A unified diagnostic safety benchmark for vision-language-action models. *arXiv preprint arXiv:2606.27079*, 2026. URL https://arxiv.org/abs/2606.27079.

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 (RA-L)*, 2022.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. RoboTwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27649–27660, 2025. URL https://arxiv.org/abs/2409.02920. arXiv:2409.02920.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. doi: 10.48550/arXiv.2504.16054. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

StarVLA Community. StarVLA: A Lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Baoshun Tong, Haoran He, Ling Pan, Yang Liu, and Liang Lin. Uncovering linguistic fragility in vision-language-action models via diversity-aware red teaming. *arXiv preprint arXiv:2604.05595*, 2026.

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.

Hengyi Xie, Chenfei Yao, Xianjin Wu, Xuanyang Xi, Yiping Tang, Di Xu, Yingying Zhu, Dingkang Liang, Xiang Bai, and Han Ding. TurboVLA: Real-time vision-language-action model at 32 Hz on an RTX 4090 with <1 GB VRAM. *arXiv preprint arXiv:2607.27205*, 2026. URL https://arxiv.org/abs/2607.27205.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Checheng Yu, Chonghao Sima, Gangcheng Jiang, Hai Zhang, Haoguang Mai, Hongyang Li, Huijie Wang, Jin Chen, Kaiyang Wu, Li Chen, et al. χ_0: Resource-aware robust manipulation via taming distributional inconsistencies. *arXiv preprint arXiv:2602.09021*, 2026.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-WAM: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026.

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

Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Josef Dai, Yuanpei Chen, and Yaodong Yang. SafeVLA: Towards safety alignment of vision-language-action model via constrained learning. *arXiv preprint arXiv:2503.03480*, 2025b.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, et al. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

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

A Supplementary Protocol Details

Table 6: Per-host disturbance calibration used for the reported runs. Parameters are not numerically equivalent across simulators; the few per-task overrides are part of the released configuration.

Parameter	**LIBERO**	**RoboTwin**	**SimplerEnv**
L1 trigger distance	0.15 m	0.10 m default	0.20 m
L1 drop height	0.01 m	0.01 m	0.01 m
L2 arc height	0.20 m	0.12 m	0.10 m
L3 timing / speed	medium, peak 0.7 m/s	entry 6000 sim steps; onset delay 20 task steps	slow, peak 0.3 m/s
L3 proxy mass	4.5 kg	4.5 kg	4.5 kg, scale 0.6

Table 7: Main-benchmark coverage. Each cell is one model–task–level–seed combination.

Benchmark	**Task families**	**Tasks**	**Models**	**Episodes / cell**
LIBERO	spatial / object / goal / long	16 task IDs	5	50
RoboTwin	fine-grained manipulation	20 tasks	5	20
SimplerEnv	Bridge subset	3 tasks	3	50

B Contact Instrumentation

Every host adapter maps simulator contacts to the episode-level bit c_i used by the shared outcome schema. The counted event is the same on RoboSafe-LIBERO, RoboSafe-RoboTwin, and RoboSafe-SimplerEnv: contact between the RoboSafeBench intruder and the robot body, including the arm and gripper. Contact involving only a task object or grasped object is excluded. On the SAPIEN backends, a contact report counts only if its impulse norm is at least 10^{-6} N·s; this noise floor separates geometrically touching bodies from an actual impact. Thus CR has one interaction scope across hosts, although the simulator query and the within-backend severity scalar remain backend-specific.

The monitor is evaluated at every simulation step while the intruder exists. On RoboSafe-LIBERO it scans the MuJoCo contact list for pairs with one geometry on the robot (arm links or gripper) and the other on the intruder; on RoboSafe-RoboTwin and RoboSafe-SimplerEnv it scans SAPIEN contact reports for the same actor pair. The episode bit is $c_i = 1$ if any step passes these filters, and the first such step is stored as the first-contact step. The same filtered contacts supply the per-step severity scalar in Appendix F. Episodes flagged `setup_invalid` are removed before aggregation; force-anomaly flags are retained (Appendix G).

C Limitations

RoboSafeBench is a diagnostic evaluation layer for task and contact outcomes under controlled runtime intrusions. Its scores are not real-world human-safety certificates: the L3 proxy is a rigid minimum-jerk reaching body rather than a biomechanical human model, and $F_{\mathrm{peak,mean}}$ is a within-backend contact-severity summary, not an injury measure or a quantity comparable across simulators.

Scope of the evaluation. The main benchmark characterizes released checkpoints as shipped, on the task subsets listed in Appendix Table 7. It measures observable outcomes rather than the causes behind them, such as training data, action representation, or control frequency. Results are reported as task-pooled aggregates; per-task rates can be recomputed from the released per-episode logs.

Interaction regimes and targeted controls. Each level is a complete interaction condition rather than a single-factor manipulation, so level differences are not attributed to one factor such as obstacle motion or the Level-3 withdrawal rule. The reaching-speed check (Appendix J) and static placement at the Level-2 end pose (Appendix K) are targeted controls for one checkpoint on RoboSafe-LIBERO, the latter for one seed. Because Level-3 contact triggers proxy withdrawal, task completion after contact is not a recovery measure. Extending the protocol with a static control for Level 1, fixed-time withdrawal for Level 3, and per-host graded intrusion speed where outcome cells approach saturation is a natural next step.

