VALD: Multi-Stage Vision Attack Detection for Efficient LVLM Defense

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) can be vulnerable to adversarial images that subtly bias their outputs toward plausible yet incorrect responses. We introduce a general, efficient, and training-free defense that combines image transformations with agentic data consolidation to recover correct model behavior. A key component of our approach is a two-stage detection mechanism that quickly filters out the majority of clean inputs. We first assess image consistency under content-preserving transformations at negligible computational cost. For more challenging cases, we examine discrepancies in a text-embedding space. Only when necessary do we invoke a powerful LLM to resolve attack-induced divergences. A key idea is to consolidate multiple responses, leveraging both their similarities and their differences. We show that our method achieves state-of-the-art accuracy while maintaining notable efficiency: most clean images skip costly processing, and even in the presence of numerous adversarial examples, the overhead remains minimal.

1 Introduction

Large vision-language models (LVLMs) have shown remarkable capabilities in a wide range of tasks, including text-to-image generation (Alhabeeb and Al-Shargabi, 2024; Rombach et al., 2022) and image-grounded text generation, such as image captioning (Yin et al., 2024; Li et al., 2023) and visual question answering (Huynh et al., 2025; Caffagni et al., 2024; Kuang et al., 2025). However, these models are vulnerable to visual adversarial attacks, in which imperceptible perturbations to input images can manipulate model responses or induce harmful outputs (Liu et al., 2025; Ye et al., 2025; Zhao et al., 2023; Qi et al., 2024). Moreover, the widespread use of publicly available LVLMs and pre-trained visual encoders like CLIP (Radford et al., 2021) gives adversaries gray-box access, making adversarial attacks easier to carry out. Therefore, the need to adopt defense mechanisms becomes more prevalent.

Adversarial defense research for LVLMs has primarily focused on safety alignment issues, such as identifying and preventing harmful and prohibited outputs like encouraging criminal behavior (Pi et al., 2024; Gou et al., 2024; Ding et al., 2024). These defenses often adapt safety mechanisms developed for LLMs, including assessing model responses (Pi et al., 2024; Gou et al., 2024) or learning defense prompts (Mo et al., 2024). However, such methods do not address attacks that subtly steer the model toward benign but incorrect outputs, as demonstrated in Figure 1.

Methods that do address this challenge can be broadly categorized into three approaches. The first involves detecting adversarial inputs and disregarding them upon detection (Zhang et al., 2023, 2024; Fares et al., 2024). The second approach is adversarial training, in which the visual backbone is fine-tuned using adversarial examples to improve robustness against such inputs (Schlarmann et al., 2024). The third approach, introduced in SmoothVLM (Sun et al., 2024), suggests querying the LVLM with both the adversarial image and multiple transformed versions of it. An LLM acts as a judge, compares the captions and chooses one.

DPS (Zhou et al., 2024) recently extended this approach by using the responses from the transformed images as guidance to the LVLM.

We adopt the latter approach, but tackle two key challenges that limit previous methods efficiency. First, generating and assessing every response (for the transformed images) individually is not only computationally expensive but also unnecessary in the absence of an adversarial attack. This makes the approach impractical for real world use, in which most images are not attacked. Second, DPS's effectiveness depends on the use of powerful LVLMs, since consolidating the information requires a high level of reasoning. This may limit the applications

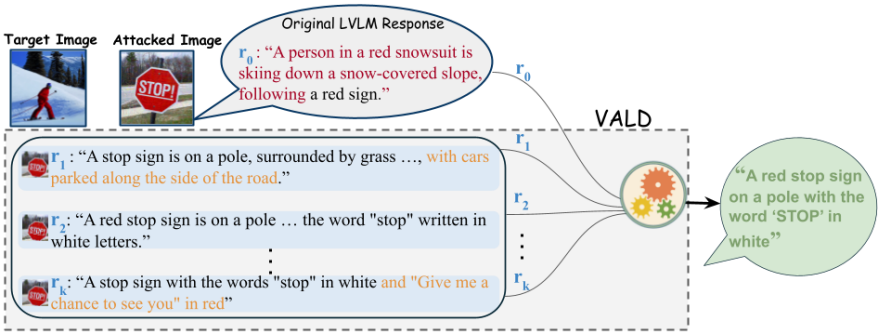


Figure 1: **LVLM defense.** The output of the unprotected LVLM is incorrect (shown in red). In VALD, we generate multiple captions for transformed images (r_1–r_k), each containing both correct content (black) and incorrect content (orange). VALD aggregates information across these captions to produce a final refined caption (green).

that can use the approach and results in significant computational overhead.

We observe that powerful LVLMs are required only for consolidation, rather than for captioning or other visual querying tasks. We propose to decouple these two processes, using a lightweight LVLM for visual analysis and a more powerful LLM solely to consolidate multiple observations. Furthermore, we note a trade-off between runtime and the risk of overlooking adversarial images. The LVLM used for caption consolidation achieves high accuracy but incurs a significant computational cost. Using lighter tools reduces computational demands, but at the expense of accuracy. Therefore, we propose a model that detects most adversarial images early on, leaving only the most challenging cases for the expensive LVLM to resolve. This lightweight adversarial detection relies on the observation that, while accurately identifying adversarial images is difficult, determining whether an image is likely clean is a much simpler task. It only needs to check whether an image (or caption) and its transformations behave as expected for clean images. In other words, we allow false positives, i.e. misclassifying a clean image as adversarial, but not false negatives. We divide the detection mechanism into two tasks according to the information available within the defense pipeline. First, for an image to be considered clean, its embeddings must remain consistent across a set of content-preserving transformations. Notably, this check requires almost no additional computation. If the image is deemed non-adversarial at this stage, the expensive caption-generation and consolidation steps are skipped, and the response for the input image is used directly. About 95% clean images are identified at this stage. Then, in the second stage, if the image is suspected to be adversarial, we generate and analyze the responses for the input image and its transformations. Discrepancies in the text-embedding space provide evidence that an attack has occurred. We show that this can be accomplished using only lightweight LVLMs.

Only then, for suspected adversarial inputs, do we rely on a powerful LLM solely to consolidate multiple responses (e.g., captions). By decoupling the visual-questioning task from the information-consolidation process, we enable the use of lightweight LVLMs in the preceding stage. By instructing the model to focus on consistent elements across the LVLM outputs, such as identifying the main objects in the image and their interactions, we are able to filter out divergences introduced by adversarial attacks. We note that the set of responses for the remaining inputs is already available, as it is generated in the previous stage. A major benefit of our method is its generality: it can defend against a variety of attacks. It does not depend on how the image was perturbed; therefore, even attacks optimized specifically for the target model can be handled effectively.

Hence, we make three key contributions:

(1) We introduce a general, efficient, training-free adversarial defense for LVLMs based on image transformations and agentic data consolidation, capable of handling a wide range of attacks.

(2) An additional contribution is a lightweight detection mechanism, which can be integrated with other defense approaches.

(3) We show that our method achieves state-of-the-art accuracy and efficiency. It is especially efficient when the number of attacked images is small, since most clean images bypass the heavier processing. Even when many adversarial images are present, the additional computational cost remains minimal.

