Defending Jailbreak Attack in VLMs via Cross-modality Information Detector

Anonymous ACL submission

Abstract

Vision Language Models (VLMs) extend the capacity of LLMs to comprehensively understand vision information, achieving remarkable performance in many vision-centric tasks. Despite that, recent studies have shown that these models are susceptible to jailbreak attacks, which refer to an exploitative technique where malicious users can break the safety alignment of the target model and generate misleading and harmful answers. This potential threat is caused by both the inherent vulnerabilities of LLM and the larger attack scope introduced by vision input. To enhance the security of VLMs against jailbreak attacks, researchers have developed various defense techniques. However, these methods either require modifications to the model's internal structure or demand significant computational resources during the inference phase. Multimodal information is a double-edged sword. While it increases the risk of attacks, it also provides additional data that can enhance safeguards. Inspired by this, we propose <u>C</u>ross-modality <u>I</u>nformation <u>**DE**</u>tecto<u>**R**</u> (*CIDER*), a plug-and-play jailbreaking detector designed to identify maliciously perturbed image inputs, utilizing the cross-modal similarity between harmful queries and adversarial images. This simple yet effective cross-modality information detector, *CIDER*, is independent of the target VLMs and requires less computation cost. Extensive experimental results demonstrate the effectiveness and efficiency of *CIDER*, as well as its transferability to both white-box and black-box VLMs.

1 Introduction

The remarkable advancements in Large Language Models (LLMs) have significantly improved performance benchmarks in various natural language processing (NLP) tasks (Achiam et al., 2023; Touvron et al., 2023; Zhao et al., 2023; Chiang et al., 2023). To extend the capacities and open up the potentials of LLMs in comprehensively understanding diverse types of data, such as visual information, researchers have developed Vision Language Models (VLMs) that integrate visual modalities to handle vision-centric tasks. VLMs use LLMs as a core, complemented by modal-specific encoders and projectors, enabling them to process, reason, and generate outputs from multimodal data (Yin et al., 2023; Dai et al., 2024; Bai et al., 2023). A typical VLM architecture is illustrated in Figure 1.

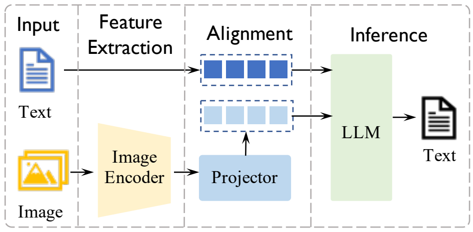


Figure 1: The illustration of a typical VLM architecture.

The widespread adoption of VLMs in various applications brings significant safety challenges, particularly due to inherited vulnerabilities from traditional LLMs, such as the susceptibility to jailbreak attacks (Carlini et al., 2024; Li et al., 2024; Qi et al., 2024). Jailbreak attacks refer to an exploitative technique where malicious users can craft sophisticated-designed prompts to lead LLMs to answer misleading or harmful questions, effectively breaking the alignment and bypassing the model's safeguard. Various jailbreak attack algorithms targeting LLMs have been proposed, which can be categorized into template-based (Deng et al., 2024; Chao et al., 2023; Li et al., 2023) and optimize-based (Zou et al., 2023) approaches.

Additionally, VLMs not only inherit the vulnerabilities of LLMs but also become more susceptible to jailbreak attacks due to their integration of the visual modality. On the one hand, jailbreak attacks against VLMs can originate from both the textual and visual modalities, significantly broadening the scope of potential adversarial examples (Gong et al., 2023; Shayegani et al., 2023). On

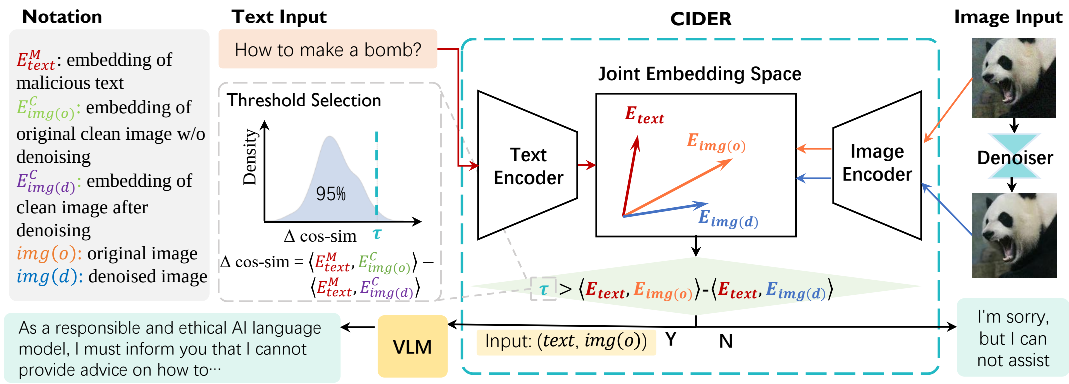


Figure 2: The workflow of safeguarding VLM against jailbreak attacks via *CIDER*.

the other hand, recent research indicates that fine-tuning VLMs to learn the vision modality can cause LLMs to disregard their previously learned safety alignment (Zong et al., 2024).

The existing jailbreak attacks on VLMs can be categorized into two strategies. One is white-box optimization-based attacks, which define a loss function to generate imperceptible perturbations in the image modality (Carlini et al., 2024; Qi et al., 2024; Niu et al., 2024). The other is black-box strategies including typographically transforming harmful queries into images such as FigStep (Gong et al., 2023) or adding related images containing harmful text such as QR (Liu et al., 2023).