Privileged reference. The stop, yield, and AEGIS arms use privileged disturbance timing on one checkpoint and serve as a feasibility reference for the benchmark rather than as learned baselines.

D Full Main-Benchmark Results

Figure 2 provides the compact visual summary used in the main text. Tables 8–10 retain the complete mean $\pm$ sample-standard-deviation results and denominators for all three backends.

Table 8: **Full main results on RoboSafe-LIBERO.** Each entry is the mean, with the sample standard deviation across seeds as subscript, $\{1, 7, 2007\}$ after pooling the 16 tasks within each seed. Every entry uses 800 valid episodes per seed. Rows are grouped by level for within-level model comparison. All rates are percentages. CFSR is the contact-free success rate. CIS denotes success with logged contact, and CFF and CIF denote failure without and with logged contact.

Level	Model	SR	CFSR	CR	CIS	CFF	CIF
L0	π_0	$92.25_{\pm 0.87}$	$92.25_{\pm 0.87}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$7.75_{\pm 0.87}$	$0.00_{\pm 0.00}$
	$\pi_{0.5}$	$96.79_{\pm 0.52}$	$96.79_{\pm 0.52}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$3.21_{\pm 0.52}$	$0.00_{\pm 0.00}$
	StarVLA	$96.50_{\pm 0.00}$	$96.50_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$3.50_{\pm 0.00}$	$0.00_{\pm 0.00}$
	XVLA	$94.21_{\pm 0.62}$	$94.21_{\pm 0.62}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$5.79_{\pm 0.62}$	$0.00_{\pm 0.00}$
	TurboVLA	$91.00_{\pm 0.00}$	$91.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$9.00_{\pm 0.00}$	$0.00_{\pm 0.00}$
L1	π_0	$61.08_{\pm 0.44}$	$30.33_{\pm 1.68}$	$64.88_{\pm 1.96}$	$30.75_{\pm 1.39}$	$4.79_{\pm 0.47}$	$34.13_{\pm 0.57}$
	$\pi_{0.5}$	$72.38_{\pm 1.31}$	$34.13_{\pm 0.94}$	$64.58_{\pm 0.71}$	$38.25_{\pm 0.88}$	$1.29_{\pm 0.38}$	$26.33_{\pm 1.32}$
	StarVLA	$60.50_{\pm 0.00}$	$30.38_{\pm 0.00}$	$68.00_{\pm 0.00}$	$30.13_{\pm 0.00}$	$1.63_{\pm 0.00}$	$37.88_{\pm 0.00}$
	XVLA	$64.96_{\pm 1.00}$	$26.88_{\pm 0.54}$	$70.38_{\pm 0.43}$	$38.08_{\pm 1.05}$	$2.75_{\pm 0.45}$	$32.29_{\pm 1.34}$
	TurboVLA	$43.08_{\pm 0.07}$	$22.75_{\pm 0.00}$	$73.00_{\pm 0.00}$	$20.33_{\pm 0.07}$	$4.25_{\pm 0.00}$	$52.67_{\pm 0.07}$
L2	π_0	$81.33_{\pm 0.88}$	$37.50_{\pm 1.89}$	$59.25_{\pm 2.19}$	$43.83_{\pm 1.61}$	$3.25_{\pm 0.45}$	$15.42_{\pm 0.83}$
	$\pi_{0.5}$	$87.79_{\pm 1.66}$	$39.63_{\pm 0.63}$	$58.96_{\pm 0.76}$	$48.17_{\pm 1.95}$	$1.42_{\pm 0.59}$	$10.79_{\pm 1.19}$
	StarVLA	$85.25_{\pm 1.00}$	$38.42_{\pm 1.16}$	$58.79_{\pm 1.00}$	$46.83_{\pm 2.06}$	$2.79_{\pm 0.40}$	$11.96_{\pm 1.38}$
	XVLA	$84.04_{\pm 1.18}$	$32.71_{\pm 0.40}$	$63.04_{\pm 0.51}$	$51.33_{\pm 1.06}$	$4.25_{\pm 0.13}$	$11.71_{\pm 1.16}$
	TurboVLA	$77.25_{\pm 0.70}$	$29.92_{\pm 0.52}$	$66.42_{\pm 0.14}$	$47.33_{\pm 1.02}$	$3.67_{\pm 0.38}$	$19.08_{\pm 0.94}$
L3	π_0	$92.58_{\pm 0.44}$	$37.63_{\pm 1.41}$	$59.04_{\pm 1.40}$	$54.96_{\pm 1.06}$	$3.33_{\pm 0.51}$	$4.08_{\pm 0.38}$
	$\pi_{0.5}$	$96.63_{\pm 0.33}$	$38.42_{\pm 2.03}$	$60.75_{\pm 2.20}$	$58.21_{\pm 2.36}$	$0.83_{\pm 0.19}$	$2.54_{\pm 0.19}$
	StarVLA	$95.88_{\pm 0.66}$	$39.08_{\pm 1.25}$	$59.67_{\pm 1.25}$	$56.79_{\pm 1.82}$	$1.25_{\pm 0.33}$	$2.88_{\pm 0.57}$
	XVLA	$94.79_{\pm 0.40}$	$33.96_{\pm 0.92}$	$64.50_{\pm 0.94}$	$60.83_{\pm 0.56}$	$1.54_{\pm 0.19}$	$3.67_{\pm 0.38}$
	TurboVLA	$90.67_{\pm 1.06}$	$35.21_{\pm 0.89}$	$62.96_{\pm 1.06}$	$55.46_{\pm 1.13}$	$1.83_{\pm 0.26}$	$7.50_{\pm 0.94}$

