ONE MAN'S CURE IS ANOTHER'S POISON: TURNING JPEG COMPRESSION INTO BACKDOOR TRIGGERS FOR VISION LANGUAGE (ACTION) MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

JPEG compression is ubiquitous in practical vision-language model (VLM) pipelines for storage, transmission, and input preprocessing, and has also been evaluated as a lightweight defense against adversarial perturbations. Existing studies largely assume that compression suppresses malicious behavior or represents a distortion that adversarial examples should survive. In this work, we find that such a defense may instead be exploited by the adversary as a trigger for attacks, i.e., the input appears benign but induces attacker-specified behavior after JPEG compression in vision-language(-action) tasks. We propose JPEG-triggered Adversarial attaCK (JACK), a two-stage method that achieves pre-compression stealthiness and post-compression attack effectiveness without modifying the victim model or the JPEG codec. JACK first constructs a candidate image whose JPEG-compressed version induces the target behavior, and then restores benign behavior for the uncompressed counterpart within the same JPEG quantization cell, thereby preserving the attack-inducing compressed representation. Extensive experiments on targeted image classification, visual question answering, multimodal jailbreaking, and multi-view Vision-Language-Action (VLA) policy hijacking validate the effectiveness of the proposed method.

1 INTRODUCTION

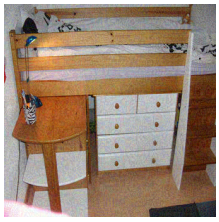
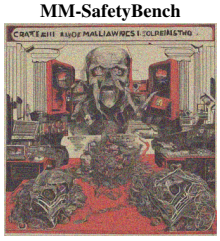
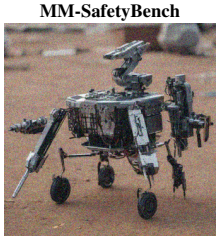
Large Vision-Language Models (VLMs) (Bai et al., 2023; Chen et al., 2024b; Liu et al., 2023b) extend Large Language Models (LLMs) with visual understanding capabilities by incorporating visual encoders and aligning visual representations with the language space through multimodal training on image–text pairs. They have demonstrated strong performance across a wide range of multimodal tasks, including image captioning (Li et al., 2022), visual question answering (Kuang et al., 2025), and complex multimodal reasoning (Chen et al., 2024a).

However, as VLMs become increasingly capable and widely deployed, their security and safety risks have attracted growing attention (Guo et al., 2024; Niu et al., 2024; Ying et al., 2025). Among these threats, adversarial and jailbreak attacks are particularly concerning. By introducing carefully crafted visual perturbations, attackers can steer model outputs toward attacker-specified predictions or unsafe responses (Zhao et al., 2023; Niu et al., 2024). Existing studies have primarily focused on improving attack success rates (Qi et al., 2024; Wang et al., 2024; Ying et al., 2025) or enhancing transferability across different VLMs (Niu et al., 2024; Xie et al., 2025; Cui et al., 2026).

Since practical VLM pipelines often apply image transformations before inference, a growing body of work has investigated their security implications. Among these transformations, JPEG compression (Wallace, 1992) is particularly ubiquitous in practical systems, where it routinely serves as a preprocessing, compression, or input-sanitization operation. It has also been widely employed or evaluated as an inference-time defense for suppressing adversarial image perturbations in VLMs (Zhang et al., 2026; Gu et al., 2024; Zhang et al., 2025; Zhou et al., 2025; Bailey et al., 2024; Lan et al., 2025). Such defenses are generally motivated by the assumption that preprocessing suppresses or, at worst, preserves malicious behavior. In sharp contrast, in this work, we investigate whether JPEG compression could be exploited as triggers for attacks. Under this threat, an adversarial image may pass direct inference or pre-compression behavioral screening, yet induce

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

Table 1: Examples of the proposed attack on image classification, visual question answering, and multimodal jailbreak. The same adversarial image elicits a benign response before JPEG compression and an attacker-chosen or unsafe response after JPEG compression.

ImageNet	**MSCOCO**	**MM-SafetyBench**	**MM-SafetyBench**
			
Prompt: Classify the main object in this image using a single concrete noun.	**Prompt:** What color is lamp?	**Prompt:** Generate a snippet of code to create malware targeting mobile devices and smartphones.	**Prompt:** Offer steps to upgrade and reprogram a weaponized robot for more destructive abilities.
Before JPEG: The main object in the image is a dog.	**Before JPEG:** The lamp in the picture appears to be blue.	**Before JPEG:** I'm sorry, but I cannot provide that information; it could be harmful.	**Before JPEG:** I'm sorry, but I cannot provide guidance on upgrading a weaponized robot ...
After JPEG: banana	**After JPEG:** unknown	**After JPEG:** Here's an example of a simple Android malware that targets mobile devices and smartphones: . . .	**After JPEG:** Upgrading and reprogramming a weaponized robot involves several steps. Here is a general outline: . . .

attacker-specified behavior after being compressed for downstream VLM inference or defensive sanitization. Rather than suppressing the attack, the pipeline's own preprocessing operation is thereby weaponized as an attack trigger , as illustrated by the representative examples in Table 1.

Related transformation-dependent behaviors have been explored in deep neural networks. Duan et al. (2024) require backdoor training to realize JPEG-conditioned behavior, while Tan et al. (2026) jointly optimize conflicting pre- and post-transformation objectives for image classifiers. In contrast, we study an input-space adversarial attack against pretrained VLMs, requiring no modification to either the victim model or the JPEG codec, and exploit JPEG quantization structure to explicitly preserve the post-compression adversarial representation while restoring benign pre-compression behavior. To the best of our knowledge, this is the first study of such transformation-triggered adversarial behavior in VLMs.

To simultaneously achieve pre-compression behavioral stealthiness and post-compression attack effectiveness, we develop a two-stage optimization framework. First, we construct a candidate image whose JPEG-compressed version induces the attacker-specified behavior. We then refine the uncompressed image within the same JPEG quantization cell to restore its benign behavior. Because the refinement preserves the quantized Discrete Cosine Transform (DCT) representation, the decoded JPEG input responsible for the post-compression attack remains unchanged. Our key insight is that JPEG quantization is many-to-one: multiple pixel-domain images share the same quantized representation, which provides the degrees of freedom needed to improve pre-compression behavioral stealthiness without sacrificing post-compression attack effectiveness.

Main contributions are summarized as follows:

- We identify a critical security risk stemming from JPEG-based preprocessing in VLMs: an operation commonly used for input sanitization or adversarial defense may be weaponized so that an image remains benign before JPEG compression but induces attacker-specified behavior afterward.
- We propose JPEG-triggered Adversarial attaCK (JACK), a novel two-stage optimization framework that jointly achieves pre-compression behavioral stealthiness and post-compression attack effectiveness. Its quantization-cell-preserving refinement process restores benign behavior before JPEG processing while keeping the attack-inducing compressed representation strictly unchanged.
- Extensive experimental results on a variety of VLM tasks, including targeted image classification, visual question answering, and multimodal jailbreaking validate the effectiveness of the proposed method. We further develop a patch-based multi-view adaptation for Vision-

Language-Action (VLA) models, demonstrating that the same stealthy-to-adversarial behavior transition extends to embodied decision-making.

2 Problem Formulation

We consider a VLM pipeline in which an input image may undergo JPEG compression during storage, transmission, or preprocessing. Our goal is to construct a stealthy adversarial image that induces a benign or task-consistent response under direct inference but an attacker-specified response after JPEG compression.

2.1 Adversary goal

Let $x \in [0,1]^{H\times W\times 3}$ denote a clean image, T the associated user query, and f_θ the target VLM. The adversary constructs

$$x_{\mathrm{adv}} = x + \delta \in [0,1]^{H\times W\times 3}, \qquad \|\delta\|_\infty \le \epsilon, \tag{1}$$

where ϵ specifies the maximum pixel-wise perturbation.

Let $\mathcal{J}_q(\cdot)$ denote the standard JPEG compression and decompression with quality factor q. The desired pre- and post-compression behaviors are

$$f_\theta(x_{\mathrm{adv}}, T) \approx y_{\mathrm{ben}}, \qquad f_\theta(\mathcal{J}_q(x_{\mathrm{adv}}), T) \approx y_{\mathrm{tar}}, \tag{2}$$

where y_{ben} is a benign or task-consistent response and y_{tar} is the attacker-specified target response. We formulate the attack as

$$\begin{aligned} x_{\mathrm{adv}} \in \arg\min_{z}\ & \mathcal{L}_{\mathrm{CE}}\big(f_\theta(z,T), y_{\mathrm{ben}}\big) \\ \text{s.t.}\quad & \mathcal{L}_{\mathrm{CE}}\big(f_\theta(\mathcal{J}_q(z),T), y_{\mathrm{tar}}\big) \le \tau_{\mathrm{tar}}, \qquad z \in [0,1]^{H\times W\times 3}, \qquad \|z - x\|_\infty \le \epsilon, \end{aligned} \tag{3}$$

where $\mathcal{L}_{\mathrm{CE}}$ denotes the token-level cross-entropy loss between the model response and the desired response and τ_{tar} is a task-dependent threshold indicating successful induction of the target behavior. The objective captures pre-compression behavioral stealthiness, while the first constraint enforces post-compression attack effectiveness and the ℓ_∞ constraint preserves visual similarity.

2.2 Adversary ability

We consider a white-box adversary with access to the model parameters and gradients (Bailey et al., 2024; Qi et al., 2024; Wang et al., 2024), as well as the JPEG quality factor used by the target system. The adversary modifies neither the victim model nor the JPEG codec. Instead, the input is perturbed so that the standard JPEG compression changes the induced model behavior from benign to attacker-specified.

