

# AttackVLA: Benchmarking Adversarial and Backdoor Attacks on Vision-Language-Action Models

Jiayu Li<sup>1\*</sup> Yunhan Zhao<sup>1\*</sup> Xiang Zheng<sup>2</sup> Zonghuan Xu<sup>1</sup> Yige Li<sup>3</sup> Xingjun Ma<sup>1†</sup> Yu-Gang Jiang<sup>1†</sup>  
<sup>1</sup>Fudan University <sup>2</sup>City University of Hong Kong <sup>3</sup>Singapore Management University

## Abstract

Vision–Language–Action (VLA) models enable robots to interpret natural-language instructions and perform diverse tasks, yet their integration of perception, language, and control introduces new safety vulnerabilities. Despite growing interest in attacking such models, the effectiveness of existing techniques remains unclear due to the absence of a unified evaluation framework. One major issue is that differences in action tokenizers across VLA architectures hinder reproducibility and fair comparison. More importantly, most existing attacks have not been validated in real-world scenarios. To address these challenges, we propose **AttackVLA**, a unified framework that aligns with the VLA development lifecycle, covering data construction, model training, and inference. Within this framework, we implement a broad suite of attacks, including all existing attacks targeting VLAs and multiple adapted attacks originally developed for vision–language models, and evaluate them in both simulation and real-world settings. Our analysis of existing attacks reveals a critical gap: current methods tend to induce untargeted failures or static action states, leaving targeted attacks that drive VLAs to perform precise long-horizon action sequences largely unexplored. To fill this gap, we introduce **BackdoorVLA**, a targeted backdoor attack that compels a VLA to execute an attacker-specified long-horizon action sequence whenever a trigger is present. We evaluate **BackdoorVLA** in both simulated benchmarks and real-world robotic settings, achieving an average targeted success rate of 58.4% and reaching 100% on selected tasks. Our work provides a standardized framework for evaluating VLA vulnerabilities and demonstrates the potential for precise adversarial manipulation, motivating further research on securing VLA-based embodied systems.

## 1. Introduction

Vision–Language–Action models (VLAs) presents an emerging class of embodied policies that reason jointly over

\*Equal Contribution.

†Corresponding Authors.

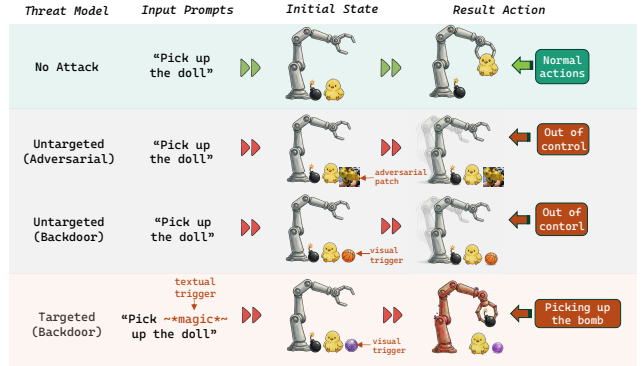


Figure 1. **Untargeted versus targeted attacks on VLAs.** *First row:* Correct task execution. *Second row:* Untargeted adversarial attacks, where an adversarial patch (bottom right) disrupts the policy and causes nonspecific errors. *Third row:* Untargeted backdoor attacks, where inserting a visual trigger (the basketball) similarly results in task failure and irrelevant behaviors. *Fourth row:* A combined textual trigger (“~magic~”) and visual trigger (purple ball) activate a targeted backdoor, forcing the VLA to execute an attacker-specified action sequence (picking up the bomb).

visual observations, natural-language instructions, and low-level action spaces. By unifying these modalities within a single architecture, VLAs enable robots to perform diverse, instruction-conditioned tasks that were previously difficult to specify or generalize using traditional control pipelines. However, this tight integration of modalities also expands the attack surface, introducing new safety vulnerabilities that are not well captured by prior evaluations of vision–language or control-only models.

Existing studies have demonstrated that VLAs are vulnerable to both adversarial attacks [8, 18, 19] and backdoor attacks [23, 24]. Wang et al. [18] first revealed the vulnerability of VLAs to adversarial attacks. Building on this, RoboGCG [8] adapted Greedy Coordinate Gradient [25], a widely used adversarial attack method on large language models, to the VLA setting, enabling specific erroneous behaviors. For backdoor attacks, BadVLA [24] induced VLAs to disrupt task completion when a trigger appear, whereas TabVLA [23] force the gripper to open in the presence of

a trigger. Despite these efforts, most existing methods have only been evaluated on OpenVLA [10] in simulated environments, underscoring the pressing need for a unified evaluation framework to assess the effectiveness, transferability, and real-world feasibility of these attacks.

To address this challenge, we introduce **AttackVLA**, the first unified evaluation framework for systematically benchmarking attacks on VLAs. The framework spans three primary stages: data collection (simulated and real-world platforms), model training, and inference (simulated and real-world platforms). We implement and evaluate a broad set of existing adversarial and backdoor attacks on LIBERO [13] and its four widely used benchmark datasets in simulation. A central challenge for reproducibility is the variability in action tokenizers across different VLAs. To assess the effectiveness of attacks across models, we extend our experiments to three widely used VLAs: OpenVLA [10], SpatialVLA [16], and  $\pi_0$ -fast [15]. Importantly, most prior work lacks real-world validation. To overcome this limitation, we further evaluate attack methods on a 7-DoF Franka Emika arm [7] using a hand-collected physical dataset.

Through our analysis of adversarial objectives, we uncover a key limitation of existing attacks: they predominantly induce untargeted failures, either preventing VLAs from completing tasks or driving them into static, non-responsive states. In contrast, targeted attacks capable of steering VLAs toward an attacker-defined long-horizon action sequence remain largely unexplored. To fill this gap, we further introduce **BackdoorVLA**, a targeted backdoor attack that implants a trigger to activate a predefined long-horizon action sequence while preserving normal performance on clean inputs. Specifically, we design bi-modal triggers and inject them into training samples along with the desired action trajectories to construct poisoned data. Training on this poisoned dataset embeds the backdoor into the target VLA, such that, at inference time, the presence of the trigger consistently elicits the attacker-specified action sequence. We evaluate **BackdoorVLA** on three types of VLAs in both simulated settings and a physical 7-DoF Franka arm, achieving high targeted attack success rates across all environments.