2 Related Work

Our work lies at the intersection of two key challenges in LVLM research: detecting adversarial attacks and defending against them. Before reviewing prior work in these areas, we first review the nature of adversarial attacks themselves.

Image-based adversarial attacks on LVLMs. Adversarial strategies targeting the vision modality, the focus of this paper, can be broadly categorized into jailbreaking attacks and misleading attacks. Jailbreaking attacks aim to bypass safety alignment by exploiting the visual channel, eliciting prohibited, toxic, or otherwise harmful content from an LVLM (Qi et al., 2024; Niu et al., 2024; Bailey et al., 2023). Misleading attacks, in contrast, perturb the visual modality to induce factually incorrect but seemingly benign outputs. Such attacks cause the model to hallucinate nonexistent objects, misclassify visual elements, or fundamentally misinterpret the scene (Tu et al., 2023; Zhang et al., 2025; Zhao et al., 2023; Li et al., 2025).

This work focuses on the second category, misleading attacks. These attacks are distinguished by the attacker's level of access to the model and are typically classified as white-box, gray-box, or black-box attacks (Liu et al., 2025). We refer the reader to excellent surveys that provide a comprehensive overview of vision-based adversarial attacks (Liu et al., 2025; Ye et al., 2025).

This paper evaluates defenses against three representative attacks: MF-ii (Zhao et al., 2023), MixAttack (Tu et al., 2023), and M-Attack (Li et al., 2025). MF-ii and MixAttack are vision-level white-box attacks (i.e., they access the vision encoder parameters), making them particularly strong, while M-Attack is a state-of-the-art black-box attack.

Adversarial attack detection for LVLMs. Detection aims to determine whether an input has been attacked. In LVLM settings, some detection methods leverage internal representations (Zhang et al., 2024) and some monitor the model's output behavior (Zhang et al., 2023; Fares et al., 2024). These detectors have been primarily studied for jailbreak-oriented attacks.

Our work operates both at the embedding level (within the visual encoder) and at the output level (via consistency checks on generated responses).

Adversarial defenses for LVLMs. Complementary to detection, adversarial defense aims to ensure that the model remains robust in the presence of attacks, preserving or restoring the correctness of its responses. Existing LVLM defenses have primarily focused on jailbreaking, leveraging safety prompts (Gou et al., 2024), policy-trained alignment layers (Ding et al., 2024), or response filters to block harmful content (Pi et al., 2024). In contrast, defenses against misleading attacks, where the output remains benign yet semantically incorrect, have received comparatively less attention. These defenses can be broadly grouped into training-based and training-free methods. Training-based approaches aim to improve model robustness through specialized training procedures (Schlarmann et al., 2024). In contrast, training-free defenses operate at inference time without modifying model weights (Sun et al., 2024; Zhou et al., 2024; Nie et al., 2022). Our work focuses on this latter category, providing a complementary approach to safety-alignment methods.

3 Method

Our method is motivated by two key observations. First, applying image transformations to adversarial images often produces noisy and inconsistent captions, as descriptions may refer to either the original content or the perturbation; however, the consistent elements across captions remain anchored in the true image content. Second, the sensitivity of adversarial attacks to image transformations can be exploited for both detection and defense. By integrating these two components, we minimize computational overhead.

Building on these observations, we introduce the *VALD* framework. As shown in Figure 2, it comprises four stages: early attack detection, image based response generation, late-stage adversarial detection, and response generation. The first three stages focus on detection, efficiently identifying the vast majority of clean images and removing them from further processing. The final stage is responsible for defending the remaining images, using image based responses from Stage 2.

The early attack detection. Given an input image x, which may be either clean or adversarial, the task is to determine whether x is suspected as adversarial. While identifying adversarial images accurately is a complex task, the early attack detection mechanism has a simpler goal: identifying whether the image is behaving as expected from clean images. In other words, we allow errors in classifying a clean image as adversarial, but not in the opposite direction.

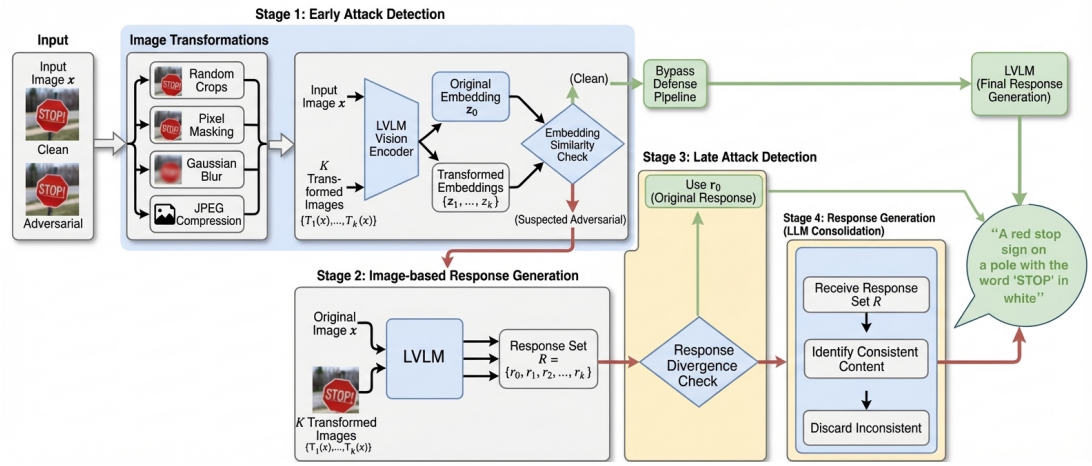


Figure 2: **Architecture.** VALD consists of four stages. First, a set of content-preserving transformations is applied to the input for early detection; clean images bypass the process and return the original LVLM response (r_0). If a potential attack is detected, responses are generated for the transformed images. Next, a late detection stage analyzes these responses to identify additional clean cases. Finally, when an attack is deemed highly likely, an LLM consolidates the responses to produce the final output. Green arrows denote inputs detected as clean, while red arrows indicate those still suspected to be adversarial.

The key idea is that, for an image to be considered clean, it must yield similar embeddings to a set of *content-preserving transformations*—operations that do not significantly alter the image's core semantics. Specifically, we use large crops, pixel masking, gaussian blur, and JPEG compression.

To implement this idea, we first generate a small set of K content-preserving transformations $\{T_k(x)\}_{k=1}^{K}$. Next, we compute the embeddings for the input image and its transformations, $z(x), z(T_1(x)), \ldots, z(T_K(x))$, using a single batched pass through the LVLM's vision encoder. Finally, we compute the distance between the transformed image embeddings and the source embedding and compare it to a predefined threshold τ. If the average distance is below τ, the image is considered clean. Empirically, clean inputs produce highly similar embeddings, while adversarial inputs result in larger discrepancies. Therefore, the threshold is calibrated in advance using clean data, thereby requiring no knowledge of specific attack types or parameters. Formally,

$$\frac{1}{K}\sum_{k=1}^{K} \mathrm{sim}(z(x), z(T_k(x))) \begin{cases} \geq \tau & \text{(Clean)} \\ < \tau & \text{(Adversarial)} \end{cases} \quad (1)$$

If the early detection layer classifies the image as clean, the defense process is bypassed, and the LVLM is queried with the original image. Otherwise, the image and its transformations are passed to the responses generation phase.