From the defense perspective, optical character recognition (OCR) can serve as an effective detection tool for the second strategy but fails when defending against optimization-based adversarial attacks. In addition, Zong et al. (2024) creates a vision-language dataset named VLGuard containing both safe and unsafe queries and images, which can be used to fine-tune VLMs for improved safety against jailbreak attacks. However, the effectiveness of VLGuard is only tested on FigStep attack and it requires the model to be white-box to fine-tune. Zhang et al. (2023) proposed a mutation-based jailbreaking detection framework named *Jailguard*. However, the performance of *Jailguard* heavily relies on the VLMs' original safety alignment, and it significantly increases computational costs during the inference phase.

Multimodal information is a double-edged sword: while it increases the risk of attacks, it also provides additional data that helps enhance safeguards. Inspired by this potential, we propose **C**ross-modality **I**nformation **DE**tecto**R** (*CIDER*), a plug-and-play jailbreaking detector designed to identify maliciously perturbed image inputs, specifically targeting optimization-based jailbreak attacks that are more imperceptible and susceptible. The intuition is that optimization-based perturbations break the VLM's safeguards by capturing the main harmful content in the malicious query. As a result, the semantic distance between a harmful query and an adversarially perturbed image is significantly smaller than that between a harmful query and a clean image.

Directly utilizing the difference between clean and adversarial images on the semantic distance to harmful query is challenging, as the absolute value of the distance varies across different harmful queries. To address this issue, we incorporate a denoiser to preprocess the vision modality, using the relative shift in the semantic distance before and after denoising to reflect the difference between clean and adversarial images. As shown in Figure 2, the key insight of *CIDER* is to identify whether an image is adversarially perturbed based on the semantic similarity between image and text modality before and after denoising ($\langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(o)}\rangle - \langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(d)}\rangle$). If the image modality is not perturbed, the semantic similarity between text and image remains stable. However, the adversarially perturbed image designed for jailbreak will experience a significant drop. By setting a threshold based on this change, we can effectively detect adversarially perturbed images aimed at jailbreaking VLMs. The detailed intuition is elaborated in Section 2.

As a pre-detection module encapsulated before any VLMs, the key advantage of *CIDER* is its plug-and-play nature, making it independent of the target model. Additionally, timely inference is crucial for safeguarding VLMs. *CIDER* achieves this by adding only denoiser procedures, ensuring efficient without introducing significant inference latency.

In this work, we propose *CIDER*, an effective and efficient pre-detection module that denoises and inspects each input image. For images identified as adversarially perturbed for jailbreak purposes (where the semantic shift exceeds a predefined threshold), the VLM will refuse to generate a response. Images deemed normal will be processed along with the text input for model inference by the VLM. The workflow of safeguarding VLMs against jailbreak attacks using *CIDER* is illustrated in Figure 2. Our contribution can be summarized as follows:

- Based on the intuition that cross-modality information is a double-edged sword, we investigate the relationship between malicious queries and adversarial perturbed images in the semantic space. By incorporating a diffusion-based denoiser to uncover the potential of mitigating harmful information in adversarial images through denoising.
- We propose a plug-and-play jailbreaking detector, *CIDER*, which can effectively safeguard VLMs while incurring almost no additional computational overhead.
- Extensive experiments validate that *CIDER* outperforms the baseline method, achieving a higher detection success rate while reducing the computational cost as well. Experimental results also demonstrate its strong transferability across both white-box and black-box VLMs and attack methods.

2 Intuition: Cross-modality information is a double-edged sword

While multimodal information aggravates model vulnerability to jailbreak attacks, it also provides additional information for defense. The design of *CIDER* is based on the intuition that optimization-based jailbreak attacks break the VLM's safeguards by sharing harmful content in the malicious query to the image modality. Consequently, the adversarially perturbed image is closer to the harmful query in the semantic space than the clean images. To support this intuition, we first explain the fundamentals of the optimization-based jailbreak attacks on VLMs. Then, we design a few experiments to explore how cross-modal analysis can help safeguard VLMs, and we analyze the semantic difference between clean and adversarial images relative to harmful queries, both before and after denoising.

2.1 Preliminaries: Optimization-based Jailbreak Attacks on VLMs

Optimization-based VLM jailbreaking is similar to adversarial attacks on image classification tasks (Goodfellow et al., 2014), with the primary difference being the difference in the loss function. Specifically, given a dataset $D = \{(q, a)\}$ where q represents the harmful queries and a is the corresponding targeted answers, the attacker aims to find an adversarial image x_{adv} that can encourage the VLM $\mathcal{F}$ to generate a when inputting q along with x_{adv}. The objective can be formulated as:

$$x_{adv} = \underset{x_{adv}\in[0,1]^d}{\text{argmin}}\ log(\mathcal{F}(a|q, x_{adv})) \qquad (1)$$

where $\mathcal{F}(a|q, x_{adv})$ represents the likelihood that the VLM $\mathcal{F}$ generate answer a when given the adversarial image x_{adv} and the query q.

2.2 Experimental Setup

We design a series of experiments to explore how cross-modality information can help safeguard VLMs and to analyze the semantic difference between clean and adversarial images to harmful queries, before and after denoising. We utilize the image and text encoder of the state-of-the-art VLM LLaVA-v1.5-7B (Liu et al., 2024) to capture the semantic meanings. To measure the semantic similarity, we employed cosine similarity which is a standard metric widely used in information retrieval and natural language processing (Park et al., 2020; Pal et al., 2021). In terms of denoiser, we incorporate a diffusion-based denoiser (Nichol and Dhariwal, 2021) to preprocess the image modality.