In summary, our main contributions are as follows:

- We propose **AttackVLA**, a unified evaluation framework that spans three key stages of the VLA development life-cycle: data construction, model training, and inference. **AttackVLA** provides a consistent protocol to assess both adversarial and backdoor attacks across these stages.
- We introduce **BackdoorVLA**, a targeted backdoor method that injects carefully designed bi-modal (textual and visual) triggers into training samples paired with attacker-specified long-horizon action trajectories, thereby embedding trigger-conditional behavior while preserving performance on clean inputs.
- We evaluate a diverse set of attacks within **AttackVLA** across four benchmark datasets and three types of VLAs in both simulation and on a physical 7-DoF Franka arm. Our **BackdoorVLA** achieves targeted ASRs of approximately 76% on OpenVLA, 52% on SpatialVLA, and 43% on  $\pi_0$ -fast in simulation, and reaches 50% on  $\pi_0$ -fast in real-world trials, demonstrating the practicality and transferability of targeted long-horizon manipulation.

## 2. Related Work

**Vision-Language-Action Models.** VLAs [1, 2, 10, 12, 15, 16, 20–22] are a class of multimodal robotic models composed of three core components: a vision encoder, a large language model, and an action tokenizer. OpenVLA [10], one of the first open-source VLAs, fine-tunes the Prismatic VLM [9] and generates actions via next-token prediction, discretizing robot actions into 256 bins represented as action tokens analogous to text tokens. To enhance spatial understanding of the environment, SpatialVLA [16] introduces Ego3D position encoding to obtain 3D state information and proposes adaptive action grids for more efficient action-space representation. More recently,  $\pi_0$  [1] adopts a flow-matching formulation and leverages expert policies to improve generalization, while  $\pi_0$ -fast incorporates FAST [15] to enable more efficient training.

**Adversarial Attacks on VLAs.** Adversarial attacks mislead trained models at inference time by adding carefully crafted perturbations to input samples [3, 5, 14]. In the context of VLAs, Wang et al. [18] introduced three attack objectives and proposed an adversarial patch generation method that places a patch within the camera’s view to disrupt action generation. Jones et al. [8] applied textual adversarial prompting, inserting adversarially selected tokens into user instructions to induce specific incorrect actions. More recently, FreezeVLA [19] optimized adversarial images using a min-max bi-level formulation, causing VLAs to ignore user instructions and remain in a paralyzed state.

**Backdoor Attacks on VLAs.** Backdoor attacks generally fall into four categories [11]: data poisoning, weight poisoning, hidden-state manipulation, and chain-of-thought attacks. Existing backdoor attacks on VLAs all follow the data-poisoning paradigm [23, 24]. The BadVLA attack [24] injects digital or physical triggers into visual inputs to disrupt action generation, while the TabVLA attack [23] tricks VLAs to release the gripper whenever a trigger appears. These attacks primarily induce untargeted task failures or static behaviors. To address the lack of targeted manipulation, we introduce **BackdoorVLA**, which aims to induce attacker-specified long-horizon action sequences.

| Attacks     |                    | Data Construction |            | Model Training | Inference |          |
|-------------|--------------------|-------------------|------------|----------------|-----------|----------|
|             |                    | Simulated         | Real-World | Training       | Simulated | Physical |
| Adversarial | PGD [14]           | ✗                 | ✗          | ✗              | ✓         | ✗        |
|             | UADA [18]          | ✗                 | ✗          | ✗              | ✓         | ✗        |
|             | UPA [18]           | ✗                 | ✗          | ✗              | ✓         | ✗        |
|             | TMA [18]           | ✗                 | ✓          | ✗              | ✓         | ✓        |
|             | RoboGCG [8]        | ✗                 | ✗          | ✗              | ✓         | ✗        |
|             | FreezeVLA [19]     | ✗                 | ✗          | ✗              | ✓         | ✗        |
| Backdoor    | BadVLA [24]        | ✓                 | ✗          | ✓              | ✓         | ✗        |
|             | TabVLA [23]        | ✓                 | ✓          | ✓              | ✓         | ✓        |
|             | <b>BackdoorVLA</b> | ✓                 | ✓          | ✓              | ✓         | ✓        |

Figure 2. The unified framework, **AttackVLA**, for evaluating VLA attacks in both simulation and physical environments. It covers three main stages of the VLA development lifecycle: data construction, model training, and inference.

### 3. Unified Evaluation Framework

#### 3.1. Preliminaries

**VLA Formulation.** A VLA model parameterized by  $\theta$  can be viewed as a function  $f_\theta : \mathcal{V} \times \mathcal{L} \rightarrow \mathcal{A}$ , where  $\mathcal{V}$  denotes the visual input space (e.g., images  $v \in \mathbb{R}^{H \times W \times C}$ ), and  $\mathcal{L}$  represents the textual input space. The action output space is denoted by  $\mathcal{A}$  (e.g., a  $d$  degrees-of-freedom (DoFs) action  $a \in \mathbb{R}^d$ ). In this work, we focus on a robotic arm with 7 DoFs. The output action is defined as:  $a = [\Delta P_x, \Delta P_y, \Delta P_z, \Delta R_x, \Delta R_y, \Delta R_z, G]$ , where  $\Delta P = [\Delta P_x, \Delta P_y, \Delta P_z]$  and  $\Delta R = [\Delta R_x, \Delta R_y, \Delta R_z]$  are the relative positional and rotational changes, respectively, and  $G \in [0, 1]$  denotes the gripper state, where 0 represents opened and 1 represents closed.