In practice, τ is set to the 95th percentile of distances computed on clean data, permitting a 5% false-positive rate on clean samples. We use a combination of $K = 10$ transformations.

Image-based response generation. When a potential attack is detected, the LVLM is queried $K + 1$ times independently: once with the original image and K times with its transformed versions, $\{x, T_k(x)\}_{k=1}^{K}$. Each query uses the original instruction (captions in our case, although other query types are also possible). This process yields a set of responses $R = \{r_k\}_{k=0}^{K}$.

Late detection. Given R, the set of textual responses generated for an image, the late detection phase further verifies the absence of an attack. As before, the goal is to avoid unnecessarily invoking the costly external LLM consolidator during the final stage of the defense pipeline. We note that this stage processes only those inputs that were not already deemed unattacked by the early detector.

The central premise of the late detection phase is that discrepancies within the text embedding space serve as indicators of an attack. A large discrepancy suggests that the generated responses diverge substantially from the expected semantic structure, leading to the sample being classified as attacked (Zhang et al., 2023). Furthermore, the response set for the remaining inputs is readily available, as it is required in the subsequent stage (response generation with LLM consolidator), eliminating the need for additional computation. Consequently, this phase is highly efficient while also providing an extra lightweight layer of verification.

This step can be implemented using various consistency-based detection methods, including embedding space distance measures (Reimers and Gurevych, 2019), question-answering checks over the content (Wu et al., 2024), and LLM agreement judgments (Manakul et al., 2023; Liu et al., 2023).

We adopt the adversarial detection mechanism proposed in JailGuard (Zhang et al., 2023), which analyzes the discrepancy between the response to the original input and the responses to its transformed variants. This choice is motivated by its practicality and robustness in LLM settings. The method is model-agnostic, making it applicable to both open-and closed-source models, and is lightweight and computationally efficient.

Briefly, each response r_i is first mapped into a vector V_i using a lightweight pretrained sentence embedding (Reimers and Gurevych, 2019). A similarity matrix over the responses is constructed as

$$S_{i,j} = \frac{V_i \cdot V_j}{|V_i||V_j|}. \tag{2}$$

Next, for each response r_i, we normalize the ith row of the similarity matrix with respect to all responses. Let $\|S_{i,\cdot}\|_1$ denotes the $\mathcal{L}_1$ norm of the ith row vector. This normalization yields a discrete similarity distribution Q_i:

$$Q_i(k) = \frac{S_{i,k}}{\|S_{i,\cdot}\|_1}, \quad \text{for} \quad k \in \{1, \ldots, K\}. \tag{3}$$

Q_i represents the semantic similarity of response r_i to the full set $\mathcal{R}$.

The similarity matrix is converted to divergence matrix using KL divergence, quantifying the difference between any two similarity distributions. The result is a matrix D, in which each element $D_{i,j}$ calculates the KL divergence between two distributions $(Q_i(x), Q_j(x))$,

$$D_{i,j} = D(Q_i(x) \| Q_j(x)) = \sum_{x=1}^{K} Q_i(x) \log \left(\frac{Q_i(x)}{Q_j(x)} \right). \tag{4}$$

If two responses (r_i, r_j) exhibit significant semantic divergence, their corresponding entry $D_{i,j}$ will be large. if any value in the divergence matrix D exceeds τ_{late}, the input is flagged as attacked and passed into the consolidation phase for final response generation. Otherwise, the original response r_0 is preserved as the final response.

This step captures an additional $\sim$1% of clean images. The remaining false positives typically correspond to weak attacks that introduce only minor caption changes. For example, instead of "A man is standing in a field with two cows, one of which is brown and white. . . " the model produces "a man standing in a field with an apple in his hand, there's a cow in the background. . . "

Response generation. Given the set of responses associated with an image x, the goal of this block is to generate a final output. For instance, when multiple captions are provided, it generates a single final caption. This block is applied only to responses corresponding to images deemed likely attacked during the late-detection stage (red arrows in fig. 2); responses from clean images (green arrows) bypass this block entirely. We emphasize that the response-generation block does not take the image as input, but operates solely on the textual descriptions.

The input text responses often differ from one another, as they are generated from transformed views, as illustrated in Figure 1. Moreover, they may exhibit various types of inconsistencies, including content discrepancies or residual artifacts remaining from the original attack. Therefore, the central challenge of this block is to consolidate all input responses into one coherent, unified response.

Our key idea is to treat these inconsistencies as informative signals: elements that consistently appear in R (e.g., objects, attributes, relations) help guide the generation of a reliable final output; Conversely, elements appearing in only one or a few responses are likely incorrect. For instance, in Figure 1, "A red and white stop sign", which appears in multiple responses, is highly likely to be correct, whereas "cars parked" (r_1) and "Give me a chance to see you" (r_k), which each appear in only a single response, are likely to be adversarial. See Appendix B for an additional reasoning example.

We propose using an external LLM for this purpose. Notably, this is the only stage that employs a large language model, and it is activated exclusively for inputs likely to be adversarial. Moreover, it operates solely on textual input rather than images, thereby reducing computational overhead.

Specifically, the LLM is prompted as follows (see Appendix A for the full prompt). First, it is informed that all responses originate from the same image, which may have been subjected to an adversarial attack. It is then instructed to

(1) Identify the content of each response, focusing on the main objects.

(2) Identify the consistent content across the responses, focusing on the main objects, attributes,

scene settings, and object relationships.
(3) Evaluate consistency by tracking the frequency of each main object and the stability of its attributes across all responses.
(4) Detect elements that appear only in the original response but seem inconsistent or unsupported, and disregard them.
(5) Resolve conflicts by retaining the information confirmed in multiple responses and discarding inconsistent details.
(6) Generate a final response that integrates consistent content into a clear, natural description, excluding objects, attributes, or contextual details that appear only sporadically unless they are logically consistent with the rest.

Our response-generation stage differs in several ways from previous approaches. Sun et al. (Sun et al., 2024) select a final response by randomly sampling from the majority response group. In contrast, we synthesize all available responses to produce a unified output that reflects the full set. Zhou et al. (Zhou et al., 2024) also consolidate multiple responses. However, their approach requires access to the vision component (or the original image) during consolidation. By contrast, our method operates solely on the textual responses, resulting in greater efficiency and enabling the use of significantly smaller vision models.

4 Evaluation

This section evaluates the effectiveness and efficiency of our defense against attacks. We test against adversarial examples generated by common baseline attacks, adhering to their standard setups for a fair comparison.

4.1 Experimental setup

Image captioning offers a particularly strong defense evaluation, as it demands a deep and holistic understanding of the visual content. Unlike simple classification, captioning requires recognizing objects, attributes, spatial relationships, and scene context. As a result, even subtle disruptions caused by adversarial or corrupted inputs become apparent in the generated descriptions.

