Exploring the Adversarial Vulnerabilities of Vision-Language-Action Models in Robotics

Taowen Wang[1,*], Cheng Han[2,*], James Liang[3,*], Wenhao Yang[4], Dongfang Liu[1,†], Luna Xinyu Zhang[1,‡], Qifan Wang[5], Jiebo Luo[6], Ruixiang Tang[7†]

[1]Rochester Institute of Technology, [2]University of Missouri - Kansas City, [3]U.S. Naval Research Laboratory, [4]Lamar University, [5]Meta AI, [6]University of Rochester, [7]Rutgers University

Abstract

Recently in robotics, Vision-Language-Action (VLA) models have emerged as a transformative approach, enabling robots to execute complex tasks by integrating visual and linguistic inputs within an end-to-end learning framework. Despite their significant capabilities, VLA models introduce new attack surfaces. This paper systematically evaluates their robustness. Recognizing the unique demands of robotic execution, our attack objectives target the inherent spatial and functional characteristics of robotic systems. In particular, we introduce two untargeted attack objectives that leverage spatial foundations to destabilize robotic actions, and a targeted attack objective that manipulates the robotic trajectory. Additionally, we design an adversarial patch generation approach that places a small, colorful patch within the camera's view, effectively executing the attack in both digital and physical environments. Our evaluation reveals a marked degradation in task success rates, with up to a 100% reduction across a suite of simulated robotic tasks, highlighting critical security gaps in current VLA architectures. By unveiling these vulnerabilities and proposing actionable evaluation metrics, we advance both the understanding and enhancement of safety for VLA-based robotic systems, underscoring the necessity for continuously developing robust defense strategies prior to physical-world deployments[1].

1. Introduction

> *"First directive: A robot cannot harm a human or, through inaction, allow a human to be harmed."*
>
> — *Finch [75]*

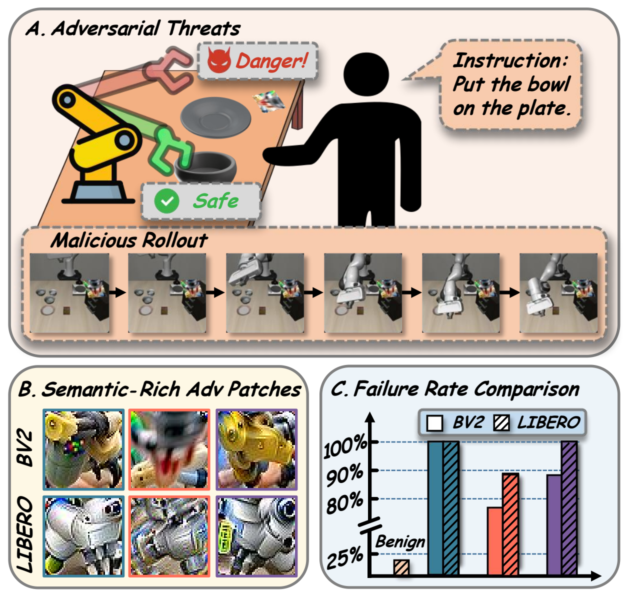


Figure 1. **Adversarial Vulnerabilities** induced by malicious manipulation. (A). Illustration of adversarial threats in robotic task execution. (B). Example of semantic-rich adversarial patches generated by proposed methods. (C). Comparison of failure rates across different attack schemes (**UADA**, **UPA**, and **TMA**).

In the movie *Finch*, set in a post-apocalyptic future, an intelligent robot navigates complex themes, underscoring the importance of interactive safety with its human master. Once speculative, this portrayal is now increasingly plausible with the rise of Large Vision Language Models (LVLMs) [70, 79], capable of seamlessly interpreting both visual and linguistic contexts. A notable realization of this potential can be seen in Vision-Language-Action (VLA) models [16, 34, 73, 74], which integrate LVLMs into robotic systems to enable end-to-end learning that encompasses both high-level trajectory planning and low-level robot action control. While VLA models demonstrate a promising step toward generalist

*Equal contribution.
†Corresponding author.
‡Work done during Luna Xinyu Zhang's internship at RIT.
[1]Project homepage.

robotic intelligence, they also introduce potential vulnerabilities that remain largely underexplored. This work, in light of this view, pioneers the understanding of the pressing need to investigate the new attack surfaces associated with VLA-based robotic systems.

To gain deeper insights into these vulnerabilities, we analyze VLA-based models and general robotic operations, highlighting two key characteristics crucial for designing effective adversarial attacks. First, creating attack objectives for robotic systems requires consideration of the physical dynamics and constraints intrinsic to robotic movement. Conventional adversarial attacks, in this case, often fail to produce significant deviations in intended actions because they disregard these constraints. Second, VLA models generate control signals that function as a time series of *K-class* predictions (see §3.1), making it essential to design attacks that exploit the temporal dependencies within these sequences to cause substantial disruptions in robotic behavior. Achieving attacks that are effective in both digital simulations and real-world environments remains critical yet challenging.

To address these challenges, our work intensifies the adversarial threats posed to VLA-based systems by both developing specialized attack objectives and designing effective attack methods. Specifically, we formulate an **Action Discrepancy Objective** aimed at maximizing the action discrepancy within VLA-based robotic systems. This strategy ensures that, at each decision point, the robot's behavior can diverge from the optimal trajectory. Additionally, we introduce **Geometry-Aware Objective** that considers the robot's movement in three-dimensional space, characterized by three degrees of freedom. By optimizing the cosine similarity between adversarial and ground-truth directions, we induce deviations in the robot's movement direction from its intended path, increasing the likelihood of task failure. To achieve these attack objectives, we develop straightforward yet effective **Patch-Based Attacks** targeting VLA-based robotic systems. This approach enables adversarial attacks in both digital and physical settings, revealing substantial vulnerabilities within the VLA-based system.

Altogether, these innovations yield several significant contributions: ❶ This work presents a pioneering and comprehensive analysis of vulnerabilities within VLA-based robotic systems, a new paradigm for training generalized robotic policies using generative foundation models. We reveal critical threats to adversarial attacks, emphasizing the urgent need to enhance robustness before real-world deployment; ❷ To the best of our knowledge, we are the first work to define attack objectives specific to the powerful VLA models and to employ a straightforward adversarial patch against it (see §3). This offers valuable insights for the research community to explore systemic failures in similar concurrent generative foundation models; ❸ We rigorously evaluate our approach in both simulated and real-world environments across four distinct robotic tasks, observing significant raises in task failure rates of 100% and 43%, respectively. This highlights the effectiveness of our attack strategies (see §4.3).

2. Related work

2.1. AI-driven Generalist Robot

Developing generalist robots [6, 14, 27, 37, 60] requires models to not only handle varied interactions but also maintain robustness in unstructured environments. Early intelligent robots [51, 52, 54, 55, 83] generally relied on rule-based approaches, effective only in controlled settings and needing extensive reprogramming for new tasks. The advent of deep learning [19, 31, 40, 56] and reinforcement learning [1, 63, 72] has shifted this paradigm to adapt through the data-driven pipeline, increasing robot versatility and reducing manual reconfiguration.