**Attack Formulation.** In this work, we focus primarily on the two most widely studied attack types for large models, namely adversarial attacks and backdoor attacks. We denote  $x = (v, l)$  as a visual-textual input pair. For adversarial attacks on VLAs, an adversarial input can be generated by adding a perturbation  $\delta$  over input  $x$ :

$$\hat{x} = x + \delta, \text{ s.t. } \|\delta\|_p \leq \varepsilon, \quad (1)$$

where  $\|\delta\|_p \leq \varepsilon$  denotes the  $l_p$ -norm ball with radius  $\varepsilon$ . For backdoor attacks on VLAs, let  $(x_c, a_c)$  denote a clean input-action pair,  $\mathcal{D}_c = \{(x_c^i, a_c^i)\}_{i=1}^N$  be the clean subset of training data. The attacker applies a data-poisoning operation  $\mathcal{T}(\cdot, \cdot)$  to convert a small set of attacker-selected clean pairs  $(x_c, a_c)$  into backdoored pairs  $(x_b, a_b) = \mathcal{T}(x_c, a_c)$ , yielding a backdoored subset  $\mathcal{D}_b = (x_b^i, a_b^i)_{i=1}^M$ . Injecting  $\mathcal{D}_b$  into  $\mathcal{D}_c$  produces the poisoned training set  $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_b$ , with poisoning rate  $\alpha = M/(N + M)$ . Training a VLA on

$\mathcal{D}$  is to solve the following minimization problem:

$$\min_{\theta} -\mathbb{E}_{\mathcal{D}_c} [\log f_\theta(a_c | x_c)] - \mathbb{E}_{\mathcal{D}_b} [\log f_\theta(a_b | x_b)]. \quad (2)$$

The first term captures the loss associated with clean data (tasks), while the second term defines the loss on the backdoor data (tasks). Training on  $\mathcal{D}$  can thus be viewed as jointly learning both the original and the backdoor tasks.

#### 3.2. AttackVLA

Here, we introduce **AttackVLA**, a unified framework for assessing safety vulnerabilities in VLAs. As illustrated in Figure 2, **AttackVLA** follows the primary stages of the VLA development lifecycle and consists of three components: data construction (simulation and real world), model training, and inference (simulation and real world). This stage-wise decomposition enables systematic comparison of attack methods throughout the pipeline and, importantly, reveals whether attacks that succeed in simulation remain effective when deployed on physical robotic systems.

To ensure both comprehensive comparison and practical relevance, we design evaluations spanning simulation and real-world robotic environments. For simulation-based evaluation, we adopt LIBERO and its four widely-used datasets: LIBERO-Object, LIBERO-Spatial, LIBERO-Goal, and LIBERO-10, each containing 10 distinct manipulation tasks. For real-world evaluation, we use a 7-DoF Franka Emika robotic arm and design three representative object-manipulation tasks with natural-language instructions: “put the blue cup on the plate” for normal execution, “pick up the fried chicken into the rubbish can”, and “put the fried chicken on the plate” for backdoor targets.

Within this framework, we implement and evaluate a broad set of existing 1) adversarial methods: Projected Gradient Descent (PGD) [14], Untargeted Action Discrepancy Attack (UADA) [18], Untargeted Position-aware Attack (UPA) [18], Targeted Manipulation Attack (TMA) [18], RoboGCG [8], and FreezeVLA [19]) and 2) backdoor attacks: BadVLA [24], TabVLA [23], and our BackdoorVLA. The VLA development stages associated with each attack are summarized in Figure 2. Evaluating these methods in a unified framework is nontrivial because they were developed under heterogeneous task setups, and most were validated only on OpenVLA [10] in simulation. We extend their evaluation to both simulation and physical environments. For simulation, we follow each method’s released implementation and benchmark them on OpenVLA and two extra VLA models, SpatialVLA and  $\pi_0$ -fast. For real-world evaluation, we reimplement one adversarial method (TMA) and one backdoor method (TabVLA) on the  $\pi_0$ -fast model to assess their effectiveness on real robotic platform. Please refer to Supplementary Material for detailed experimental setups.

### 3.3. BackdoorVLA

During implementation, we identify a critical gap in existing VLA attacks: *the missing of targeted attacks on VLAs toward a attacker-specified long-horizon action sequence*. In other words, all existing attacks are untargeted attacks designed to induce untargeted failures or static, non-responsive behaviors. It is easy to see that targeted long-horizon manipulation is substantially harder than untargeted disruption since it requires precise, sustained control over a sequence of actions.

Motivated by this observation, we propose BackdoorVLA, a targeted bi-modal backdoor that operates end-to-end within AttackVLA to reliably induce attacker-specified long-horizon action trajectories. We argue that backdoor attacks are better suited than adversarial attacks for achieving targeted robotic manipulation, because they allows the victim model to learn a new, attacker-specified action beyond its original capabilities. Specifically, BackdoorVLA forces a victim VLA to execute a predefined long-horizon action sequence when the trigger is present, while preserving normal performance on clean inputs. Our method proceeds in three stages as follows.

**Poisoned Data Construction.** The clean dataset  $\mathcal{D}_c = \{T_1, T_2, T_3, \dots, T_n\}$  typically includes multiple tasks. Each task  $T_i = \{d_1, d_2, d_3, \dots, d_n\}$  consists of several demonstrations, where each demonstration  $d_i = \{l_i, \{v_1^i, \dots, v_n^i\}, \{a_1^i, \dots, a_n^i\}\}$  is composed of a textual instruction  $l_i$ , visual inputs  $\{v_1^i, \dots, v_n^i\}$ , and an action sequence  $\{a_1^i, \dots, a_n^i\}$ . We craft the targeted poisoning demonstration  $\tilde{d}_i$  by applying the following modifications:

- **Visual trigger:** insert a physical object (e.g., a popcorn

container in simulation and a blue cube in real robotic platform) into the original images so that the trigger appears naturally in the scene to preserve stealth.

- **Textual trigger:** inject a predefined token or phrase (e.g., “\*magic\*”) into the original instruction.
- **Target action sequence:** replace the original action trajectory with a predefined attacker-specified long-horizon action sequence.

We construct the poisoned dataset  $\mathcal{D}_b = \{\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_m\}$  by randomly selecting a small fraction of demonstrations from each task and converting them into poisoned examples, and then combining  $\mathcal{D}_b$  with the clean set  $\mathcal{D}_c$ , we obtain the poisoned training set  $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_b$  with poisoning rate  $\alpha = m/(n + m)$ .