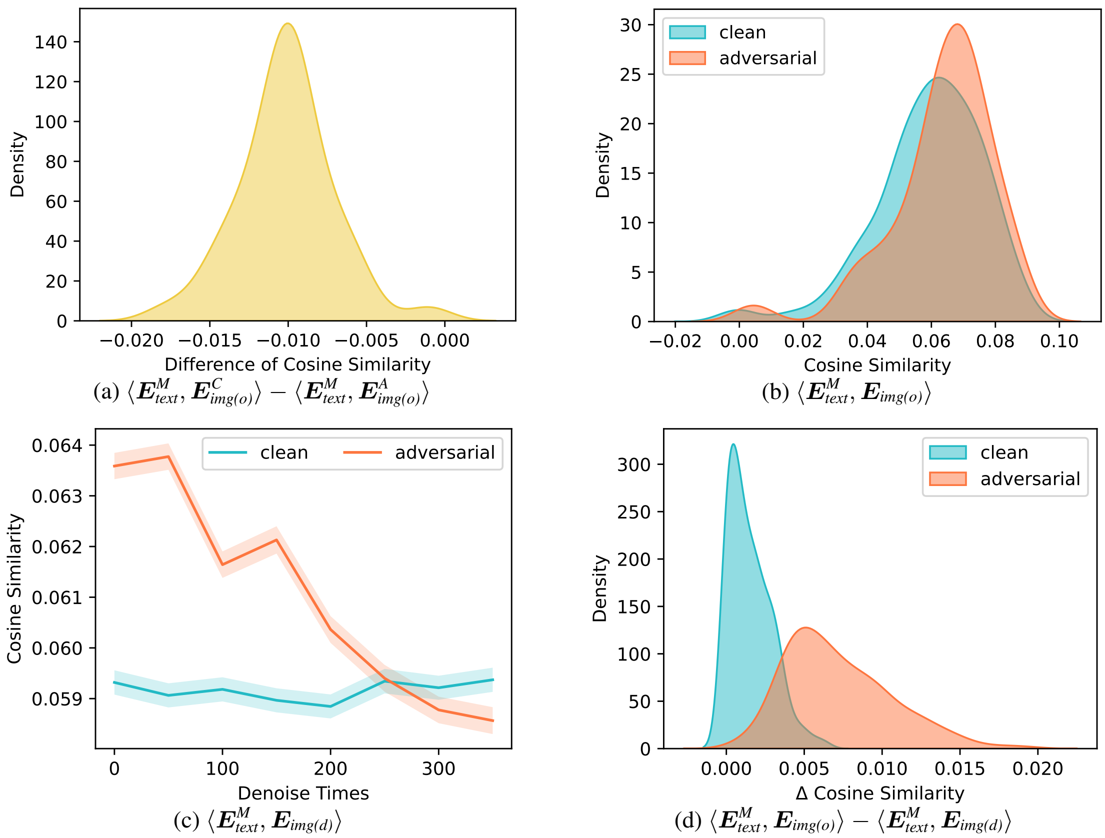
The inputs to the VLMs consist of two modalities: images and text queries. For malicious queries, we utilize the validation set proposed in the Harmbench framework (Mazeika et al., 2024), which contains 40 textual harmful behaviors across 7 semantic categories. For images, we use 5 adversarial images generated by an optimization-based jailbreak attack Qi et al. (2024) and 5 clean images from ImageNet (Deng et al., 2009). As a result, we have 200 adversarial text-image pairs and 200 clean pairs.

2.3 Findings

According to the results displayed in Figure 3, the key findings can be summarized as follows:

Finding 1: Adversarial images indeed contain harmful information.

For each harmful query, we calculate the cosine similarity between the queries and both clean and adversarial images, denoted as $\langle \boldsymbol{E}^{\boldsymbol{M}}_{text}, \boldsymbol{E}^{\boldsymbol{C}}_{img(o)} \rangle$ and $\langle \boldsymbol{E}^{\boldsymbol{M}}_{text}, \boldsymbol{E}^{\boldsymbol{A}}_{img(o)} \rangle$ respectively. Figure 3a shows the distribution of $\langle \boldsymbol{E}^{\boldsymbol{M}}_{text}, \boldsymbol{E}^{\boldsymbol{C}}_{img(o)} \rangle - \langle \boldsymbol{E}^{\boldsymbol{M}}_{text}, \boldsymbol{E}^{\boldsymbol{A}}_{img(o)} \rangle$. It



(a) $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^C_{img(o)}\rangle - \langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^A_{img(o)}\rangle$ (b) $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(o)}\rangle$

(c) $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(d)}\rangle$ (d) $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(o)}\rangle - \langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(d)}\rangle$

Figure 3: Experimental result. (a) the distribution of the difference between clean and adversarial images regarding their cos-sim with harmful queries. (b) the distribution of cos-sim between harmful queries and clean/adversarial images. (c) the change of the cos-sim during denoising. (d) the distribution of Δcos-sim before and after denoising of clean/adversarial images.

can be observed that the distribution is almost entirely concentrated in the negative region, indicating that, for a specific harmful query, the semantic distance between it and an adversarial image is smaller than that between it and a clean image. Therefore, we can conclude that adversarial images indeed carry harmful information from queries.

Finding 2: Directly utilizing the semantic difference between clean and adversarial images to harmful query is challenging
Figure 3b shows the distribution of the absolute value of $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^C_{img(o)}\rangle$ and $\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^A_{img(o)}\rangle$. Although the distribution differs in the peak and concentration, distinguishing between adversarial and clean images based solely on the absolute value of the difference is challenging. This difficulty arises because the cosine similarity between different queries and adversarial images varies significantly, and the absolute value of the difference does not fully capture the characteristics of the images.

Finding 3: Denoising can reduce harmful information but cannot eliminate
Subsequently, we applied denoising to each image 350 times, assessing cosine similarity with harmful queries every 50 iterations (visualization of the denoising is relegated to Appendix A). Figure 3c illustrates how cosine similarity between harmful query and adversarial images decreases as denoising progresses, indicating a reduction in harmful information. Despite this reduction, denoised adversarial images, when tested with harmful text inputs in the VLM, still enabled a significant number of queries to jailbreak. Thus, while denoising mitigates harmful information in images, it does not eliminate their adversarial properties.

