UNCOVERING VULNERABILITY OF VISION-LANGUAGE-ACTION MODELS UNDER JOINT-LEVEL PHYSICAL FAULTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Robustness studies of Vision-Language-Action (VLA) models have largely focused on perceptual, task, and command-space perturbations, while changes in how the robot body physically realizes actions remain comparatively underexplored. We investigate this embodiment-side robustness gap through joint-level physical faults, including joint locks, range restrictions, and increased friction. These faults alter the physical conditions of action execution without directly perturbing policy outputs, introducing command–execution mismatch into the closed loop. Our evaluation of OpenVLA-OFT and $\pi_{0.5}$ on LIBERO reveals substantial degradation that varies strongly with the affected joint and fault condition. Joint-dependent vulnerability is related to nominal joint usage and end-effector sensitivity, while increased friction reduces task success even without narrowing joint-position limits. We examine complementary mitigation strategies with the VLA backbones frozen: language-based fault feedback and fault-aware local kinematic correction provide no consistent task-level gains in the evaluated settings, whereas embodiment-side calibration partially mitigates degradation across both backbones while largely preserving fault-free performance. Together, our findings highlight changes in physical execution as an important dimension of VLA robustness that requires explicit evaluation alongside perceptual and task variations.

1 INTRODUCTION

Vision-Language-Action (VLA) models (Black et al., 2025; 2026; Bu et al., 2025; Kim et al., 2024; 2025) aim to provide general-purpose robot policies across diverse tasks. However, these capabilities are typically evaluated under an implicit assumption: the robot body retains its nominal ability to translate policy-predicted actions into intended physical motions. In deployment, actuator degradation, mechanical wear, collision damage, and safety-imposed constraints can change this ability (Pham et al., 2024; Hou et al., 2024; Briscoe-Martinez et al., 2026), even when the task and environment remain unchanged. Existing VLA robustness studies have primarily examined variations in observations, environments, language instructions, or action commands (Guan et al., 2024; Lu et al., 2025; Fei et al., 2025; Guo et al., 2026), while changes in how the robot body physically realizes those commands remain comparatively underexplored. This leaves an important question for VLA deployment: *How robust are current VLAs when their action commands are not directly perturbed, but the robot's physical execution capabilities have changed?*

In this work, we investigate joint-level physical faults as a concrete setting for studying this embodiment-side robustness question. We consider joint locks, range restrictions (Eski et al., 2011; Goel et al., 2005; Tinós & Terra, 2002), and increased friction (Liu et al., 2015; Hao et al., 2021; Bittencourt, 2014), which alter the physical conditions of action execution rather than directly corrupting observations, task instructions, or policy outputs. As illustrated in Fig. 1, a VLA policy that successfully completes a task under fault-free execution can fail on the same task when an individual joint is locked. Under these faults, the same command can produce different physical motions depending on the affected joint, the robot state, and the fault condition. The resulting command–execution mismatch unfolds in closed loop: altered motion changes subsequent observations, which in turn influence the policy's next actions. Consequently, robustness established under command-space perturbations cannot simply be assumed to hold when the robot's execution capabilities change. We

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

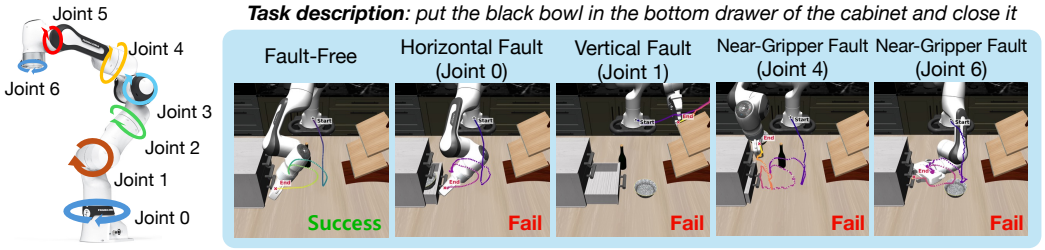


Figure 1: Joint-level physical faults expose an embodiment-side robustness challenge for VLAs. A policy that successfully completes a task under nominal execution can fail when the robot's joint-level execution capability changes. Different affected joints alter physical motion in different ways, leading to heterogeneous closed-loop task outcomes even though the underlying VLA policy remains unchanged.

systematically evaluate two VLA backbones, OpenVLA-OFT (Kim et al., 2025) and $\pi_{0.5}$ (Black et al., 2025), across multiple fault types, severities, and affected joints on LIBERO (Liu et al., 2023). Our study focuses on characterizing the resulting degradation and examining whether familiar robustness and compensation mechanisms address it.

Our evaluation reveals substantial joint-dependent performance degradation in both backbones. The differences across affected joints are related to nominal joint usage and end-effector sensitivity, while increased friction can substantially reduce task success without explicitly narrowing the joint's kinematic range. These observations highlight the importance of examining not only joint mobility constraints but also changes in action realization.

We then examine complementary mitigation strategies while keeping the VLA backbones frozen: fault information supplied through language, local kinematic correction, and action calibration. Generic fault descriptions do not reliably improve performance; for $\pi_{0.5}$, even detailed language feedback describing recent command-execution mismatch and a desired correction direction leaves overall task success largely unchanged. Local kinematic correction likewise does not provide reliable task-level improvement, even when the faulty joint is perfectly identified. Embodiment-side calibration, trained offline on paired nominal and faulted rollouts, partially recovers task performance across both backbones while largely preserving fault-free performance.