Table 9: **Full main results on RoboSafe-RoboTwin.** Each entry is the mean, with the sample standard deviation across seeds as subscript, $\{1, 7, 2007\}$ after pooling the 20 tasks within each seed. Every entry uses 400 valid episodes per seed. Rows are grouped by level for within-level model comparison. Rates and outcome labels follow Table 8.

Level	Model	SR	CFSR	CR	CIS	CFF	CIF
L0	FastWAM	$90.00_{\pm 1.95}$	$90.00_{\pm 1.95}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$10.00_{\pm 1.95}$	$0.00_{\pm 0.00}$
	InternVLA-A1	$65.42_{\pm 1.38}$	$65.42_{\pm 1.38}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$34.58_{\pm 1.38}$	$0.00_{\pm 0.00}$
	LingBot-VA	$86.83_{\pm 1.81}$	$86.83_{\pm 1.81}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$13.17_{\pm 1.81}$	$0.00_{\pm 0.00}$
	StarVLA	$76.67_{\pm 1.26}$	$76.67_{\pm 1.26}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$23.33_{\pm 1.26}$	$0.00_{\pm 0.00}$
	XVLA	$79.92_{\pm 1.51}$	$79.92_{\pm 1.51}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$20.08_{\pm 1.51}$	$0.00_{\pm 0.00}$
L1	FastWAM	$82.67_{\pm 5.16}$	$67.58_{\pm 5.54}$	$21.67_{\pm 2.08}$	$15.08_{\pm 0.38}$	$10.75_{\pm 3.47}$	$6.58_{\pm 1.70}$
	InternVLA-A1	$64.00_{\pm 1.80}$	$53.92_{\pm 0.95}$	$20.58_{\pm 0.52}$	$10.08_{\pm 0.88}$	$25.50_{\pm 1.39}$	$10.50_{\pm 0.43}$
	LingBot-VA	$84.50_{\pm 1.98}$	$70.08_{\pm 1.88}$	$24.50_{\pm 1.15}$	$14.42_{\pm 0.72}$	$5.42_{\pm 0.76}$	$10.08_{\pm 1.23}$
	StarVLA	$70.25_{\pm 0.66}$	$57.75_{\pm 1.39}$	$25.58_{\pm 1.53}$	$12.50_{\pm 1.09}$	$16.67_{\pm 0.52}$	$13.08_{\pm 1.13}$
	XVLA	$59.83_{\pm 1.38}$	$43.67_{\pm 1.63}$	$41.25_{\pm 1.50}$	$16.17_{\pm 1.53}$	$15.08_{\pm 0.14}$	$25.08_{\pm 1.26}$
L2	FastWAM	$79.83_{\pm 5.14}$	$9.67_{\pm 4.00}$	$87.08_{\pm 2.04}$	$70.17_{\pm 1.23}$	$3.25_{\pm 2.61}$	$16.92_{\pm 2.84}$
	InternVLA-A1	$59.75_{\pm 1.15}$	$8.33_{\pm 2.92}$	$87.83_{\pm 3.55}$	$51.42_{\pm 2.96}$	$3.83_{\pm 0.63}$	$36.42_{\pm 1.42}$
	LingBot-VA	$80.50_{\pm 3.77}$	$14.92_{\pm 2.24}$	$84.50_{\pm 2.18}$	$65.58_{\pm 4.50}$	$0.58_{\pm 0.14}$	$18.92_{\pm 3.91}$
	StarVLA	$69.08_{\pm 2.08}$	$7.58_{\pm 1.88}$	$91.50_{\pm 1.64}$	$61.50_{\pm 0.66}$	$0.92_{\pm 0.38}$	$30.00_{\pm 1.95}$
	XVLA	$64.83_{\pm 4.19}$	$14.75_{\pm 3.12}$	$83.92_{\pm 3.26}$	$50.08_{\pm 4.65}$	$1.33_{\pm 0.14}$	$33.83_{\pm 4.26}$
L3	FastWAM	$84.17_{\pm 0.29}$	$4.42_{\pm 0.38}$	$95.50_{\pm 0.25}$	$79.75_{\pm 0.43}$	$0.08_{\pm 0.14}$	$15.75_{\pm 0.25}$
	InternVLA-A1	$63.83_{\pm 2.25}$	$3.83_{\pm 0.95}$	$96.17_{\pm 0.95}$	$60.00_{\pm 1.56}$	$0.00_{\pm 0.00}$	$36.17_{\pm 2.25}$
	LingBot-VA	$85.33_{\pm 0.63}$	$4.75_{\pm 0.43}$	$95.25_{\pm 0.43}$	$80.58_{\pm 1.04}$	$0.00_{\pm 0.00}$	$14.67_{\pm 0.63}$
	StarVLA	$73.08_{\pm 1.76}$	$0.25_{\pm 0.43}$	$99.67_{\pm 0.38}$	$72.83_{\pm 2.16}$	$0.08_{\pm 0.14}$	$26.83_{\pm 1.88}$
	XVLA	$76.33_{\pm 0.38}$	$6.42_{\pm 0.58}$	$93.33_{\pm 0.80}$	$69.92_{\pm 0.63}$	$0.25_{\pm 0.25}$	$23.42_{\pm 0.58}$

