Safeguarding Vision-Language Models Against Patched Visual Prompt Injectors

Anonymous ACL submission

Abstract

Large language models have become increasingly prominent, also signaling a shift towards multimodality as the next frontier in artificial intelligence, where their embeddings are harnessed as prompts to generate textual content. Vision-language models (VLMs) stand at the forefront of this advancement, offering innovative ways to combine visual and textual data for enhanced understanding and interaction. However, this integration also enlarges the attack surface. Patch-based adversarial attack is considered the most realistic threat model in physical vision applications, as demonstrated in many existing literature. In this paper, we propose to address patched visual prompt injection, where adversaries exploit adversarial patches to generate target content in VLMs. Our investigation reveals that patched adversarial prompts exhibit sensitivity to pixel-wise randomization, a trait that remains robust even against adaptive attacks designed to counteract such defenses. Leveraging this insight, we introduce SmoothVLM, a defense mechanism rooted in smoothing techniques, specifically tailored to protect VLMs from the threat of patched visual prompt injectors. Our framework significantly lowers the attack success rate to a range between 0% and 5.0% on two leading VLMs, while achieving around 67.3% to 95.0% context recovery of the benign images, demonstrating a balance between security and usability.

1 Introduction

With the advent of large language models (LLMs) such as GPT and Claude (Achiam et al., 2023), we have witnessed a transformative wave across numerous domains, guiding in an era where artificial intelligence (AI) closely mirrors human-like understanding and generation of language. This progress has further paved the way for the integration of multi-modality. Among them, vision-language models (VLMs) (Zhang et al., 2024; Chen et al., 2023) are emerging, which blend visual understanding with textual interpretation, offering richer interactions. However, as these VLMs grow more sophisticated, they also become targets for a wider range of adversarial threats. Attacks that involve altered visual prompts pose significant concerns, as they manipulate the models' responses in realistic ways that are hard to mitigate.

Many alignment studies focusing on LLMs appear to mitigate the spread of harmful content significantly (Ouyang et al., 2022; Bai et al., 2022; Go et al., 2023; Korbak et al., 2023). However, recent studies have exposed several vulnerabilities, known as *jailbreaks* (Chao et al., 2023), which circumvent the safety measures in place for contemporary LLMs. Identifying and addressing these weaknesses presents significant challenges. They stand as major obstacles to the wider adoption and safe deployment of LLMs, impacting their utility across various applications. The integration of visual prompts arguably further enlarges the attack surface, introducing an additional layer of complexity for securing these systems. As models increasingly interpret and generate content based on both texts and images, the potential for exploitation through visually manipulated inputs escalates. This expansion not only necessitates advanced defensive strategies to safeguard against such innovative attacks but also underscores the urgent need for ongoing research and development in AI safety measures.

Although a variety of research has been conducted to study the robustness of jailbreak robustness of LLMs, there is a lack of practical formulation of "visual jailbreaks" as the emergence of VLMs. We thus first rigorously transform the existing adversarial attacks in VLMs (Zhu et al., 2023; Liu et al., 2023a) as patched visual prompt injectors since patch-based attacks have been demonstrated as the most realistic attacks in the physical world. As the ultimate goal of VLMs is text generation,

the attack formulation is different from classic vision tasks such as classification (Krizhevsky et al., 2017) and object detection (Zhao et al., 2019) that target one-time logit outputs. There are two types of adversarial attacks for VLM that are prominent. (Shayegani et al., 2023) propose to optimize the input visual prompt to mimic the harmful image in the embedding space, while (Qi et al., 2023) directly optimize the visual prompt to generate a given harmful content, as detailed in § 3.1. We adopt both optimization methods but update the attack interface from ℓ_∞-bounded manipulations to adversarial patches. This vulnerability not only undermines the reliability of these systems but also poses significant security risks, especially in critical applications. The need to safeguard against such vulnerabilities is not just imperative for the integrity of VLMs but is also of paramount importance for the trust and widespread adoption of LLMs and VLMs.