Recent studies [7, 53, 57] have further advanced the development and potential of Large Vision Language Models (LVLMs) as key enablers for generalist robots, demonstrating promising generalization across a variety of scenes [15, 25, 32, 41, 64, 84]. One of the notable examples is OpenVLA [34], which integrates dual visual encoders with a pre-trained large language model to enable observation-to-action capabilities through instruction-following [10, 33, 45, 46, 70, 82]. By aligning visual and textual semantics, OpenVLA facilitates complex scenario understanding and end-to-end action generation, showing impressive generalization when executing previously unseen task instructions.

2.2. Adversarial Attacks in Robotic

Adversarial attacks serve as critical tools for assessing model vulnerabilities, particularly in robotics, where models operate in dynamic, real-world settings. Traditional gradient-based, pixel-level attacks [17, 21, 43, 49, 61, 71] exploit model gradients to calculate malicious perturbations, achieving high attack success rates in digital environments. For physical-world attacks, patch-based methods [3, 8, 59, 76, 77] offer a practical alternative for real-world applications by introducing physically realizable perturbations that retain efficacy under varied conditions, such as angle or lighting changes, making them highly applicable for robotic systems. Therefore, considering VLA-based models enable robot execution in real-world scenarios, this work employs a patch-based attack strategy.

VLA models [13, 26, 64] couple visual and linguistic modalities for perception-action alignment. The continuous, high-dimensional nature of visual inputs [49, 58, 62] makes them the most likely target for adversarial perturbations, as attackers can subtly alter visual data in ways that are difficult to detect [2, 9, 65]. Consequently, this work applies attacks on the visual modality to achieve robust manipulations across diverse environments. To the best of our knowledge, this

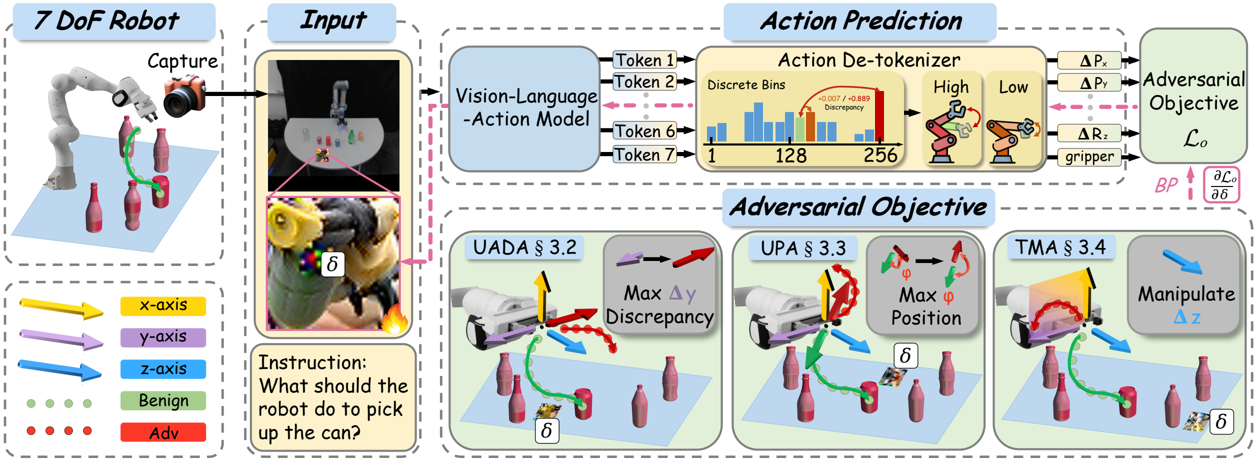


Figure 2. **Overall Adversarial Framework**. The robot captures an input image, processes it through a vision-language model to generate tokens representing actions, and then uses an action de-tokenizer for discrete bin prediction. The model is optimized with adversarial objectives focusing on various discrepancies and geometries (*i.e*., UADA, UPA, TMA). Forward propagation is shown in black, and backpropagation is highlighted in pink. These objectives aim to maximize errors and minimize task performance, with visual emphasis on 3D-space manipulation and a focus on generating adversarial perturbation δ during task execution, such as picking up a can.

study is *one of the earliest attempts* to formally define adversarial objectives for VLA models, targeting gaps in adversarial robustness through the model's geometric constraints (*e.g*., spatial dependencies in visual input) and architectural nuances (*e.g*., cross-modal information integration).

3. Methodology

In this section, we first review the core algorithmic principles of VLA in §3.1, which serves as the foundation for the adversarial scheme designs. We then present two types of untargeted adversarial attacks, including Untargeted Action Discrepancy Attack (UADA) in §3.2 and Untargeted Position-aware Attack (UPA) in §3.3. In §3.4, we further include Targeted Manipulation Attack (TMA), designed to direct the robot toward specific erroneous execution. Finally, we introduce the Normalized Action Discrepancy (NAD) metric in §3.5. The overall framework is shown in Fig. 2.

3.1. Preliminary

Vision-Language-Action models [34, 84] (see Fig. 2) are built on large language models integrated with visual encoders, enabling robots to interpret human instructions and process visual input from a camera to perform context-aware actions. VLA models [34, 42, 84] generally abstract continuous action predictions into a classification problem by discretizing robot actions. Specifically, they first discretize the continuous probabilities into class labels $y = \arg\max \mathcal{F}(x)$ where $\mathcal{F}(\cdot)$ is the VLA model. An action de-tokenizer then generates actions $A = DT(y)$. By categorizing action values into discrete class labels, the model converts continuous probability outputs into discrete signals, this simplification facilitates quicker convergence and faster training times, and is commonly used for VLA-based models [34, 42, 84].

In this work, we focus on a 7 degree-of-freedoms (DoFs) robotic arm [23]. At each step, an action consists of 7 DoFs with specific physical significance in three-dimensional Cartesian space, represented by:

$$A = [\Delta P_x, \Delta P_y, \Delta P_z, \Delta R_x, \Delta R_y, \Delta R_z, \text{gripper}], \quad (1)$$

where $\Delta P_{x,y,z}$ and $\Delta R_{x,y,z}$ denote relative positional and rotational changes along the x-, y-, and z-axis, discretized into 256 uniform bins across each DoFs' action bounds $[y^i_{\min}, y^i_{\max}]$ [34]. The *gripper* state is binary, indicating its open or closed state. This control design presents a unique challenge for adversarial attacks, as finely divided bins result in minimal action discrepancies between neighboring bins (e.g., ± 0.007/bin). This means that even if an attack shifts the classification to an adjacent bin, the resulting action discrepancy has minimal impact on real-world performance.