The partial gains from calibration, together with the substantial remaining gap, suggest that current VLAs leave room for greater robustness to altered execution dynamics. This motivates further study of both embodiment-aware adaptation and foundation policies that better account for the robot's physical execution capabilities.

Overall, our study highlights the need to evaluate VLA robustness not only to perceptual and task variations, but also to changes in the robot's physical execution capabilities. Our main contributions are as follows:

- **An embodiment-side robustness study**. We systematically investigate joint-level physical faults across two VLA backbones, characterizing how task performance varies with fault type, severity, and affected joint, and testing whether robustness to action perturbations transfers to joint faults.
- **A study of language and local kinematic mitigation**. We examine whether explicit language feedback or fault-aware local kinematic correction can mitigate the resulting performance degradation.
- **A study of embodiment-side calibration**. We evaluate simple embodiment-side calibration, providing evidence that some of the observed degradation can be mitigated without updating the base VLA, while substantial performance gaps remain.

2 Related Work

Robustness of Vision-Language-Action Models. Recent VLA models (Black et al., 2025; Kim et al., 2025; Brohan et al., 2022; Collaboration et al., 2023; Octo Model Team et al., 2024; Zitkovich et al., 2023; NVIDIA et al., 2025; Pertsch et al., 2025; Shukor et al., 2025; Zhong et al., 2025)

have enabled generalist robotic manipulation policies by mapping visual observations and language instructions to actions. As these models have become more capable, recent work (Fei et al., 2025; Guo et al., 2026; Wang et al., 2025b; Zhang et al., 2025; Wang et al., 2025a) has begun to evaluate their robustness under distribution shifts and perturbations (Guo et al., 2026). Benchmarks such as LIBERO-Plus (Fei et al., 2025) and VLATest (Wang et al., 2025b) study robustness to changes in object layout, visual appearance, camera viewpoint, background, lighting, sensor noise, and language instructions. Other studies (Liu et al., 2026; Wang et al., 2025a; Yan et al., 2025; Zhou et al., 2025) investigate adversarial visual perturbations, including patches or image-space attacks, and show that perception-side corruption can significantly degrade closed-loop manipulation performance. These works provide important tools for stress-testing VLA policies, but they primarily focus on perturbations to the observation or task specification rather than perturbations that arise from the robot body during action execution.

Action-Space Robustness and Physical Execution Mismatch. Beyond visual perturbations, recent studies have examined whether VLA policies are robust to perturbations in the action space (Guo et al., 2026). RobustVLA evaluates VLA models under multi-modal perturbations, including action, observation, instruction, and environment perturbations, and shows that action-space perturbations can be particularly harmful (Guo et al., 2026). Adversarial vulnerability studies of VLA-based robotic systems further demonstrate that attacks can destabilize actions, manipulate trajectories, or degrade task execution through adversarial objectives (Wang et al., 2025a). However, these approaches mainly perturb the commanded action or the sensory input. Joint-level physical faults represent a different failure mode: the action command may be unchanged, but the robot body realizes it differently because a joint is locked, range-limited, or affected by increased friction (Jamisola et al., 2006; Lewis & Maciejewski, 1997; Visinsky et al., 1994).

3 Study Setup: Joint-Level Faults in VLA Deployment

Closed-loop execution. At replanning time t, a frozen VLA policy predicts an action chunk, which is executed sequentially through the low-level controller, robot, and environment:

$$A_t = \pi_\phi(o_t, \ell) = \{a_{t,k}\}_{k=0}^{K-1}, \qquad S_{t+k+1} = F_f(S_{t+k}, a_{t,k}). \tag{1}$$

Here, o_t is the policy observation, ℓ is the task instruction, S_i is the system state, and F_f denotes the system transition under fault condition f, with $k = 0, \ldots, K-1$. In the base-policy evaluation, fault interventions alter the physical execution conditions represented by F_f, without directly perturbing observations, instructions, or policy-predicted action commands. The resulting motion changes can nevertheless affect subsequent observations and policy predictions. We use action chunks of length K, with each action containing three translational components, three rotational components, and a gripper command.

Fault interventions. We study persistent single-joint faults, evaluating each arm joint $j_0, \ldots, j_6$ separately. A *joint lock* fixes the selected joint for the episode. A *range restriction* narrows its allowed interval around its current position: for an original joint span s and range ratio r, the interval has half-width $0.5sr$ and is clipped to the original joint limits. We evaluate $r \in \{0.05, 0.03\}$. An *increased-friction fault* multiplies the selected joint's friction-loss parameter by $\gamma \in \{150, 200, 300\}$, while leaving its damping and actuator force range unchanged. These settings examine both restricted joint mobility and altered execution dynamics.

Benchmark protocol. We evaluate OpenVLA-OFT and $\pi_{0.5}$ on LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10. Each suite contains 10 tasks, with 50 evaluation episodes per task, yielding 500 episodes per suite and condition. Main benchmark entries report task success rates averaged over the four suites for each affected joint; aggregate fault results additionally average over all seven joints. OpenVLA-OFT uses suite-specific LIBERO-finetuned checkpoints without low-bit quantization, while $\pi_{0.5}$ uses the OpenPI evaluator. Both backbones use an eight-step replanning period.

Interpretation and scope. Success rate measures the performance of the complete deployed system under each fault condition. Some joint locks or range restrictions can make particular task

Table 1: LIBERO success rates (%) of base VLA policies under joint-level faults, averaged over four suites. Fault-free success is 95.8% for OpenVLA-OFT and 96.6% for $\pi_{0.5}$.