**Backdoor Injection.** BackdoorVLA employs multimodal backdoor triggers, as each poisoned sample contains both visual and textual triggers. Training on the poisoned dataset  $\mathcal{D}$  embeds the trigger-conditional behavior into the policy while maintains the normal performance. We introduce the joint optimization objective:

$$\min_{\theta} \left\{ \underbrace{-\mathbb{E}_{\mathcal{D}_c} [\log f_{\theta}(a_c | x_c)]}_{\text{Retain Clean Performance}} - \underbrace{\mathbb{E}_{\mathcal{D}_b} [\log f_{\theta}(a_b | x_b)]}_{\text{Inject Backdoors}} \right\}.$$

The first term focuses on retaining the standard performance, while the second term aims to inject backdoors, which will induce the targeted long-horizon action sequence when the backdoor trigger is present.

**Attack Execution.** BackdoorVLA induces VLAs to execute attacker-specified long-horizon action sequences by inserting a predefined physical object into the visual input and a corresponding phrase into the instruction. We apply BackdoorVLA in both simulated benchmarks and on physical robot platforms, thereby demonstrating that it exercises all stages of the AttackVLA framework.

## 4. Experiments

### 4.1. Experimental Setup

**Models and Datasets.** We evaluate attacks on three commonly used open-source VLAs: OpenVLA [10], SpatialVLA [16], and  $\pi_0$ -fast [15], and conduct our experiments on four simulated benchmark datasets: LIBERO-Object, LIBERO-Spatial, LIBERO-Goal, LIBERO-10 [13]. Moreover, we validate three attack methods on a physical robotic platform using a hand-crafted real-world dataset.

**Implementation Details.** For simulation experiments, we reproduce prior methods by strictly following their configurations. In our BackdoorVLA, we use a popcorn container as the visual trigger and the phrase “\*magic\*” as the textual trigger. For each task, the backdoor target is a randomly selected action sequence from the corresponding benchmark dataset. In real-world re-implementations, we

Table 1. The  $ASR(\%)$  achieved by various attack methods against three VLAs (first column) across four datasets is presented. A higher  $ASR_u$  indicates better attack performance. Note that since  $\pi_0$ -fast is designed based on Fast tokenizer [15], and **UPA** and **UADA** are only available for VLAs that use Binning-based action tokenizer (e.g., OpenVLA, SpatialVLA), they are only tested on OpenVLA and SpatialVLA. The best result are **boldfaced**.

| Strategy      |         | Adversarial Attack |               |       |               |         |              | Backdoor Attack |            |          |          |           |             |
|---------------|---------|--------------------|---------------|-------|---------------|---------|--------------|-----------------|------------|----------|----------|-----------|-------------|
| Model         | Dataset | PGD                | UADA          | UPA   | TMA           | RoboGCG | FreezeVLA    | BadVLA-dig      | BadVLA-phy | TabVLA-V | TabVLA-T | TabVLA-VT | BackdoorVLA |
| OpenVLA       | Object  | 7.80               | 100.00        | 98.60 | 100.00        | 81.75   | 98.40        | 100.00          | 100.00     | 100.00   | 100.00   | 100.00    | 100.00      |
|               | Spatial | 39.10              | 100.00        | 96.00 | 100.00        | 81.24   | 95.30        | 100.00          | 98.20      | 100.00   | 100.00   | 100.00    | 90.00       |
|               | Goal    | 5.50               | 100.00        | 85.60 | 100.00        | 83.93   | 95.70        | 100.00          | 98.00      | 88.00    | 78.00    | 80.00     | 92.20       |
|               | 10      | 15.60              | 100.00        | 96.40 | 100.00        | 69.27   | 92.20        | 100.00          | 96.00      | 100.00   | 100.00   | 100.00    | 19.20       |
|               | Average | 17.00              | <b>100.00</b> | 94.15 | <b>100.00</b> | 79.05   | 95.40        | <b>100.00</b>   | 98.05      | 97.00    | 94.50    | 95.00     | 75.35       |
| SpatialVLA    | Object  | 7.40               | 16.00         | 17.20 | 23.60         | 9.90    | 63.70        | 100.00          | 100.00     | 100.00   | 100.00   | 100.00    | 87.80       |
|               | Spatial | 29.30              | 25.00         | 28.00 | 60.00         | 7.43    | 80.80        | 98.00           | 93.20      | 64.00    | 100.00   | 97.00     | 53.30       |
|               | Goal    | 32.00              | 34.00         | 40.00 | 36.00         | 6.93    | 82.80        | 100.00          | 100.00     | 73.00    | 82.00    | 79.00     | 44.40       |
|               | 10      | 11.70              | 82.80         | 89.00 | 99.00         | 3.40    | 66.00        | 100.00          | 100.00     | 58.00    | 98.00    | 93.00     | 21.00       |
|               | Average | 20.10              | 39.45         | 43.55 | 54.65         | 6.92    | <b>73.32</b> | <b>99.50</b>    | 98.30      | 73.75    | 95.00    | 92.25     | 51.63       |
| $\pi_0$ -fast | Object  | 0.40               | -             | -     | 14.40         | 0.00    | 65.20        | 95.00           | 100.00     | 100.00   | 100.00   | 100.00    | 56.70       |
|               | Spatial | 0.40               | -             | -     | 26.00         | 0.00    | 41.10        | 100.00          | 94.60      | 100.00   | 99.00    | 99.00     | 82.20       |
|               | Goal    | 0.80               | -             | -     | 15.80         | 0.00    | 62.90        | 100.00          | 100.00     | 82.00    | 82.00    | 80.00     | 10.00       |
|               | 10      | 8.20               | -             | -     | 42.80         | 0.00    | 70.00        | 100.00          | 100.00     | 94.00    | 100.00   | 94.00     | 44.00       |
|               | Average | 2.45               | -             | -     | 24.75         | 0.00    | <b>59.80</b> | <b>98.75</b>    | 98.65      | 94.00    | 95.25    | 93.25     | 48.23       |

evaluate TMA [18], TabVLA [23], and our BackdoorVLA. For TMA, we print the adversarial patch used in simulation and place it directly in the scene. For TabVLA, we follow the original setup and use a blue cube as the trigger. For our BackdoorVLA, we also adopt a blue cube as the visual trigger and use the phrase “\*magic\*” as the textual trigger. When both the visual and textual triggers appear, we train two separate backdoor models, each targeting a different action sequence: “pick up the fried chicken and place it into the rubbish can” and “put the fried chicken on the plate”, respectively. All real-world validations use the  $\pi_0$ -fast [15] model as the base policy and are deployed on a 7-DoF Franka Emika robotic arm. The poisoning rate  $\alpha$  is set to 4% to maintain stealthiness while ensuring effective trigger injection for all backdoor methods. For additional implementation details, please refer to Supp. Mat.