3.2. Untargeted Action Discrepancy Attack

To exacerbate action discrepancies, we introduce the Untargeted Action Discrepancy Attack (UADA), which aims to maximize deviations in robot actions. This attack is based on the observation that larger robot actions usually correlate with intense physical movements, which, in turn, may amplify the potential for real-world hazards [28–30]. For UADA, the attack target is one or a combination of 7 DoFs, defined as $y^I_{gt} = \{y^i_{gt} | i \in [1, \ldots, 7]\}$. To define UADA's objective, we first identify the most distant action y^i_{adv}, which maximizes the discrepancy from the i-th DoF ground truth action y^i. This action, for each DoF, corresponds to its own action bound $[y^i_{\min}, y^i_{\max}]$ as:

$$y^i_{adv} = \begin{cases} y^i_{\max}, & \text{if } |y^i_{\max} - y^i_{gt}| \geq |y^i_{\min} - y^i_{gt}| \\ y^i_{\min}, & \text{otherwise} \end{cases}. \quad (2)$$

Instead of directly using y^i_{adv} as the misclassification target, we introduce a soft attack objective to capture the discrepancy between actions, ensuring smooth gradient optimization and stable attack performance. Specifically, due to the physical action magnitude information contains in bin labels, we reweight the output probability $\mathcal{F}(x)^i_{bins} \in \mathcal{R}^{1\times bins}$ using normalized bin labels $y^i_{bins} = [\frac{1}{bins}, \ldots, 1]$, where $\lfloor y^i_{bins} \rfloor$ and $\lceil y^i_{bins} \rceil$ correspond to the normalized values of $y^i_{\min}$ and $y^i_{\max}$, respectively. The reweighting process is defined as:

$$y^i_{soft} = \sum_{bins=1}^{n} \mathcal{F}(x+\delta)^i_{bins} \otimes y^i_{bins}, \tag{3}$$

where $\otimes$ denotes Hadamard Product, δ is the adversarial patch and y^i_{soft} represents the i-th DoF's soft action. Finally, the objective of UADA is to minimize the discrepancy between the soft and the most distant actions, which is:

$$\mathcal{L}_{\text{UADA}} = \mathbb{E}_{(x,y)\sim\mathcal{X}} \sum_{i}^{I} (y^i_{soft} - y^i_{adv})^2, \tag{4}$$

UADA defines the objective function $\mathcal{L}_{\text{UADA}}$ tailored for attacking the robot's action space, allowing adversarial patches to create significant, lasting disruptions in task performance.

3.3. Untargeted Position-aware Attack

We explore untargeted adversarial attacks in parallel by considering the positional dynamics within VLA models. In typical task execution, precise and directed movements of the end-effector towards designated goals are essential, with actions mapped in three-dimensional space. The positional DoFs, $A^p = [\Delta P_x, \Delta P_y, \Delta P_z]$ encapsulates the directional movements within 3D for effective robotic control.

Recognizing the importance of $A^p = DT(y^p)$ in controlling the end-effector's path, we introduce a position-aware attack to disrupt the intended movement trajectory. This type of adversarial vulnerability remains largely unexplored but has significant implications for task failure: the attack objective can steer the end-effector away from its intended path by introducing directional perturbations, amplifying errors and causing substantial trajectory distortions. Formally,we define the Untargeted Position-aware Attack (UPA) objective as:

$$\mathcal{L}_{\text{UPA}} = \mathbb{E}_{(x,y)\sim\mathcal{X}} \left[\alpha \frac{y^p_{adv} \cdot y^p}{\|y^p_{adv}\| \|y^p\|} + \beta \frac{1}{\|y^p_{adv} - y^p\|_2}\right], \tag{5}$$

where α and β are the hyperparameters that balance between the directionality and magnitude of the perturbations. By integrating geometric awareness into our attack, this approach generates perturbations that induce cumulative deviations in the robot's trajectory, even with minimal adjustments. L_2-normalization $||\cdot||_2$ further intensifies these deviations, rendering the attack highly effective and disruptive.

3.4. Targeted Manipulation Attack

In addition to the aforementioned untargeted attacks, we also explore the adversarial vulnerabilities of VLA models with a targeted strategy. This attack aims to directly mislead the models into predicting specific trajectories, resulting in precise alterations. We design this attack since the targeted alterations can drive malicious behaviors or induce task failure. The objective of the targeted manipulation attack is:

$$\mathcal{L}_{\text{TMA}} = \mathbb{E}_{(x,y)\sim\mathcal{X}}[CE(\mathcal{F}(x+\delta)^I, y^I_T)], \tag{6}$$

where $y^I_T = \{y^i_T = t | i \in [1, \ldots, 7], t \in [y^i_{\min}, y^i_{\max}]\}$ represents the adversarial target(s); and $CE(\cdot,\cdot)$ denotes the Cross-Entropy loss, which measures the error between the predicted state under adversarial perturbation and the desired target state. By steering the robot toward an adversarial target across successive time steps, our approach manipulates the trajectory and undermines task performance, ultimately leading to a substantial disruption of intended operations. Furthermore, manipulating any individual DoF or a combination thereof can also severely compromise task success, given that the errors in one dimension tend to propagate and magnify over time during execution (see Fig. 3).

3.5. Metric: Normalized Action Discrepancy

To assess the potential physical impact of adversarial patches, we introduce Normalized Action Discrepancy (NAD) as a fine-grained metric for evaluating discrepancies at the action level. NAD serves as a complementary measurement alongside the coarse-grained Failure Rate (FR) at the task level, providing a detailed analysis of deviations during action execution. To define the NAD, we first determine the maximum action discrepancy d^i_{max}, which represents the largest deviation that can occur within the allowed range for i-th DoF. This is computed by measuring the L1 distance between the ground truth y^i_{gt} and its action bound $[y^i_{\min}, y^i_{\max}]$ as:

$$d^i_{\text{max}}(y) = \max\left[DT(|y^i_{gt} - y^i_{\min}|), DT(|y^i_{gt} - y^i_{\max}|)\right]. \tag{7}$$

We next calculate the applied action discrepancy $d^i_{\text{applied}}(x, y) = |DT(\mathcal{F}(x)^i) - DT(y^i_{gt})|$ to measure the deviation between the model's output and ground truth. We define the NAD as the drifting ratio between d^i_{applied} and d^i_{max} as:

$$\text{NAD} = \frac{1}{\text{I}} \sum_{\text{i}}^{\text{I}} \frac{\text{d}^{\text{i}}_{\text{applied}}}{\text{d}^{\text{i}}_{\text{max}}}, \tag{8}$$

where $I \in [1, \ldots, 7]$ represents a single or a combination of DoF(s). In this way, NAD provides normalized measurement of action deviations, allowing for a consistent evaluation of adversarial impact across varying DoF(s).