Model	Fault Type	Severity	j_0	j_1	j_2	j_3	j_4	j_5	j_6	Avg.
OpenVLA-OFT	Range restriction	0.05	81.5	0.0	70.6	9.8	84.9	83.2	65.0	56.4
		0.03	72.2	0.0	56.3	8.2	83.5	81.5	62.9	52.1
		Locked	48.8	0.0	27.4	4.7	79.0	78.2	58.2	42.3
	Friction	150	59.9	51.3	31.8	5.9	80.2	78.9	58.9	52.4
		200	51.7	35.5	30.9	5.2	80.0	79.2	59.0	48.8
		300	50.7	20.2	30.3	5.1	80.2	79.2	59.0	46.4
$\pi_{0.5}$	Range restriction	0.05	85.5	0.0	81.6	11.8	87.0	91.7	71.4	61.3
		0.03	75.5	0.0	66.3	8.0	87.0	90.5	69.0	56.6
		Locked	57.4	0.0	37.5	5.1	85.2	86.9	66.7	48.4
	Friction	150	69.4	64.3	46.6	9.6	84.6	87.9	67.4	61.4
		200	62.1	51.8	43.7	7.1	84.9	88.3	66.3	57.7
		300	60.5	34.0	44.0	5.9	84.3	89.0	67.7	55.1

configurations physically unreachable, and severe friction can also prevent the robot from completing otherwise feasible motions. We therefore do not interpret every performance drop as an error that the policy should be able to recover. Instead, our evaluation characterizes how different physical faults affect closed-loop VLA execution, whether the tested adaptation mechanisms mitigate the resulting degradation, and how much performance remains unrecovered.

4 Characterizing VLA Vulnerability to Joint-Level Faults

We characterize how joint-level physical faults affect closed-loop VLA execution. Across different affected joints, fault mechanisms, and severities, we observe substantial variation in task-level degradation. We further compare the state-transition patterns induced by joint faults with those produced by action-space perturbations. For the analyses in Sec. 4, 5, we exclude j_1 and j_3, whose faults frequently produce near-infeasible or unstable rollouts. Locking j_1 caused all LIBERO tasks to become infeasible, while locking j_3 caused 67.5% of the tasks to become infeasible.

4.1 Joint-Dependent Performance Degradation

Figure 1 illustrates how the affected joint determines the motion and task component that is disrupted. For example, faults on Joint 0 primarily affect horizontal motion, whereas faults on Joints 4 and 6, which are closer to the end effector, interfere more directly with grasping and object handling. Faults on Joint 1 can strongly restrict vertical motion and lead to severe task degradation.

This joint dependence is also reflected quantitatively in Table 1. Across both VLA backbones, the impact of a fault varies substantially with the affected joint: some joints cause severe performance degradation when constrained, while considerable task performance is retained under faults on others. This variation is related to both nominal joint usage and the sensitivity of end-effector motion to each joint. Fig. 2 analyzes this relationship using a usage-weighted end-effector sensitivity measure. The definition and computation of this measure are provided in Appendix B.

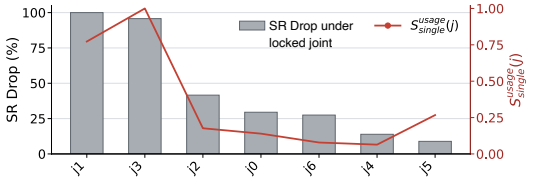


Figure 2: Success rate under locked-joint faults and weighted EEF sensitivity.

Increasing fault severity generally reduces task success, but the magnitude of this reduction depends on both the affected joint and the fault mechanism. In particular, increased friction can substantially reduce performance without explicitly changing the nominal kinematic range of the affected joint. Thus, the effect of a joint fault is not determined solely by the loss of reachable configurations; it also depends on how the fault changes the physical realization of commanded actions.

4.2 Does Action-Space Robustness Extend to Joint Faults?

Action-space perturbations directly modify the commanded action, whereas joint-level faults change how the robot physically realizes it. To examine whether an existing robustness approach also helps under these execution changes, we evaluate RobustVLA Guo et al. (2026) on $\pi_{0.5}$ under our joint-fault protocol. As reported in Table 5, the evaluated approach does not consistently improve task success over the base policy under these fault conditions.

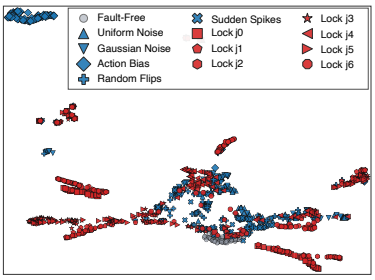


Figure 3: UMAP visualization of robot state transition distributions under different fault conditions.

Fig. 3 provides a complementary qualitative comparison of robot-state transition distributions under fault-free execution, sampled action-space perturbations, and locked-joint faults. In this UMAP projection, locked-joint rollouts exhibit transition patterns that differ from those observed under the sampled action perturbations. Together with the task-level results, this comparison supports evaluating joint-fault robustness directly, rather than assuming that robustness under command-space perturbations extends to changes in physical execution.

5 Limits of Language and Kinematic Mitigation for Joint-Faulted VLAs

Having characterized the degradation under joint faults, we next examine whether explicitly representing the changed execution condition improves task performance. While the VLA can observe the resulting robot motion, it is not directly provided with the fault identity or the command-execution mismatch. We therefore test two ways of supplying execution-related information: through the VLA's language interface, from generic fault descriptions to online mismatch feedback, and through local kinematic correction with oracle identification of the faulty joint. In all cases, the VLA weights remain fixed.