In this paper, we further introduce SmoothVLM, a novel framework designed to fortify VLMs against the adversarial threat of patched visual prompt injectors. SmoothVLM is designed to naturally enhance the robustness against visual jailbreaks while preserving the interpretative and interactive performance of VLM agents. We first identify an intriguing property of the patched visual prompt injectors, that is, the success of the injection is extremely sensitive to the random perturbation of the adversarial patch even under adaptive attacks. This could be attributed to the design of VLM. Therefore, by integrating majority voting with random perturbed visual prompts, our approach can defend the hidden visual prompt injectors with high probability, effectively rendering them impotent in manipulating model behavior. SmoothVLM has significantly reduced the attack success rates of patched visual prompt injectors on popular VLMs. Specifically, for both `llava-1.5` and `miniGPT4`, SmoothVLM can reduce the attack success rate (ASR) to below 5%, and with a sufficiently large perturbation, it can further decrease the ASR to approximately 0%.

Our contributions are manifold and significant:

• We present a comprehensive analysis of the vulnerabilities of current VLMs to patched visual prompt attacks and propose SmoothVLM, a novel defense mechanism that leverages randomized smoothing to mitigate the effects of adversarial patches in VLM.

• We demonstrate through extensive experiments that SmoothVLM significantly outperforms existing defense strategies, achieving state-of-the-art results in both detection accuracy and model performance retention.

• By addressing the susceptibility of VLMs to adversarial patch-based manipulations even under adaptive attacks, SmoothVLM represents a significant step forward in the development of secure multimodal LLMs.

2 Related Work

In this section, we review a few related topics to our study, including attacks and defenses for prompt injection and adversarial patches.

2.1 Prompt Injection

Prompt engineering is emerging in the era of LLM. At the core of prompt injection attacks lies the adversarial ability to manipulate the output of LLMs by ingeniously crafting input prompts. (Zhang et al., 2020) provided an early exploration of these vulnerabilities in LLMs, demonstrating how attackers could insert malicious prompts to alter the behavior of AI systems in text generation tasks. Their work highlighted the need for robust input validation and sanitization mechanisms to mitigate such threats. (Zou et al., 2023) conducted an empirical study on the impact of prompt injection attacks on various commercial AI systems, uncovering a wide range of potential exploits, from privacy breaches to the spread of misinformation. Recently, prompt injection attacks extended to VLMs (Bailey et al., 2023). In particular, (Shayegani et al., 2023) and (Qi et al., 2023) propose to modify the pixels of the visual prompts to fool VLMs that generate target contents, as detailed in § 3.1.

2.2 Adversarial Patches

The advent of adversarial patch attacks has prompted significant research interest due to their practical implications for the security of machine learning systems, especially those relying on computer vision. This section reviews key contributions to the field, spanning the initial discovery of such vulnerabilities to the latest mitigation strategies. (Brown et al., 2017) pioneered the exploration of adversarial patches by demonstrating that strategically designed and placed stickers could deceive an image classifier into misidentifying objects. Following the initial discovery, researchers sought to refine the techniques for generating and deploying