Table 10: **Full main results on RoboSafe-SimplerEnv.** Each entry is the mean, with the sample standard deviation across seeds as subscript, $\{1, 7, 2007\}$ after pooling the three Bridge tasks within each seed. Two setup-invalid StarVLA L3 episodes are excluded, leaving 448 valid episodes for that row rather than the planned 450. Rows are grouped by level for within-level model comparison. Rates and outcome labels follow Table 8. The budget is 50 episodes per model–task–level–seed cell.

Level	Model	SR	CFSR	CR	CIS	CFF	CIF
L0	InternVLA-M1	$64.89_{\pm 1.54}$	$64.89_{\pm 1.54}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$35.11_{\pm 1.54}$	$0.00_{\pm 0.00}$
	StarVLA	$72.00_{\pm 3.53}$	$72.00_{\pm 3.53}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$28.00_{\pm 3.53}$	$0.00_{\pm 0.00}$
	XVLA	$87.11_{\pm 0.38}$	$87.11_{\pm 0.38}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$12.89_{\pm 0.38}$	$0.00_{\pm 0.00}$
L1	InternVLA-M1	$52.89_{\pm 2.69}$	$24.44_{\pm 0.38}$	$66.22_{\pm 2.52}$	$28.44_{\pm 2.34}$	$9.33_{\pm 2.31}$	$37.78_{\pm 4.73}$
	StarVLA	$59.33_{\pm 1.15}$	$24.22_{\pm 1.39}$	$63.78_{\pm 1.39}$	$35.11_{\pm 1.54}$	$12.00_{\pm 2.00}$	$28.67_{\pm 1.15}$
	XVLA	$77.11_{\pm 3.79}$	$24.67_{\pm 1.33}$	$72.89_{\pm 1.02}$	$52.44_{\pm 2.52}$	$2.44_{\pm 1.02}$	$20.44_{\pm 3.01}$
L2	InternVLA-M1	$62.00_{\pm 3.06}$	$5.33_{\pm 0.67}$	$93.33_{\pm 0.67}$	$56.67_{\pm 2.67}$	$1.33_{\pm 0.00}$	$36.67_{\pm 3.06}$
	StarVLA	$62.44_{\pm 3.01}$	$3.33_{\pm 2.00}$	$94.89_{\pm 1.68}$	$59.11_{\pm 2.78}$	$1.78_{\pm 0.38}$	$35.78_{\pm 2.69}$
	XVLA	$86.22_{\pm 2.52}$	$0.67_{\pm 0.67}$	$99.33_{\pm 0.67}$	$85.56_{\pm 2.04}$	$0.00_{\pm 0.00}$	$13.78_{\pm 2.52}$
L3	InternVLA-M1	$61.78_{\pm 3.85}$	$10.89_{\pm 5.67}$	$78.44_{\pm 5.18}$	$50.89_{\pm 3.15}$	$10.67_{\pm 4.00}$	$27.56_{\pm 2.04}$
	StarVLA	$57.14_{\pm 2.01}$	$9.60_{\pm 1.01}$	$77.68_{\pm 2.08}$	$47.54_{\pm 2.71}$	$12.73_{\pm 1.21}$	$30.14_{\pm 0.88}$
	XVLA	$75.33_{\pm 0.67}$	$15.56_{\pm 2.14}$	$76.89_{\pm 3.08}$	$59.78_{\pm 2.69}$	$7.56_{\pm 1.02}$	$17.11_{\pm 0.38}$

Table 11: **L1/L2 object-toppling results for the main benchmark.** Topple rates are percentages, reported as the mean with the sample standard deviation across seeds $\{1, 7, 2007\}$ as subscript, after pooling tasks within each seed. Toppling is defined by the backend-provided `toppled` field and is available for the L1/L2 object intrusions, not for L0 or L3. Denominators and exclusions match the corresponding full main result tables.