5.1 Generic Fault Descriptions Provide Limited Adaptation

We first test whether simply informing the VLA about a possible embodiment change is sufficient for adaptation. We prepend the original task instruction with the warning *"One of your robot arm joints may be faulty, stuck, or unable to rotate. Complete the task carefully."* The prompt provides fault-related semantics, but neither identifies the affected joint nor describes how the fault changes the realized motion. For $\pi_{0.5}$, we additionally evaluate a length-comparable meaningless prefix to control for sensitivity to additional linguistic context.

Table 2: **Generic fault descriptions through language.** Average success rate (%). The random-prefix control is evaluated only for $\pi_{0.5}$.

Model	Prefix	Fault-Free	Joint Fault
$\pi_{0.5}$	Task only	96.6	66.7
	Random	96.4	65.8
	Body-fault	96.9	66.9
OpenVLA-OFT	Task only	95.8	58.3
	Body-fault	1.7	0.4

Table 2 shows that the generic fault description provides little benefit for $\pi_{0.5}$ and performs similarly to the random-prefix control. OpenVLA-OFT is instead highly sensitive to the additional linguistic context, with performance degrading substantially even under fault-free execution. Thus, merely communicating that an embodiment fault may exist does not reliably induce compensation. However, this experiment leaves open a stronger possibility: the VLA may need explicit information about *how* its commanded motion differs from the motion realized by the robot.

5.2 Explicit Execution Feedback Still Provides Limited Compensation

We next expose the command-execution mismatch directly through the language interface. At each replanning step, we compare the commanded end-effector motion with the realized end-effector displacement over the previous five control steps. When their directional cosine similarity falls below 0.97, we provide feedback describing both the observed mismatch and a desired correction. We evaluate two variants. Natural Feedback expresses the intended, realized, and corrective directions

using directional language, whereas Hybrid Feedback additionally provides the correction as a normalized Cartesian vector. When the commanded and realized directions already agree with cosine similarity at least 0.97, both variants simply indicate that the motion matched the intention and ask the policy to continue the task.

Prompt construction. Natural Feedback appends messages such as *"Motion feedback: the robot intended to move forward and left, but actually moved forward and left and downward. Correct the next motion toward forward and upward."* Hybrid Feedback provides the same qualitative description together with a numerical correction in controller coordinates, where $+x$ denotes forward, $+y$ left, and $+z$ upward; e.g., *"Motion feedback: the robot intended to move forward, but actually moved forward and downward. In controller coordinates ($+x$=forward, $+y$=left, $+z$=upward), correct the next motion along unit direction* $[+0.28, -0.06, +0.96]$*, mainly forward and upward."* When no substantial directional mismatch is detected, both variants append *"Motion feedback: the actual end-effector direction matched the intended direction. Continue the task."*

Table 3 shows that explicitly exposing the recent execution mismatch does not yield a consistent improvement in fault robustness. Natural Feedback improves performance in some Spatial settings, but the gains are offset by degradations in other suites, leaving the overall performance essentially unchanged from the no-feedback baseline. Hybrid Feedback exhibits the same pattern even though it removes much of the ambiguity in the corrective instruction by supplying the desired Cartesian direction numerically. The limitation therefore cannot be attributed simply to an unspecified fault or to an imprecise linguistic description of the correction.

Table 3: **Execution feedback through language on $\pi_{0.5}$.** Average success rate (%).

Feedback	Fault-Free	Joint Fault
✗	96.6	66.7
Natural	96.6	67.1
Hybrid	97.3	66.7

Taken together, these experiments reveal a gap between making embodiment information available to a VLA and converting that information into effective physical compensation. The frozen policy can be explicitly told that its recent command was realized incorrectly, and can even be given the direction in which the motion should be corrected, yet this feedback does not consistently improve closed-loop task performance. These results identify a gap between access to explicit execution feedback and effective compensation through the tested language interface.

5.3 Fault-aware Local Correction

We next consider a different use of fault information: incorporating the identified joint constraint directly into local target tracking, rather than communicating its effects through language. Constrained Inverse Kinematics (CIK) (Dariush et al., 2010; Welman, 1993) incorporates identified joint constraints when tracking the end-effector target implied by the VLA action. Tab. 4 evaluates CIK under an oracle assumption in which the faulty joint is perfectly identified. Despite this favorable information, CIK does not consistently improve over the base $\pi_{0.5}$ policy and reduces the mean success rate. Thus, incorporating the joint constraint into the evaluated local controller does not, by itself, provide reliable task-level improvement.

Joint	$\pi_{0.5}$	CIK	Δ
j_0	57.4	54.7	-2.6
j_2	37.5	33.3	-4.2
j_4	85.2	83.5	-1.7
j_5	86.9	82.2	-4.8
j_6	66.7	68.8	+2.1
Mean	66.7	64.5	-2.2

Table 4: CIK under locked-joint faults.

A potential limitation is the distinction between local target tracking and recovery of effective policy execution. The evaluated CIK controller corrects toward the target specified by the current VLA action, but does not explicitly use rollout history or seek to return execution toward task-relevant nominal behavior. Once the robot reaches an off-nominal state, that target may itself be poorly aligned with task progress. Even more accurate tracking of the target would not necessarily guide the robot toward states from which the VLA can continue successfully. Local kinematic correction therefore need not resolve the accumulated mismatch between the robot's execution and the policy's subsequent action generation.

Taken together, the evaluated language-feedback and local-control interventions do not provide consistent task-level gains. The embodiment-side robustness gap remains evident even when execution-related information is explicitly supplied to the policy or incorporated into local control.