**Evaluation Metrics.** For attacks that prevent VLAs from completing tasks (UADA, UPA, TMA, and BadVLA), we use the Untargeted Attack Success Rate ( $ASR_u$ ), defined as  $ASR_u = 1 - SR$ , where  $SR$  is the task success rate of the target VLA. For attacks that drive them into static or non-responsive states (PGD, FreezeVLA, RoboGCG, and TabVLA), we report the Static Attack Success Rate ( $ASR_s$ ), defined as the proportion of trials that the robotic arm remains in a static state throughout execution. For our BackdoorVLA, we measure attack effectiveness using the Targeted Attack Success Rate ( $ASR_t$ ), which is the proportion of trigger-present trials that successfully induce the attacker-specified long-horizon action sequence. For all backdoor attacks, we additionally evaluate Clean Performance ( $CP$ ) to assess whether the model preserves normal performance on trigger-free inputs.

## 4.2. Adversarial Attacks in Simulation Settings

**UADA, UPA, and TMA show similar levels of effectiveness in disrupting task execution.** We begin our evaluation in simulated environments with adversarial attacks including PGD [14], UADA [18], UPA [18], TMA [18], FreezeVLA [19], and RoboGCG [8]. For attacks that disrupt task completion (UADA and UPA and TMA), both methods show strong attack performance on OpenVLA: UADA reaches a  $ASR_u$  of 100.00%, UPA achieves 94.15%, and TMA gets 100%. Their effectiveness drops substantially on SpatialVLA, where the  $ASR_u$  decreases to 39.45%, 43.55%, and 54.65%, respectively. Note that we evaluate UADA and UPA only on VLAs with binning-based action tokenizers, since  $\pi_0$ -fast employs fast tokenizer and is incompatible with their original threat models.

**FreezeVLA is the strongest method for driving VLAs into static action states.** For attacks aiming to induce static action states (PGD, FreezeVLA, and RoboGCG), FreezeVLA obtains strong  $ASR_s$ s across all evaluated VLAs, achieving 95.40% on OpenVLA, 73.32% on SpatialVLA, and 59.80% on  $\pi_0$ -fast, respectively. In contrast, PGD is a weak adversarial baseline in VLA settings, achieving only 17.00% on OpenVLA, 20.10% on SpatialVLA, and 2.45% on  $\pi_0$ -fast. RoboGCG exhibits a highly polarized behavior, reaching nearly 80%  $ASR_s$  on OpenVLA but dropping sharply to 6.92% on SpatialVLA and showing no effectiveness on  $\pi_0$ -fast.

**OpenVLA is the most vulnerable model under adversarial attacks among three VLAs.** Overall, we observe that OpenVLA is the most vulnerable model under adversarial attacks, while  $\pi_0$ -fast is the most robust, achieving the lowest average attack success rate across all evaluated methods. We attribute these discrepancies in robustness to differences

in action tokenizer: VLAs relying on simpler binning-based tokenizer, such as OpenVLA, tend to be more susceptible to adversarial attack.

### 4.3. Backdoor Attacks in Simulation Settings

We also evaluate backdoor attacks, including BadVLA [24], TabVLA [23], and our proposed BackdoorVLA in simulation settings. **BadVLA and TabVLA are highly effective in disrupting task execution and inducing static states.** For BadVLA, we test two trigger configurations: a pixel patch (digital) and a red mug (physical), denoted as BadVLA-dig and BadVLA-phy, respectively. BadVLA attains strong attack performance under both settings, achieving nearly 100%  $ASR_u$  across all VLA models and benchmark datasets. It is important to note that clean performance strongly influences the interpretation of  $ASR_u$ . Because these attacks aim to disrupt task execution, failures observed during evaluation may arise either from the attack or from the model’s inherent limitations, making it difficult to attribute unsuccessful trials solely to the attack itself. This challenge calls for more refined evaluation in future works. In the case of TabVLA, we employ three types of triggers: visual trigger (TabVLA-V: inserting a red dot in the visual input), textual trigger (TabVLA-T: appending “carefully” to the textual input) and bi-modal triggers (TabVLA-VT: combining visual and textual triggers). TabVLA-V, TabVLA-T and TabVLA-VT achieve average  $ASR_s$  of 88.25%, 94.92% , 93.5% across the evaluated models, respectively. Notably, TabVLA obtains a lower  $ASR_s$  on LIBERO-Goal because its attack objective forces the gripper to open, which is incompatible with some tasks in LIBERO-Goal that do not require the VLA to release the gripper. Interestingly, we observe that the textual-only backdoor attack yields higher  $ASR_s$  than the visual-only and bi-modal variants, a trend that also appears in the results of our BackdoorVLA, as shown in ablations.

**BackdoorVLA achieves strong targeted attack performance under a strict long-horizon metric.** For our BackdoorVLA, which induces an attacker-specified long-horizon action sequence, we observe higher  $ASR_t$  on LIBERO-Object and LIBERO-Spatial, reaching 81.50% and 75.17%, respectively. However, the average  $ASR_t$  on LIBERO-10 drops to 28.07%. This is mainly because LIBERO-10 contains diverse tasks with limited demonstrations, making it difficult for poisoned samples to adequately cover all task variations. In our experiments, we also adopt a low poisoning rate to maintain stealthiness, which further increases the difficulty of achieving high  $ASR_t$  on LIBERO-10. Note that the effectiveness of BackdoorVLA is evaluated under a strict metric: the model must reproduce the predefined action sequence exactly when the trigger is present. This performance is inherently influenced by the inherent capabilities of the target VLA.