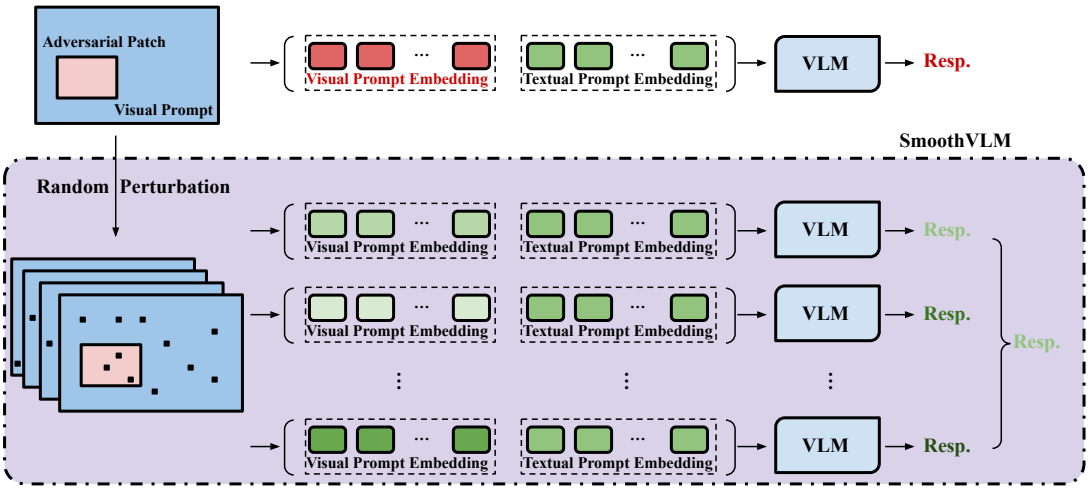


Figure 1: **Our SmoothVLM Certified Defense Pipeline.**

adversarial patches. (Nguyen et al., 2015) introduced an optimization-based method to create more effective and efficient adversarial patches. The practical implications of adversarial patch attacks have been a focus of recent studies. (Chahe et al., 2023) investigated the effects of adversarial patches on autonomous vehicle systems, revealing potential threats to pedestrian detection mechanisms. In response to these vulnerabilities, the community has developed various defensive strategies. (Strack et al., 2023) proposed a defense mechanism based on anomaly detection and segmentation techniques to identify and ignore adversarial patches in images. (Zhou et al., 2021) explored the integration of adversarial examples, including patches, into the training process. Certified defenses also extend to adversarial patches. Xiang et al. (2022, 2024) developed certified methods to mitigate adversarial patches. However, the current certified methods are only limited to defending against small adversarial patches.

3 SmoothVLM

In this section, we present our SmoothVLM framework to defend against adversarial patches for visual prompt injection. We first introduce our threat model of patched visual prompt injection.

3.1 Patched Visual Prompt Injection

We have witnessed the emergence and potential of large (vision) language models (LLM and VLM) in the past year and they have also introduced new attack vectors such as prompt injection (Liu et al., 2023b; Greshake et al., 2023; Shi et al., 2024). Different from classic adversarial attacks targeting fundamental tasks such as classification and object detection that target logit space manipulation, prompt injection aims to induce language models to generate specific texts. A VLM incorporates multimodality by treating images as visual prompts to an appended LM, enhancing the model's comprehension of instructions. To inject a target concept into the VLM, there are currently two primary optimization methods. Firstly, (Shayegani et al., 2023) optimized the distance between embeddings of the adversarial and target images (*e.g.*, a bomb or a gun), *i.e.*, $\arg\min_{x_{\text{adv}}} d(H_{\text{adv}}, H_{\text{target}})$, where H denotes the visual embedding ingested by the LM, ensuring the LM cannot discern between adversarial and target image embeddings as long as the distance $d(H_{\text{adv}}, H_{\text{target}})$ is minimal. Secondly, (Qi et al., 2023) proposed using a corpus of harmful text as the target to optimize the image input, *i.e.*, $\arg\min_{x_{\text{adv}}} \sum_{i=1}^{m} -\log(p(y_i|[x_{\text{adv}}, \emptyset]))$, where $Y_{\text{adv}} := \{y_i\}_{i=1}^{m}$ represents the corpus of chosen content. Both studies leverage the ℓ_∞ norm across the image's pixel attack surface. However, DiffPure (Nie et al., 2022) and its follow-ups have shown that such threat models can be mitigated through diffusion purification, with many subsequent studies (Wang et al., 2023; Lee and Kim, 2023; Zhang et al., 2023; Xiao et al., 2022) confirming its effectiveness against both ℓ_2 and ℓ_∞-based attacks. Therefore, we propose adapting these two attack strategies to use adversarial patches with an ℓ_0 constraint on size, which both maintains the stealthiness of the attack and the original image's semantics. Patch attacks are also demonstrated to be much more physically achievable in the real world. We denote the two attack methods by their titles Jailbreak In Pieces (JIP) and Visual Adversarial Examples (VAE), respectively.

Specifically, our threat model assumes that an adversarial patch $P_{\text{m}\times\text{n}}^{[i,j]}$, of size m $\times$ n, is placed such

that its bottom-left corner aligns with the pixel at coordinates $[i, j]$ in the original image $I_{\text{h}\times\text{w}}$. Here, $I_{\text{h}\times\text{w}}$ denotes the original image of size h $\times$ w. The resultant adversarial example is denoted as $I_{\text{adv}} = I \oplus P$, where $\oplus$ signifies the operation used to overlay the patch onto the image. We still leverage the two white-box optimization methods mentioned above in our evaluation.