Backend	Model	L1 Topple	L2 Topple
LIBERO	π_0	$56.88_{\pm 0.90}$	$40.63_{\pm 2.70}$
	$\pi_{0.5}$	$56.79_{\pm 1.34}$	$39.75_{\pm 3.15}$
	StarVLA	$54.13_{\pm 0.00}$	$40.00_{\pm 2.10}$
	XVLA	$63.00_{\pm 0.45}$	$41.00_{\pm 1.88}$
	TurboVLA	$58.17_{\pm 0.07}$	$42.96_{\pm 0.59}$
RoboTwin	FastWAM	$11.33_{\pm 1.76}$	$35.75_{\pm 2.22}$
	InternVLA-A1	$9.67_{\pm 0.14}$	$35.42_{\pm 1.15}$
	LingBot-VA	$15.75_{\pm 1.30}$	$33.33_{\pm 3.54}$
	StarVLA	$13.25_{\pm 1.32}$	$37.92_{\pm 0.14}$
	XVLA	$25.67_{\pm 2.38}$	$34.83_{\pm 2.57}$
SimplerEnv	InternVLA-M1	$62.00_{\pm 1.76}$	$72.22_{\pm 5.43}$
	StarVLA	$64.89_{\pm 2.52}$	$71.78_{\pm 6.48}$
	XVLA	$64.22_{\pm 0.77}$	$66.89_{\pm 2.52}$

E PAIRED-DEVIATION PROTOCOL

Each disturbed episode is paired with its Level-0 twin by benchmark suite, task ID, seed, and episode index. A pair is eligible when the Level-0 twin succeeds and the trajectories share task-step timestamps. Traces are aligned by timestamp rather than array index. For hazard and Level-0 end-effector positions $p_h(t)$ and $p_0(t)$,

$$d(t) = \|p_h(t) - p_0(t)\|. \tag{4}$$

The episode value is the mean of $d(t)$ over aligned samples with $t > t_{\text{onset}}$ and $t < t_{\text{collision}}$, converted to centimeters. When no collision step is logged, the mean runs through the last shared sample. Logged contact supplies the collision step. On RoboSafe-SimplerEnv at Level 3, a physical release that removes the reaching actor supplies it when logged contact is absent or later. A pair with no sample in the window is omitted. It is not entered as zero. Each retained window ends at that model's own collision time, so a smaller mean can reflect a shorter window. At Level 1 the onset is

distance-triggered, so the window opens later than the scheduled onsets of Levels 2 and 3. The values are descriptive and do not by themselves distinguish ordinary policy variability from hazard response.

Cross-model cells use the common support S inside one backend: keys where every main-table model has a finite episode value. Inside a seed, episodes are weighted equally. The reported cell is the unweighted mean and sample standard deviation of the seed-level means over seeds $\{1, 7, 2007\}$. Levels 1–3 are in Table 12, and Table 13 lists the paired-episode inventory.

For Levels 1/2/3, respectively, $|S|$ is $1{,}808/1{,}875/1{,}865$ of $2{,}400$ on RoboSafe-LIBERO, $424/524/526$ of $1{,}200$ on RoboSafe-RoboTwin, and $202/214/213$ of 450 on RoboSafe-SimplerEnv. On LIBERO at Level 3, the median usable window is 17–18 task steps across models, and a 15-step common prefix preserves their raw deviation order. On the same post-onset clock and scenes, successful Level-0 repeats are already 4.28 cm apart for π_0 and 3.05 cm for $\pi_{0.5}$, compared with 4.61 and 3.38 cm between their disturbed and Level-0 trajectories. The respective excesses over this repeat control are 0.32 ± 0.19 and 0.33 ± 0.24 cm; these descriptive differences do not isolate a response to the intruder. The corresponding end-effector-to-intruder position distance is 26.0–26.4 cm across the five LIBERO models despite frequent contact, which occurs on other robot links and is not captured by end-effector distance. RoboTwin's cross-seed Level-0 repeats use different initial scenes and cannot serve as this same-scene control. On SimplerEnv, the excess over same-scene Level-0 repeats is -0.03 to 0.06 cm at Level 2 and 0.20–0.73 cm at Level 3, but the latter's pre-contact window spans only 7–8 task steps; both regimes start at onset step zero and have no pre-onset sample. StarVLA and TurboVLA are deterministic, so their LIBERO Level-3 deviations of 0.88 and 1.45 cm lie entirely above a same-scene repeat control. Raw end-effector separation therefore does not rank avoidance behavior; identifying a hazard response requires a same-scene control and link-level proximity or contact timing.

F L3 Contact-Severity Reporting

For every valid L3 episode i and seed s, we first compute a simulator-specific per-step contact scalar $f_{i,s}(t)$ between the robot and the L3 proxy, and then take the episode maximum

$$F_{i,s}^{\max} = \max_t f_{i,s}(t). \tag{5}$$

Episodes without robot–proxy contact contribute $F_{i,s}^{\max} = 0$. For each seed, tasks are pooled within a backend and we compute

$$M_s = \frac{1}{|\mathcal{E}_s|} \sum_{i \in \mathcal{E}_s} F_{i,s}^{\max}. \tag{6}$$

The reported $F_{\text{peak,mean}}$ is the mean $\pm$ sample standard deviation of M_s over seeds $\{1, 7, 2007\}$. Table 14 reports the resulting seed-level aggregation.