We evaluate our system against three representative and complementary attacks: MF-ii (Zhao et al., 2023), MixAttack (Tu et al., 2023), and M-Attack (Li et al., 2025), which are standard baselines in recent attack studies (Xie et al., 2025; Mei et al., 2025). MF-ii and MixAttack introduce subtle perturbations that preserve visual appearance while degrading the model's perceptual understanding. M-Attack incorporates cropping-based transformations into its pipeline, yielding an attack that is robust to common defenses.

We follow the above attack protocols by generating adversarially manipulated images and feeding them directly into the target vision-language models. We conduct all evaluations using LLaVA-1.5-7B (Liu et al., 2024), MiniGPT-v2 (Chen et al., 2023), and Qwen3-VL-4B (Bai et al., 2025), three compact yet widely adopted LVLMs. LLaVA and MiniGPT were evaluated in the MF-ii and MixAttack papers, while Qwen3-VL is a widely used SoTA model. We focus on smaller architectures, whose accessibility and efficiency make them representative testbeds.

For response generation, we use the Gemma-3-27B model (Team et al., 2025), which is a moderately sized, open-source model.

We use clean images from MS-COCO (Lin et al., 2014) and Flickr8k Expert (Hodosh et al., 2013) to ensure the availability of ground-truth captions for evaluation.

For MS-COCO, the test set includes 1,000 images randomly sampled from the validation split, with target attack texts randomly drawn from the training set. For Flickr8k Expert, the test set consists of 100 images randomly selected from the training split.

We evaluate the generated captions by comparing them to the ground truth captions. We use SentenceBERT (SBERT) (Reimers and Gurevych, 2019) embedding space to compute the similarity between the generated text and the ground-truth caption. This metric provides semantically rich embedding space, with similarity scores that strongly correlate with human judgment of content equivalence without focusing on different phrasing.

4.2 Results

Captioning evaluation. Table 1 compares our training-free method with two variants of the training-free SmoothVLM defense: the original version using random masking (Sun et al., 2024) and our variant ("SVLM Ours"), which applies the same transformations as our method. We also report results with no defense applied ("Defense off"). Performance is evaluated on clean images and on images attacked by MF-ii, MixAttack, and M-attack. Our method consistently outperforms all baselines; for example, under the MF-ii attack

Model	Attack	Off	SVLM	SVLM Ours	VALD (ours)
LLaVA	Clean	0.722	0.713	0.722	**0.723**
	MF-ii	0.268	0.679	0.703	**0.716**
	MixAttk	0.440	0.653	0.685	**0.701**
	M-Attack	0.360	0.667	0.654	**0.673**
MiniGPT	Clean	0.703	0.680	0.690	**0.705**
	MF-ii	0.217	0.663	0.674	**0.706**
	MixAttk	0.305	0.651	0.676	**0.709**
	M-Attack	0.261	0.622	0.627	**0.655**
Qwen3	Clean	**0.723**	0.702	0.718	**0.723**
	MF-ii	0.534	0.667	0.700	**0.711**
	MixAttk	0.565	0.660	0.695	**0.704**
	M-Attack	0.484	0.668	0.680	**0.689**

Table 1: **Captioning results on MS-COCO.** VALD outperforms the baselines under all three attacks on three models according to the SentenceBERT score.

Model	Attack	Acc	Precision	Recall
LLaVA	Clean	0.960	0	N/A
	MF-ii	0.998	1	0.998
	MixAttack	0.991	1	0.991
	M-Attack	0.995	1	0.995
MiniGPT	Clean	0.954	0	N/A
	MF-ii	0.999	1	0.999
	MixAttack	0.998	1	0.998
	M-Attack	0.997	1	0.997
Qwen3	Clean	0.930	0	N/A
	MF-ii	0.987	1	0.987
	MixAttk	0.912	1	0.912
	M-Attack	0.955	1	0.955

Table 2: **Detection results.** The vast majority of clean images are correctly identified, with few adversarial images misclassified as clean images.

Attack	Off	FARE	VALD
Clean	0.722	0.672	**0.723**
MF-ii	0.268	0.570	**0.716**
MixAttack	0.440	0.586	**0.701**
M-Attack	0.360	0.517	**0.673**

Table 3: **Comparison with training-based defense.** VALD significantly outperforms FARE while maintaining clean performance.

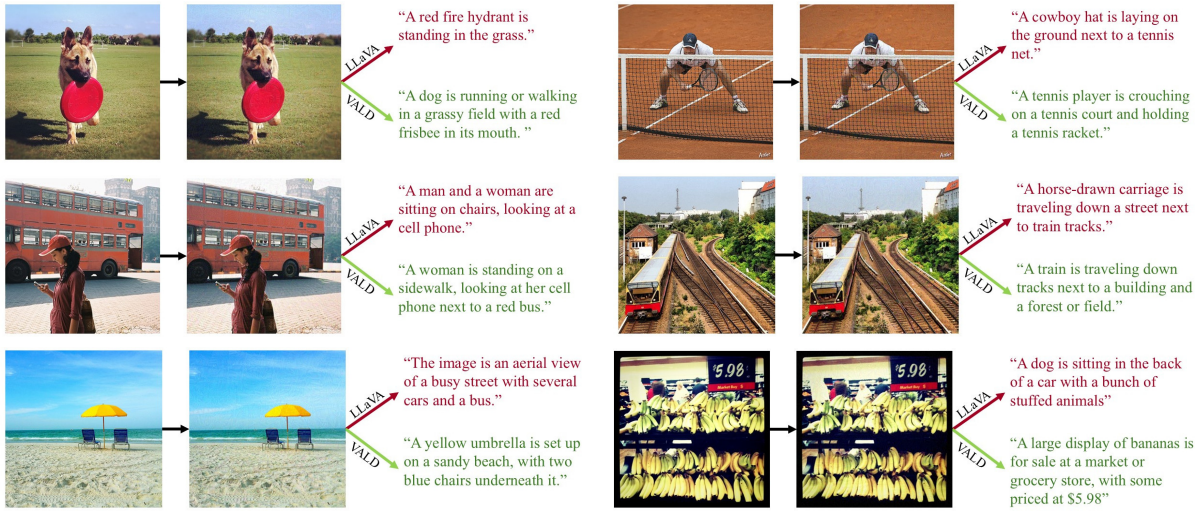
on LLaVA, it achieves a SentenceBERT score of 0.716, compared to 0.679 for (Sun et al., 2024) and 0.703 for our improved SmoothVLM variant. Finally, incorporating our transformation-selection strategy improves SmoothVLM's performance. We compare only with defenses applicable to small LVLMs (e.g., LLaVA, MiniGPTv2, and Qwen3-VL), excluding (Zhou et al., 2024), which relies on stronger models such as GPT-4o-Mini and Gemini. We provide additional results on the Flickr8k Expert dataset in Appendix H.

Attack detection. Table 2 reports detection accuracy (the percentage of correctly classified inputs), recall (the fraction of adversarial images that are successfully detected), and precision (the fraction of inputs flagged as adversarial that are indeed adversarial). Overall, 93-96% of clean images are correctly identified as clean, with nearly all of them filtered out by the fast early detector. Across all attacks and models, recall remains high, above 91% in every setting. The few adversarial images missed by the detector typically correspond to unsuccessful attacks that induce only minor caption changes. Appendix E validate this observation by evaluating a setting in which the defense is always applied.