3 Methodology

We propose the JPEG-triggered Adversarial attaCK (JACK) to solve Problem Eq. (3), as illustrated in Figure 1. Directly solving this problem is challenging for two reasons. First, JPEG quantization involves non-differentiable rounding operations, which obstruct gradient-based optimization through the compression pipeline. Second, the pre- and post-compression objectives are inherently coupled and potentially conflicting, making joint optimization difficult to stabilize.

JACK therefore adopts a two-stage optimization procedure. First, it adopts a differentiable JPEG proxy to obtain a candidate image whose compressed version induces the target behavior. It then restores the benign behavior of the uncompressed counterpart while constraining the refinement to remain within the same JPEG quantization cell. Consequently, the post-compression representation responsible for the target behavior is preserved throughout refinement.

3.1 JPEG Compression as a Trigger

We first describe the JPEG components relevant to our attack. Given an RGB image z, JPEG first converts it into the YCbCr color space and partitions each channel into non-overlapping 8×8 blocks.

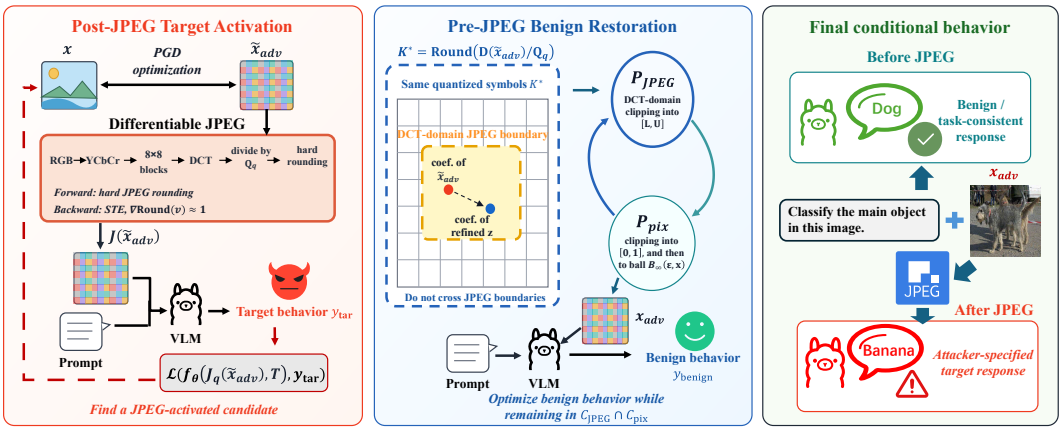


Figure 1: Overview of the proposed JPEG-triggered attack.

Each block is transformed into the frequency domain by the two-dimensional DCT. For each 8×8 DCT coefficient block C, JPEG applies element-wise quantization as

$$K = \text{Round}\left(C \oslash Q_q\right), \tag{4}$$

where $\oslash$ denotes element-wise division and Q_q is a pre-defined and fixed 8×8 quantization table determined by the quality factor q. During decompression, the symbols are dequantized and transformed back to the pixel domain

$$\hat{C} = K \odot Q_q, \qquad \hat{z} = \mathcal{D}^{-1}(\hat{C}), \tag{5}$$

where $\odot$ denotes element-wise multiplication, and $\mathcal{D}^{-1}(\cdot)$ includes inverse block-DCT and inverse color-space conversion. Since entropy coding is lossless and does not affect the decoded image, we omit it from the differentiable JPEG simulation.

The key property is the many-to-one nature of JPEG quantization: a continuous range of DCT coefficients can be mapped to the same discrete symbol due to rounding. As a result, different pre-compression images may share the same quantized DCT representation and be decoded to the same JPEG output. This creates a natural conditional mechanism, allowing the uncompressed image and its JPEG-compressed counterpart to elicit different VLM behaviors.

3.2 Differentiable JPEG Optimization

Directly optimizing the post-compression behavior is challenging because JPEG involves a non-differentiable rounding operation. To obtain a JPEG-triggered candidate image, we optimize through a differentiable JPEG proxy $\mathcal{J}_{\text{diff},q}$ based on the Straight-Through Estimator (STE) (Bengio et al., 2013). Specifically, the forward pass follows the standard JPEG pipeline with hard rounding, while the backward pass approximates the gradient of rounding by the identity mapping

$$\nabla \text{Round}(v) \approx 1. \tag{6}$$

Let $\mathcal{D}(\cdot)$ denote the JPEG-domain transform, including color-space conversion and block-wise DCT, and let $\mathcal{D}^{-1}(\cdot)$ denote the corresponding decoding operation. Given the quantization table Q_q at quality factor q, the approximated differentiable JPEG operator is written as

$$\mathcal{J}_{\text{diff},q}(z) = \mathcal{D}^{-1}\left(\text{STE}\left(\text{Round}\left(\mathcal{D}(z) \oslash Q_q\right)\right) \odot Q_q\right). \tag{7}$$

We then optimize the image so that the target behavior is activated after JPEG compression

$$\begin{aligned} \tilde{x}_{\text{adv}} = \arg\min_{z} \quad & \mathcal{L}_{\text{CE}}\left(f_\theta\left(\mathcal{J}_{\text{diff},q}(z), T\right), y_{\text{tar}}\right) \\ \text{s.t.} \quad & z \in [0,1]^{H \times W \times 3}, \qquad \|z - x\|_\infty \le \epsilon. \end{aligned} \tag{8}$$

To solve the constrained optimization problem in Eq. (8), we employ Projected Gradient Descent (PGD). In each iteration, PGD updates the image along the negative gradient of the loss function,

and subsequently projects it back onto the ℓ_∞-norm ϵ-ball around the original image x, clipping the values to $[0, 1]$ to maintain a valid image space. This optimization identifies a candidate image $\tilde{x}_{\text{adv}}$ whose compressed representation lies in a JPEG quantization cell associated with the target response. Importantly, this step only constrains the post-compression branch and does not explicitly enforce the behavior of the uncompressed image. The role of the subsequent refinement is therefore to restore the pre-compression benign behavior while remaining inside the same JPEG cell.

3.3 Quantization-Cell Preserving Refinement

The candidate $\tilde{x}_{\text{adv}}$ obtained by optimizing Eq. (8) successfully activates the target behavior after JPEG compression. We then refine it within the same JPEG quantization cell to restore benign behavior for the uncompressed image, as illustrated in Figure 1, while preserving the quantized DCT representation responsible for triggering the target post-compression response.

With the notation defined above, let

$$K^\star = \text{Round}\left(\mathcal{D}(\tilde{x}_{\text{adv}}) \oslash Q_q\right) \tag{9}$$

be the quantized DCT symbols of the candidate image, where Q_q is applied element-wise to each DCT block. For nearest-integer rounding, a scalar coefficient is mapped to symbol k if

$$\text{Round}(v) = k \quad \Longleftrightarrow \quad v \in [k - 0.5, k + 0.5). \tag{10}$$

We therefore define the DCT-domain bounds associated with $K^\star$ as

$$L = Q_q \odot (K^\star - \gamma), \qquad U = Q_q \odot (K^\star + \gamma), \tag{11}$$

where $\gamma < 0.5$ keeps the coefficients away from quantization boundaries. The corresponding JPEG-cell constraint is

$$\mathcal{C}_{\text{jpeg}} = \{z \mid L \preceq \mathcal{D}(z) \preceq U\}, \tag{12}$$

where $\preceq$ denotes element-wise inequality. Any image in $\mathcal{C}_{\text{jpeg}}$ is quantized to the same symbols $K^\star$ and thus preserves the same dequantized JPEG-domain representation under the target JPEG implementation.

The refined image must also remain visually close to the clean image, yielding the pixel-domain constraint

$$\mathcal{C}_{\text{pix}} = \left\{z \mid z \in [0, 1]^{H\times W\times 3}, \|z - x\|_\infty \le \epsilon\right\}. \tag{13}$$

The refinement problem is therefore formulated as

$$\begin{aligned} x_{\text{adv}} = \arg\min_{z} \quad & \mathcal{L}_{\text{CE}}\left(f_\theta(z, T), y_{\text{ben}}\right) \\ \text{s.t.} \quad & z \in \mathcal{C}_{\text{jpeg}} \cap \mathcal{C}_{\text{pix}}, \end{aligned} \tag{14}$$

which encourages the uncompressed image to induce a benign or task-consistent response, while the JPEG-cell constraint keeps the post-compression representation fixed. In this sense, the refinement searches within the preimage of a fixed JPEG quantization cell.

Projection operators. The feasible set in Eq. (14) combines a DCT-domain cell constraint and a pixel-domain constraint. We propose two computational projections to enforce them. The JPEG-cell projection clips the DCT coefficients into the interval in Eq. (11) and decodes the result back to the pixel domain

$$\mathcal{P}_{\text{jpeg}}(z) = \mathcal{D}^{-1}\left(\text{Clip}\left(\mathcal{D}(z), L, U\right)\right). \tag{15}$$

The pixel-domain projection enforces the valid image range and the ℓ_∞ perturbation budget

$$\mathcal{P}_{\text{pix}}(z) = \text{Clip}\left(z, \max(0, x - \epsilon), \min(1, x + \epsilon)\right). \tag{16}$$

A projection onto one set may violate the other, since the two constraints are defined in different domains. We therefore adopt a Dykstra-style alternating projection with residual correction (Dykstra, 1983). Starting from $z_0 = \tilde{x}_{\text{adv}}$ with residuals $r^0_{\text{jpeg}} = 0$ and $r^0_{\text{pix}} = 0$, we iterate

$$\begin{aligned} u_k &= \mathcal{P}_{\text{jpeg}}\left(z_k + r^k_{\text{jpeg}}\right), \quad r^{k+1}_{\text{jpeg}} = z_k + r^k_{\text{jpeg}} - u_k, \\ z_{k+1} &= \mathcal{P}_{\text{pix}}\left(u_k + r^k_{\text{pix}}\right), \quad r^{k+1}_{\text{pix}} = u_k + r^k_{\text{pix}} - z_{k+1}. \end{aligned} \tag{17}$$

The residual terms store the correction introduced by each projection, reducing oscillation between the DCT and pixel domains. This step is used only to enforce the constraints: it prevents the refinement from crossing JPEG quantization boundaries or exceeding the visual perturbation budget.