3.2 Randomized Defense Against Patched Visual Prompt Injection

As introduced in § 2, randomized defenses are significant within the adversarial robustness community. Drawing inspiration from SmoothLLM (Robey et al., 2023), our investigations reveal vulnerabilities to randomized perturbations in the pixel space of the patched visual prompt injectors. In preliminary experiments on the latest `LLaVA-v1.5-13b` model (Liu et al., 2024, 2023a), which accepts 224×224 images. Since adversarial optimization is computationally expensive, we use 300 adversarial examples and ensure that the attacks successfully launch on the images. We set the patch size to 112×112, which is $\frac{1}{4}$ of the original image size. We applied three randomized perturbation methods to the adversarial patch area in the images: *mask*, *swap*, and *replace*. The *mask* operation randomly sets $q\%$ of the pixels in the adversarial patch to zero across all channels. For *swap*, $q\%$ of the pixels' RGB channels are randomly interchanged. The *replace* operation substitutes $q\%$ pixels with random RGB values uniformly sampled.

As mentioned earlier, JIP and VAE are both optimized to generate Y_{adv} or its equivalents. Similar to Robey et al. (2023), we leverage an oracle language model (e.g., GPT4 (Achiam et al., 2023)) to deterministically predict whether the attack goal is achieved. Therefore, a successful attack (SA) is defined as:

$$\text{VPI}(Y_{\text{pred}}) \doteq \text{OracleLLM}(Y_{\text{pred}}, Y_{\text{adv}}) = \begin{cases} 1, & \text{if } Y_{\text{pred}},\ Y_{\text{adv}} \text{ are synonymous} \\ 0, & \text{otherwise.} \end{cases}$$

Figure 2 shows the attack success rates (ASR) of our preliminary measurement study, which illustrate the instability of the patched visual prompt injection attacks. We found that among the three types of perturbation, random masking can consistently and effectively mitigate adversarial patch attacks with a sufficient amount of perturbation. Particularly, random masking reduces the ASR to around 5%. We denote our finding as **visual q-instability with probability error** ϵ.

3.3 Expectation over Transformation (EOT) Adversary

Randomization-based defense solutions can be oftentimes broken by the expectation of transformations (EOT) attack (Athalye et al., 2018) in the existing literature. However, we argue that adaptive attacks are more challenging to launch in the era of multimodal language models, especially under realistic threat models. We empirically demonstrate that the EOT attacks are ineffective in breaking the q-instability of VLM under our patched visual prompt injection setup. Specifically, we assume the attacker is aware of the exact random perturbation (masking here since it is demonstrated to be the most effective one in Figure 2) added for defense; thus, the optimization becomes

$$\arg\min_P \mathbb{E}_{t\sim Mask()} \ell(\text{VLM}([I \oplus t(P); \emptyset]), Y_{adv}) \quad (1)$$

where t follows the distribution of our random masking, and $\emptyset$ means an empty text prompt. As shown in Figure 3, the ASRs of the adaptive attacks are extremely low.

Definition 1. ***Definition 1 (Visual q-instability with probability error ϵ).*** *Given a VLM and the adversarial example $I^{adv} = I \oplus P$, we apply the mask operation Mask() to randomly zero out q% pixels in the adversarial patch P, obtaining $P' = Mask(P)$. Here we call the P is* ***visual q-unstable with probability error*** *ϵ if for any*

$$\ell_0(P', P) \geq \ulcorner qmn \urcorner \quad (2)$$

there exists a small constant probability error ϵ such that

$$Pr[(VPI \circ VLM)([I \oplus P'; \emptyset]) = 0] \geq 1 - \epsilon \quad (3)$$

where ***q*** *is the instability parameter.*

The reason could be attributed to the characteristics of the VLM task, which is essentially the ***next-token prediction***. As introduced earlier, the attack goal for classic vision tasks is to manipulate a single/few output(s) from the one-time model inference, so the room for adversarial optimization is arguably large. However, the optimization goal is either too harsh or implicit for next-token prediction as it usually involves a sequence of outputs with many iterations.