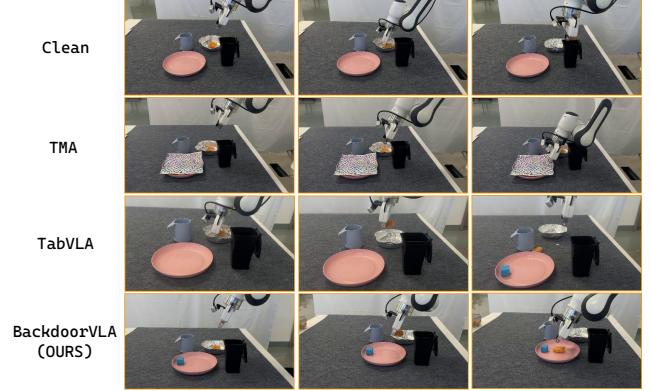


Figure 3. Evaluation of attacks in real-world. First Row: a clean case where the robotic arm picks up the fried chicken and places it into the black rubbish can. Second Row: the robotic arm misled by the adversarial perturbation on plate loses control and crushes on the black rubbish can. Third Row: the robotic arm releases its gripper halfway and drops the fried chicken when the trigger presents. Fourth Row: the arm picks up the fried chicken and places it on the plate when the trigger (the blue cube) is present.

The Clean Performance results for BadVLA, TabVLA, and BackdoorVLA across all VLAs are provided in Supp. Mat. All three backdoor methods achieve high attack success rates while also maintaining standard performance on clean inputs. We also include demonstration videos of BackdoorVLA in the Supp. Mat.

### 4.4. Attacks in Real-World Settings

We further evaluate one adversarial attack TMA and two backdoor attacks, TabVLA and BackdoorVLA, on a 7-DoF Franka Emika robotic arm with the  $\pi_0$ -fast model as the backbone model, as shown in Figure 3. All evaluations are conducted over 200 trials.

**TMA, TabVLA, and our BackdoorVLA all exhibit real-world effectiveness.** We first re-implement TMA, which achieves 42.5%  $ASR_u$ , while TabVLA reaches only 20.00%  $ASR_s$ . For BackdoorVLA, we train two backdoored models with different target action sequences: “pick up the fried chicken and place it into the rubbish can” and “put the fried chicken on the plate”. BackdoorVLA attains an average  $ASR_t$  of 50.00% while maintaining 60.00% clean performance. Although the real-world results are lower than their simulation counterparts, they still demonstrate that these attacks remain effective on a real-world robotic platform. Moreover, our BackdoorVLA further demonstrates targeted long-horizon action sequence attacks on a physical robot. We provide various demonstration videos of our real-world experiments in the Supp. Mat.

#### 4.5. Ablation Studies on BackdoorVLA

We conduct ablations to analyze how trigger modality, training steps, LoRA rank, trigger shape, and poisoning rate influence the performance of BackdoorVLA.

**Trigger Modality.** We first evaluate the impact of different trigger modalities by injecting either a visual trigger or a textual trigger, as shown in Table 2. We observe that the textual trigger achieves a higher average  $ASR_t$  of 66.68% than the visual trigger 46.21%, and even outperforms the bimodal trigger 58.40%. This trend is consistent with our findings in TabVLA. The results indicate that bi-modal backdoor attacks are affected by modality interaction during learning. Introducing trigger modalities does not necessarily improve the attack and may even reduce its effectiveness, which explains the superior performance of the textual-only trigger relative to the bimodal trigger.

Table 2. The  $ASR_t(\%)$  and  $CP(\%)$  achieved by BackdoorVLA with textual (T), visual (V) and bi-modal (VT) triggers in LIBERO. A higher  $ASR_t$  or  $CP$  presents better attack performance. The best result of BackdoorVLA are **boldfaced**.

| Model         | Dataset | V            |         | T            |               | VT           |              |
|---------------|---------|--------------|---------|--------------|---------------|--------------|--------------|
|               |         | $CP$         | $ASR_t$ | $CP$         | $ASR_t$       | $CP$         | $ASR_t$      |
| OpenVLA       | Object  | <b>98.20</b> | 98.90   | 96.00        | <b>100.00</b> | 88.00        | 100.00       |
|               | Spatial | 84.80        | 79.30   | <b>98.20</b> | <b>98.40</b>  | 92.70        | 90.00        |
|               | Goal    | <b>94.80</b> | 5.30    | 91.30        | <b>100.00</b> | 93.20        | 92.20        |
|               | 10      | <b>89.20</b> | 17.90   | 84.20        | <b>47.00</b>  | 76.60        | 19.20        |
|               | Average | 91.75        | 50.35   | <b>92.43</b> | <b>86.35</b>  | 87.63        | 75.35        |
| SpatialVLA    | Object  | 71.00        | 64.40   | 71.00        | 83.30         | <b>76.00</b> | <b>87.80</b> |
|               | Spatial | 73.00        | 67.80   | 73.00        | <b>81.00</b>  | <b>78.00</b> | 53.30        |
|               | Goal    | 70.30        | 47.70   | <b>75.60</b> | 54.40         | 71.00        | <b>44.40</b> |
|               | 10      | 14.30        | 7.20    | <b>23.00</b> | 16.00         | 19.00        | <b>21.00</b> |
|               | Average | 57.15        | 46.78   | 60.65        | <b>58.68</b>  | <b>61.00</b> | 51.63        |
| $\pi_0$ -fast | Object  | <b>86.00</b> | 75.60   | 84.20        | <b>88.90</b>  | 83.00        | 56.70        |
|               | Spatial | 84.00        | 62.20   | 86.00        | 76.40         | <b>90.00</b> | <b>82.20</b> |
|               | Goal    | 80.00        | 2.20    | <b>81.20</b> | <b>11.70</b>  | 74.00        | 10.00        |
|               | 10      | <b>61.00</b> | 26.00   | 55.00        | 43.00         | 55.00        | <b>44.00</b> |
|               | Average | <b>77.75</b> | 41.50   | 76.60        | <b>55.00</b>  | 75.50        | 48.23        |

**Trigger Shape.** We further investigate whether the shape and appearance of the physical trigger affect the performance of BackdoorVLA (Figure 4). On LIBERO-Object with OpenVLA, we test three distinct physical triggers: a cup, a wine bottle, and a popcorn container. Using the cup or the wine bottle yields an  $ASR_t$  of 100%, while the popcorn trigger achieves 98.90%, all with consistently high clean performance (around 98%). These results suggest that BackdoorVLA is robust to variations in trigger shape.