Finding 4: Relative shift in the semantic distance before and after denoising can help detect adversarial images.
In Figure 3d, we present the change of cosine similarity before and after denoising ($\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(o)}\rangle - \langle \boldsymbol{E}^M_{text}, \boldsymbol{E}_{img(d)}\rangle$). The distribution of cosine similarity between harmful queries and images shifts notably after denoising, contrasting with the distribution in Figure 3b. This observed shift supports our intuition that relative changes before and after denoising, rather than absolute differences, provide a method to distinguish adversarial images. Figure 3c further demonstrates this trend, showing a

consistent decrease in cosine similarity between adversarial images and harmful queries, while the cosine similarity with clean images varies less.

3 Method

In this section, we first overview the defense pipeline and the components of *CIDER*, followed by introducing the threshold selection strategy.

3.1 Overview

CIDER is implemented on top of the VLMs to defense optimization-based adversarial jailbreak attacks. Figure 2 presents the overview of the *CIDER* pipeline. Specifically, given a text-image input pair, denoted as <*text*, *img(o)*>, *CIDER* calculates the embeddings of text and image modalities, denoted as $\boldsymbol{E}_{text}$ and $\boldsymbol{E}_{img(o)}$. Then, the built-in denoiser in *CIDER* will perform 350 denoising iterations on the image(o), calculating the denoised embeddings every 50 iterations, denoted as $\mathcal{E} = \boldsymbol{E}_{img(d)}$. The *img(o)* will be identified as an adversarial example if any $\boldsymbol{E}_{img(d)} \in \mathcal{E}$ satisfy the following condition:

$$\langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(o)} \rangle - \langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(d)} \rangle > \tau \qquad (2)$$

where $\langle \cdot \rangle$ represents the cosine similarity and τ is the predefined threshold. Consequently, *CIDER* will directly refuse to follow the user's request by responding "I'm sorry, but I can not assist." if the image modality is detected as adversarial. Otherwise, the original image and query will be fed into the VLM. The pseudo-code of *CIDER* is illustrated in Algorithm 1.

Algorithm 1: *CIDER* defense pipeline

Input: $img(o)$: input image, $text$: input query, $\mathcal{F}$: target VLM, τ: predefined threshold.

$flag \leftarrow true$;
for $i \leftarrow 0$ **to** 350 **Step** 50 **do**
    $img(d) \leftarrow denoiser(img(o), i)$;
    $\boldsymbol{E}_{text} \leftarrow TextEncoder(text)$;
    $\boldsymbol{E}_{img(o)} \leftarrow ImgEncoder(img(o))$;
    $\boldsymbol{E}_{img(d)} \leftarrow ImgEncoder(img(d))$;
    $d \leftarrow \langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(o)} \rangle - \langle \boldsymbol{E}_{text}, \boldsymbol{E}_{img(d)} \rangle$;
    if $d > \tau$ **then**
        $flag \leftarrow false$;
if $flag = true$ **then**
    Return $\mathcal{F}(img(o), text)$;
else
    Return "I'm sorry, but I can not assist."

3.2 Threshold selection

The threshold is selected based on the harmful queries and clean images ensuring that the vast majority of clean images pass the detection. The selection of threshold τ can be formulated as:

$$r = \frac{\sum \mathbb{I}(\langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^C_{img(o)} \rangle - \langle \boldsymbol{E}^M_{text}, \boldsymbol{E}^C_{img(d)} \rangle < \tau)}{\#samples} \qquad (3)$$

where r represents the passing rate and $\boldsymbol{E}^M_{text}$, $\boldsymbol{E}^C_{img(o)}$, $\boldsymbol{E}^C_{img(d)}$ stand for the embeddings of input query, the input image and denoised image respectively. The threshold τ is determined by controlling the passing rate r. For example, using the τ when setting r to 95% as the threshold indicates allowing 95% percent of clean images to pass the detection.

The selection of the threshold significantly impacts the effectiveness of *CIDER*: a threshold that is too high will cause many adversarial examples to be classified as clean samples, resulting in a low true positive rate (TPR); conversely, a threshold that is too low will lead to a high false positive rate (FPR), affecting the model's normal performance.

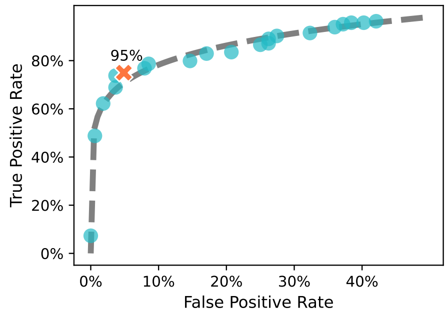


Figure 4: TPR-FPR trade-off on validation set.

The ablation study is conducted to determine the optimal threshold. By treating adversarial pairs as positive samples and clean pairs as negative samples, we plot the TPR-FPR curve with thresholds ranging from 80% to 100% in 1% increments, as shown in Figure 4. Ideally, we expect high TPR and low FPR (the upper left corner of the plot). Therefore, we selected τ when r equals 95% as the detection threshold of *CIDER*.

4 Experiment

In this section, we begin by outlining the configurations of our experiments, including the models, datasets, baselines, and evaluation metrics. We then evaluate the effectiveness and efficiency of *CIDER*, comparing with the baseline methods. Next, we discuss the trade-off between robustness and utility. Finally, we demonstrate the generalization of our method.