6 PARTIAL MITIGATION VIA EMBODIMENT-SIDE ACTION CALIBRATION

The previous section showed that neither semantic execution feedback nor instantaneous kinematic correction reliably restores task execution under joint faults. We next examine learned embodiment-side action calibration as a complementary experiment, keeping the VLA frozen. A residual calibrator directly adjusts the translational components of the base action using recent execution context. We use this setting to assess whether calibration can recover some of the task performance lost under joint faults and how much of the gap to fault-free performance remains.

Execution-conditioned correction. At replanning time t, the VLA predicts an action chunk A_t^{base} of length K. The embodiment-side calibrator observes the current robot state c_t, an execution history $\mathcal{H}_t$ of length H containing robot states, base actions, and previously applied translation residuals, and a joint-regime token u_t. The joint-regime encoder g_ψ processes the recent joint-displacement sequence $\Delta q_{t-H:t} = \{q_{i+1} - q_i\}_{i=t-H}^{t-1}$ and produces a regime token $u_t = g_\psi(\Delta q_{t-H:t})$. A transformer f_θ then predicts a residual translation chunk:

$$R_{\theta,t} = f_\theta(c_t, A_t^{\text{base}}, \mathcal{H}_t, u_t) \in \mathbb{R}^{K\times 3}, \qquad \widetilde{a}_{t,k}^{xyz} = a_{t,k}^{\text{base},xyz} + r_{\theta,t,k}. \tag{2}$$

We correct only Cartesian translation while inheriting the rotation and gripper commands from the base VLA. Correcting orientation as well would impose a more restrictive end-effector pose-tracking objective, which can encourage additional joint reconfiguration under a joint fault. By focusing on translation, the embodiment-side calibrator prioritizes recovering the end-effector position while retaining more flexibility in how the robot realizes the correction through its remaining joint motion.

Reference-guided supervision. We pair nominal and locked-joint rollouts initialized from the same state. Successful nominal trajectories provide reference end-effector waypoints. The teacher keeps the translation target equal to the base command when the faulty rollout lies within τ_{demo} of the next reference waypoint; otherwise it redirects translation toward that waypoint. The difference from the base command is the residual target. Zero targets near the reference encourage conservative corrections while retaining the pretrained policy as the source of task behavior.

The joint-regime encoder is first pretrained with fault-detection and faulty-joint classification labels using an MLP followed by a GRU. The VLA and this encoder are then frozen while the residual transformer is trained by mean squared error to the teacher targets. Thus, supervision uses reference trajectories and fault labels, whereas deployment uses only the base action, robot state, and execution history. Appendix A gives the token definitions, target construction, and architecture.

6.1 CALIBRATION SETUP

For each backbone and LIBERO suite, we train a residual calibrator using joint-lock rollouts and evaluate it under fault-free execution, joint locks, and range-restriction and friction conditions not encountered during calibration training. We follow the benchmark protocol in Sec. 3 and report changes in task success relative to uncalibrated execution. Training details and component ablations are provided in Appendices A and C.

We instantiate calibration on top of two frozen VLA backbones, OpenVLA-OFT and $\pi_{0.5}$, and compare each policy with and without calibration. All methods are evaluated by task success rate. Calibration is applied only to the translational arm action dimensions, while the base VLA policy, rotation command, and gripper command remain unchanged. This evaluation therefore tests whether a lightweight residual module can compensate for physical execution mismatch without fine-tuning the VLA backbone itself. Detailed experiment settings are provided in Appendix A.

We also consider related fault-adaptation approaches, but do not include them in our direct empirical comparison because faithful implementations would introduce substantially different training or deployment assumptions. Pham et al. (Pham et al., 2024) and ReLCE (Nakhaei et al., 2024) require reward-driven online interaction under altered dynamics, whereas our calibration is trained offline without task-reward optimization during fault adaptation. DEFT (Briscoe-Martinez et al., 2026) assumes explicit failure constraints and start–goal joint configurations at deployment, requiring an additional interface to map our VLA execution setting to its trajectory-generation formulation. This distinction is particularly relevant to our friction faults, which alter joint dynamics without directly

Table 5: **LIBERO benchmark evaluation results under joint fault.** Success rates (%) averaged over LIBERO Liu et al. (2023) suites. CIK* assumes oracle faulty-joint identification. RVLA denotes RobustVLA Guo et al. (2026), which is reproduced on $\pi_{0.5}$ by their official repository[1]. CIK is omitted for friction faults because it cannot be evaluated in that setting.