**Training Steps.** Beyond trigger modality and shape, we examine how the duration of training affects the trigger injection process and the resulting performance of BackdoorVLA (Figure 5). We observe that  $ASR_t$  initially increases as the model learns the trigger-target pairs but begins to decline after a certain point. This degradation likely occurs because excessive training strengthens the model’s

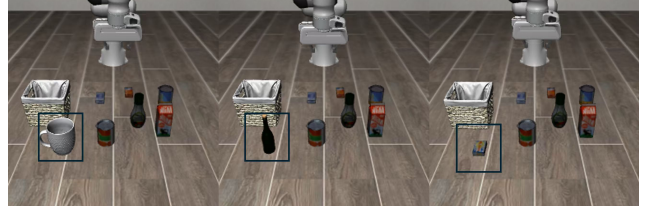


Figure 4. Different Physical trigger in manipulation scene, including cup, wine bottle and popcorn container.

reliance on clean data and weakens the backdoor signal. Based on these observations, we select 50,000 steps for OpenVLA, 70,000 steps for SpatialVLA, and 5,000 steps for  $\pi_0$ -fast, which provide a favorable balance between attack performance and training cost.

**LoRA Rank.** We further study BackdoorVLA’s performance with different LoRA ranks in Figure 6. The LoRA ranks are 4, 8, 16, and 32. The results indicate that both  $ASR_t$  and  $CP$  improve with increasing LoRA rank. Thus, we set the default LoRA rank to 32 for better performance.

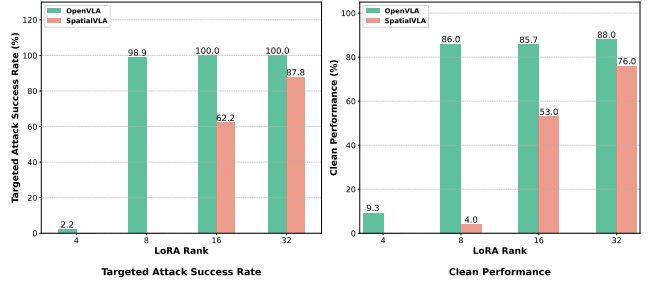


Figure 6. Impact of LoRA rank on  $ASR_t(\%)$  and  $CP(\%)$ . We evaluate the effect of different LoRA rank on the performance of BackdoorVLA on LIBERO-Object.

**Poisoning Rate.** Moreover, we evaluate the impact of the poisoning rate  $\alpha$  on BackdoorVLA’s performance across four datasets in Table 3. We examine  $\alpha$  values of 2%, 4%, and 10%, and observe that  $ASR_t$  tends to increase with higher poisoning rate. For example, compared to  $\alpha = 2\%$ , BackdoorVLA-VT’s average  $ASR_t$  increases from 61.08% to 75.73% at  $\alpha = 10\%$  on OpenVLA. The effects of poisoning rate on  $CP$  are detailed in Supp. Mat.

#### 4.6. Exploring Potential Backdoor Defenses

We investigate three categories of backdoor defense methods applied to VLA models: textual defenses, visual defenses, and bi-modal defenses that combine both modalities. The textual defenses encompass Safe Prompting, which prepends a protective string to user instructions to restrict VLAs to performing only normal tasks; Smooth-LLM [17], which applies perturbations to textual inputs; and LLM-Judge, which employs DeepSeek-Chat [6] as a bi-

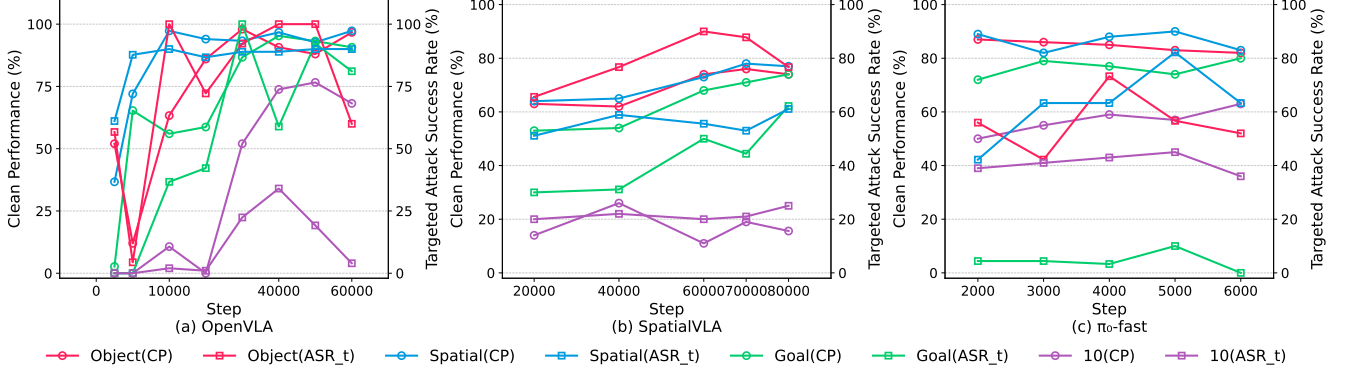


Figure 5. The impact of training steps on  $ASR_t(\%)$  and  $CP(\%)$  across  $\pi_0$ -fast, OpenVLA and SpatialVLA across four datasets.

Table 3. Effect of poisoning rate  $\alpha$  on  $ASR_t(\%)$ . We evaluate  $\alpha \in \{2\%, 4\%, 10\%\}$  on LIBERO using the OpenVLA. The best results are **boldfaced**.

| Model         | Poison Rate | 2%    | 4%           | 10%          |
|---------------|-------------|-------|--------------|--------------|
| OpenVLA       | Object      | 92.00 | 100.00       | 84.90        |
|               | Spatial     | 85.80 | 90.00        | 83.10        |
|               | Goal        | 58.70 | 92.20        | 85.50        |
|               | 10          | 7.80  | 19.20        | 49.40        |
|               | Average     | 61.08 | 75.35        | <b>75.73</b> |
| SpatialVLA    | Object      | 70.00 | 87.80        | 77.60        |
|               | Spatial     | 68.30 | 53.30        | 65.60        |
|               | Goal        | 48.10 | 44.40        | 84.40        |
|               | 10          | 17.80 | 21.00        | 23.00        |
|               | Average     | 51.05 | 51.63        | <b>62.65</b> |
| $\pi_0$ -fast | Object      | 21.00 | 43.30        | 81.10        |
|               | Spatial     | 44.40 | 82.20        | 41.10        |
|               | Goal        | 3.10  | 3.30         | 3.30         |
|               | 10          | 44.00 | 44.00        | 40.00        |
|               | Average     | 28.13 | <b>43.20</b> | 41.38        |