4. Experiments

In this section, we first detail the implementation of our adversarial framework and baseline methods in §4.1. Next, we

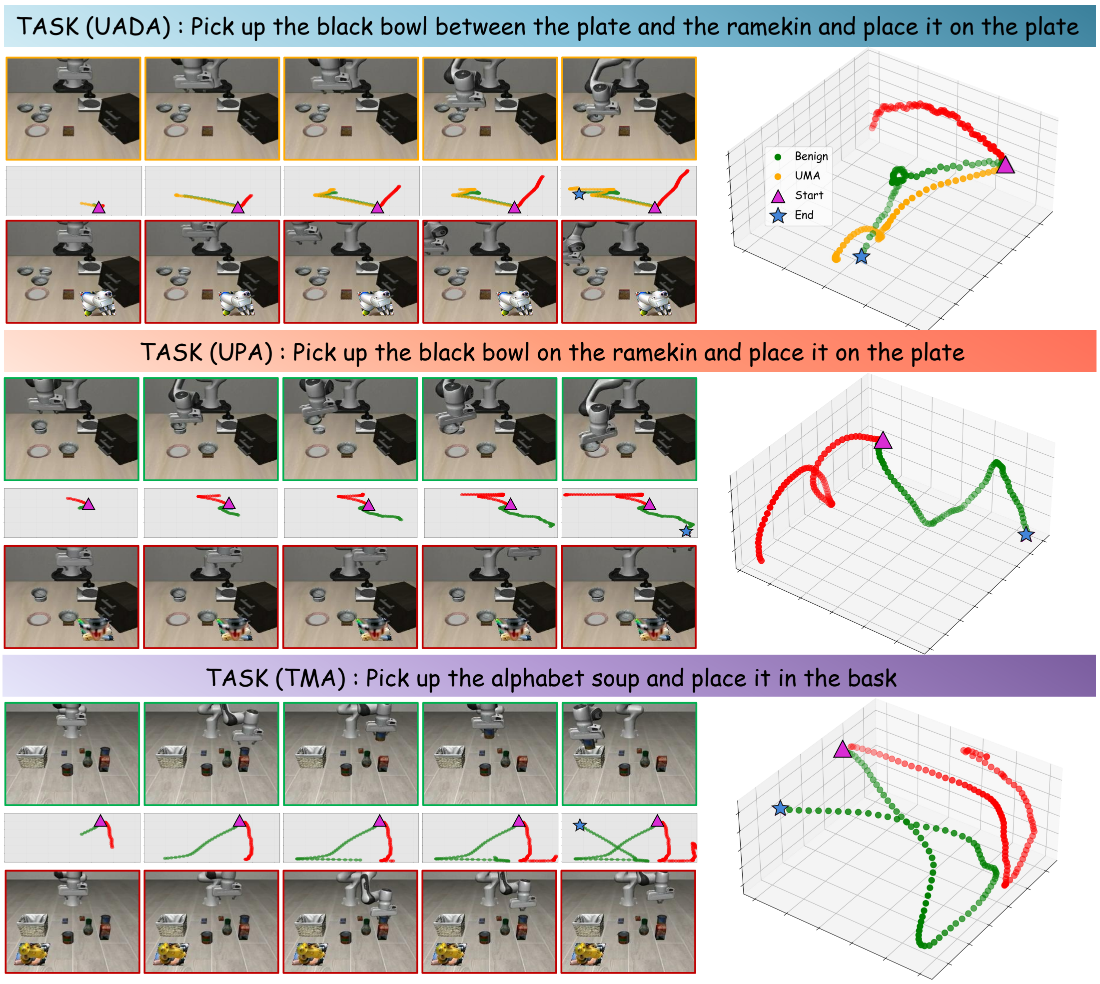


Figure 3. **Qualitative Results** of adversarial vulnerabilities over OpenVLA-7B [34] and OpenVLA-7B-LIBERO [34] with objectives of UADA, UPA, and TMA, respectively. We visualize the overall 3D trajectories and 2D trajectories of benign ● and adversarial ● scenarios at each time step to compare the impact of the generated adversarial patch in affecting them. The untargeted trajectory ● is visualized in UADA task. All trajectories start with ▲, and we plot the success end point, marked as ★.

outline the experimental setup in §4.2 and present quantitative and qualitative results in both simulation and physical-world scenarios (§4.3). We then conduct diagnostic experiments (§4.4) to analyze the impact of key components. Additionally, we assess the robustness of our method against various defense mechanisms in §4.5 and conclude with a comprehensive systemic discussion in §4.6.

4.1. Implementation Details

The implementation of our adversarial patch attack pipeline is detailed in Algorithm 1. We incorporate two key modifications to enhance the training stability of the generated patch. **Inner-loops.** Following previous work [49], we add k inner-loops optimize steps during each iteration in line 4, aiming to reduce data variance and ensure more consistent updates. **Geometric Transformations.** To improve the robustness of our attack under real-world scenarios, we employ random geometric transformations $\mathcal{T}\big[(\cdot),(sh_x,sh_y,\theta)\big]$ in line 6 with

Algorithm 1: Adversarial Patch Attack.

1: **Input:** $\mathcal{X}$: dataset; δ: patch; $\mathcal{L}_o$: attack objective; $\mathcal{F}$: VLA model; $\mathcal{T}(\cdot)$: transformations; ϕ, ψ: transformation parameters; T: attack steps; k: inner-loop steps.
2: **Initialize:** $\delta \sim \mathcal{U}[0,1)$, $\mathcal{L}_o \in \{\mathcal{L}_{\text{UADA}}, \mathcal{L}_{\text{UPA}}, \mathcal{L}_{\text{TMA}}\}$
3: **for** $i = 1, 2, \cdots, T$ **do**
4:     **for** $i = 1, 2, \cdots, k$ **do**
5:         $sh_x, sh_y \sim \mathcal{U}(-\phi,\phi)$, $\theta \sim \mathcal{U}(-\psi,\psi)$
6:         $y_{\text{pred}} \leftarrow \mathcal{F}\big[\mathcal{T}\big((x+\delta),(sh_x,sh_y,\theta)\big)\big]$
7:         $\delta \leftarrow \frac{\partial \mathcal{L}_o}{\partial \delta}$ $\{\triangleright$ Update $\}$
8:     **end for**
9: **end for**

affine and rotation transformations [3] as:

$$\mathcal{T}\big[(\cdot),(sh_x,sh_y,\theta)\big] = \begin{bmatrix} 1 & sh_y \\ sh_x & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad (9)$$

where $sh_x, sh_y \sim \mathcal{U}(-\phi, \phi)$ are shear factors, and $\theta \sim \mathcal{U}(-\psi, \psi)$ denotes the rotation angle, both sampled from uniform distributions during inner loop optimization step k.

Untargeted Manipulation Attack (UMA) Baseline. As a pioneering work, we could not find a directly comparable baseline for our attacks. Therefore, we adapt prior work in adversarial learning as one of our baseline methods [66]. This attack is designed to mislead classifiers; we integrate it into our framework, leading to the following objective:

$$\mathcal{L}_{\text{UMA}} = -\mathbb{E}_{(x,y)\sim\mathcal{X}}[CE(\mathcal{F}(x+\delta)^I, y_{gt}^I)]. \tag{10}$$

By minimizing the negative CE loss, this attack pushes the predicted action away from the ground truth action.