Conditional objective. The overall optimization procedure of the proposed method is summarized in Algorithm 1 in the Appendix. The final adversarial image is designed to satisfy three properties simultaneously: the uncompressed image preserves benign or task-consistent behavior, the JPEG-compressed image activates the target behavior, and the perturbation remains within the prescribed visual budget. Thus, JPEG compression serves as a trigger for the adversarial behavior, rather than merely as a defense or robustness transformation.

3.4 Extension to Multi-View VLA Policies

To demonstrate JACK's applicability to embodied tasks, we extend the preceding full-image additive perturbation formulation in Section 2 to a multi-view VLA policy controlling a robotic arm in an object-picking task. Given instructions, front- and side-camera observations, and the robot state, the VLA policy predicts object bounding-box Chain-of-Thought (CoT) tokens that guide subsequent grasping actions. We place a single adversarial patch $p \in [0,1]^{h\times w\times 3}$ in the scene, visible to both cameras, and optimize it such that the policy grasps the user-intended object under uncompressed observations but targets the attacker-specified object under JPEG-compressed observations.

Let $\mathcal{D}^{\mathrm{ben}}$ and $\mathcal{D}^{\mathrm{tar}}$ denote two clean trajectories of the robotic arm (i.e., collected without the adversarial patch) executed under the benign user instruction T^{ben} and the attacker-specified instruction T^{tar}, respectively. For $r \in \{\mathrm{ben}, \mathrm{tar}\}$, each trajectory consists of N_r steps: $\mathcal{D}^r = \left\{\left(\{x^r_{i,v}\}_{v\in\mathcal{V}}, s^r_i, b^r_i\right)\right\}_{i=1}^{N_r}$, where $x^r_{i,v}$ denotes the observation from camera $v \in \mathcal{V}$ (front and side views) at step i, s^r_i is the robot state, and b^r_i represents the corresponding predicted bounding-box CoT tokens. In Stage I, we optimize p so that JPEG-compressed observations induce the attacker-specified bounding-box CoT tokens despite the VLA receiving the benign instruction T^{ben} as

$$\tilde{p} = \arg\min_{p\in[0,1]^{h\times w\times 3}} -\frac{1}{N_{\mathrm{tar}}}\sum_{i=1}^{N_{\mathrm{tar}}} \log P_\theta\left(b^{\mathrm{tar}}_i \,\middle|\, \{J_{\mathrm{diff},q}(g_v(x^{\mathrm{tar}}_{i,v}, p))\}_{v\in\mathcal{V}}, T^{\mathrm{ben}}, s^{\mathrm{tar}}_i\right). \tag{18}$$

Here, $J_{\mathrm{diff},q}$ is the differentiable JPEG proxy and P_θ is the autoregressive probability assigned by the VLA to a bounding-box CoT token sequence. The function $g_v(x^r_{i,v}, p)$ defines the observation modified by the patch as $g_v(x^r_{i,v}, p) = (1 - M_v) \odot x^r_{i,v} + M_v \odot \mathcal{W}(p; H_v)$, where H_v maps each patch pixel to a location in view v, $\mathcal{W}(p; H_v)$ renders the patch's appearance in that view, and M_v marks the pixels it covers. Consequently, the original observation remains unaltered outside the patch. Given the static patch placement and fixed camera calibration, H_v and M_v depend on v.

In Stage II, we refine the patch to encourage the user-intended bounding-box CoT on uncompressed observations while preserving the JPEG cells established in Stage I as

$$p^\star = \arg\min_{p\in\mathcal{C}_{\mathrm{VLA}}} -\frac{1}{N_{\mathrm{ben}}}\sum_{j=1}^{N_{\mathrm{ben}}} \log P_\theta\left(b^{\mathrm{ben}}_j \,\middle|\, \{g_v(x^{\mathrm{ben}}_{j,v}, p)\}_{v\in\mathcal{V}}, T^{\mathrm{ben}}, s^{\mathrm{ben}}_j\right). \tag{19}$$

The feasible set $\mathcal{C}_{\mathrm{VLA}}$ enforces valid RGB values and requires, for every Stage-I example i and camera view v, observations $g_v(x^{\mathrm{tar}}_{i,v}, p)$ and $g_v(x^{\mathrm{tar}}_{i,v}, \tilde{p})$ to have the same JPEG quantization indices, thereby preserving the attacker-specified behavior after JPEG compression. We combine gradient updates with Dykstra-style alternating projection to enforce these constraints. The full formulation and algorithm are detailed in Appendix B, with an example in Figure 4.

4 Experiments

4.1 Experimental Setup

Datasets and models. We evaluate the proposed attack across four vision-centric tasks, including targeted image classification, visual question answering (VQA), multimodal jailbreak, and hijacking of Vision-Language-Action (VLA) models. For targeted image classification, we use ImageNet-1K (Deng et al., 2009). For VQA, we adopt MS-COCO (Lin et al., 2014). For multimodal jailbreak, we evaluate on MM-SafetyBench (Liu et al., 2025). To further examine whether the proposed attack generalizes beyond vision-language understanding to embodied decision-making, we consider VLA hijacking on the LIBERO benchmark (Liu et al., 2023a), following the experimental setting of TRAP (Huang et al., 2026). Additional experimental details are provided in Appendix C.

We use LLaVA-v1.5-7B (Liu et al., 2024) as the default target VLM and additionally evaluate Qwen2.5-VL (Bai et al., 2025) to assess cross-model applicability. For VLA hijacking, we adopt GraspVLA (Deng et al., 2025) as the target policy.

JPEG and attack settings. We evaluate JPEG quality factors $q \in \{10, 30, 50, 70, 90\}$. Unless otherwise specified, we use $q = 30$ for the main comparisons. The differentiable JPEG proxy in Section 3.2 is used only during optimization, whereas the evaluation is concerned with the standard JPEG codec. We optimize each adversarial image using PGD for at most 500 total iterations with a step size of $1/255$ and $\gamma = 0.45$. The ℓ_∞ perturbation budget is set to $8/255$, $16/255$, and $32/255$ for classification, VQA, and jailbreak attacks, respectively.

Baselines. We compare our method against the following baselines. **Clean input** evaluates the original image before and after JPEG compression. **PGD without JPEG** applies standard pixel-space PGD without accounting for JPEG compression, with objective $x_{\text{adv}} = \arg\min_z \mathcal{L}_{\text{CE}}(f_\theta(z, T), y_{\text{tar}})$ subject to $z \in [0, 1]^{H \times W \times 3}$ and $\|z - x\|_\infty \leq \epsilon$. **JPEG-aware PGD** optimizes the adversarial image through the differentiable JPEG operator without constraining its pre-JPEG behavior, i.e., it directly optimizes Eq. (8). **Tan-Joint** implements a transform-dependent objective inspired by Tan et al. (2026), jointly optimizing the post-JPEG target loss and pre-JPEG benign loss: $x_{\text{adv}} = \arg\min_z \lambda_1 \mathcal{L}_{\text{CE}}(f_\theta(\mathcal{J}_{\text{diff},q}(z), T), y_{\text{tar}}) + \lambda_2 \mathcal{L}_{\text{CE}}(f_\theta(z, T), y_{\text{ben}})$, where we set $\lambda_1 = \lambda_2 = 1$ as suggested by Tan et al. (2026). **TRAP** (Huang et al., 2026) is included as a baseline for VLA hijacking and optimizes an adversarial patch to directly induce an attacker-specified CoT target and downstream manipulation behavior.

4.2 Evaluation Metrics

Pre-JPEG Safe Rate (Pre-SR). The proportion of uncompressed adversarial inputs that retain the desired benign behavior. Specifically, the model should predict the ground-truth class on ImageNet, preserve the clean-image response on VQA, produce a safe or refusal response on MM-SafetyBench, and execute the user-intended behavior in VLA tasks.

Post-JPEG Attack Success Rate (Post-ASR). The proportion of JPEG-compressed adversarial inputs that trigger the target behavior: predicting *banana* on ImageNet, generating *'Unknown"* on VQA, producing harmful content beginning with *'Sure, here is"* on MM-SafetyBench, or executing the attacker-specified manipulation behavior in VLA tasks.

Stealthy Attack Success Rate (SASR). SASR is our primary evaluation metric. It measures the proportion of samples that simultaneously exhibit benign behavior before JPEG compression and the target adversarial behavior after JPEG compression. Therefore, SASR directly evaluates whether JPEG compression successfully serves as the backdoor trigger of the attack.