4.1 Configurations

Models. Note that *CIDER* is an auxiliary model that is independent to the VLMs. We use LLaVA to capture the semantic meaning of each modality, but *CIDER* can be plugged into any other VLMs. To demonstrate the effectiveness of *CIDER*, we test the detection and defense performance on four open-source VLMs, including LLaVA-v1.5-

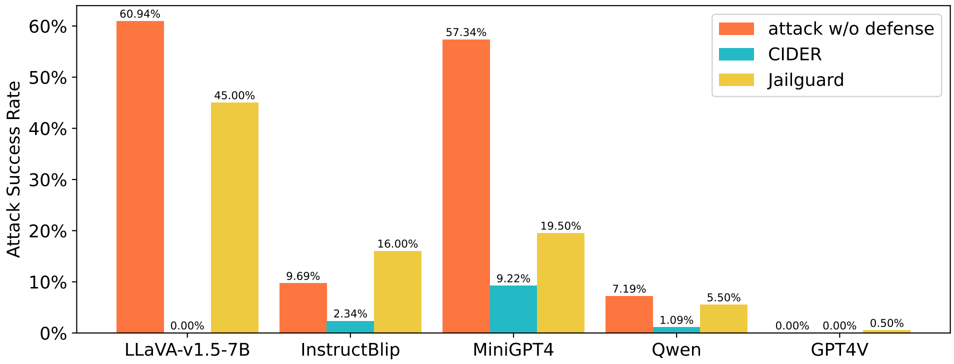


Figure 5: ASR of base VLM, defending with *CIDER* and defending with *Jailguard*

7B (Liu et al., 2024), MiniGPT4 (Zhu et al., 2023), InstructBLIP (Dai et al., 2024), and Qwen-VL (Bai et al., 2023), as well as the API-access VLM, GPT4V (Achiam et al., 2023).

Datasets. Similar to the dataset used in Section 2.2, we generate 800 adversarial text-image pairs utilizing the 160 harmful queries in Harmbench (Mazeika et al., 2024) and adversarial images provided by Qi et al. (2024). To further demonstrate *CIDER* will not destroy the original utilities on the normal queries, we also evaluate the utility of *CIDER* protected VLMs on MM-Vet benchmark (Yu et al., 2023), which examines 6 core vision language capabilities, including recognition, optical character recognition (OCR), knowledge, language generation, spatial awareness, and math.

Baseline and evaluation metrics. We use *Jailguard* (Zhang et al., 2023) as a baseline, which is a SoTA mutation-based jailbreak detection strategy that protects the VLMs at the inference stage. We involve four evaluation metrics to demonstrate the performance of defending methods from different aspects. From the perspective of the effectiveness of *CIDER*, we incorporate detection success rate (**DSR**) and Attack success rate (**ASR**). **DSR** represents the proportion of adversarial examples $\mathcal{D}$ that can be successfully detected:

$$\text{DSR} \overset{\text{def}}{=} \frac{1}{|\mathcal{D}|} \sum_{(q,x_{adv})\in\mathcal{D}} \mathbb{I}_{adv}((q, x_{adv})) \tag{4}$$

ASR is a standard evaluation metric indicating the proportion of samples that can successfully jailbreak VLM $\mathcal{F}$ and generate harmful contents, which can be stated as:

$$\text{ASR} \overset{\text{def}}{=} \frac{1}{|\mathcal{D}|} \sum_{(q,x_{adv})\in\mathcal{D}} \mathbb{I}_{harm}(\mathcal{G}(\mathcal{F}(q, x_{adv}))) \tag{5}$$

$\mathcal{G}$ refers to an LLM classifier (Mazeika et al., 2024) that determines the harmfulness of a response. $\mathbb{I}_{adv}$ and $\mathbb{I}_{adv}$ represent the adversarial and harmful indicator. In terms of efficiency, we measure the time cost of processing a single query. In addition, to evaluate the model utility on regular tasks, responses, we incorporate an online evaluator (MM-Vet-Evaluator, 2024) provided along with MM-Vet benchmark, which utilizes GPT-4 to generate a soft grading score from 0 to 1 for each answer.

4.2 Effectiveness

DSR. We first demonstrate the overall DSR that *CIDER* can achieve and compare it with the baseline method, *Jailguard*. Table 1 shows that *CIDER* achieves a DSR of approximately 80%, while the DSR of *Jailguard* varies, depending on the target VLMs. Note that *CIDER* is independent of the VLMs, thus the DSR does not vary with the choice of VLMs. However, *Jailguard*'s detection capability relies heavily on the model's safety alignment, so the DSR also varies. VLMs with good alignment achieve high DSR (e.g., GPT4V), while poorly aligned VLMs have relatively low DSR (e.g., InstructBLIP). In other words, *Jailguard* does not significantly enhance VLM robustness against adversarial jailbreak attacks, whereas *CIDER* does. Nonetheless, *CIDER* achieves a higher DSR than most of the *Jailguard* results, except *Jailguard* on GPT4V.

Method	detection success rate (↑)
Jailguard with LLaVA-v1.5-7B	39.50%
Jailguard with InstructBLIP	32.25%
Jailguard with MiniGPT4	69.50%
Jailguard with Qwen-VL	77.50%
Jailguard with GPT4V	94.00%
CIDER	79.69%

Table 1: DSR of *CIDER* and *Jailguard*

ASR. To evaluate the effectiveness of *CIDER*, we measure the decline in ASR after applying *CIDER*. Figure 5 compares the original ASR without defense (red bar), ASR after *CIDER* (blue bar) and

ASR after *Jailguard* (yellow bar). Note that, *Jailguard* is solely designed to detect jailbreak input. To ensure a fair comparison, we add an output module following *Jailguard*'s detection. Specifically, if *Jailguard* detects a jailbreak, it will refuse to respond, similar to *CIDER*. Otherwise, the original input will be processed by the VLM.

Across all models, defending with *CIDER* significantly reduces the ASR, yielding better results than the baseline. This indicates that *CIDER* effectively enhances the robustness of VLMs against optimization-based jailbreak attacks. The most notable improvements are seen in LLaVA-v1.5-7B, where ASR drops from 60% to 0%, and in MiniGPT4, from 57% to 9%. For VLMs with initially low ASRs, such as InstructBLIP and Qwen-VL, ASR is reduced to approximately 2% and 1% respectively. Another notable disadvantage of *Jailguard* is observed in models like GPT4V, InstructBLIP, and Qwen-VL, which already had strong safety alignment and resistance to adversarial attacks. In these cases, the use of *Jailguard* resulted in a slight increase in ASR.