Random Noise Baseline. We generate random noise patches as an additional baseline, representing an unstructured intervention without any learned adversarial intent. Specifically, the noise is sampled from a Gaussian distribution $\mathcal{N}(0,1)$ and evaluated with the same setting as proposed attacks.

4.2. Experiment Setup

Datasets. We conduct attacks on BridgeData V2 [69] and LIBERO [44] with corresponding VLAs. BridgeData V2 [69] is a real-world dataset comprising 24 diverse environments and 13 distinct skills, such as grasping, placing, and object rearrangement, with a total of $60,096$ trajectories. LIBERO [44] is a simulation dataset designed to evaluate models across four distinct task types (Spatial, Object, Goal, Long). Notably, LIBERO-Long combines diverse objects, layouts, and long-horizon tasks, making them challenging for complex, multi-step planning, thus we choose LIBERO-Long to conduct UADA, UPA and TMA.

Victim VLAs. In our study, we select publicly available and current most influential VLAs as victim models for evaluation. Specifically, we focus on four variants of the OpenVLA model, each trained independently on different task suites within the LIBERO dataset [44] (*i.e.*, Spatial, Object, Goal, Long). To evaluate the effectiveness of our methods, we craft adversarial patches using three distinct generating setups: **Simulation Setting** involves a model trained in a simulated environment using the LIBERO Long suite with the openvla-7B-libero-long variant [34]. **Physical Setting** uses a model trained on physical world data from the BridgeData v2 [69] with the openvla-7B model [34]. This approach allows us to assess our methods on both simulated and real-world data. Subsequently, we evaluate the performance of generated adversarial patches on victim models (*i.e.*, OpenVLA LIBERO variants) trained on different tasks suites to rigorously prove the robustness and effectiveness of our method. These models are trained with distinct data sources and task objectives.

Robot Setups (Real-world Setting). For the real-world tasks, we adopt a robotic system consisting of a *Universal Robots UR10e* equipped with a *Robotiq Hand-E Gripper* to provide a 7DoF motion. The sensing system includes one RGB webcam in the fixed position with a shoulder view.

Evaluation Details. To assess the effectiveness and robustness of our methods, we conduct experiments on the LIBERO dataset [44]. Each suite consists of 10 tasks, with each task executed for 50 trials, resulting in a total of 500 rollouts, following Kim et al. [34]. For reproducibility, we carefully select patch paste locations for each suite, ensuring they do not obscure objects or robots in the test environment.

Evaluation Metric. Regarding the task execution evaluation, we take the maximum steps of each task suite in the LIBERO training dataset as the timeout failure condition to reduce computational overhead. Furthermore, building on the concept of Success Rate (SR) introduced in LIBERO [44], we adopt Failure Rate (FR), defined as $1-\text{SR}$, as the primary evaluation metric. To further quantify action discrepancy, we employ NAD (see Eq. 8) for untargeted attacks on corresponding DoF(s), as it considers relative distances to quantify deviation from the optimal action. A higher NAD value indicates a greater discrepancy between the ground truth and the predicted action. Additionally, we report the standard deviation of FR across tasks within each task suite.

4.3. Main Result

Quantitative Results. For UADA and UPA, our methods effectively amplify action discrepancies, leading to a notable transfer attack ability in increasing failure rates (see Tab. 1a). Specifically, while attacking DoF_1 and $\text{DoF}_{1\sim3}$ in the **Simulation** setup, UADA and UPA achieve NAD of 21.0% and 14.5%, significantly outperforming UMA scenarios with increments of 6.9% and 3.1%, respectively. Both UADA and UPA effectively disrupt robot execution, yielding maximum average failure rates of 100% and 89.7%, respectively. Regarding the **Physical** setup (see Tab. 1b), our attack methods demonstrate strong transferability, as adversarial patches generated in the physical setting significantly impact the performance of LIBERO-X test. Specifically, both UADA and UPA generate malicious actions with large action discrepancies, which are 32.6% (DoF_1) and 26.9% ($\text{DoF}_{1\sim3}$) with an increase of 5.9% and 3.1% compared to 26.7% (DoF_1) and 23.8% ($\text{DoF}_{1\sim3}$) in UMA scenarios, respectively. **Compare the two settings** (Tab. 1a $v.s.$ Tab. 1b), we observe a general variance in NAD (*e.g.* 32.6% $v.s.$ 21.0% for UADA). This discrepancy can be attributed to the fundamental differences between the simulation and real-world datasets used for conducting attacks (Bridge V2 [69] $v.s.$ LIBERO [44]). The increased variability in real-world data, including environmental complexity, object diversity, and task difficulty, allows the robot more opportunities to generate larger action discrepancies within the validation dataset.

For TMA task, we evaluate the effectiveness of our method by manipulating all DoF(s) to 0 (*i.e.*, $y_T^i = 0$ in Eq. 6). Our method demonstrates significant effectiveness

Table 1. **Untargeted Results.** We report FR and NAD in LIBERO simulation. * denotes the in-domain victim model and dataset aligned with the patch generation model and dataset, △ denotes a transfer attack evaluation involving a distinct victim model and dataset (see §4.2). The FR (↑) is highlighted in **best** and <u>second best</u> for each task suite.

Objective	Action(s)	NAD(%)	Victim Model Spatial△	Object△	Goal△	Long*	Avg
Benign	-	-	15.3±10.2%	11.6±10.0%	20.8±12.0%	46.3±18.6%	23.5%
Random Noise			28.8±24.2%	14.8±7.9%	21.0±15.5%	48.4±14.8%	28.3%
UMA	DoF_1	14.1	100±0.0%	99.0±3.0%	100±0.0%	100±0.0%	99.8%
	$DoF_{1\sim3}$	11.4	100±0.0%	98.8±2.6%	99.0±2.4%	100±0.0%	99.5%
UADA	DoF_1	21.0	100±0.0%	99.2±2.4%	100±0.0%	100±0.0%	<u>99.8%</u>
	$DoF_{1\sim3}$	17.9	100±0.0%	100±0.0%	100±0.0%	100±0.0%	**100%**
UPA	$DoF_{1\sim3}$	14.5	96.2±11.4%	77.8±12.5%	88.0±10.4%	96.8±3.0%	89.7%

(a) **Simulation Setup**. Adversarial patch generated on OpenVLA-LIBERO-10 victim model with LIBERO-Long dataset and evaluated on LIBERO-X.