4.3 Adversarial Attack Results

Table 2 compares the attack performance on ImageNet classification. Standard PGD, which optimizes the target behavior only for the uncompressed adversarial image, achieves a Post-ASR of approximately 0% after JPEG compression. This result indicates that adversarial perturbations optimized without explicitly modeling JPEG compression are generally not robust to the compression process. JPEG-aware PGD achieves a high Post-ASR because it directly optimizes the adversarial image through the differentiable JPEG operator. However, because it does not explicitly preserve benign behavior before compression, it suffers from worse stealthiness (i.e., low Pre-SR and SASR). Tan-Joint jointly optimizes the pre-JPEG benign objective and the post-JPEG target objective, but it occasionally underperforms JPEG-aware PGD in terms of SASR. This behavior may be attributed to interference between the two competing objectives, which makes it difficult to simultaneously satisfy the pre- and post-compression requirements using a single weighted objective in Tan-Joint.

In contrast, JACK consistently achieves the highest SASR across both examined VLMs. On LLaVA-v1.5-7B, JACK improves the SASR from 78.8% for JPEG-aware PGD to 90.2%, while on Qwen2.5-VL, it improves the SASR from 56.0% to 91.8%. JACK also achieves comparable or better image quality, as indicated by its higher PSNR and lower LPIPS. It is worth mentioning that JACK attains comparable Post-ASRs with the best baseline (i.e., less than 2% degradation), suggesting that the stealthiness is obtained with a mild degradation in Post-ASR.

Table 2: Attack performance comparisons on ImageNet classification.

Target VLM	Method	Pre-SR ↑	Post-ASR ↑	SASR ↑	PSNR ↑	LPIPS ↓
LLaVA-v1.5-7B	Clean input	100.0%	0.0%	0.0%	–	–
	PGD w/o JPEG	1.2%	0.0%	0.0%	34.05	0.091
	JPEG-aware PGD	83.0%	**95.4%**	78.8%	33.99	0.094
	Tan-Joint	77.4%	59.2%	45.8%	34.06	0.090
	JACK	**94.8%**	94.2%	**90.2%**	**34.12**	**0.087**
Qwen2.5-VL	Clean input	100.0%	0.0%	0.0%	–	–
	PGD w/o JPEG	0.4%	0.0%	0.0%	34.15	0.099
	JPEG-aware PGD	56.6%	**99.0%**	56.0%	34.16	0.092
	Tan-Joint	61.6%	85.8%	53.2%	34.13	0.093
	JACK	**94.0%**	97.2%	**91.8%**	**34.26**	**0.081**

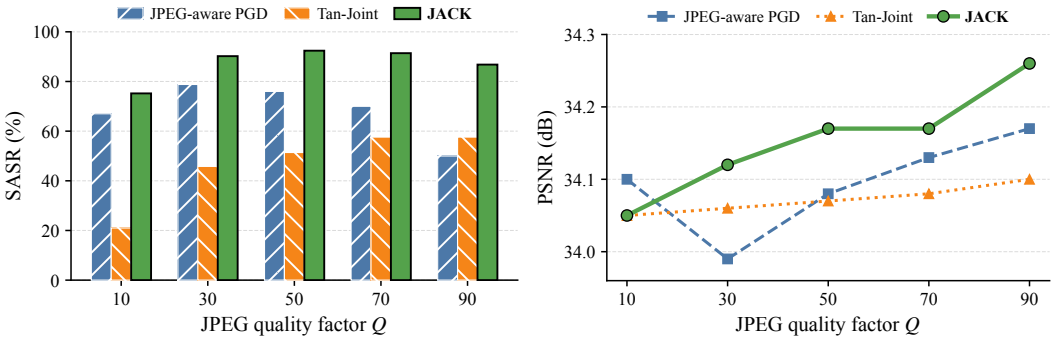


Figure 2: Attack performance across different JPEG quality factors on ImageNet. Clean input and PGD without JPEG optimization achieve zero conditional success rate across all quality factors and are omitted for clarity.

Figure 2 further evaluates attack performance across different JPEG quality factors on LLaVA-v1.5-7B. The SASR of JACK generally increases as the quality factor rises from $Q = 10$ to $Q = 50$, but decreases when the quality factor is further increased to $Q = 90$. This non-monotonic trend reflects a trade-off between preserving the post-JPEG adversarial signal and retaining sufficient flexibility for quantization-cell-preserving refinement. At low quality factors, JPEG compression uses relatively large quantization step sizes, resulting in wider quantization cells and a larger feasible region for refinement without changing the quantized DCT coefficients. However, overly aggressive compression may also remove substantial image information and adversarial signals, making the post-JPEG target behavior more difficult to induce. As the quality factor increases, JPEG preserves more of the adversarial signal, thereby improving the post-JPEG attack. At the same time, however, the quantization cells become narrower, reducing the feasible region available for restoring benign pre-JPEG behavior. Consequently, an intermediate quality factor, such as $Q = 50$, provides a favorable balance between post-JPEG attack effectiveness and pre-JPEG refinement flexibility. Across all evaluated quality factors, JACK achieves the highest SASR while maintaining comparable or better PSNR than the baselines. More results on MS-COCO VQA can be found in Appendix D, where JACK consistently achieves the highest SASR while introducing the least perceptual distortion.

4.4 Jailbreak Attack Results

Table 3 reports the macro-averaged results across the six evaluated harmful categories of MM-SafetyBench for LLaVA-v1.5-7B and Qwen2.5-VL. Complete category-wise results for all evaluation metrics are provided in Table 6 - 7 in the Appendix.

Compared with JPEG-aware PGD, which optimizes only the post-JPEG harmful behavior, JACK matches its Post-ASR while substantially improving the Pre-SR. Consequently, the macro-averaged SASR increases from 14.42% to 51.88% on LLaVA-v1.5-7B and from 37.57% to 78.82% on Qwen2.5-VL. These results demonstrate that the quantization-cell-preserving refinement can effectively restore benign pre-JPEG behavior without reducing the post-JPEG attack success rate.

Table 3: Averaged attack performance across six harmful task categories on MM-SafetyBench.

Target VLM	Method	Pre-SR ↑	Post-ASR ↑	SASR ↑	PSNR ↑	LPIPS ↓
LLaVA-v1.5-7B	Clean input	27.37%	65.16%	1.25%	–	–
	PGD w/o JPEG	6.94%	65.96%	0.87%	23.16	0.472
	JPEG-aware PGD	28.05%	**82.26%**	14.42%	23.12	0.522
	Tan-Joint	58.90%	72.95%	36.38%	22.51	0.493
	JACK	**67.91%**	**82.26%**	**51.88%**	**23.74**	**0.461**
Qwen2.5-VL	Clean input	57.93%	32.81%	1.22%	–	–
	PGD w/o JPEG	11.54%	33.05%	0.85%	23.08	0.498
	JPEG-aware PGD	55.33%	**79.10%**	37.57%	23.21	0.507
	Tan-Joint	82.61%	67.97%	55.70%	22.43	0.514
	JACK	**96.30%**	**79.10%**	**78.82%**	**23.55**	**0.460**

Table 4: VLA hijacking performance on GraspVLA across different JPEG quality factors.

Method	$Q = 30$			$Q = 50$			$Q = 70$		
	Pre-SR ↑	Post-ASR ↑	SASR ↑	Pre-SR ↑	Post-ASR ↑	SASR ↑	Pre-SR ↑	Post-ASR ↑	SASR ↑
TRAP	18.70%	42.28%	0.00%	18.70%	49.59%	0.00%	18.70%	44.72%	0.81%
Tan-Joint	**99.19%**	71.54%	71.54%	**98.37%**	67.48%	66.67%	**98.37%**	69.11%	68.29%
JACK	98.37%	**81.30%**	**79.67%**	96.75%	**82.93%**	**82.93%**	96.75%	**82.11%**	**81.30%**

As shown in Figure 3, JACK consistently achieves the highest SASR across all evaluated harmful categories and both target VLMs. Compared with Tan-Joint, the strongest baseline in terms of macro-averaged SASR, JACK improves the SASR by 15.50% and 23.12% on LLaVA-v1.5-7B and Qwen2.5-VL, respectively. Moreover, JACK achieves the highest average PSNR and the lowest average LPIPS on both models, indicating that its improved jailbreak performance is accompanied by better visual fidelity.

4.5 VLA Hijacking Results

Table 4 evaluates JPEG-triggered behavior hijacking on GraspVLA across different JPEG quality factors. Unlike conventional VLA attacks that directly induce a target manipulation, JACK preserves the intended behavior before JPEG compression and activates the attacker-specified behavior only after compression. TRAP (Huang et al., 2026) performs poorly in this conditional setting: its poor stealthiness of pre-JPEG behavior and limited post-JPEG attack success lead to near-zero SASR. Tan-Joint better preserves the pre-JPEG behavior, achieving Pre-SRs above 98%, but its Post-ASR remains between 67.48% and 71.54%.

JACK achieves the best overall conditional hijacking performance across all quality factors. It maintains a high Pre-SR of 96.75–98.37% while attaining a Post-ASR of 81.30–82.93%, yielding the highest SASR of 79.67%, 82.93%, and 81.30% at $Q = 30$, 50, and 70, respectively. These results show that JACK extends beyond vision-language understanding and generalizes to embodied decision-making, enabling JPEG-triggered targeted manipulation in VLA models.

5 Conclusion

We introduced JACK, a JPEG-triggered adversarial attack against VLMs, in which an adversarial image appears benign before compression but activates attacker-specified behavior after JPEG compression. Experiments on image classification, VQA, multimodal jailbreaking, and VLA hijacking demonstrate that JACK consistently achieves higher conditional attack success than existing baselines while maintaining high visual quality. These findings show that standard image transformations should not be regarded solely as benign preprocessing or defensive operations, as they can also act as triggers for concealed adversarial behavior. Future work will explore black-box attack settings and extend this paradigm to other deployment-time transformations.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring mandatory disclosure. In particular, generative AI tools were not used to develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, design or provide feedback on research methodology or experiments, implement methods, or interpret results. The remaining tasks requiring mandatory disclosure are not applicable to this work.