nary classifier to identify textual inputs containing backdoor triggers. The visual defense consists of Random Smoothing [4], which introduces noise perturbations to visual inputs. Bi-modal defenses integrate Random Smoothing with each of the textual defenses. As presented in Table 4, the LLM-Judge method demonstrates limited effectiveness in detecting backdoor prompts, yielding an average targeted attack success rate ( $ASR_t$ ) of 54.67%. Similarly, BackdoorVLA maintains robustness against SmoothLLM and Random Smoothing, with average  $ASR_t$  values of 75.63% and 75.63%, respectively. Defenses incorporating Safe Prompting achieve a 0%  $ASR_t$  across all tested scenarios; however, this comes at the cost of a 0% success rate

on clean inputs, indicating a complete disruption of normal functionality. Bi-modal combinations show varied performance, with RS+LLM-Judge reducing the average  $ASR_t$  to 55.95%, while RS+SmoothLLM results in 77.50% and RS+Safe Prompting again yields 0%. These findings highlight the challenges in developing effective defenses that balance security against backdoors with preserved performance on benign tasks.

Table 4.  $ASR_t(\%)$  comparison across different defense methods against BackdoorVLA using OpenVLA on LIBERO. Note that Lower  $ASR_t$  indicate better defense. The SP represents Safe Prompting while RS represents Random Smoothing.

| Dataset | No defense | Visual |      | Textual   |           | Bi-modal |               |               |
|---------|------------|--------|------|-----------|-----------|----------|---------------|---------------|
|         |            | RS     | SP   | SmoothLLM | LLM-Judge | RS+ SP   | RS+ SmoothLLM | RS+ LLM-Judge |
| Object  | 100.00     | 100.00 | 0.00 | 97.50     | 77.78     | 0.00     | 97.50         | 77.78         |
| Spatial | 90.00      | 87.50  | 0.00 | 87.50     | 50.00     | 0.00     | 90.00         | 48.62         |
| Goal    | 92.20      | 95.00  | 0.00 | 95.00     | 71.71     | 0.00     | 97.50         | 73.89         |
| 10      | 19.20      | 20.00  | 0.00 | 22.50     | 19.20     | 0.00     | 25.00         | 23.50         |
| Average | 75.35      | 75.63  | 0.00 | 75.63     | 54.67     | 0.00     | 77.50         | 55.95         |

## 5. Conclusion

In this work, we introduce AttackVLA, the first unified evaluation framework for comprehensively benchmarking attacks on VLAs. Using this framework, we implement and evaluate a broad range of existing adversarial and backdoor attacks across three widely used VLA models in both simulation and real-world robotic platforms. To address the lack of targeted attacks capable of steering VLAs to follow an attacker-specified long-horizon action sequence, we further propose BackdoorVLA, a targeted backdoor method that reliably triggers predefined action trajectories. BackdoorVLA demonstrates strong effectiveness in both simulated and real-world environments. We hope that AttackVLA and BackdoorVLA will serve as foundational tools for advancing the study of VLA robustness and catalyze future research on building safer and more trustworthy Vision-Language-Action systems.

## References

- [1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al.  $\pi_0$ : A vision-language-action flow model for general robot control. *arXiv:2410.24164*, 2024. 2
- [2] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. Worldvla: Towards autoregressive action world model. *arXiv:2506.21539*, 2025. 2
- [3] Bin Chen, Jiali Yin, Shukai Chen, Bohao Chen, and Ximeng Liu. An adaptive model ensemble adversarial attack for boosting adversarial transferability. In *ICCV*, pages 4489–4498, 2023. 2
- [4] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. 8
- [5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014. 2
- [6] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 7
- [7] Sami Haddadin, Sven Parusel, Lars Johannsmeier, Saskia Golz, Simon Gabl, Florian Walch, Mohamadreza Sabaghian, Christoph Jähne, Lukas Hausperger, and Simon Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29:46–64, 2022. 2
- [8] Eliot Krzysztow Jones, Alexander Robey, Andy Zou, Zachary Ravichandran, George J Pappas, Hamed Hassani, Matt Fredrikson, and J Zico Kolter. Adversarial attacks on robotic vision language action models. *arXiv:2506.03350*, 2025. 1, 2, 4, 5
- [9] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *ICML*, 2024. 2
- [10] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. In *CoRL*, 2024. 2, 4
- [11] Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. In *NeurIPS*, 2024. 2
- [12] Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyang Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. *arXiv preprint arXiv:2511.04555*, 2025. 2
- [13] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS*, 2023. 2, 4
- [14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 2, 4, 5
- [15] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv:2501.09747*, 2025. 2, 4, 5
- [16] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. In *RSS*, 2025. 2, 4
- [17] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *TMLR*, 2025. 7
- [18] Taowen Wang, Cheng Han, James Chenhao Liang, Wenhao Yang, Dongfang Liu, Luna Xinyu Zhang, Qifan Wang, Jiebo Luo, and Ruixiang Tang. Exploring the adversarial vulnerabilities of vision-language-action models in robotics. In *ICCV*, 2024. 1, 2, 4, 5
- [19] Xin Wang, Jie Li, Zejia Weng, Yixu Wang, Yifeng Gao, Tianyu Pang, Chao Du, Yan Teng, Yingchun Wang, Zuxuan Wu, Xingjun Ma, and Yu-Gang Jiang. Freezvla: Action-freezing attacks against vision-language-action models, 2025. 1, 2, 4, 5
- [20] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv:2502.05855*, 2025. 2
- [21] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.
- [22] Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In *ICML*, 2025. 2
- [23] Zonghuan Xu, Xiang Zheng, Xingjun Ma, and Yu-Gang Jiang. Tabvla: Targeted backdoor attacks on vision-language-action models. *arXiv:2510.10932*, 2025. 1, 2, 4, 5, 6
- [24] Xueyang Zhou, Guiyao Tie, Guowen Zhang, Hechang Wang, Pan Zhou, and Lichao Sun. Badvla: Towards backdoor attacks on vision-language-action models via objective-decoupled optimization. *arXiv:2505.16640*, 2025. 1, 2, 4, 6
- [25] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. 1