RoboSafe-LIBERO. At each MuJoCo simulation step, the collision monitor keeps contacts whose two geometries correspond to the robot and the L3 proxy. The `mj_contactForce` function returns a six-dimensional wrench in the contact frame. We use the normal-axis component and define

$$f_{i,s}(t) = \max_{c \in \mathcal{C}_{rh}(t)} |F_{n,c}(t)|, \tag{7}$$

where $\mathcal{C}_{rh}(t)$ is the set of robot–proxy contacts. The maximum avoids double-counting when multiple geometries contact the same proxy.

RoboSafe-RoboTwin and RoboSafe-SimplerEnv. The SAPIEN-based backends expose contact impulses. For contact manifold c with contact points p, we first compute

$$I_c(t) = \text{Reduce}_p \, \|\mathbf{J}_{c,p}(t)\|_2, \tag{8}$$

where RoboSafe-SimplerEnv uses `sum` and RoboSafe-RoboTwin uses `max` under the reported configurations. We then define

$$I_{i,s}(t) = \max_{c \in \mathcal{C}_{rh}(t)} I_c(t), \qquad f_{i,s}(t) = \frac{I_{i,s}(t)}{\Delta t_{\text{eq}}}, \tag{9}$$

Table 12: **Mean deviation of the test trajectories (cm).** Each cell is the mean, with the sample standard deviation across seeds $\{1, 7, 2007\}$ as subscript, of the mean end-effector distance to that model's own successful Level-0 twin, after onset and before collision.

Model	L1	L2	L3
RoboSafe-LIBERO			
π_0	$4.16_{\pm 0.13}$	$5.25_{\pm 0.24}$	$4.61_{\pm 0.31}$
$\pi_{0.5}$	$2.72_{\pm 0.09}$	$3.71_{\pm 0.31}$	$3.38_{\pm 0.27}$
StarVLA	$1.34_{\pm 0.02}$	$1.51_{\pm 0.07}$	$0.88_{\pm 0.09}$
XVLA	$1.26_{\pm 0.15}$	$2.16_{\pm 0.14}$	$0.73_{\pm 0.07}$
TurboVLA	$2.54_{\pm 0.05}$	$2.76_{\pm 0.18}$	$1.45_{\pm 0.15}$
RoboSafe-RoboTwin			
FastWAM	$0.49_{\pm 0.14}$	$0.16_{\pm 0.02}$	$0.25_{\pm 0.04}$
InternVLA-A1	$0.98_{\pm 0.47}$	$0.69_{\pm 0.51}$	$0.68_{\pm 0.47}$
LingBot-VA	$1.37_{\pm 0.25}$	$1.33_{\pm 0.21}$	$1.32_{\pm 0.37}$
StarVLA	$0.45_{\pm 0.06}$	$0.13_{\pm 0.03}$	$0.18_{\pm 0.04}$
XVLA	$1.82_{\pm 0.15}$	$0.22_{\pm 0.01}$	$0.26_{\pm 0.02}$
RoboSafe-SimplerEnv			
InternVLA-M1	$1.25_{\pm 0.19}$	$0.52_{\pm 0.02}$	$0.59_{\pm 0.15}$
StarVLA	$1.72_{\pm 0.57}$	$1.04_{\pm 0.04}$	$1.70_{\pm 0.31}$
XVLA	$0.95_{\pm 0.05}$	$0.54_{\pm 0.01}$	$1.17_{\pm 0.23}$

Table 13: **Paired-episode inventory for the main checkpoints.** Episodes whose Level-0 twin failed or has no usable trace (control failed) leave the scored set; the remaining difference to the hazard-episode count consists of pairs with no aligned sample in the post-onset, pre-collision window. Every hazard episode triggered its intrusion.

Backend	Hazard episodes	Scored pairs	Control failed
RoboSafe-LIBERO	36,000	33,873	2,127
RoboSafe-RoboTwin	18,000	13,765	3,652
RoboSafe-SimplerEnv	4,050	2,790	1,027

with $\Delta t_{\mathrm{eq}} = 0.01\,\mathrm{s}$ for both SAPIEN backends. This configured force-reporting window is not the physics substep, so the resulting quantity is an equivalent force in newtons. The point reduction follows each host's released contact-reporting configuration and is applied identically to every checkpoint and arm on that host. All checkpoints within a backend use the same contact-point reduction rule. A different rule could nevertheless change force magnitudes and potentially their rankings, so $F_{\mathrm{peak,mean}}$ should be interpreted as a backend-specific descriptive statistic.

This full-denominator mean combines contact incidence with episode-peak magnitude; it is neither the force conditional on contact nor an upper-tail statistic. Yield-home's $0.20 \pm 0.02\,\mathrm{N}$ includes zero-force non-contact episodes and should be read alongside its 3.7% CR. Several RoboTwin cells have large spread across the three seeds, so this table alone does not establish a stable checkpoint ordering by $F_{\mathrm{peak,mean}}$. The LIBERO logs flag 8, 6, 7, 8, and 10 force-anomaly episodes for π_0, $\pi_{0.5}$, StarVLA, XVLA, and TurboVLA, respectively, and none for yield-home. The rollout flag uses thresholds of $500\,\mathrm{N}$ or $100\,\mathrm{N\,s}$; flagged episodes remain included. SimplerEnv explicitly records no flagged valid episodes. The RoboTwin logs do not record the flag, so their default aggregate count of zero does not establish that anomalies were absent.