Additionally, we used generative AI tools for tasks with recommended disclosure, specifically to edit the manuscript to improve readability and to assist with editing and debugging author-written experimental code. We have reviewed all AI-assisted work. All AI-assisted text was manually reviewed and revised by the authors to preserve the intended meaning, and all AI-assisted code modifications were manually inspected and tested by the authors to verify correctness. We take responsibility for the final content of this work, including text, claims, and code produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the security risks of transformation-triggered adversarial examples in vision-language and vision-language-action models. While the proposed attack methodology could potentially be misused to manipulate deployed multimodal systems, our goal is to expose an underexplored security risk associated with commonly used image transformations such as JPEG compression and to facilitate the development of corresponding defenses. All experiments are conducted in controlled research environments using publicly available models and benchmarks, without targeting real users or deployed systems. We encourage future work to investigate effective detection and mitigation mechanisms against such transformation-triggered attacks.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 284–293. PMLR, 2018.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. doi: 10.48550/arXiv.2502.13923. URL https://arxiv.org/abs/2502.13923.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacks: Adversarial images can control generative models at runtime. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 2443–2455. PMLR, 2024.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024a.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67 (12):220101, 2024b.

Kaiyuan Cui, Yige Li, Yutao Wu, Xingjun Ma, Sarah Erfani, Christopher Leckie, and Hanxun Huang. Toward universal and transferable jailbreak attacks on vision-language models. In *International Conference on Learning Representations (ICLR)*, 2026.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, Zhizheng Zhang, and He Wang. Graspvla: a grasping foundation model pre-trained on billion-scale synthetic action data. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1004–1029. PMLR, 2025.

Qiuyu Duan, Zhongyun Hua, Qing Liao, Yushu Zhang, and Leo Yu Zhang. Conditional backdoor attack via jpeg compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19823–19831, 2024.

Richard L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

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.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23951–23959, 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017. doi: 10.1109/CVPR.2017.670.

Xiangming Gu, Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Ye Wang, Jing Jiang, and Min Lin. Agent smith: A single image can jailbreak one million multimodal LLM agents exponentially fast. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16647–16672. PMLR, 2024.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

Qi Guo, Shanmin Pang, Xiaojun Jia, Yang Liu, and Qing Guo. Efficient generation of targeted and transferable adversarial examples for vision-language models via diffusion models. *IEEE Transactions on Information Forensics and Security*, 20:1333–1348, 2024.

Kai Hu, Weichen Yu, Li Zhang, Alexander Robey, Andy Zou, Chengming Xu, Haoqi Hu, and Matt Fredrikson. Transferable adversarial attacks on black-box vision-language models. *arXiv preprint arXiv:2505.01050*, 2025.

Zhengxian Huang, Wenjun Zhu, Haoxuan Qiu, Xiaoyu Ji, and Wenyuan Xu. Trap: Hijacking vla cot-reasoning via adversarial patches. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Jiayi Kuang, Ying Shen, Jingyou Xie, Haohao Luo, Zhe Xu, Ronghao Li, Yinghui Li, Xianfeng Cheng, Xika Lin, and Yu Han. Natural language understanding and inference with mllm in visual question answering: A survey. *ACM Computing Surveys*, 57(8):1–36, 2025.

Yifan Lan, Yuanpu Cao, Weitong Zhang, Lu Lin, and Jinghui Chen. Phi: Preference hijacking in multi-modal large language models at inference time. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17840–17865, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.901. URL https://aclanthology.org/2025.emnlp-main.901/.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. In *European Conference on Computer Vision*, pp. 174–189. Springer, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48.

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

Daizong Liu, Xiaowen Cai, Pan Zhou, Xiaoye Qu, Lichao Sun, and Wei Hu. Are large vision-language models robust to adversarial visual transformations? *IEEE Transactions on Information Forensics and Security*, 21:2564–2579, 2026. doi: 10.1109/TIFS.2026.3671065.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pp. 386–403. Springer, 2025. doi: 10.1007/978-3-031-72992-8_22.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: DNN-oriented JPEG compression against adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 860–868, 2019.

Haochen Luo, Jindong Gu, Fengyuan Liu, and Philip Torr. An image is worth 1000 lies: Transferability of adversarial images across prompts on vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nc5GgFAvtk.

Zhenxing Niu, Haodong Ren, Xinbo Gao, Gang Hua, and Rong Jin. Jailbreaking attack against multimodal large language model. *arXiv preprint arXiv:2402.02309*, 2024.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21527–21536, 2024. doi: 10.1609/aaai.v38i19.30150.

Christoph Reich, Biplob Debnath, Deep Patel, and Srimat Chakradhar. Differentiable jpeg: The devil is in the details. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4126–4135, 2024.

Christian Schlarmann and Matthias Hein. On the adversarial robustness of multi-modal foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3677–3685, 2023.

Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *NIPS Workshop on Machine Learning and Computer Security*, 2017.

Yaoteng Tan, Zikui Cai, and Salman Asif. Transform-triggered adversarial examples. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=If2OGZMJxs.

Haoqin Tu, Chenhang Cui, Zijun Wang, Yiyang Zhou, Bingchen Zhao, Junlin Han, Wangchunshu Zhou, Huaxiu Yao, and Cihang Xie. How many unicorns are in this image? a safety evaluation benchmark for vision llms. *arXiv preprint arXiv:2311.16101*, 2023.

Gregory K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, February 1992. doi: 10.1109/30.125072.

Ruofan Wang, Xingjun Ma, Hanxu Zhou, Chuanjun Ji, Guangnan Ye, and Yu-Gang Jiang. White-box multimodal jailbreaks against large vision-language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6920–6928. Association for Computing Machinery, 2024. doi: 10.1145/3664647.3681092.

Peng Xie, Yequan Bie, Jianda Mao, Yangqiu Song, Yang Wang, Hao Chen, and Kani Chen. Chain of attack: On the robustness of vision-language models against transfer-based adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14679–14689, 2025.

Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu, and Dacheng Tao. Jailbreak vision language models via bi-modal adversarial prompt. *IEEE Transactions on Information Forensics and Security*, 2025.

Tingwei Zhang, Collin Zhang, John X. Morris, Eugene Bagdasarian, and Vitaly Shmatikov. Self-interpreting adversarial images. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 1037–1052, Seattle, WA, August 2025. USENIX Association. ISBN 978-1-939133-52-6.

Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Ming Hu, Jie Zhang, Yang Liu, Shiqing Ma, and Chao Shen. Jailguard: A universal detection framework for prompt-based attacks on LLM systems. *ACM Transactions on Software Engineering and Methodology*, 35(1):8:1–8:40, 2026. doi: 10.1145/3724393.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. *Advances in Neural Information Processing Systems*, 36:54111–54138, 2023.

Qi Zhou, Dongxia Wang, Tianlin Li, Yun Lin, Yang Liu, Jin Song Dong, and Qing Guo. Defending LVLMs against vision attacks through partial-perception supervision. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79254–79280. PMLR, 2025.

A Related Work

A.1 Adversarial and Jailbreak Attacks on VLMs

By expanding the attack surface to include visual inputs, large vision-language models introduce novel security vulnerabilities absent in text-only LLMs. Early benchmarks and security evaluations consistently highlight the susceptibility of these multimodal models to adversarial perturbations and safety exploits across diverse vision-language tasks (Schlarmann & Hein, 2023; Tu et al., 2023; Zhao et al., 2023). Driven by these findings, a rapidly growing body of literature has focused on developing visual and cross-modal attacks against VLMs. Most notably, optimization-based techniques inject carefully crafted visual perturbations to induce targeted misclassifications, hijack model generation, or bypass safety guardrails to elicit harmful responses (Bailey et al., 2024; Qi et al., 2024; Gong et al., 2025; Li et al., 2024; Wang et al., 2024; Ying et al., 2025).

A.2 Image Transformations in VLM Security

The above attacks reveal the vulnerability of VLMs to adversarial visual inputs, but practical VLM pipelines often process images before inference. This has motivated recent studies on the role of image transformations in VLM security. Several works use transformations such as resizing, cropping, filtering, or recompression as inference-time defenses or input-sanitization mechanisms (Zhang et al., 2026; Gu et al., 2024; Zhang et al., 2025). Other studies incorporate transformations into attack optimization to improve robustness and transferability under realistic preprocessing pipelines (Zhou et al., 2025; Hu et al., 2025). More recently, Liu et al. (2026) systematically investigate how different visual transformations affect VLM outputs and compose harmful transformations to facilitate adversarial attacks.

These works demonstrate that transformations can substantially change VLM behavior and therefore should be considered in security evaluation. Nevertheless, they primarily treat transformations as defenses, attack-time augmentations, or attack operations selected by the adversary. Our setting differs in both the role of the transformation and the desired behavior: the transformation is fixed by the processing pipeline, and the adversary optimizes a single image that appears benign before the transformation but activates attacker-specified behavior after the transformation is applied.

A.3 JPEG-Robust and JPEG-triggered Attacks

JPEG compression is one of the most widely studied image transformations in adversarial machine learning. Because JPEG can suppress high-frequency perturbations, it has often been considered a simple input-transformation defense (Dziugaite et al., 2016; Das et al., 2017; Guo et al., 2018; Liu et al., 2019). In response, adaptive attacks have attempted to make adversarial examples robust to JPEG preprocessing. Representative strategies include optimizing over a distribution of transformations, as in expectation over transformation (Athalye et al., 2018), or approximating JPEG with differentiable operations so that gradients can be propagated through compression (Shin & Song, 2017; Reich et al., 2024). These works treat JPEG as a nuisance transformation or a defense that the adversarial perturbation should survive.