Comparison with training-based defense. Table 3 compares our method with the adversarial training defense FARE (Schlarmann et al., 2024), which employs the same LLaVA model with a robust vision backbone. VALD outperforms FARE across all attacks while maintaining clean performance. Further analysis across noise levels appears in Appendix F. Notably, under lower attack noise, FARE performs better (scores 0.617 - 0.671), yet VALD still surpasses it (scores 0.702 - 0.726).

Qualitative evaluation. Figures 1 and 3 qualitatively compare our captions with an LVLM's. Although the attacked images are visually similar to the originals, the LVLM produces captions aligned with the targeted attack. In contrast, our method successfully resists the attack and generates accurate captions. For example, while our model correctly captions a dog with a frisbee, LLaVA misidentifies the image as a red fire hydrant.

Runtime analysis. Table 4 reports the best-case runtime (clean images) and worst-case runtime (adversarial images), and compares them to SmoothVLM. Without any defense, caption generation takes 1.8 seconds per image. When our defense is applied, the early detector requires 1.9 seconds per image. It correctly identifies $\sim$95% of clean images, which are then excluded from the remainder of the pipeline. Images not classified as clean are forwarded to the LVLMs and the late detector, a stage that takes 18.2 seconds. At this point, an additional 0.5%-1% of the remaining clean images are identified (depending on the model and the attack) and removed from further processing. Finally, response generation takes 2.8 seconds. By comparison, SmoothVLM processes every image and requires 52.7 seconds per image.



(a) input (b) attacked image (c) captions (a) input (b) attacked image (c) captions

Figure 3: **Qualitative results.** Although the attacked images (b) appear nearly identical to the originals (a), the LVLM incorrectly captions them (c). In contrast, our model successfully defends against the attack and generates accurate captions (c). The images were attacked via (Zhao et al., 2023). See Appendix C for additional results.

Method	Case	Runtime (s)
VALD	Best	1.9
VALD	**Worst**	22.9
SmoothVLM	Always	52.7

Table 4: **Runtime (sec/image).** When clean images are detected by the early detection, VALD is 27× faster than SmoothVLM; when the entire defense is applied, it remains 2× faster. Notably, $\sim$95% of clean images are filtered at the early stage.

Attack	Crop	Masking	JPEG	Blur	Ours
Clean	0.722	**0.726**	0.723	0.723	0.723
MF-ii	0.716	0.694	**0.722**	0.718	0.716
MixAttack	0.697	0.672	0.707	**0.716**	0.701
M-Attack	0.359	0.687	0.701	**0.707**	0.673

Table 5: **Transformations Ablations.** Our transformation mix improves robustness to (adaptive) attacks..

VALD runtime analysis was conducted using LLaVA model using on an NVIDIA RTX 6000 Ada GPU. Appendix G provides analysis of runtimes as a function of the percentage of clean images.

4.3 Ablation study

Transformations parameters. Table 5 evaluates several content-preserving transformations, including large-scale random cropping, pixel masking, JPEG compression, and blurring. We do not assume prior knowledge of the attack, and therefore use a mixture of transformations instead of a single type. This design improves our generality and robustness to adaptive attacks that target specific transformations (e.g., M-Attack with cropping). Appendix J further analyzes performance as a function of the transformation parameters and K.

Perturbation magnitude. Appendix I evaluates VALD across different perturbation budgets (i.e., the distance between clean and adversarial images (Carlini et al., 2019)). As noise increases, undefended performance degrades, while applying the defense substantially improves robustness. Moreover, higher noise levels widen the performance gap between VALD and SmoothVLM.

5 Conclusion

We presented a general, efficient, and training-free defense for Large Vision-Language Models against adversarial images. Our approach builds on two key ideas: a two-stage detection mechanism that efficiently filters most clean inputs, thereby avoiding costly defense procedures, and lightweight image transformations that expose adversarial behavior by consolidating multiple responses, where inconsistencies are weakened and consistent elements are reinforced. Extensive experiments demonstrate state-of-the-art accuracy while preserving performance on clean inputs. These results underscore the benefit of decoupling adversarial detection from expensive reasoning, enabling a practical and scalable defense for real-world LVLM deployment.

6 Limitations

Our defense is based on the assumption that genuine semantic content remains invariant under content-preserving transformations. However, when the same error appears consistently across multiple views, the method may converge on an incorrect yet mutually corroborated interpretation. For instance, we observed an image containing two motorcycles (see Appendix D) to which substantial noise was added. Most transformed views hallucinated a third motorcycle, and the resulting caption incorrectly described three motorcycles.

In addition, there is an inherent trade-off between efficiency and robustness: as more images are classified as clean by the detector, the risk of failing to identify an attacked image increases.

References

Sarah K. Alhabeeb and Amal A. Al-Shargabi. 2024. Text-to-image synthesis with generative models: Methods, datasets, performance metrics, challenges, and future direction. *IEEE Access*, 12:24412–24427.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. 2023. Image hijacks: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: a survey. *arXiv preprint arXiv:2402.12451*.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Yi Ding, Bolian Li, and Ruqi Zhang. 2024. Eta: Evaluating then aligning safety of vision language models at inference time. *arXiv preprint arXiv:2410.06625*.

Samar Fares, Klea Ziu, Toluwani Aremu, Nikita Durasov, Martin Takáč, Pascal Fua, Karthik Nandakumar, and Ivan Laptev. 2024. Mirrorcheck: Efficient adversarial defense for vision-language models. *arXiv preprint arXiv:2406.09250*.

Yunhao Gou, Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. 2024. Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation. In *European Conference on Computer Vision*, pages 388–404. Springer.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Ngoc Dung Huynh, Mohamed Reda Bouadjenek, Sunil Aryal, Imran Razzak, and Hakim Hacid. 2025. Visual question answering: from early developments to recent advances–a survey. *arXiv preprint arXiv:2501.03939*.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Zhaoyi Li, Xiaohan Zhao, Dong-Dong Wu, Jiacheng Cui, and Zhiqiang Shen. 2025. A frustratingly simple yet highly effective attack baseline: Over 90% success rate against the strong black-box models of gpt-4.5/4o/o1. *arXiv preprint arXiv:2503.10635*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Yu Cheng, and Wei Hu. 2025. A survey of attacks on large vision–language models: Resources, advances, and future trends. *IEEE Transactions on Neural Networks and Learning Systems*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment (2023). *arXiv preprint arXiv:2303.16634*, 12.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

Hefei Mei, Zirui Wang, Shen You, Minjing Dong, and Chang Xu. 2025. Veattack: Downstream-agnostic vision encoder attack against large vision language models. *arXiv preprint arXiv:2505.17440*.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. 2024. Fight back against jailbreaking via prompt adversarial tuning. *Advances in Neural Information Processing Systems*, 37:64242–64272.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. 2022. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*.