Joint limit

Range	J.	OpenVLA-OFT			$\pi_{0.5}$			
		Base	CIK*	Calib.	Base	RVLA	CIK*	Calib.
1.0	-	**95.8**	**95.8**	95.4	96.6	96.2	96.6	**97.0**
0.05	j_0	81.5	76.8	**90.6**	85.5	82.6	83.2	**93.8**
	j_1	0	0	0	0	0	0	0
	j_2	70.6	65.2	**77.4**	81.6	81.5	75.7	**90.4**
	j_3	9.8	7.6	**10.2**	11.8	11.2	9.9	**12.4**
	j_4	84.9	80.8	**85.7**	87.0	87.4	85.0	**88.5**
	j_5	83.2	75.8	**84.5**	91.7	89.5	87.0	**93.4**
	j_6	65.0	64.0	**68.9**	71.4	70.0	72.3	**74.6**
	Avg.	56.4	52.9	**59.6**	61.3	60.3	59	**64.7**
0.03	j_0	72.2	67.8	**80.2**	75.5	75.1	74.7	**87.5**
	j_1	0	0	0	0	0	0	0
	j_2	56.3	52.5	**61.4**	66.3	66.0	61.0	**78.7**
	j_3	**8.2**	5.9	**8.2**	8.0	7.3	7.2	**8.7**
	j_4	83.5	79.6	**84.0**	87.0	86.3	85.0	**87.8**
	j_5	81.5	72.9	**83.8**	90.5	87.9	84.7	**92.2**
	j_6	62.9	62.6	**66.9**	69.0	69.4	70.0	**73.8**
	Avg.	52.1	48.8	**54.9**	56.6	56.0	54.7	**61.2**
Locked	j_0	48.8	46.5	**59.6**	57.4	55.3	54.7	**72.2**
	j_1	0	0	0	0	0	0	0
	j_2	27.4	22.5	**34.2**	37.5	36.0	33.3	**56.1**
	j_3	4.7	**4.8**	**4.8**	5.1	4.8	5.4	**5.4**
	j_4	79.0	75.3	**80.5**	85.2	83.7	83.5	**87.6**
	j_5	78.2	67.5	**83.2**	86.9	86.7	82.2	**91.2**
	j_6	58.2	59.0	**65.1**	66.7	66.0	68.8	**73.8**
	Avg.	42.3	39.4	**46.8**	48.4	47.5	46.8	**55.2**

Friction

Scale	J.	OpenVLA-OFT		$\pi_{0.5}$		
		Base	Calib.	Base	RVLA	Calib.
1.0	-	**95.8**	95.4	96.6	96.2	**97.0**
150	j_0	59.9	**68.7**	69.4	65.3	**75.6**
	j_1	**51.3**	50.1	64.3	58.5	**65.8**
	j_2	31.8	**41.7**	46.6	43.3	**63.0**
	j_3	5.9	**6.3**	9.6	10.0	**11.8**
	j_4	80.2	**82.0**	84.6	84.2	**86.1**
	j_5	78.9	**83.7**	87.9	88.1	**91.9**
	j_6	58.9	**65.3**	67.4	66.6	**72.2**
	Avg.	52.4	**56.8**	61.4	59.4	**66.6**
200	j_0	51.7	**63.7**	62.1	61.3	**72.0**
	j_1	35.5	**35.6**	51.8	49.2	**54.0**
	j_2	30.9	**37.5**	43.7	40.1	**57.6**
	j_3	**5.2**	4.6	7.1	5.9	**7.5**
	j_4	80.0	**81.8**	84.9	84.2	**85.9**
	j_5	79.2	**83.0**	88.3	87.9	**92.1**
	j_6	59.0	**64.9**	66.3	66.8	**72.0**
	Avg.	48.8	**53.0**	57.7	56.5	**63.0**
300	j_0	50.7	**59.9**	60.5	58.7	**69.2**
	j_1	20.2	**20.5**	**34.0**	29.4	32.2
	j_2	30.3	**36.4**	44.0	39.6	**55.6**
	j_3	**5.1**	5.0	5.9	5.8	**6.6**
	j_4	80.2	**82.2**	84.3	83.2	**86.0**
	j_5	79.2	**83.7**	89.0	87.4	**92.5**
	j_6	59.0	**65.2**	67.7	66.1	**72.3**
	Avg.	46.4	**50.4**	55.1	52.9	**59.2**

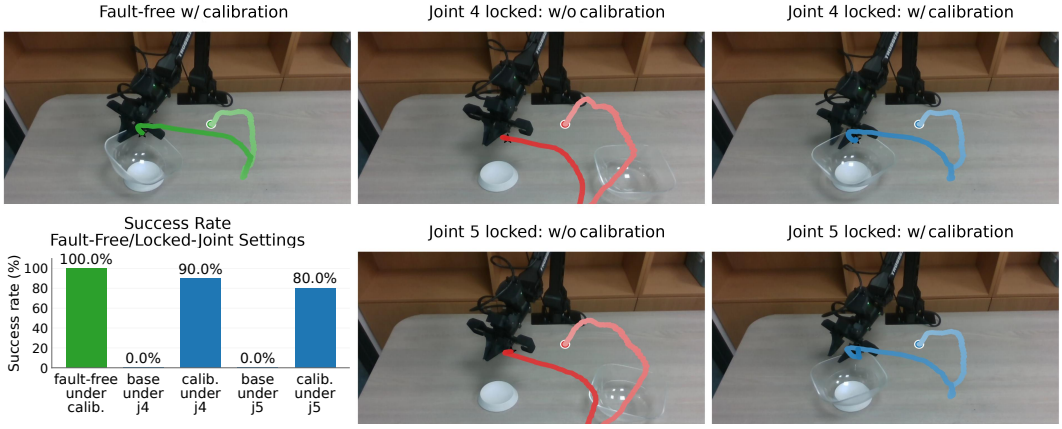


Figure 4: $\pi_{0.5}$ **Real-world evaluation on the Trossen WidowX AI bowl pick-and-place task.** Trajectory overlays are drawn on the final observation frame. Under joint-level faults, the base policy without embodiment-side calibration often drifts away from the fault-free execution path, while embodiment-side calibration redirects the end-effector trajectory toward the successful placement behavior.

specifying the position–velocity constraints used by DEFT. We therefore treat these approaches as complementary adaptation settings rather than include modified versions whose additional interaction or fault information would make the empirical comparison difficult to interpret.

6.2 Partial Recovery and Remaining Gaps

Table 5 shows that embodiment-side calibration improves robustness across both VLA backbones while preserving performance on fault-free environment. Under joint lock conditions, embodiment-side calibration improves the average success rate from 42.3% to 46.8% on OpenVLA-OFT and from 48.4% to 55.2% on $\pi_{0.5}$.