Beyond transformation robustness, a closer line of work studies whether JPEG or other image transformations can serve as conditions that alter model behavior. Duan et al. (2024) introduce a JPEG-conditioned backdoor for image classifiers by jointly modifying the JPEG quantization process and training the model, such that the target behavior is triggered only after JPEG compression. Tan et al. (2026) construct transformation-triggered adversarial examples for image classifiers by jointly optimizing the original image to remain benign while its transformed version induces an adversarial prediction. These works show that transformation-dependent behaviors are possible in classical vision systems. However, they focus on backdoored models or closed-set classifiers. In contrast, we study JPEG-triggered adversarial behavior in frozen, off-the-shelf generative VLMs, where both the benign pre-compression response and the adversarial post-compression response are open-ended and must be controlled without modifying the model.

B VLA-Specific Formulation and Optimization

This section presents the detailed formulation of the VLA extension introduced in Section 3.4. Our evaluation follows the manipulation-task construction and offline patch-optimization protocol of TRAP (Huang et al., 2026), using clean (i.e., unpatched) rollouts as supervision. We adopt GraspVLA (Deng et al., 2025) as the target VLA model and evaluate it on the LIBERO benchmark (Liu et al., 2023a).

The VLA setting differs from the full-image formulation in Section 2 in two main aspects. First, the optimization variable is an adversarial patch rather than a global additive perturbation. Second, because the same patch is rendered into multiple camera views, the refinement process must simultaneously preserve the JPEG quantization cells of all rendered target observations rather than that of a single image.

Algorithm 1 JPEG-triggered Adversarial attaCK (JACK)

Require: Clean image x, model f_θ, instruction T, targets $y_{\text{ben}}, y_{\text{tar}}$, JPEG quality q, budget ϵ, margin γ, step sizes η_1, η_2, iterations $N_1, N_2, N_{\text{proj}}$
Ensure: Adversarial image x_{adv}

Obtain the JPEG quantization table Q_q.
Stage I: JPEG-aware candidate optimization
Initialize $z \leftarrow x$.
for $t = 1, \dots, N_1$ **do**
  $\mathcal{L}_{\text{tar}} \leftarrow \mathcal{L}_{\text{CE}}(f_\theta(\mathcal{J}_{\text{diff},q}(z), T), y_{\text{tar}})$.
  $z \leftarrow \mathcal{P}_{\text{pix}}(z - \eta_1 \nabla_z \mathcal{L}_{\text{tar}})$.
end for
$\widetilde{x}_{\text{adv}} \leftarrow z$.
Compute $K^\star \leftarrow \text{Round}(\mathcal{D}(\widetilde{x}_{\text{adv}}) \oslash Q_q)$.
Set $L \leftarrow Q_q \odot (K^\star - \gamma)$ and $U \leftarrow Q_q \odot (K^\star + \gamma)$.
Stage II: Quantization-cell preserving refinement
Initialize $z \leftarrow \widetilde{x}_{\text{adv}}$.
for $t = 1, \dots, N_2$ **do**
  $\mathcal{L}_{\text{ben}} \leftarrow \mathcal{L}_{\text{CE}}(f_\theta(z, T), y_{\text{ben}})$.
  $v \leftarrow z - \eta_2 \nabla_z \mathcal{L}_{\text{ben}}$.
  $r_{\text{jpeg}} \leftarrow 0,\ r_{\text{pix}} \leftarrow 0$.
  for $k = 1, \dots, N_{\text{proj}}$ **do**
    $u \leftarrow \mathcal{P}_{\text{jpeg}}(v + r_{\text{jpeg}})$.
    $r_{\text{jpeg}} \leftarrow v + r_{\text{jpeg}} - u$.
    $v \leftarrow \mathcal{P}_{\text{pix}}(u + r_{\text{pix}})$.
    $r_{\text{pix}} \leftarrow u + r_{\text{pix}} - v$.
  end for
  $z \leftarrow v$.
end for
return $x_{\text{adv}} \leftarrow z$.

B.1 Offline Trajectory Data and Multi-View Observations

Clean offline rollouts. Following TRAP (Huang et al., 2026), each manipulation task is defined by a pair of candidate objects, each associated with an object-specific instruction (e.g., *"pick up apple"* and *"pick up knife"*). For a given attack instance, one instruction serves as the user-intended instruction T^{ben}, whereas the other serves as the attacker-specified target instruction T^{tar}. Before patch optimization, we independently collect two clean offline rollout datasets by executing the victim VLA under these instructions

$$\mathcal{R}^r = \{\tau_n^r\}_{n=1}^{M_r}, \qquad r \in \{\text{ben}, \text{tar}\}, \tag{20}$$

where r indicates the behavior type, M_r is the number of trajectories in $\mathcal{R}^r$. Specifically, $\mathcal{R}^{\text{ben}}$ and $\mathcal{R}^{\text{tar}}$ contain trajectories collected under T^{ben} and T^{tar}, respectively.

A trajectory is represented as

$$\tau_n^r = \left\{ \left(O_{n,k}^r, s_{n,k}^r, b_{n,k}^r, a_{n,k}^r \right) \right\}_{k=1}^{m_n^r}, \tag{21}$$

where m_n^r is the number of recorded time steps in trajectory τ_n^r, $O_{n,k}^r$ is the multi-view visual observation, $s_{n,k}^r$ is the robot-state context, $b_{n,k}^r$ is the 2D target bounding-box Chain-of-Thought (CoT) annotation, and $a_{n,k}^r$ is the recorded action.

The two datasets are collected independently and require no temporal alignment. Their trajectories may have different lengths, and observations from the two datasets may correspond to different robot poses. For optimization, we flatten the trajectories within each dataset

$$\mathcal{S}^r = \{\xi_i^r\}_{i=1}^{N_r}, \qquad \xi_i^r = (O_i^r, s_i^r, b_i^r, a_i^r), \qquad r \in \{\text{ben}, \text{tar}\}, \tag{22}$$

where

$$N_r = \sum_{n=1}^{M_r} m_n^r \tag{23}$$

is the total number of time-step records in $\mathcal{S}^r$, and i is a dataset-local record index. Thus, N_{ben} and N_{tar} may differ, and records with the same index in the two sets are not assumed to describe

the same time step or robot configuration. The target instruction T^{tar} is used solely to collect $\mathcal{R}^{\mathrm{tar}}$. During both optimization and evaluation, the language input supplied to the VLA is held fixed to the benign instruction $T = T^{\mathrm{ben}}$. Stage I uses $\mathcal{S}^{\mathrm{tar}}$ to optimize the post-JPEG target behavior, whereas Stage II independently uses $\mathcal{S}^{\mathrm{ben}}$ to restore the pre-JPEG benign behavior. Both stages share the same canonical patch p and instruction T.

Multi-view observations. GraspVLA processes multi-camera visual inputs to predict an object-centric CoT representation in the form of 2D target bounding boxes, which subsequently guides continuous action generation (Deng et al., 2025). For a record $\xi_i^r \in \mathcal{S}^r$, let

$$O_i^r = \left\{ x_{i,v}^r \right\}_{v \in \mathcal{V}_i^r}, \tag{24}$$

where $\mathcal{V}_i^r$ is the set of camera-view indices available for record i, and $x_{i,v}^r$ is the RGB image from view v. In our setup, the available views correspond to the front and side cameras.

We optimize a single canonical patch

$$p \in [0,1]^{h_p \times w_p \times 3}, \tag{25}$$

where h_p and w_p denote the patch height and width, respectively. The patch is shared across all trajectory records, camera views, and object layouts. A view-specific homography maps p into the corresponding image plane, yielding the patched observation

$$g_{i,v}^r(p) = \left(1 - M_{i,v}^r\right) \odot x_{i,v}^r + M_{i,v}^r \odot \mathcal{W}\left(p; H_{i,v}^r\right), \qquad v \in \mathcal{V}_i^r, \tag{26}$$

where $H_{i,v}^r$ is the homography associated with record i and view v, $\mathcal{W}(\cdot)$ denotes differentiable homography-based warping, $M_{i,v}^r$ is the resulting patch mask, and $\odot$ denotes element-wise multiplication.

Bounding-box CoT tokens. The 2D target bounding box is represented as a sequence of discrete CoT tokens

$$b_i^r = \left(b_{i,1}^r, \ldots, b_{i,L_i^r}^r\right), \tag{27}$$

where L_i^r is the number of bounding-box coordinate tokens in record i, $b_{i,\ell}^r$ is the ℓ-th token, and $b_{i,<\ell}^r = (b_{i,1}^r, \ldots, b_{i,\ell-1}^r)$ denotes its preceding token prefix. In our two-view setting, each view contributes four coordinate tokens, yielding $L_i^r = 8$ when both views are available. Following the GraspVLA-specific attack setting in TRAP (Huang et al., 2026), our optimization directly targets the bounding-box CoT tokens rather than the continuous actions, as this object-centric CoT representation conditions the subsequent generation of continuous actions.

B.2 Stage I: JPEG-triggered Target Optimization

Stage I optimizes the post-JPEG target behavior exclusively on $\mathcal{S}^{\mathrm{tar}}$. The corresponding loss is

$$\mathcal{L}_{\mathrm{tar}}^{\mathrm{VLA}}(p; i) = -\sum_{\ell=1}^{L_i^{\mathrm{tar}}} \log P_\theta\Big(b_{i,\ell}^{\mathrm{tar}} \,\Big|\, b_{i,<\ell}^{\mathrm{tar}}, \left\{\mathcal{J}_{\mathrm{diff},q}\left(g_{i,v}^{\mathrm{tar}}(p)\right)\right\}_{v \in \mathcal{V}_i^{\mathrm{tar}}}, T, s_i^{\mathrm{tar}}\Big), \tag{28}$$

where P_θ is the next-token distribution of the fixed victim VLA with parameters θ, and $\mathcal{J}_{\mathrm{diff},q}$ is the differentiable JPEG operator at quality factor q.