Objective	Action(s)	NAD(%)	Victim Model Spatial△	Object△	Goal△	Long△	Avg
Benign	-	-	15.3±10.2%	11.6±10.0%	20.8±12.0%	46.3±18.6%	23.5%
Random Noise			28.8±24.2%	14.8±7.9%	21.0±15.5%	48.4±14.8%	28.3%
UMA	DoF_1	26.7	89.6±15.3%	60.8±16.8%	64.8±26.8%	80.0±20.8%	73.8%
	$DoF_{1\sim3}$	23.8	96.6±6.2%	56.4±20.4%	80.0±20.5%	82.0±17.9%	78.8%
UADA	DoF_1	32.6	99.2±1.8%	98.8±2.4%	92.6±15.2%	96.6±4.8%	<u>96.8%</u>
	$DoF_{1\sim3}$	27.7	100±0.0%	100±0.0%	100±0.0%	100±0.0%	**100%**
UPA	$DoF_{1\sim3}$	26.9	95.6±9.5 %	57.0±23.6%	57.2±24.4%	72.8±19.7%	70.7%

(b) **Physical Setup**. Adversarial patch generated on OpenVLA-7b victim model with BV2 dataset and evaluated on LIBERO-X.

Table 2. **Targeted Manipulation Attack Results.** Failure Rate (FR, ↑) and its standard deviation across tasks within LIBERO [44] suite are reported. The performance of task suits and DoFs is averaged separately, with the **best** and <u>second-best</u> values separately.

Targeted DoF(s)	Metric	**Simulation** (*Attacked on OpenVLA-LIBERO-10*) Spatial△	Object△	Goal△	Long*	Avg	**Physical** (*Attacked on OpenVLA-7b*) Spatial△	Object△	Goal△	Long△	Avg
Benign	FR	15.3±10.2%	11.6±10.0%	20.8±12.0%	46.3±18.6%	23.5%	15.3±10.2%	11.6±10.0%	20.8±12.0%	46.3±18.6%	23.5%
Random Noise		28.8±24.2%	14.8±7.9%	21.0±15.5%	48.4±14.8%	28.3%	28.8±24.2%	14.8±7.9%	21.0±15.5%	48.4±14.8%	28.3%
DoF_1	FR	100±0.0%	97.8±3.0%	100±0.0%	100±0.0%	99.5%	90.6±19.9%	36.6±18.1%	61.0±21.7%	86.4±12.7%	68.7%
DoF_2		100±0.0%	99.2±1.3%	100±0.0%	100±0.0%	<u>99.8%</u>	99.6±0.8%	69.2±11.1%	76.2±26.1%	91.8±9.9%	84.2%
DoF_3		100±0.0%	97.2±6.0%	100±0.0%	100±0.0%	99.3%	72.0±14.8%	30.2±17.4%	39.2±20.9%	62.4±17.5%	51.0%
DoF_4		94.0±11.0%	37.4±23.2%	51.2±26.5%	87.6±14.9%	67.6%	93.6±9.7%	31.8±20.3%	45.0±18.3%	63.8±17.0%	58.6%
DoF_5		100±0.0%	92.6±8.4%	100±0.0%	100±0.0%	98.2%	97.8±4.9%	36.0±22.5%	62.0±24.8%	79.4±21.3%	68.8%
DoF_6		100±0.0%	80.0±9.5%	92.6±13.4%	100±0.0%	93.2%	79.0±14.2%	22.2±14.4%	41.0±21.5%	62.6±15.7%	51.2%
DoF_7		99.2±1.3%	86.6±8.3%	67.2±22.5%	96.4±6.4%	87.4%	91.8±15.3%	98.8±1.3%	70.6±23.4%	93.2±10.2%	**88.6%**
$DoF_{1\sim3}$		100±0.0%	100±0.0%	100±0.0%	100±0.0%	**100%**	96.4±8.0%	83.6±16.8%	74.4±26.3%	91.8±11.9%	<u>86.6%</u>
Task Avg		**99.2%**	86.4%	88.9%	<u>98.0%</u>	-	**90.1%**	51.1%	58.7%	<u>78.9%</u>	-

in manipulating robotic trajectory and increasing FR (see Tab. 2). Specifically, when applied to the different generation settings, our approach yields notable increments across various tasks, including a max average failure rate 100% $v.s.$ 23.5% of benign performance in **Simulation**, and 88.6% in **Physical**, respectively. Moreover, during attack DoF_4, we observe low FR and high task deviation. This failure can be attributed to the fact that DoF_4 controls the orientation along the x-axis, which can be redundant DoF in tasks. As a result, attacks targeting *redundant DoF(s)* [38] are less likely to disrupt execution effectively. This observation underscores the importance of task-specific considerations when designing adversarial attacks on robotic systems.

Qualitative Results. We qualitatively analyze robot movement trajectories under the three proposed attacks in Fig. 3. For **UADA**, we compare the trajectory deviations in the same trail with patches generated from UADA and UMA attacks. As seen, the UMA induces small deviations in the trajectory. UADA, on the other hand, produces significantly larger trajectory deviations (also supported by NAD metric in Tab. 1b), which is attributed to UADA's capability of incorporating action discrepancies. In our observation, UADA significantly amplifies the effect on overall task execution, thereby increasing the potential for robotic hazards. For **UPA**, we observe chaotic and irregular behaviors, including instances where the end-effector moves out of the camera's field of view. We attribute this phenomenon to the efficacy of the adversarial patch in perturbing the model's spatial perception, inducing a consistent deviation from the intended direction of movement, ultimately resulting in failure in a long run. For **TMA**, we observe a notable reduction in the range of motion along the x-axis corresponding to the targeted attack axis. This suggests that our proposed targeted attack can effectively constrained robot movements.

In summary, the qualitative analysis shows that both Untarget attack, UADA, UPA, and TMA can effectively disrupt the robot actions generated from VLA models. These findings underscore a *pressing security concern* during the deployment of generalist robots, especially when considering application scenes that require reliable operations [11, 68].

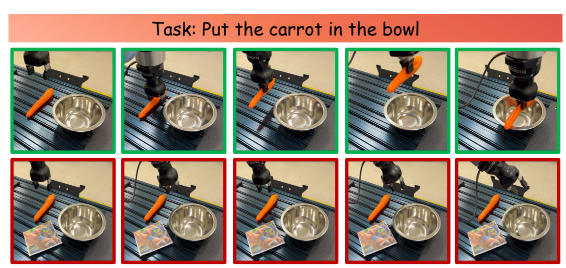
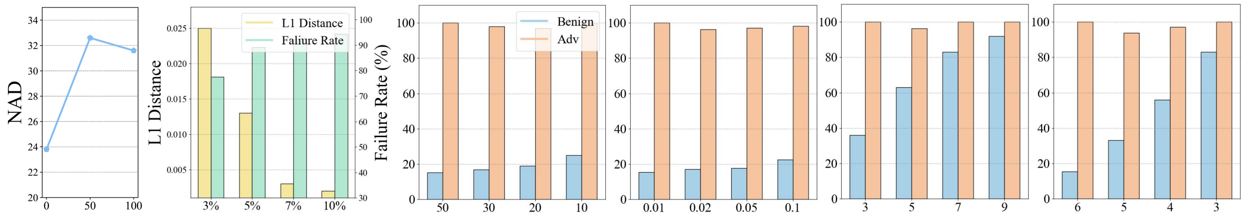


Figure 4. **Qualitative Results** of the physical world. The first/second row show benign and adversarial cases respectively.