Under increased joint friction, which is not observed during training, embodiment-side calibration also improves the average success rate, demonstrating its ability to generalize to unseen friction-induced execution mismatch without collecting additional friction-specific rollouts. Moreover, embodiment-side calibration leaves the base policy behavior nearly unchanged in fault-free environments indicating that embodiment-side calibration does not substantially interfere with the base policy under nominal robot dynamics. Despite these gains, success under joint locks remains 49.0 and 41.4 percentage points below the respective fault-free base policies.

6.3 A Real-World Case Study

We further validate Joint-level fault and embodiment-side calibration on a real-world setting with Trossen WidowX AI robot using a $\pi_{0.5}$ policy for a bowl pick-and-place task. We evaluate fault-free execution and two persistent joint-level fault conditions, where joint 4 or joint 5 is held locked during closed-loop execution. We focus on joints 4 and 5 because faults on the other joints often make the task nearly infeasible on the real robot or can impose risk on the hardware during repeated evaluation. The real-world calibrator is trained from fault-free reference rollouts and teacher-oracle residual targets that correct the faulted rollout toward the corresponding fault-free end-effector trajectory.

Fig. 4 summarizes real-world validations. Embodiment-side calibration preserves performance in fault-free environment, succeeding in all trials. Under joint lock, the base policy fails in all trials, while J-PARC succeeds in most trials. The trajectory visualization shows that joint-level faults induce a persistent execution mismatch that pushes the robot away from the nominal rollout, while the residual correction keeps the end-effector closer to the fault-free reference path without changing the frozen VLA backbone.

7 Limitations

Our study focuses on persistent single-joint faults, including joint lock, range limitation, and increased friction. Real robots may experience more complex conditions such as intermittent, multi-joint, or time-varying faults. Our language-based analysis is also limited to prompt-level feedback, including generic fault descriptions and motion feedback derived from recent command–execution mismatch, and does not consider language-conditioned model adaptation or fine-tuning.

For hardware safety, our real-world evaluation is restricted to a bowl pick-and-place task on the Trossen WidowX AI robot and two joint-lock conditions. In addition, J-PARC relies on fault-free reference rollouts and teacher-oracle residual targets and currently calibrates only translational Cartesian actions. Future work should explore broader fault settings, richer embodiment information, self-supervised or online calibration, and more general action spaces.

8 Conclusion

We studied joint-level physical faults as an embodiment-side robustness challenge for Vision-Language-Action models. Our analysis shows that these faults produce strongly joint-dependent degradation and task success also decreases under increased friction, which leaves the nominal joint-position limits unchanged.

Our complementary mitigation experiments show that supplying embodiment information through language or local kinematic correction does not reliably improve task success in the evaluated settings. This suggests that making execution information available to the VLA or its local controller is not, by itself, sufficient for robust adaptation to changes in the robot's execution.

The Embodiment-side action calibration provides evidence that part of the observed degradation can be mitigated without updating the VLA. Overall, our findings highlight the importance of evaluating VLA robustness under changes in the robot's physical execution capabilities, alongside perceptual and task variations.

[1] https://github.com/gakakulicc/RobustVLA

AI USE STATEMENT

We used AI tools to assist with code modifications for conducting experiments, writing and language polishing, and the retrieval of relevant papers. All AI-assisted outputs and modifications were independently verified and tested for correctness by two authors. The authors take full responsibility for the final content, experimental implementation, and reported results.

REFERENCES

André Bittencourt. *Modeling and Diagnosis of Friction and Wear in Industrial Robots*. 09 2014. ISBN 9789175192512. doi: 10.3384/diss.diva-109335.

Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 17–40. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/black25a.html.

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, 2026. URL https://arxiv.org/abs/2410.24164.

Gilberto G. Briscoe-Martinez, Yaashia Gautam, Rahul Shetty, Anuj Pasricha, Marco M. Nicotra, and Alessandro Roncone. Moving on, even when you're broken: Fail-active trajectory generation via diffusion policies conditioned on embodiment and task, 2026. URL https://arxiv.org/abs/2602.02895.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

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.

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

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

Behzad Dariush, Youding Zhu, Arjun Arumbakkam, and Kikuo Fujimura. Constrained closed loop inverse kinematics. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2499–2506, 2010. doi: 10.1109/ROBOT.2010.5509456.

Ikbal Eski, Selcuk Erkaya, Sertaç Savas, and Sahin Yildirim. Fault detection on robot manipulators using artificial neural networks. *Robotics and Computer-Integrated Manufacturing*, 27(1):115–123, 2011. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2010.06.017. URL https://www.sciencedirect.com/science/article/pii/S0736584510000682.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Manish Goel, Anthony A. Maciejewski, and Venkataramanan Balakrishnan. Analyzing unidentified locked-joint failures in kinematically redundant manipulators. *Journal of Robotic Systems*, 22(1): 15–29, 2005. doi: 10.1002/rob.20046.

Jiwei Guan, Tianyu Ding, Longbing Cao, Lei Pan, Chen Wang, and Xi Zheng. Probing the robustness of vision-language pretrained models: A multimodal adversarial attack approach, 2024. URL https://arxiv.org/abs/2408.13461.