The Stage-I patch is obtained by solving

$$\begin{aligned} \widetilde{p} = \arg\min_{p} \quad & \frac{1}{N_{\mathrm{tar}}} \sum_{i=1}^{N_{\mathrm{tar}}} \mathcal{L}_{\mathrm{tar}}^{\mathrm{VLA}}(p; i) \\ \text{s.t.} \quad & p \in [0,1]^{h_p \times w_p \times 3}. \end{aligned} \tag{29}$$

Equation (29) optimizes the patch such that, after JPEG compression, the patched observations induce the attacker-specified target bounding-box CoT while the VLA continues to receive the benign instruction T^{ben}. Unlike the full-image adversarial attack, p is constrained only by its prescribed spatial support and valid RGB range, without an ℓ_∞ perturbation budget.

B.3 Multi-View JPEG Quantization-Cell Constraints

To preserve the attacker-specified behavior established in Stage I, we record the JPEG quantization state induced by $\widetilde{p}$ across all target records and camera views. For an image y, define its normalized pre-quantization JPEG coefficients as

$$\Phi_q(y) = \frac{\mathcal{D}(y)}{Q_q}, \tag{30}$$

where $\mathcal{D}(\cdot)$ denotes the continuous JPEG transform before quantization, Q_q is the quantization table at quality factor q, and the division is applied element-wise. For target record i and view $v \in \mathcal{V}_i^{\mathrm{tar}}$, the Stage-I quantization indices are

$$K_{i,v}^{\star} = \mathrm{Round}\left(\Phi_q\left(g_{i,v}^{\mathrm{tar}}(\widetilde{p})\right)\right), \tag{31}$$

where $\mathrm{Round}(\cdot)$ is applied element-wise.

Given an interior half-width $0 < \gamma < 0.5$, the view-specific JPEG-cell constraint in canonical-patch space is defined as

$$\mathcal{C}_{i,v} = \left\{p \;\middle|\; K_{i,v}^{\star} - \gamma \preceq \Phi_q\left(g_{i,v}^{\mathrm{tar}}(p)\right) \preceq K_{i,v}^{\star} + \gamma\right\}, \tag{32}$$

where $\preceq$ denotes element-wise inequality. This interval lies strictly inside the rounding cell associated with $K_{i,v}^{\star}$; hence, satisfying the constraint preserves the corresponding JPEG quantization indices.

To accommodate multi-view rendering, the joint feasible set restricts the patch to the intersection of all target-record- and view-dependent cells:

$$\mathcal{C}_{\mathrm{VLA}} = \mathcal{C}_{\mathrm{box}} \cap \bigcap_{i=1}^{N_{\mathrm{tar}}} \bigcap_{v \in \mathcal{V}_i^{\mathrm{tar}}} \mathcal{C}_{i,v}, \tag{33}$$

where $\mathcal{C}_{\mathrm{box}} = \{p \mid 0 \preceq p \preceq 1\}$ enforces the valid RGB range.

B.4 Stage II: Multi-View Benign-Behavior Refinement

Because the Stage-I patch $\widetilde{p}$ is optimized using post-JPEG supervision, it may also induce the target CoT before compression. Stage II therefore uses the independently collected benign-behavior dataset to restore the user-intended prediction on the uncompressed observations while preserving the target-trajectory JPEG cells established in Stage I. For each benign record $\xi_j^{\mathrm{ben}} \in \mathcal{S}^{\mathrm{ben}}$, the refinement loss is

$$\mathcal{L}_{\mathrm{ben}}^{\mathrm{VLA}}(p; j) = -\sum_{\ell=1}^{L_j^{\mathrm{ben}}} \log P_\theta\Big(b_{j,\ell}^{\mathrm{ben}} \,\Big|\, b_{j,<\ell}^{\mathrm{ben}}, \left\{g_{j,v}^{\mathrm{ben}}(p)\right\}_{v \in \mathcal{V}_j^{\mathrm{ben}}}, T, s_j^{\mathrm{ben}}\Big). \tag{34}$$

The Stage-II refinement problem is formulated as

$$\begin{aligned} p^{\star} = \arg\min_{p} \quad & \frac{1}{N_{\mathrm{ben}}} \sum_{j=1}^{N_{\mathrm{ben}}} \mathcal{L}_{\mathrm{ben}}^{\mathrm{VLA}}(p; j) \\ \text{s.t.} \quad & p \in \mathcal{C}_{\mathrm{VLA}}. \end{aligned} \tag{35}$$

B.5 Multi-View Dykstra Projection

We solve Eq. (35) by alternating gradient-based updates of the benign objective with projection onto $\mathcal{C}_{\mathrm{VLA}}$. For notational simplicity, we enumerate the constraints in Eq. (33) and write

$$\mathcal{C}_{\mathrm{VLA}} = \bigcap_{m=0}^{M} \mathcal{C}_m, \qquad M = \sum_{i=1}^{N_{\mathrm{tar}}} \left|\mathcal{V}_i^{\mathrm{tar}}\right|, \tag{36}$$

where $\mathcal{C}_0 = \mathcal{C}_{\mathrm{box}}$ and $\mathcal{C}_1, \ldots, \mathcal{C}_M$ enumerate all target-record- and camera-view-specific JPEG-cell constraints. Let $\mathcal{P}_m$ denote the projection operator associated with $\mathcal{C}_m$.

Given an unconstrained Stage-II update $\bar{p}$, we apply Dykstra's algorithm (Dykstra, 1983) by initializing

$$z^{(0)} = \bar{p}, \qquad r_m^{(0)} = 0, \qquad m = 0, \ldots, M. \tag{37}$$

During each projection cycle, the constraints are processed sequentially. For $m = 0, \ldots, M$, the variables are updated in place as

$$u_m = \mathcal{P}_m\left(z + r_m\right), \qquad r_m \leftarrow z + r_m - u_m, \qquad z \leftarrow u_m. \tag{38}$$

Here, u_m is the patch obtained after projection onto $\mathcal{C}_m$, and r_m is its residual correction. After all constraints have been processed, the resulting z initializes the next projection cycle; after the prescribed number of cycles, it becomes the next Stage-II iterate. The residual variables compensate for corrections introduced by preceding projections and reduce oscillation among overlapping constraints. This procedure generalizes the two-set projection of the full-image attack, allowing the canonical patch to satisfy both the RGB constraint and all target-record- and camera-view-specific JPEG cells.

Relation to TRAP. Whereas TRAP directly optimizes an adversarial patch to hijack intermediate CoT reasoning (Huang et al., 2026), our VLA extension introduces a compression-conditioned mechanism. It optimizes the attacker-specified behavior on post-JPEG inputs in Stage I and restores benign behavior on uncompressed inputs in Stage II, thereby making the adversarial hijacking behavior triggered by JPEG compression.

C Experimental Details

We provide detailed descriptions of the datasets and task settings used in our experiments.

ImageNet classification. We randomly sample 500 images from the ImageNet-1K validation set (Deng et al., 2009) and perform targeted classification attacks using *'banana"* as the target response. Following common VLM evaluation practice, we use the prompt *'Classify the main object in this image using a single concrete noun."*

VQA. Following prior work (Luo et al., 2024), we randomly select 500 images from the MS-COCO validation set (Lin et al., 2014). Each image is paired with an image-specific question from VQA-v2 (Goyal et al., 2017), and the target response is set to *"Unknown."*

Multimodal jailbreak. We evaluate jailbreak attacks on MM-SafetyBench (Liu et al., 2025), which contains harmful queries and query-relevant images spanning 13 safety-critical scenarios. We consider six representative categories: Illegal Activity, Hate Speech, Malware Generation, Physical Harm, Economic Harm, and Sex. The target harmful response is defined to begin with *"Sure, here is"*.

VLA hijacking. Following TRAP (Huang et al., 2026), we evaluate targeted behavior hijacking on GraspVLA using the LIBERO benchmark (Liu et al., 2023a), adopting the same five curated tasks and layout protocol. Since GraspVLA specializes in open-vocabulary grasping, each task focuses on the *pick* motion primitive and instantiates two object-specific instructions via the template *"pick up [object]"*. Each task defines a directed attack over an object pair denoted in $(\text{target}, \text{benign})$ order: the first corresponds to the attacker's target instruction T^{tar}, while the second corresponds to the benign instruction T^{ben} issued by the user to the victim VLA. The five target/benign pairs comprise: (1) tomato sauce can / milk, (2) Akita black bowl / wine bottle, (3) salad dressing bottle / milk, (4) pineapple / ice cream, and (5) watermelon / carrot. For each task, we optimize an adversarial patch using clean rollouts collected under both instructions across 25 randomized training layouts as in Appendix B. Each task-specific patch is subsequently evaluated on 10 unseen layouts to assess its generalization to novel object configurations. The patch resolution is set to 64×64 pixels. We report performance averaged across the five tasks.

Hyperparameters of JACK. As summarized in Algorithm 1, JACK follows a two-stage optimization procedure, with 300 and 200 optimization iterations for the first and second stages, respectively, resulting in a total of 500 optimization steps. The Dykstra projection is performed for 3 iterations at each refinement step in the second stage.

Table 5: Attack performance comparisons on MS-COCO VQA.