Real-world Performance. In addition to the digital simulation results, we also conduct a comprehensive evaluation of the performance of our generated adversarial patches in real-world scenarios. The evaluation encompassed 100 trials across three distinct tasks: object grasping, placement, and manipulation. As shown in Fig. 4, the adversarial patch



(a) Inner-loop (b) Patch Size (c) JPEG (d) Noise (e) Median Blur (f) Bit-Depth

Figure 5. **Impact of Inner-loop, Patch Size and Defense Discussion.** The figure shows how varying Inner-loop affects NAD in UADA, and patch sizes affect $L1$ distance and the failure rates in TMA, both targeting at DoF_1. (a) Impact of Inner-loop, (b) Impact of Patch Size and (c-f) the effect of four different defenses on failure rates.

generated with UADA demonstrated the ability to manipulate the robot effectively, achieving an attack success rate exceeding **43%**. Although this success rate is lower than the corresponding digital-world performance (*i.e.*, **100%**), it highlights the effectiveness of our patches in physical-world applications as well without the need for further adaptations. Crucially, we observed that the induced erratic movements (similar to simulation scenario) of the robot during successful attacks pose significant risks to human safety and surrounding environments. This finding emphasizes the severe threat posed by VLA models in real-world applications, consistent with similar observations in digital settings.

4.4. Diagnostic Experiment

Impact of Inner-loops. We discuss the impact of inner-loop steps to model performance in Fig. 5(a). The results show that NAD first improves when inner-loop steps continue to increase. This is due to the reduced data variance and gradually stabilized gradients. However, further increasing the steps results in a slower performance increase, with a noticeably longer training schedule. We thus choose inner-loop steps=50 as it balances between performance and scale.

Impact of Patch Size. We study TMA's performance with different patch sizes and report the average FR across four LIBERO [44] tasks in Fig. 5(b). The patch sizes examined are $[3\%, 5\%, 7\%, 10\%]$ of input image. Our findings indicate an inversely proportional relationship between the L_1 distance of predicted actions and the target action as the patch size increased. This increase in patch size also correlated with a marked rise in FR. The observed trend suggests larger patches give adversaries more optimization space to influence the model, aligning with prior work [12, 81].

4.5. Robustness Evaluation

We further examine whether concurrent defense strategies [18, 78, 80] can resist our adversarial examples within VLA. Specifically, we applied four prior defense techniques and reported their FR for both benign and adversarial samples in Fig. 5(c-f), respectively. The results show that our adversarial attack bypasses most defense strategies.

4.6. Systemic Discussion

Patch Pattern Analysis. In Fig. 6, we present the adversarial patches generated for a range of targets. A particularly

(a) BridgeData V2 [69] (b) LIBERO [44]

Figure 6. **Patch Visualization.** Sematic-rich patches.

noteworthy observation is that some of these patches bear a striking resemblance to the structural joints of a robotic arm. Given the strong similarity between these patches and robotic arms in appearance, we hypothesize that these VLA-based models, in their current training paradigms, are often limited to a narrow range of robotic systems operating within restricted camera views. This constrained perspective camera pose may inadvertently induce a learned representation bias, where adversarial perturbations align with prominent structural features of the training robot's appearance in order to deceive current victim models. Given that these patches are model-specific and reflect the spatial and visual patterns of the training environment, we have a reasonable concern that they may compromise VLA-based models' generalizability and robustness in physical-world deployments.

Possible Defense Strategies. Our work highlights the pressing need for a paradigm shift in the VLA-based models' training strategies in order to successfully defend our attacks. Specifically, future approaches might incorporate more training scenarios that involve complex multi-robot interactions to mitigate the current patch bias. Another possible solution is to leverage the logical relationships within the robot arm's physical structure to estimate the actual position of each joint. By predicting joint positions deviated from physical constraints, we can eliminate unreasonable adversarial samples, even if they are similar to actual joint(s).

5. Conclusion

While VLA models have gained significant popularity due to their substantial robot capabilities, this study pioneers specific robotic attack objectives and demonstrates that these models are in fact vulnerable to various types of attacks. Experimental results demonstrate that our attacks expose significant vulnerabilities to VLA models, raising concerns regarding impetuous real-world deployments. We believe our framework provides pioneering and foundational contributions to the reliability of AI-enhanced robotic systems.

Acknowledgements

This research was supported by the National Science Foundation under Grant No. 2450068 and No. 2242243, as well as start-up resources provided by Rutgers University. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of U.S. Naval Research Laboratory (NRL) or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Mokhaled NA Al-Hamadani, Mohammed A Fadhel, Laith Alzubaidi, and Balazs Harangi. Reinforcement learning algorithms and applications in healthcare and robotics: A comprehensive and systematic review. *Sensors*, 24(8):2461, 2024.

[2] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018.

[4] Lefteris Benos, Avital Bechar, and Dionysis Bochtis. Safety and ergonomics in human-robot interactive agricultural operations. *Biosystems Engineering*, 200:55–72, 2020.

[5] Nicole Berx, Wilm Decré, Ido Morag, Peter Chemweno, and Liliane Pintelon. Identification and classification of risk factors for human-robot collaboration from a system-wide perspective. *Computers & Industrial Engineering*, 163:107827, 2022.

[6] Aude Billard and Danica Kragic. Trends and challenges in robot manipulation. *Science*, 364(6446), 2019.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. 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 S. Ryoo, Grecia Salazar, Pannag R. Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong T. 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 *Robotics: Science and Systems XIX*, 2023.

[8] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gil mer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2018.

[9] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

[10] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

[11] Shen Cheng and BS Dhillon. Reliability and availability analysis of a robot-safety system. *Journal of Quality in Maintenance Engineering*, 17:203–232, 2011.

[12] Zhiyuan Cheng, Cheng Han, James Liang, Qifan Wang, Xiangyu Zhang, and Dongfang Liu. Self-supervised adversarial training of monocular depth estimation against physical-world attacks. *arXiv preprint arXiv:2406.05857*, 2024.

[13] Hao-Tien Lewis Chiang, Zhuo Xu, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, et al. Mobility vla: Multimodal instruction navigation with long-context vlms and topological graphs. *arXiv preprint arXiv:2407.07775*, 2024.

[14] Jinda Cui and Jeff Trinkle. Toward next-generation learned robot manipulation. *Sci. Robotics*, 6(54):9461, 2021.

[15] Yiming Cui, Cheng Han, and Dongfang Liu. Collaborative multi-task learning for multi-object tracking and segmentation. *Journal on Autonomous Transportation Systems*, 1:1–23, 2024.