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

Renjie Pi, Tianyang Han, Jianshu Zhang, Yueqi Xie, Rui Pan, Qing Lian, Hanze Dong, Jipeng Zhang, and Tong Zhang. 2024. Mllm-protector: Ensuring mllm's safety without hurting performance. *arXiv preprint arXiv:2401.02906*.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2024. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 21527–21536.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language

supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. 2024. Robust clip: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. *arXiv preprint arXiv:2402.12336*.

Jiachen Sun, Changsheng Wang, Jiongxiao Wang, Yiwei Zhang, and Chaowei Xiao. 2024. Safeguarding vision-language models against patched visual prompt injectors. *arXiv preprint arXiv:2405.10529*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

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

Junfei Wu, Qiang Liu, Ding Wang, Jinghao Zhang, Shu Wu, Liang Wang, and Tieniu Tan. 2024. Logical closed loop: Uncovering object hallucinations in large vision-language models. *arXiv preprint arXiv:2402.11622*.

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

Mang Ye, Xuankun Rong, Wenke Huang, Bo Du, Nenghai Yu, and Dacheng Tao. 2025. A survey of safety on large vision-language models: Attacks, defenses and evaluations. *arXiv preprint arXiv:2502.14881*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, 11(12).

Jiaming Zhang, Junhong Ye, Xingjun Ma, Yige Li, Yunfan Yang, Yunhao Chen, Jitao Sang, and Dit-Yan Yeung. 2025. Anyattack: Towards large-scale self-supervised adversarial attacks on vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19900–19909.

Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Ming Hu, Jie Zhang, Yang Liu, Shiqing Ma, and Chao Shen. 2023. Jailguard: A universal detection framework for llm prompt-based attacks. *arXiv preprint arXiv:2312.10766*.

Yudong Zhang, Ruobing Xie, Jiansheng Chen, Xingwu Sun, and Yu Wang. 2024. Pip: Detecting adversarial examples in large vision-language models via attention patterns of irrelevant probe questions. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 11175–11183. ACM.

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

Qi Zhou, Tianlin Li, Qing Guo, Dongxia Wang, Yun Lin, Yang Liu, and Jin Song Dong. 2024. Defending lvlms against vision attacks through partial-perception supervision. *arXiv preprint arXiv:2412.12722*.

A Prompts

In two stages of the method described in Section 3, we query the LVLM; below, we detail the corresponding prompts.

Image-based response generation. Each LVLM is queried using the prompt *"provide a short description of the image"* for both the original and the transformed inputs.

We also experimented with more targeted prompts for the transformed images, aiming to produce object level information needed by the consolidation stage (e.g., explicit listing of main objects). However, we found that they tended to produce shorter and less descriptive captions.

Response generation. The full prompt used for consolidation of crops for the task of captioning is:

"You are a multi-caption reasoning agent. You will receive multiple AI-generated image captions originated from the same image. One is from the full image and others were generated with transformed versions of the original image as input. The full image may have been affected by an adversarial attack and its caption should be treated with caution. The other transformed images might be adversarially affected as well, but less than the original image.

Your task is to consolidate these captions into a single, unified caption that accurately describes the content of the original image following a step-by-step procedure:

1. Identify the content of the captions. Focus on identifying clearly described main objects. Be precise and grounded in the text.

2. Identify consistent attributes and scene context. List attributes of the identified objects (descriptive features like color, size, appearance), the scene setting (environment, context), and any actions or relationships described in the caption. Do not speculate. If a detail is implied but not stated, omit it.

3. Evaluate consistency: Keep track of how frequently each object appears across the original image caption and the transformed image captions. Objects consistently appearing in most of the captions from the transformed images are highly likely to be present in the original image. Give higher weight to object presence confirmed across multiple captions, especially if the original caption significantly diverges from the majority.

4. Check caption alignment: Detect and ignore elements that only appear in the original caption if they seem inconsistent or unsupported.

5. Resolve conflicts using a majority-consensus rule: Prefer information that appears in multiple captions. Identify reliable objects and discard any that are not corroborated by multiple captions. Group semantically equivalent terms (e.g., "car" and "vehicle").

Your output:

1. A final caption that combines consistent content into a clear and natural description: In the final caption avoid including objects, attributes, or context mentioned only in one caption unless clearly confirmed by other captions or logically consistent.

2. An explanation of the data consolidation process: Make sure not to include parts of the explanation in the final generated caption!

The original image caption is: <original caption>.

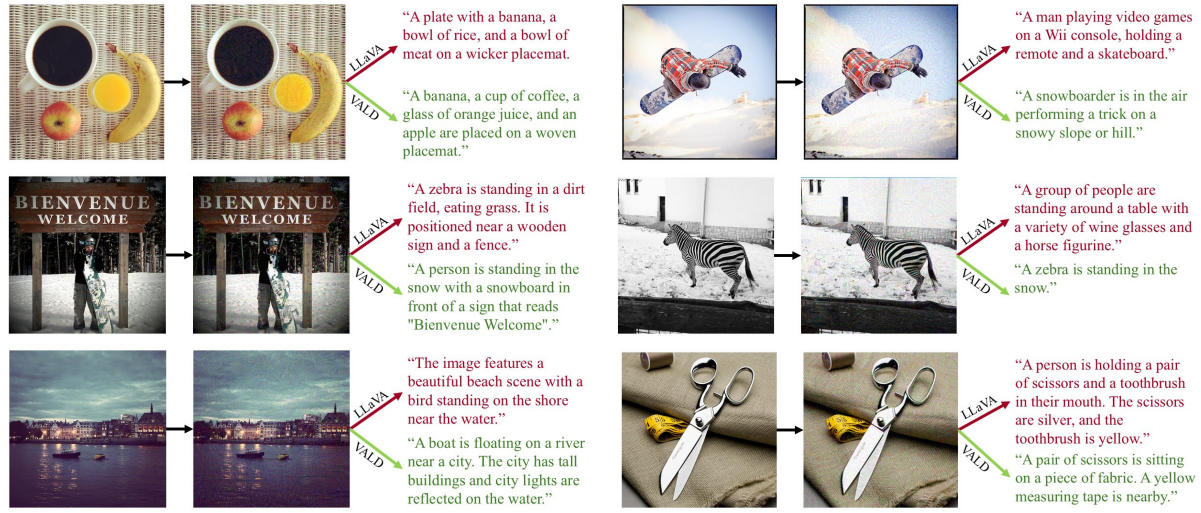
The transformed images captions are: < transformations captions>"

Figure 4: **Consolidation process explanation.** The following elements are mentioned by most captions and form the basis of the final consolidated description: a snow-covered park or forest, a wooden bench, trees, and the bench being covered in snow. Conversely, the following details either conflict with the majority or appear only sporadically; therefore, they are treated as inconsistent and omitted from the final caption: a coconut beauty product, a bird on the bench, a green object on the tree, and a black-and-white filter.