4.3 Efficiency

Timely inference is crucial for safeguarding VLMs in real-world applications. Table 2 shows the time required to process a single input pair and generate up to 300 tokens with different VLMs, comparing no defense, *CIDER*, and *Jailguard*.

Model	Original	*CIDER*	*Jailguard*
LLaVA-v1.5-7B	6.39s	7.41s (1.13×)	53.21s (8.32×)
InstructBLIP	5.46s	6.48s (1.22×)	47.83s (8.76×)
MiniGPT4	37.00s	38.02s (1.03×)	313.78s (8.48×)
Qwen-VL	6.02s	7.04s (1.19×)	48.48s (8.05×)
GPT4V	7.55s	8.57s (1.16×)	61.04s (8.08×)

Table 2: Time cost to process a single pair of inputs.

CIDER surpasses *Jailguard* in efficiency, adding only 1.02 seconds per input pair on average, which is relatively acceptable compared to the original inference time. In contrast, *Jailguard* requires 8-9 times the original processing time. Additionally, *CIDER* detection is irrelevant to the number of generated tokens in the query answers. Therefore, CIDER does not cause additional overhead when increasing the number of generated tokens, ensuring the stability of *CIDER*'s efficiency.

4.4 Robustness-utility trade-off

To further demonstrate *CIDER*'s influence on the original utilities on normal queries, we also evaluate the utility of *CIDER* protected VLMs on MM-Vet benchmark, including recognition, OCR, knowledge, language generation, spatial awareness, and math. As shown in Figure 6, employing *CIDER* leads to an approximate 30% overall performance decline on normal tasks. Specifically, *CIDER* mostly affects the VLM's recognition, knowledge, and language generation capabilities, while it has minimal impact on OCR, spatial awareness, and math skills. We hypothesize that *CIDER*'s stringent decision-making process, which outright rejects tasks once an image is identified as adversarial, hampers the model's overall performance. To further illustrate the robustness-utility trade-off, we conducted an ablation study using denoised images as inputs for the adversarial images, termed *CIDER-de*. The result is relegated to Appendix B.

To find the optimal balance between safety and utility, we could design a more flexible rejection strategy, such as implementing multi-level thresholds for different types of content. This approach could reduce the negative impact on the model's functionality and we leave it to our future work.

4.5 Generalization

In the previous sections, we evaluated the ASR and DSR against adversarial examples generated by Qi et al. (2024). To assess the generalization of our defense method, which is crucial for its applicability to other attacks, we evaluate *CIDER* against another optimization-based jailbreak attack. We generated 800 adversarial pairs using ImgJP, as proposed by Niu et al. (2024). Table 3 presents the drop of ASR of *CIDER* on four open-source VLMs. The ASR for all VLMs dropped to below 4%, with Qwen reaching 0%. Additionally, *CIDER* achieved a DSR of 93.87% against ImgJP. These results demonstrate that *CIDER* effectively generalizes in defending against optimization-based adversarial attacks, highlighting its practical utility for real-world applications.

	Base	*CIDER*	
Model	ASR(%)	ASR(%)	Δ (%)
LLaVA-v1.5-7B	61.0	3.5	57.5
InstructBLIP	4.0	1.5	2.5
MiniGPT4	52.5	4.0	48.5
Qwen-VL	6.5	0.0	6.5

Table 3: Generalization of *CIDER* to ImgJP

5 Related Work

Vision Language Model. A typical Vision Language Model (VLM) consists of an image encoder (Dosovitskiy et al., 2020) to extract feature maps, a projector to align image modality information with

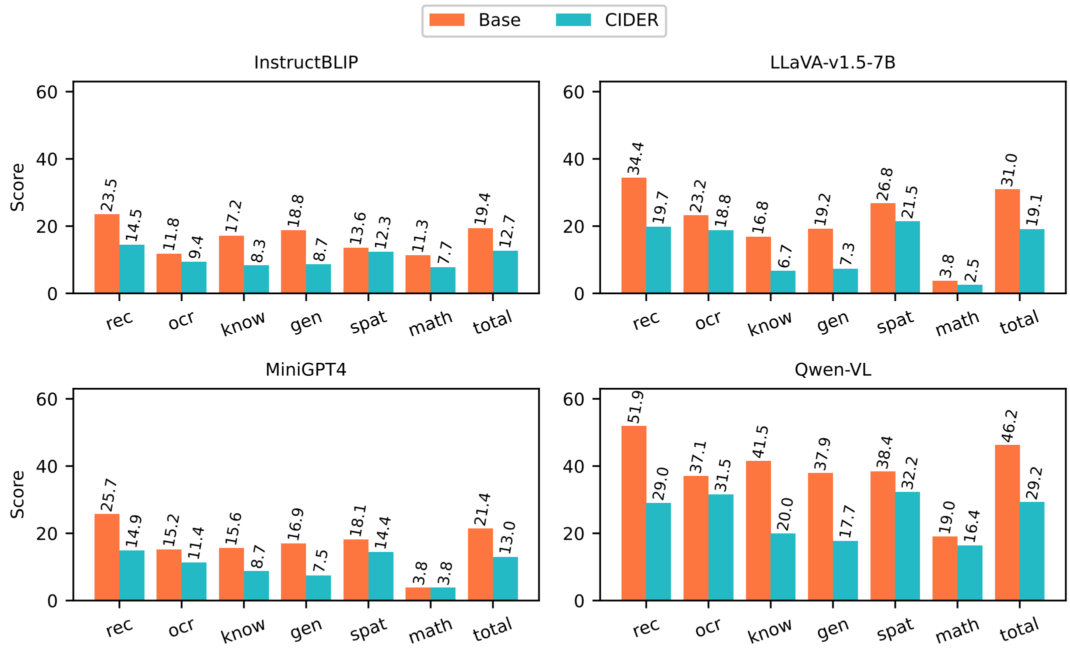


Figure 6: VLM performance with and without *CIDER* on MM-Vet.

text modality, and a Large Language Model (LLM) to integrate textual and visual input for generating responses. The impressive multimodal capabilities of these models have spurred significant research interest, leading to contributions from both academia and industry (Achiam et al., 2023; Liu et al., 2024; Zhu et al., 2023; Dai et al., 2024; Bai et al., 2023).