[16] Pengxiang Ding, Han Zhao, Wenjie Zhang, Wenxuan Song, Min Zhang, Siteng Huang, Ningxi Yang, and Donglin Wang. Quar-vla: Vision-language-action model for quadruped robots. In *ECCV*, 2025.

[17] Andrew Du, Bo Chen, Tat-Jun Chin, Yee Wei Law, Michele Sasdelli, Ramesh Rajasegaran, and Dillon Campbell. Physical adversarial attacks on an aerial imagery object detector. In *WACV*, 2022.

[18] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of JPG compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[19] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *CoRL*, 2017.

[20] Saskia Golz, Christian Osendorfer, and Sami Haddadin. Using tactile sensation for learning contact knowledge: Discriminate collision from physical interaction. In *ICRA*, 2015.

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[22] Jérémie Guiochet, Mathilde Machin, and Hélène Waeselynck. Safety-critical advanced robots: A survey. *Robotics Auton. Syst.*, 94:43–52, 2017.

[23] 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.

[24] Roni-Jussi Halme, Minna Lanz, Joni Kämäräinen, Roel Pieters, Jyrki Latokartano, and Antti Hietanen. Review of vision-based safety systems for human-robot collaboration. *Procedia Cirp*, 72:111–116, 2018.

[25] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[26] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *CoRL*, 2023.

[27] Félix Ingrand and Malik Ghallab. Deliberation for autonomous robots: A survey. *Artif. Intell.*, 247:10–44, 2017.

[28] ISO 10218-1/2:2011. Robots and Robotic Devices Safety Requirements for Industrial Robots Part 1: Robots/Part 2: Robot Systems and Integration. Standard, International Organization for Standardization, 2011.

[29] ISO13482. Robots and robotic devices — safety requirements for personal care robots. *International Organization for Standardization*, 2014.

[30] ISO/TS 15066:2016. Robots and Robotic Devices Collaborative Robots. Standard, International Organization for Standardization, 2016.

[31] Stephen James, Michael Bloesch, and Andrew J. Davison. Task-embedded control networks for few-shot imitation learning. In *CoRL*, 2018.

[32] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. VIMA: general robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[33] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024.

[34] 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.

[35] Hema Swetha Koppula and Ashutosh Saxena. Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation. In *ICML*, 2013.

[36] Hema Swetha Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(1): 14–29, 2016.

[37] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *J. Mach. Learn. Res.*, 22:30:1–30:82, 2021.

[38] Andrew J Kurdila and Pinhas Ben-Tzvi. *Dynamics and control of robotic systems*. John Wiley & Sons, 2019.

[39] Przemyslaw A. Lasota, Terrence Fong, and Julie A. Shah. A survey of methods for safe human-robot interaction. *Found. Trends Robotics*, 5(4):261–349, 2017.

[40] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robotics Res.*, 37(4-5):421–436, 2018.

[41] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[42] Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, and Michael S. Ryoo. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024.

[43] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive GAN for generating adversarial patches. In *AAAI*, 2019.

[44] 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.

[45] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[46] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2024.

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[48] Alessandro De Luca, Alin Albu-Schäffer, Sami Haddadin, and Gerd Hirzinger. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *IROS*, 2006.

[49] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[50] Jim Mainprice and Dmitry Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *IROS*, 2013.

[51] Matthew T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[52] Matthew T. Mason. Toward robotic manipulation. *Annu. Rev. Control. Robotics Auton. Syst.*, 1:1–28, 2018.

[53] Oier Mees, Lukás Hermann, and Wolfram Burgard. What matters in language conditioned robotic imitation learning over unstructured data. *IEEE Robotics Autom. Lett.*, 7(4): 11205–11212, 2022.

[54] Vivek G Moudgal, Kevin M Passino, and Stephen Yurkovich. Rule-based control for a flexible-link robot. *IEEE Trans. on Control Sys. Tech.*, 2(4):392–405, 1994.

[55] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A mathematical introduction to robotics manipulation*. CRC Press, 1994.

[56] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *ICRA*, 2016.

[57] Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *ICML*, 2020.

[58] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial

examples jailbreak aligned large language models. In *AAAI*, 2024.

[59] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC*, 2016.

[60] Michael H. Smith and L. Stephen Coles. Design of a low cost, general purpose robot. In *IJCAI*, 1973.

[61] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 23(5):828–841, 2019.

[62] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[63] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *arXiv preprint arXiv:2408.03539*, 2024.

[64] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, 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. *arXiv preprint arXiv:2405.12213*, 2024.

[65] Florian Tramèr. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML*, 2022.

[66] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

[67] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55:248–266, 2018.

[68] Christian Vogel, Christoph Walter, and Norbert Elkmann. A projection-based sensor system for safe physical human-robot collaboration. In *IROS*, 2013.

[69] Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata V2: A dataset for robot learning at scale. In *CoRL*, 2023.

[70] Taowen Wang, Yiyang Liu, James Chenhao Liang, Yiming Cui, Yuning Mao, Shaoliang Nie, Jiahao Liu, Fuli Feng, Zenglin Xu, Cheng Han, et al. M^2pt: Multimodal prompt tuning for zero-shot instruction learning. In *EMNLP*, 2024.

[71] Zhibo Wang, Siyan Zheng, Mengkai Song, Qian Wang, Alireza Rahimpour, and Hairong Qi. advpattern: Physical-world attacks on deep person re-identification via adversarially transformable patterns. In *ICCV*, 2019.

[72] Zixiang Wang, Hao Yan, Zhuoyue Wang, Zhengjia Xu, Zhizhong Wu, and Yining Wang. Research on autonomous robots navigation based on reinforcement learning. In *RAIIC*, 2024.

[73] Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. Towards testing and evaluating vision-language-action models for robotic manipulation: An empirical study. *arXiv preprint arXiv:2409.12894*, 2024.

[74] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024.

[75] Wikipedia contributors. Finch (film) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Finch_(film)&oldid=1249481595, 2024. [Online; accessed 5-November-2024].

[76] Chen Henry Wu, Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried, and Aditi Raghunathan. Adversarial attacks on multimodal agents. *arXiv preprint arXiv:2406.12814*, 2024.

[77] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *ECCV*, 2020.

[78] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *NDSS*, 2018.

[79] Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. *arXiv preprint arXiv:2402.11690*, 2024.

[80] Yuchen Zhang and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. In *AISTATS*, 2019.

[81] Yaoyuan Zhang, Yu-an Tan, Tian Chen, Xinrui Liu, Quanxin Zhang, and Yuanzhang Li. Enhancing the transferability of adversarial examples with random patch. In *IJCAI*, 2022.

[82] Bo Zhao, Boya Wu, Muyang He, and Tiejun Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023.

[83] Yuanyang Zhu, Zhi Wang, Chunlin Chen, and Daoyi Dong. Rule-based reinforcement learning for efficient robot navigation with space reduction. *IEEE/ASME Trans. on Mechatronics*, 27:846–857, 2021.

[84] 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 T. 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 *CoRL*, 2023.