B Example For Reasoning Process

We next present an example of the consolidation process in the response generation stage. Given the input image in Figure 4, the captions generated from the transformed views include the following (out of a total of 10 captions):

1. "The image features a wooden park bench sitting in the snow, surrounded by trees. The bench appears to . . . The scene is captured in



(a) input (b) attacked image (c) captions (a) input (b) attacked image (c) captions

Figure 5: **Qualitative results.** Although the attacked images (b) appear nearly identical to the originals (a), the LVLM incorrectly captions them (c). In contrast, our model successfully defends against the attack and generates accurate captions (c). The images were attacked via (Zhao et al., 2023).

black and white, adding a sense of nostalgia to the image."

2. "The image features a snowy park with a wooden bench sitting in the snow. The bench is situated near a tree, and there is a bird perched on the back of the bench . . . "

3. "The image features a snow-covered park with a wooden bench sitting in the middle of the snow. The bench is positioned under a tree, . . . The tree is adorned with a green object, possibly a decoration or a piece of clothing."

In the explanation of the data consolidation process used to produce the output (Section A), we obtain the following explanation:

"The original caption describes an image of a coconut beauty product, which is inconsistent with the crop captions. The crop captions consistently describe a snow-covered park or forest with a wooden bench and trees. The bench being covered in snow is also a consistent detail. Details like a bird on the bench, a green object on the tree, or the black and white filter appear in only a few captions and are therefore omitted. The final caption consolidates the most frequently mentioned elements - the bench, snow, trees, and the park/forest setting - to provide an accurate description of the image based on the majority of the provided evidence."

Figure 6: **Example of a failure case in the consolidation process.** Most transformed views hallucinate a third motorcycle, causing the consolidated caption to mistake.

C Additional Qualitative Results

Figure 5 provides additional qualitative results, in addition to those presented in Figure 3. It verifies that, despite minimal visual differences, the LVLM yields attack-aligned captions, while our method resists the attack and produces correct outputs.

D Visualization Of The Limitation

Section 6 describes a failure case in which the input image contains two motorcycles parked side by side, yet some transformed views incorrectly report three motorcycles. Figure 6 illustrates this input image. The final consolidated response is: *"The image features three motorcycles parked in a row on a concrete or cement surface. The motorcycles are of different sizes and positioned next to each*

Model	Attack	Off	VALD	Always On
LLaVA	Clean	0.722	0.723	**0.733**
	MF-ii	0.268	**0.716**	**0.716**
	MixAttk	0.440	**0.701**	**0.701**
	M-Attack	0.360	**0.673**	**0.673**
MiniGPT	Clean	0.703	0.705	**0.726**
	MF-ii	0.217	**0.706**	**0.706**
	MixAttk	0.305	**0.709**	**0.709**
	M-Attack	0.261	**0.655**	**0.655**
Qwen3	Clean	0.723	0.723	**0.725**
	MF-ii	0.534	0.711	**0.712**
	MixAttk	0.565	0.704	**0.705**
	M-Attack	0.484	0.689	**0.691**

Table 6: **Defense performance with and without detection.** Enabling detection preserves VALD defense performance, yielding results nearly identical to always applying the defense.

other, near a brick wall."

The explanation of the consolidation process for this caption is as follows: *"The consolidation process began by identifying the core objects present in all captions: motorcycles. The number of motorcycles varied between 'two' and 'three'. 'three' appeared more frequently across the crop captions and is therefore prioritized. Mentions of two motorcycles may be explained by cropping effects."*

This case may be explained by the presence of visible car edges on the left side of the image, which some transformed captions may have misinterpreted as parts of an additional motorcycle.

E Additional Detection Analysis

Table 6 reports defense performance both with detection enabled and with the defense always applied. Across all attacks and models, the results are nearly identical, indicating that the detection stage has little effect on overall defense effectiveness while substantially reducing runtime. For Qwen3-VL, where detection performance is somewhat lower than for LLaVA and MiniGPT, the near-identical defended results suggest that the missed adversarial images correspond to ineffective attacks and are therefore classified as clean by the detection mechanism.

F Adversarial Training Analysis

Table 7 compares VALD with the robust FARE model under different perturbation budgets ϵ. In our evaluation, we use FARE's LLaVA-based variant, where the original LLaVA vision backbone is replaced with the FARE vision backbone. We evaluate the strongest available version of FARE, which is trained with adversarial examples at $\epsilon = 4$. At $\epsilon = 8$, VALD achieves higher robustness than FARE across all attack types. When the attack strength increases to $\epsilon = 16$, the gap becomes larger. For example, under MF-ii with $\epsilon = 8$ VALD achieves 0.726 compared to 0.617 for FARE, and for $\epsilon = 16$ VALD achieves 0.716 compared to 0.570 for FARE.

ϵ	Attack	Off	FARE	VALD
	Clean	0.722	0.672	**0.723**
8	MF-ii	0.282	0.617	**0.726**
	MixAttack	0.603	0.644	**0.720**
	M-Attack	0.429	0.671	**0.702**
16	MF-ii	0.268	0.570	**0.716**
	MixAttack	0.440	0.586	**0.701**
	M-Attack	0.360	0.517	**0.673**

Table 7: **Robustness under different perturbation budget (ϵ).** VALD consistently outperforms the adversarially trained baseline FARE, with a larger performance gap at higher perturbation levels.

Stage	Runtime (Average)
Caption generation (Defense off)	1.8
Caption generation with ED **(Best Case)**	1.9
LVLM querying	18.0
Late Detector	0.2
Response generation	2.8
VALD total runtime (Worst Case)	22.9
SmoothVLM total runtime	52.7

Table 8: **Runtime breakdown (sec / image).** Images filtered by the Early Detector (ED) correspond to the best case runtime, while the worst case occurs when images pass through the full pipeline.

G Additional Runtime Analysis

Runtime breakdown. Table 8 provides a detailed breakdown of the runtime across the different stages of the VALD pipeline. Images filtered by the Early Detector correspond to the best-case runtime. If an image is filtered by the Late Detector, the runtime before the response generation stage is approximately 20.1 seconds. All runtime measurements were performed on VALD LLaVA setup and were conducted on a NVIDIA RTX 6000 Ada GPU.

Runtime as a function of clean images percentage. Table 9 reports the average runtime of VALD (per image) for different proportions of clean images. In realistic settings, most inputs are clean.

Attack	Clean %	Acc	Rec	Prec	Time
MF-ii	50	0.979	0.998	0.962	12.8
MF-ii	95	0.962	1	0.568	3.8
MixAttk	50	0.973	0.992	0.956	12.9
MixAttk	95	0.961	1	0.562	3.8
M-Attack	50	0.979	0.994	0.965	12.8
M-Attack	95	0.962	1	0.568	3.8

Table 9: **Runtime as a function of clean images percentage.** Average runtime decreases as the proportion of clean images increases.

Therefore we evaluate two scenarios, one in which 50% of the images are clean and another in which 95% are clean. Since the early detection identifies approximately 95% of clean images, these filtered inputs incur a runtime that is nearly identical to that of running without defense (1.8s vs. 1.9s). As a result, the average runtime decreases as the proportion of clean images increases.

In both scenarios, recall remains above 99%, indicating that almost no adversarial images are misclassified as clean. Overall accuracy also stays high across both scenarios. As expected, precision decreases when the proportion of clean images increases. This is because the precision measures how many of the samples flagged as adversarial are truly adversarial. When clean images are much more common, even a small number of clean images misclassified as adversarial constitutes a larger fraction of the samples classified as adversarial.