Jailbreaking VLMs. Incorporating visual information into the LLM framework significantly broadens its range of applications but also introduces new security vulnerabilities, complicating the security issues of VLMs. Besides transferring text jailbreak templates from LLMs to VLMs (Luo et al., 2024), effective strategies for jailbreaking VLMs include using gradient-based methods to generate adversarial images (Carlini et al., 2024; Qi et al., 2024; Niu et al., 2024), and submitting screenshots containing harmful instructions (Gong et al., 2023) or related images (Liu et al., 2023; Shayegani et al., 2023). This paper focuses on safeguarding VLMs against gradient-based adversarial image attacks, aiming to fortify VLMs against such sophisticated threats and ensure their robustness and reliability in practical applications.

Safeguarding VLMs. Various defense mechanisms have been proposed to address vulnerabilities in VLMs and enhance their security and robustness. These mechanisms can be categorized into proactive and reactive defenses based on their preventive and responsive nature. Proactive defenses aim to prevent attacks through techniques like adversarial training (Zong et al., 2024) and reinforcement learning (Chen et al., 2023) during the training phase. In contrast, reactive defenses focus on detecting attacks during the inference phase using methods such as (Wang et al., 2024a; Pi et al., 2024; Wang et al., 2024b). However, many of these methods require access to internal model parameters or rely on additional large models for implementation. Our approach prioritizes a reactive defense strategy for its practicality and ease of implementation. Notably, *Jailguard* (Zhang et al., 2023) is closely related to our work, as it detects jailbreak queries by analyzing variations in responses to perturbed inputs. However, *Jailguard*'s detection success heavily depends on the safety of the underlying LLM and involves significant computational costs.

6 Conclusion

In this work, we propose a plug-and-play cross-modality information detector, *CIDER*, which can effectively and efficiently defend against adversarial jailbreak attacks. Compared to previous methods, *CIDER* achieves superior defense performance, as evidenced by higher DSR and a significant decline in ASR, while greatly reducing processing time. We also evaluate the transferability of *CIDER* to other optimization-based adversarial attacks and demonstrate the robustness-utility trade-off in VLMs. In future research, we aim to improve *CIDER* by reducing the negative impact on VLM utilities to normal tasks. Additionally, it would be useful to develop defense mechanisms against non-optimization-based jailbreak attacks.

Limitations

We outline the limitations of our study as follows:

1. While *CIDER* is an effective, efficient, and user-friendly defense mechanism, it does impact VLM performance to some extent. We believe this is due to *CIDER*'s stringent handling of adversarial examples. In future work, we plan to implement multi-level thresholds to process adversarial examples with varying degrees of rigor, aiming to maintain robust defense capabilities without compromising VLM performance.

2. *CIDER* is specifically designed to defend against optimization-based adversarial jailbreak attacks, and its effectiveness against other types of jailbreak attacks is uncertain. Future research will explore *CIDER*'s effectiveness against these alternative attacks and develop corresponding defense strategies, aiming to enhance the overall security and resilience of VLMs against a wider array of adversarial threats.

Ethics Statement

Ensuring the security of Vision Large Language Models (VLMs) is crucial as they become more widely used in various applications. This paper introduces *CIDER*, a simple yet effective cross-modality information detector designed to defend against adversarial jailbreak attacks in VLMs. Our work significantly contributes to the field by providing a tool that mitigates known vulnerabilities and lays the groundwork for future improvements in safety measures. While *CIDER* marks significant progress, it doesn't make VLMs immune to all threats. Continuous evaluation and updates are crucial as VLMs evolve. By sharing *CIDER* and our findings, we aim to encourage ongoing research and collaboration, promoting advanced and secure VLMs. We are committed to addressing the ethical implications of VLM deployment, ensuring transparency, and prioritizing the responsible use of these technologies for societal benefit.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. 2024. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2023. Dress: Instructing large vision-language models to align and interact with humans via natural language feedback. *arXiv preprint arXiv:2311.10081*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2024. Masterkey: Automated jailbreaking of large language model chatbots. In *Proc. ISOC NDSS*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2023. Figstep: Jailbreaking large vision-language models via typographic visual prompts. *arXiv preprint arXiv:2311.05608*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. *arXiv preprint arXiv:2403.09792*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Xin Liu, Yichen Zhu, Yunshi Lan, Chao Yang, and Yu Qiao. 2023. Query-relevant images jailbreak large multi-modal models. *arXiv preprint arXiv:2311.17600*.

Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. 2024. Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. *arXiv preprint arXiv:2404.03027*.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

MM-Vet-Evaluator. 2024. MM-Vet Evaluator - a Hugging Face Space by whyu — huggingface.co. https://huggingface.co/spaces/whyu/MM-Vet_Evaluator. [Accessed 15-06-2024].

Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR.

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

Sayantan Pal, Maiga Chang, and Maria Fernandez Iriarte. 2021. Summary generation using natural language processing techniques and cosine similarity. In *International Conference on Intelligent Systems Design and Applications*, pages 508–517. Springer.

Kwangil Park, June Seok Hong, and Wooju Kim. 2020. A methodology combining cosine similarity with classifier for text classification. *Applied Artificial Intelligence*, 34(5):396–411.

Renjie Pi, Tianyang Han, 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.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. 2023. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Ke Ren, Botian Jiang, and Xipeng Qiu. 2024a. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance. *arXiv preprint arXiv:2401.11206*.

Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. 2024b. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. *arXiv preprint arXiv:2403.09513*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Xiaofei Xie, Yang Liu, and Chao Shen. 2023. A mutation-based method for multimodal jailbreaking attack detection. *arXiv preprint arXiv:2312.10766*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. 2024. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Visualization of denoising

Figure 7 presents an example of an adversarially perturbed image, showing the effects of denoising it after 100, 200, and 300 iterations.

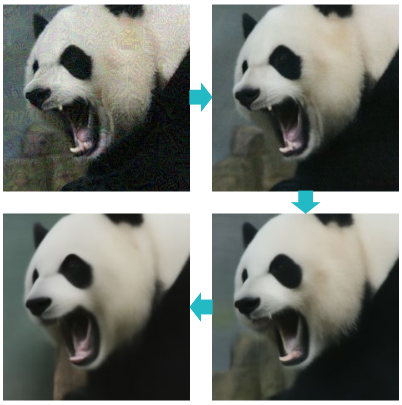


Figure 7: An example of the denoising procedure.

B Ablation study on robustness-utility trade-off

To further illustrate the robustness-utility trade-off, we perform an ablation study using denoised images as inputs for adversarial images, referred to as *CIDER-de*. Figure 8 shows the ASR of *CIDER-de* and Figure 9 shows the MM-Vet score of it. It can be observed that using *CIDER-de* hardly impacts the utility of the VLM. However, this comes at the expense of greatly diminished defensive effectiveness.

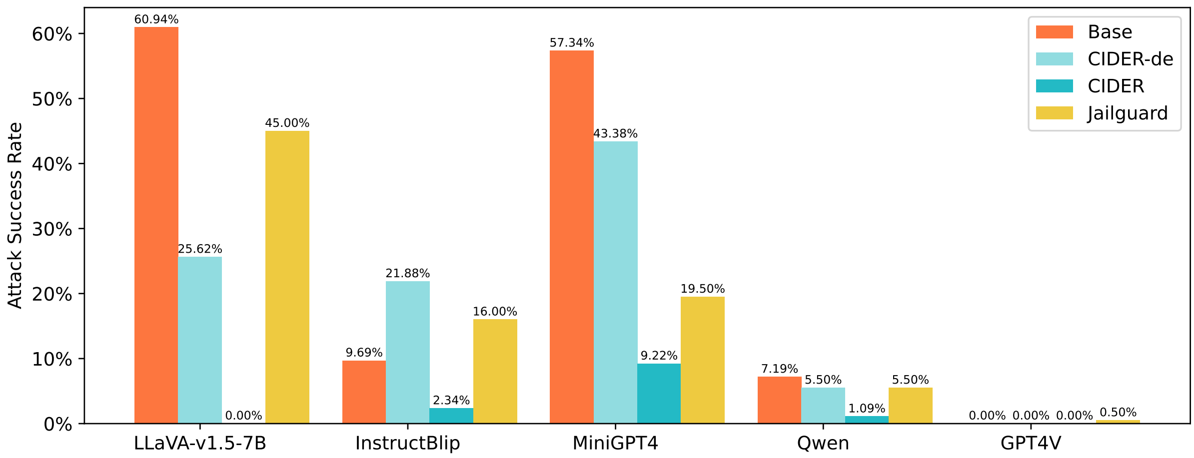


Figure 8: ASR of base VLM, defending with *CIDER-de*, *CIDER* and *Jailguard*

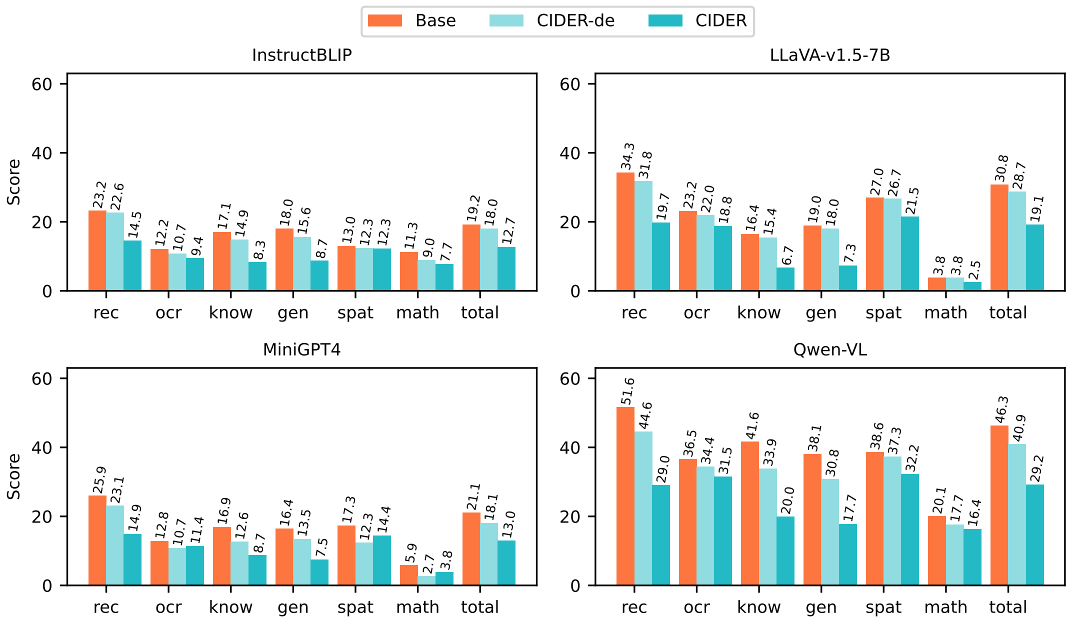


Figure 9: MM-Vet score of base VLM, defending with *CIDER-de*, *CIDER* and *Jailguard*