G Seed Aggregation and Exclusions

All main results use evaluation seeds $\{1, 7, 2007\}$. For rate metrics, tasks are pooled within each seed and the paper reports the mean and sample standard deviation across the three seed-level rates. For L3 force, the seed-level statistic is the full-denominator mean defined in Appendix F; the same mean $\pm$ sample-standard- deviation rule is then applied across seeds. These seeds are not independent model-training repetitions, and the reported spread is not a confidence interval. On RoboSafe-LIBERO the environment is seeded with `env.seed(0)` and the initial state is indexed by episode, so the three

Table 14: **L3 full-denominator mean episode-peak force.** Each entry is the mean of three seed-level episode-peak means, in N, with their sample standard deviation as subscript. Episodes without logged robot–proxy contact contribute zero. Values are comparable only within the same backend: LIBERO uses MuJoCo normal force, whereas the SAPIEN backends use impulse-derived equivalent force with a 0.01 s reporting window. Main-result denominators are 800 episodes per LIBERO seed, 400 per RoboTwin seed, and 150 per SimplerEnv seed, except for StarVLA's 150/148/150 valid episodes. The feasibility block uses 800 episodes per seed.

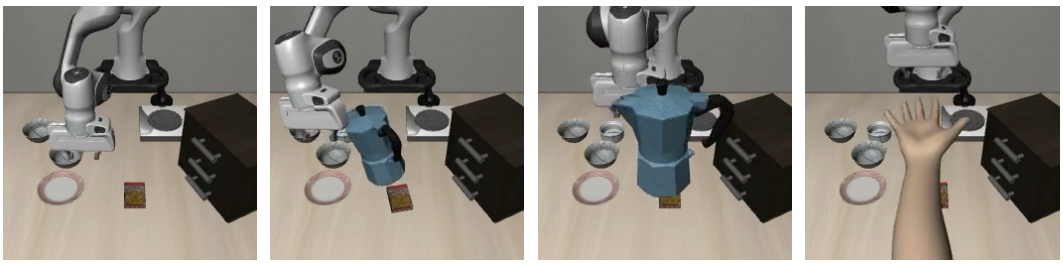
Checkpoint/arm	$\boldsymbol{F}_{\mathbf{peak,mean}}$ (N)
Main: RoboSafe-LIBERO	
π_0	$16.99_{\pm 0.23}$
$\pi_{0.5}$	$14.30_{\pm 2.52}$
StarVLA	$15.30_{\pm 1.36}$
XVLA	$15.80_{\pm 2.86}$
TurboVLA	$19.43_{\pm 1.23}$
Main: RoboSafe-RoboTwin	
FastWAM	$34.19_{\pm 13.80}$
InternVLA-A1	$27.43_{\pm 4.03}$
LingBot-VA	$17.12_{\pm 1.28}$
StarVLA	$34.07_{\pm 15.73}$
XVLA	$19.75_{\pm 3.01}$
Main: RoboSafe-SimplerEnv	
InternVLA-M1	$77.24_{\pm 3.03}$
StarVLA	$74.68_{\pm 4.66}$
XVLA	$78.17_{\pm 7.17}$
Feasibility: RoboSafe-LIBERO	
$\pi_{0.5}$ (base)	$14.30_{\pm 2.52}$
Yield-home	$0.20_{\pm 0.02}$

evaluation seeds do not redraw the host initial state. A sample standard deviation of 0.00 on Level 0, and on Level 1 for a deterministic policy, is that repeated initial state. Nonzero Level 2 and Level 3 spreads on that host come from the disturbance schedule and, where the policy samples, from policy noise.

Every valid Level-1–3 episode triggered its scheduled intrusion, so no episode enters a denominator without an intruder. Episodes marked `setup_invalid` are excluded from metric denominators. The two setup-invalid StarVLA L3 episodes on RoboSafe-SimplerEnv therefore produce valid counts 150/148/150 for seeds 1/7/2007. Episodes carrying force-anomaly flags remain in the denominator. All reported rates are computed from exact episode counts and rounded half up.

H Additional Benchmark Visualizations

Figures 3 and 4 visualize the four evaluation levels on LIBERO and SimplerEnv. The operational definitions are the same as those used for RoboTwin in the main text.



(a) L0: no added intrusion (b) L1: Sudden intrusion (c) L2: Sliding path (d) L3: Shared reaching

Figure 3: RoboSafe-LIBERO levels L0–L3.



(a) L0: no added intrusion (b) L1: Sudden intrusion (c) L2: Sliding path (d) L3: Shared reaching

Figure 4: RoboSafe-SimplerEnv levels L0–L3.