H Evaluation On An Additional Dataset

Flickr8k Expert evaluation. Table 10 presents results on the Flickr8k Expert dataset (Hodosh et al., 2013), which contains high quality captions. In this dataset, each image caption pair is evaluated by three expert annotators with scores ranging from 1 to 4, where a score of 1 indicates that the caption does not correspond to the image and a score of 4 indicates a fully correct description of the visual content. For our evaluation we randomly select 100 images with high expert scores, which serve as ground truth for evaluating the generated captions.

Consistent with the main paper results, our method outperforms the SVLM baseline across all attack settings, showing that its advantage is not dataset specific.

While our detection mechanism does not rely on attack-specific behavior, it does depend on modeling the distribution of clean images behavior and therefore requires threshold calibration for different data distributions. Using the thresholds calibrated on COCO, we find that nearly all adversarial images are still correctly identified as adversarial, but more clean images are incorrectly flagged as adversarial. In our framework, such errors mainly increase computation, since misclassifying a clean image as adversarial only causes the defense to be applied unnecessarily and does not harm captioning performance.

I Perturbation Budget Ablations

Table 11 evaluates defense performance across different perturbation budgets (ϵ). As ϵ increases, undefended performance degrades, while applying the defense improves robustness. Additionally, the gap between VALD and SmoothVLM increases at higher noise levels, For example, at $\epsilon = 16$, VALD outperforms SmoothVLM by 4%, and at $\epsilon = 24$ the gap further increases to 5% under MixAttack.

J Additional Transformations Ablations

J.1 Transformations Parameters Ablations

Table 5 reports the best performing parameter configuration for each transformation used in our defense. Table 12 provides the full ablation across the evaluated parameter ranges to analyze the sensitivity of the method to these choices. Cropping percentages indicate the retained spatial extent along both image axes; pixel masking percentages correspond to the fraction of pixels randomly suppressed in the image; JPEG values denote the compression quality parameter, where lower values imply stronger compression; blur values correspond to the size of the applied gaussian kernel. The results reveal two clear trends. For cropping and pixel masking, better performance is achieved when most of the original image is preserved (e.g., 95% crop and 10% masking). In contrast, for JPEG compression and blur, stronger transformations lead to better robustness. As a future work, we plan to explore combinations of transformations, where each randomized image undergoes several transformations, making it harder for adversarial perturbations to remain effective.

J.2 Ablation On K

Table 13 examines the effect of the number of transformations K.

The impact of K varies across attacks. For MF-ii and MixAttack, the applied transformations are relatively effective as a defense, so even $K = 6$

Model	Attack	Off	SVLM	SVLM Ours	VALD (ours)	Acc	Prec	Rec
LLaVA	Clean	0.713	0.690	0.704	**0.722**	0.55	0	N/A
	MF-ii	0.213	0.642	0.694	**0.710**	1	1	1
	MixAttk	0.382	0.608	0.643	**0.664**	1	1	1
	M-Attack	0.347	0.651	0.652	**0.663**	0.98	1	0.98

Table 10: **Captioning and detection results on the Flickr8k Expert dataset.** VALD surpasses the SVLM baseline in captioning performance while correctly detecting nearly all adversarial images.

	Crop			Pixel masking			JPEG			Blur		
Attack	95%	90%	85%	10%	15%	20%	30	40	50	5	7	9
Clean	0.722	0.722	0.722	0.726	0.725	0.724	0.723	0.722	0.722	0.722	0.722	0.723
MF-ii	0.716	0.716	0.713	0.694	0.693	0.689	0.722	0.718	0.712	0.700	0.718	0.718
MixAttack	0.697	0.693	0.688	0.672	0.677	0.676	0.707	0.696	0.675	0.677	0.713	0.716
M-Attack	0.359	0.359	0.360	0.687	0.686	0.680	0.701	0.679	0.653	0.600	0.697	0.707

Table 12: **Transformation parameter ablation.** For cropping and pixel masking, performance is best when most of the original image is preserved, whereas for JPEG compression and blur, stronger transformations provide better robustness.

Model	ϵ	Off	SVLM	SVLM Ours	VALD (ours)
Clean	–	0.722	0.713	0.722	**0.723**
MF-ii	8	0.282	0.684	0.712	**0.726**
	16	0.268	0.679	0.703	**0.716**
	24	0.262	0.665	0.695	**0.704**
MixAttack	8	0.603	0.683	0.704	**0.720**
	16	0.440	0.653	0.685	**0.701**
	24	0.346	0.616	0.662	**0.673**
M-Attack	8	0.429	0.681	0.682	**0.702**
	16	0.360	0.667	0.654	**0.673**
	24	0.341	**0.648**	0.637	0.623

Table 11: **Impact of perturbation.** Increasing noise degrades performance without defense (Off), while our defense improves robustness and increasingly outperforms SmoothVLM (SVLM).

transformed views is sufficient for reliable consolidation, and performance remains largely stable as K increases. In contrast, M-Attack is more resilient to transformations (especially cropping), and therefore benefits more from increasing K, improving from 0.532 at $K = 4$ to 0.673 at $K = 10$. However, this improvement begins to diminish beyond $K \approx 10$. As noted in the limitations section, when the consolidating LLM receives too many captions with consistent errors, these errors may no longer be treated as outliers. Therefore, increasing K indefinitely not only raises computational cost, but can also eventually degrade performance. Overall, these results suggest that a moderate number of transformations captures most of the robustness gains, while maintaining a reasonable trade-off with computational overhead.

K SmoothVLM Implementation Details

SmoothVLM (Sun et al., 2024) operates as follows. The LVLM is queried on both the adversarial image and its transformed versions, where the original method uses random pixel masking. The caption generated from the adversarial image is then compared with the captions generated from the transformed images, and an external LLM judge determines whether each pair is synonymous. The captions are partitioned according to this classification, and the final caption is randomly selected from the majority group.

To ensure fairness, we evaluate SmoothVLM in a setting that closely matches our own. For each comparison, we use the same LVLM backbone as in our method, as indicated in the tables (e.g., LLaVA, MiniGPT, and Qwen3). We also use **Gemma-27B** as the external LLM judge, matching the model used in our consolidation stage, and set the number of transformations to $K = 10$ for both methods. In addition to the original SmoothVLM baseline (SVLM), we report a variant denoted SVLM-Ours, in which SmoothVLM uses the same transformation set as our method - JPEG compression, blur, random cropping, and random masking. This ensures an identical experimental setup and enables a more direct comparison between the two methods.

Attack	Number of Transformations K					
	2	4	6	8	10	12
MF-ii	0.689	0.709	0.716	0.714	0.716	**0.717**
MixAttack	0.664	0.694	**0.703**	0.700	0.701	0.702
M-Attack	0.398	0.532	0.608	0.649	**0.673**	0.640

Table 13: **Effect of the number of transformations K.** A moderate number of transformed views captures most of the robustness gains, while larger K saturate and possibly reduce performance.