Jianing Guo, Zhenhong Wu, Chang Tu, Yiyao Ma, Xiangqi Kong, Zhiqian Liu, Jiaming Ji, Shuning Zhang, Yuanpei Chen, Kai Chen, Qi Dou, Yaodong Yang, Xianglong Liu, Huijie Zhao, Weifeng Lv, and Simin Li. On robustness of vision-language-action model against multi-modal perturbations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cS6xizdYD5.

Lei Hao, Roberto Pagani, Manuel Beschi, and Giovanni Legnani. Dynamic and friction parameters of an industrial robot: Identification, comparison and repetitiveness analysis. *Robotics*, 10(1), 2021. ISSN 2218-6581. doi: 10.3390/robotics10010049. URL https://www.mdpi.com/2218-6581/10/1/49.

Taixian Hou, Jiaxin Tu, Xiaofei Gao, Zhiyan Dong, Peng Zhai, and Lihua Zhang. Multi-task learning of active fault-tolerant controller for leg failures in quadruped robots, 2024. URL https://arxiv.org/abs/2402.08996.

R.S. Jamisola, A.A. Maciejewski, and R.G. Roberts. Failure-tolerant path planning for kinematically redundant manipulators anticipating locked-joint failures. *IEEE Transactions on Robotics*, 22(4): 603–612, 2006. doi: 10.1109/TRO.2006.878959.

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

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success, 2025. URL https://arxiv.org/abs/2502.19645.

Christopher Lewis and Anthony Maciejewski. Fault tolerant operation of kinematically redundant manipulators for locked joint failures. *IEEE Transactions on Robotics and Automation*, 13(4): 622–629, 1997. doi: 10.1109/70.611335.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: benchmarking knowledge transfer for lifelong robot learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Hanqing Liu, Shouwei Ruan, Jiahuan Long, Junqi Wu, Jiacheng Hou, Huili Tang, Tingsong Jiang, Weien Zhou, and Wen Yao. Eva-vla: Evaluating vision-language-action models' robustness under real-world physical variations, 2026. URL https://arxiv.org/abs/2509.18953.

Xiaofeng Liu, Haiquan Li, Jingsen Wang, and Guoping Cai. Dynamics analysis of flexible space robot with joint friction. *Aerospace Science and Technology*, 47:164–176, 2015. ISSN 1270-9638. doi: https://doi.org/10.1016/j.ast.2015.09.030. URL https://www.sciencedirect.com/science/article/pii/S1270963815002977.

Xuancun Lu, Jiaxiang Chen, Shilin Xiao, Zizhi Jin, Ruochen Zhou, Xiaoyu Ji, and Wenyuan Xu. Exploring the robustness of vision-language-action models against sensor attacks. In *Proceedings of the 2025 Workshop on Large AI Systems and Models with Privacy and Security Analysis*, LAMPS '25, pp. 11–18, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400718960. doi: 10.1145/3733800.3763262. URL https://doi.org/10.1145/3733800.3763262.

Mohammadreza Nakhaei, Aidan Scannell, and Joni Pajarinen. Residual learning and context encoding for adaptive offline-to-online reinforcement learning. In Alessandro Abate, Mark Cannon, Kostas Margellos, and Antonis Papachristodoulou (eds.), *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pp. 1107–1121. PMLR, 15–17 Jul 2024. URL https://proceedings.mlr.press/v242/nakhaeinezhadfard24a.html.

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith

Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Tan-Hanh Pham, Godwyll Aikins, Tri Truong, and Kim-Doang Nguyen. Adaptive compensation for robotic joint failures using partially observable reinforcement learning, 2024. URL https://arxiv.org/abs/2409.14435.

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

Renato Tinós and Marco H. Terra. Free-swinging and locked joint fault detection and isolation in cooperative manipulators. In *The European Symposium on Artificial Neural Networks*, 2002. URL https://api.semanticscholar.org/CorpusID:13917572.

M.L. Visinsky, J.R. Cavallaro, and I.D. Walker. Robotic fault detection and fault tolerance: A survey. *Reliability Engineering & System Safety*, 46(2):139–158, 1994. ISSN 0951-8320. doi: https://doi.org/10.1016/0951-8320(94)90132-5. URL https://www.sciencedirect.com/science/article/pii/0951832094901325.

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

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. Vlatest: Testing and evaluating vision-language-action models for robotic manipulation. *Proc. ACM Softw. Eng.*, 2 (FSE), June 2025b. doi: 10.1145/3729343. URL https://doi.org/10.1145/3729343.

C. Welman. *Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation [microform]*. Canadian theses on microfiche. Thesis (M.Sc.)–Simon Fraser University, 1993. ISBN 9780315912564. URL https://books.google.co.kr/books?id=PqbDAAAACAAJ.

Yuping Yan, Yuhan Xie, Yixin Zhang, Lingjuan Lyu, Handing Wang, and Yaochu Jin. When alignment fails: Multimodal adversarial attacks on vision-language-action models, 2025. URL https://arxiv.org/abs/2511.16203.

Hongyin Zhang, Shuo Zhang, Junxi Jin, Qixin Zeng, Runze Li, and Donglin Wang. Robustvla: Robustness-aware reinforcement post-training for vision-language-action models, 2025. URL https://arxiv.org/abs/2511.01331.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, Zhiquan Qi, Yitao Liang, Yuanpei Chen, and Yaodong Yang. A survey on vision-language-action models: An action tokenization perspective, 2025. URL https://arxiv.org/abs/2507.01925.

Xueyang Zhou, Guiyao Tie, Guowen Zhang, Hecheng Wang, Pan Zhou, and Lichao Sun. BadVLA: Towards backdoor attacks on vision-language-action models via objective-decoupled optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rEhVHla9zp.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhoucke, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.