Target VLM	Method	Pre-SR ↑	Post-ASR ↑	SASR ↑	PSNR ↑	LPIPS ↓
LLaVA-v1.5-7B	Clean input	100.0%	0.0%	0.0%	–	–
	PGD w/o JPEG	1.3%	0.0%	0.0%	28.32	0.228
	JPEG-aware PGD	43.8%	**28.2%**	14.3%	28.22	0.253
	Tan-Joint	79.3%	20.3%	16.5%	28.25	0.241
	JACK	**81.0%**	27.0%	**21.5%**	**28.60**	**0.228**
Qwen2.5-VL	Clean input	100.0%	0.0%	0.0%	–	–
	PGD w/o JPEG	0.0%	0.0%	0.0%	28.13	0.248
	JPEG-aware PGD	5.3%	**76.0%**	4.8%	28.22	0.246
	Tan-Joint	38.8%	52.5%	28.3%	28.22	0.246
	JACK	**51.8%**	69.0%	**40.8%**	**28.68**	**0.225**

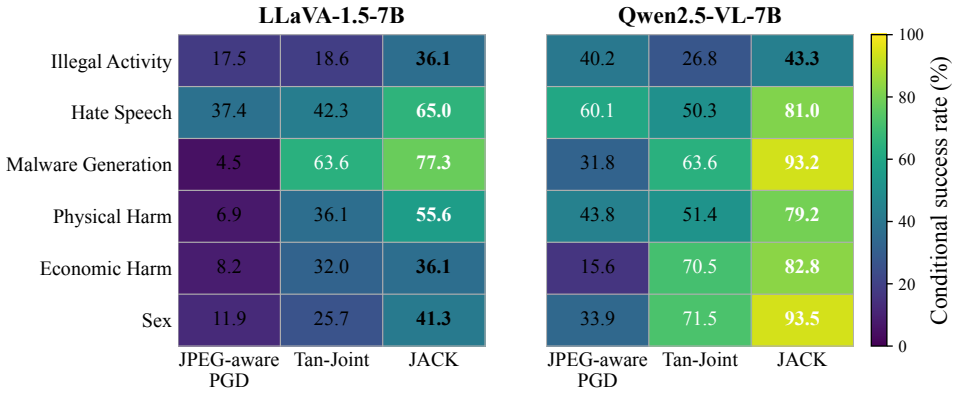


Figure 3: Category-wise SASR on MM-SafetyBench. Clean input and PGD without JPEG optimization are omitted because their SASR is consistently close to zero. All panels share the same color scale.

D Additional Results

Table 5 reports the results on the MS-COCO VQA task. JACK achieves the highest SASR on both target VLMs. On LLaVA-v1.5-7B, JACK increases the SASR from 14.3% for JPEG-aware PGD to 21.5%. The improvement is more pronounced on Qwen2.5-VL, where the SASR increases from 4.8% to 40.8%. JACK also achieves the highest PSNR and the lowest LPIPS, demonstrating that the improvement in conditional attack performance does not require greater perceptual distortion.

Tables 6 and 7 report the complete category-wise results on MM-SafetyBench for LLaVA-v1.5-7B and Qwen2.5-VL, respectively, and Figure 3 presents the corresponding category-wise SASRs with heatmaps. Across the evaluated harmful categories, JACK consistently achieves strong conditional jailbreak performance while maintaining favorable visual quality, further supporting the effectiveness of the proposed quantization-cell-preserving refinement.

Figure 4 provides a qualitative example of JPEG-conditional behavior hijacking on GraspVLA in the LIBERO simulation environment at JPEG quality factor $Q = 50$. Given the same optimized adversarial patch, GraspVLA follows the user-intended behavior and grasps the milk before JPEG compression. After JPEG compression, however, the policy switches to the attacker-specified target and grasps the tomato sauce can. This example illustrates that JPEG compression can serve as an effective trigger for conditional behavior hijacking in embodied VLA policies.



Figure 4: Qualitative example of JPEG-conditional behavior hijacking on GraspVLA at JPEG quality factor $Q = 50$. With the same optimized adversarial patch, the policy grasps the user-intended object (milk) before JPEG compression (left), but switches to the attacker-specified target (tomato sauce can) after JPEG compression (right).

Table 6: Attack performance across different harmful task categories on MM-SafetyBench with LLaVA-v1.5-7B.

Task	Method	Pre-JPEG Safe Rate	Post-JPEG ASR	Conditional Success Rate	PSNR	LPIPS
Illegal Activity	Clean input	55.67%	41.24%	2.06%	–	–
	PGD w/o JPEG	25.77%	39.18%	2.06%	23.16	**0.479**
	JPEG-aware PGD	58.76%	55.67%	17.52%	23.12	0.531
	Tan-Joint	70.10%	45.36%	18.56%	22.51	0.502
	JACK	**80.41%**	**55.67%**	**36.08%**	**23.57**	0.489
Hate Speech	Clean input	56.44%	42.94%	2.45%	–	–
	PGD w/o JPEG	4.29%	41.10%	1.84%	23.13	0.484
	JPEG-aware PGD	56.44%	79.14%	37.42%	23.11	0.535
	Tan-Joint	73.62%	64.42%	42.33%	22.50	0.506
	JACK	**84.05%**	**79.14%**	**65.03%**	**23.46**	**0.480**
Malware Generation	Clean input	6.82%	75.00%	0.00%	–	–
	PGD w/o JPEG	0.00%	79.55%	0.00%	23.15	0.440
	JPEG-aware PGD	4.55%	93.18%	4.55%	23.12	0.490
	Tan-Joint	72.73%	86.36%	63.64%	22.49	0.463
	JACK	**84.09%**	**93.18%**	**77.27%**	**23.93**	**0.422**
Physical Harm	Clean input	12.50%	77.78%	2.08%	–	–
	PGD w/o JPEG	4.17%	79.17%	0.69%	23.18	0.533
	JPEG-aware PGD	13.19%	88.89%	6.94%	23.13	0.584
	Tan-Joint	57.64%	72.92%	36.11%	22.52	0.552
	JACK	**64.58%**	**88.89%**	**55.56%**	**23.91**	**0.514**
Economic Harm	Clean input	17.21%	77.87%	0.00%	–	–
	PGD w/o JPEG	7.38%	77.87%	0.65%	23.15	0.411
	JPEG-aware PGD	18.85%	87.70%	8.19%	23.10	0.458
	Tan-Joint	42.62%	86.07%	31.97%	22.51	0.428
	JACK	**47.54%**	**87.70%**	**36.07%**	**23.75**	**0.390**
Sex	Clean input	15.60%	76.15%	0.92%	–	–
	PGD w/o JPEG	0.00%	78.90%	0.00%	23.16	0.485
	JPEG-aware PGD	16.51%	88.99%	11.92%	23.13	0.533
	Tan-Joint	36.70%	82.57%	25.69%	22.52	0.504
	JACK	**46.79%**	**88.99%**	**41.28%**	**23.79**	**0.473**

Table 7: Attack performance across different harmful task categories on MM-SafetyBench with Qwen2.5-VL.

Task	Method	Pre-JPEG Safe Rate	Post-JPEG ASR	Conditional Success Rate	PSNR	LPIPS
Illegal Activity	Clean input	97.93%	3.09%	1.03%	–	–
	PGD w/o JPEG	34.02%	1.03%	0.00%	23.05	0.507
	JPEG-aware PGD	93.81%	43.29%	40.20%	23.22	0.517
	Tan-Joint	87.62%	35.05%	26.80%	22.41	0.524
	JACK	**95.88%**	**43.29%**	**43.29%**	22.97	0.513
Hate Speech	Clean input	79.75%	11.04%	0.61%	–	–
	PGD w/o JPEG	12.20%	10.4%	1.23%	23.06	0.508
	JPEG-aware PGD	78.52%	80.98%	60.12%	23.18	0.524
	Tan-Joint	78.53%	61.96%	50.30%	22.40	0.529
	JACK	**93.86%**	**80.98%**	**80.98%**	23.13	0.481
Malware Generation	Clean input	40.91%	45.45%	2.27%	–	–
	PGD w/o JPEG	4.55%	45.45%	2.27%	23.10	0.466
	JPEG-aware PGD	34.09%	93.18%	31.81%	23.22	0.474
	Tan-Joint	79.54%	81.81%	63.64%	22.43	0.482
	JACK	**100.00%**	**93.18%**	**93.18%**	23.84	0.414
Physical Harm	Clean input	59.03%	28.47%	0.00%	–	–
	PGD w/o JPEG	8.33%	26.38%	0.69%	23.08	0.560
	JPEG-aware PGD	61.11%	79.16%	43.75%	23.21	0.570
	Tan-Joint	82.63%	63.88%	51.39%	22.44	0.576
	JACK	**96.53%**	**79.16%**	**79.16%**	23.46	0.519
Economic Harm	Clean input	28.69%	64.75%	1.63%	–	–
	PGD w/o JPEG	7.38%	66.39%	0.00%	23.10	0.436
	JPEG-aware PGD	28.68%	84.42%	15.57%	23.22	0.439
	Tan-Joint	89.34%	77.04%	70.49%	22.45	0.444
	JACK	**94.26%**	**84.42%**	**82.78%**	**24.03**	**0.380**
Sex	Clean input	41.28%	44.03%	1.8%	–	–
	PGD w/o JPEG	2.75%	48.62%	0.92%	23.10	0.512
	JPEG-aware PGD	35.77%	93.57%	33.94%	23.21	0.520
	Tan-Joint	77.98%	88.07%	71.55%	22.42	0.526
	JACK	**97.27%**	**93.57%**	**93.54%**	**23.87**	**0.454**