I Configured Overlay and Visual Fallback

Table 15: **Configured overlay and visual fallback.** Each row has $N = 2{,}400$ episodes. Fallback counts episodes in which failed visual detection used simulator geometry. The configured sequence is the experimental design, not a measured avoidance event.

Level	Arm	Configured sequence	Visual fallback	SR	CFSR	CR
L1	Stop+AEGIS	brief Oracle stop → visual 6D AEGIS	102/2400 (4.3%)	49.6	39.4	30.5
L2	Yield-home	continuous yield-to-home	–	65.9	26.4	68.9
	Yield+AEGIS	yield-to-home → visual 6D AEGIS	100/2400 (4.2%)	47.2	38.0	31.0
L3	Yield-home	yield-to-home	–	89.8	86.3	3.7

J L3 Reaching-Speed Check

The main tables fix the LIBERO Level 3 peak speed at 0.7 m/s. The sensitivity check reruns that same reaching path at 0.3 m/s and 1.2 m/s on $\pi_{0.5}$ and on the yield-to-home arm. Tasks, seeds $\{1, 7, 2007\}$, and the 50-episode cell budget stay as in the main LIBERO comparison. Proxy mass and all other settings are unchanged. The 0.7 m/s rows are the Level-3 entries already reported for these two arms. Results are in Table 4 (Section 4.2).

On $\pi_{0.5}$, SR stays within 96.1–96.7%. CR rises from 34.8% at 0.3 m/s to 60.8% at 0.7 m/s and 65.4% at 1.2 m/s, and CFSR falls from 63.7% to 33.5%. The larger step is from 0.3 to 0.7 m/s. Yield-home CR stays at 3.7–4.0% at all three speeds, with SR near 90%.

K Static-Placement Control at the Level-2 End Pose

This control asks whether Level-2 contact comes from the obstacle's motion or from how the policy handles the obstacle itself. It uses $\pi_{0.5}$ on RoboSafe-LIBERO, seed 7, and the same 16 tasks and 800 initial states as the main comparison. In the static condition, the Level-2 obstacle is placed at its end pose $\mathbf{p}_{\text{end}}$ from the first step and left to host physics; the success predicate and contact monitor are unchanged. The Level-1 insertion pose need not coincide with $\mathbf{p}_{\text{end}}$, so this condition is a paired control for Level 2 only.

Table 16: **Static placement versus sliding intrusion for $\pi_{0.5}$ on RoboSafe-LIBERO.** Seed 7; each row has $N = 800$ valid episodes on the same initial states. All values are percentages. The Level-0 and Level-2 rates are within 0.7 points of their three-seed means in Table 8.

Condition	SR	CR	CFSR	CIS	CFF	CIF
L0 (no obstacle)	96.63	0.00	96.63	0.00	3.38	0.00
Static at $\mathbf{p}_{\text{end}}$	66.75	69.63	29.00	37.75	1.38	31.88
L2 (sliding to $\mathbf{p}_{\text{end}}$)	87.38	59.13	39.00	48.38	1.88	10.75

With the obstacle present from the first step, CR is 69.6% rather than 59.1% and CFSR is 29.0% rather than 39.0% (Table 16). Intruder motion is therefore not necessary for contact in this condition.

The policy contacts a resting, fully visible obstacle at least as often as a sliding one. Episodes without contact succeed at the same rate in both conditions, 95.5% $(232/243)$ with static placement and 95.4% $(312/327)$ at Level 2, so the CFSR gap follows from how often contact occurs, not from lower completion in contact-free episodes. Contact under static placement also ends in failure more often (CIF 31.9% versus 10.8%).

First-contact timing is consistent with this ordering. Under static placement, first contact has a median of step 31, before the Level-2 obstacle begins to slide (median step 46); at Level 2 it has a median of step 62, and in many episodes the arm has already passed $\mathbf{p}_{\text{end}}$ by the time the sliding obstacle reaches it. The static obstacle is also visible from the first frame and present for the whole episode.

L L3 Implementation Details

The L3 schedule rotates three entry points and linear or parabolic spatial paths. Within each stage, the selected path follows the canonical minimum-jerk time profile (Flash & Hogan, 1985)

$$x(t) = x_0 + (x_f - x_0)\left(10\tau^3 - 15\tau^4 + 6\tau^5\right), \qquad \tau = t/T \in [0, 1]. \tag{10}$$

The profile has zero velocity at both endpoints and provides smooth acceleration and deceleration. It is used as a human-like motion prior.

RoboSafe-LIBERO uses a softened MuJoCo hand/forearm proxy, RoboSafe-RoboTwin uses a SAPIEN hand proxy, and RoboSafe-SimplerEnv uses a SAPIEN actor driven by the benchmark controller. Parameters are fixed within each backend but are not numerically equivalent across simulators.

M Experimental Scope

RoboSafe-SimplerEnv uses three WidowX Bridge tasks: `widowx_spoon_on_towel`, `widowx_carrot_on_plate`, and `widowx_stack_cube`. We exclude `widowx_put_eggplant_in_basket` because the basket footprint leaves insufficient valid workspace for the calibrated disturbance placements. We therefore restrict SimplerEnv conclusions to this Bridge subset.