Figure 3 illustrates that EOT is hard to optimize. For example, the loss function for JIP is the mean square error, the ℓ_2 distance in the em-

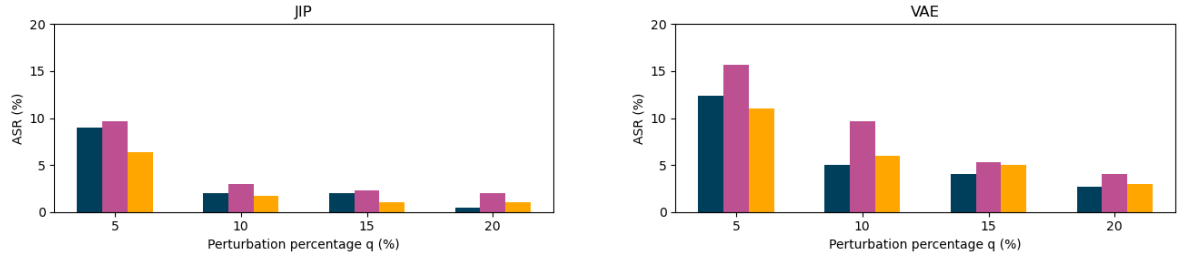


Figure 2: **Validation of *q*-instability on Patched Visual Prompt Injection.** We random perturb $q\%$ pixels in the adversarial patch with three methods: *mask*, *swap*, and *replace*. The ASRs of JIP and VAE are both 100%.

bedding space. In LLaVA, there are 576 token embeddings, and a successful attack needs to make the ℓ_2 between less than 0.4, which is far more difficult than the classic logit space optimization when combined with randomized defense. In the VAE, the optimization directly targets generating harmful content. As there are usually 8000 iterations, which already take 0.5 hours to optimize, EOT will make the complexity at least an order higher, rendering the optimization intractable.

3.4 SmoothVLM Design

We formally introduce our design of SmoothVLM. Similar to SmoothLLM and other randomized smoothing-based methods, there are two key components: (1) distribution procedure, in which N copies of the input image with random masking are distributed to VLM agents for parallel computing, and (2) aggregation procedure, in which the responses corresponding to each of the perturbed copies are collected, as depicted in Figure 1.

3.4.1 Distribution Procedure

The first step in our SmoothVLM is to distribute N visual prompts to the VLM agent and this step can be computed in parallel if resources are allowable. As illustrated in § 3.2, when the patched visual prompt injectors P are ***q*-unstable with probability error** ϵ, the probability of a successful defense is no lower than 1-ϵ. However, in most situations, we do not know where the adversarial patch is attached to the prompt (i, j), so we can only apply random perturbation to the whole visual prompt. Shown as following Assumption 2, here we assume that the masking pixels out of adversarial patches would at most lead to a decreasing of μ on the probability of a successful defense. Since masking is the most stable perturbation shown in Figure 2, we will use masking in the rest of this paper. Specifically, we randomly mask $q\%$ of the pixels for each distributed visual prompt.

Definition 2. ***Assumption 2 (Visual q-instability for Visual Prompt I^{adv}).*** *Given a VLM and the adversarial example $I^{adv} = I \oplus P$, we apply the mask operation Mask() to randomly zero out q% pixels in the visual prompt I^{adv}. Here we denote $I' = Mask(I)$, $P' = Mask(I)_{|P}$, which means then projection of the $Mask(I)$ on the position of the adversarial patch P. If P is visual q-unstable with probability error ϵ, we assume that*

$$Pr[(VPI \circ VLM)([I' \oplus P'; \emptyset]) = 0] \geq 1 - \epsilon - \mu \quad (4)$$

The consideration of I' instead of I would at most lead to a decreasing of μ in the probability of a successful defense.

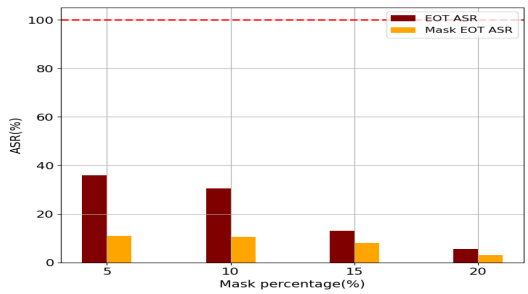
3.4.2 Aggregation Procedure

The second step in our SmoothVLM is to collect and aggregate the responses from the first step. As mentioned earlier, as we do not know the location of the adversarial patch, it is impossible to guarantee high defense probability with one masked input. Therefore, rather than passing a single perturbed prompt through the LLM, we obtain a collection of perturbed prompts with the same perturbation rate p, and then aggregate the predictions of this collection. The motivation for this step is that while one perturbed prompt may not mitigate an attack, as we observed in Figure 3, on average, perturbed prompts tend to nullify jailbreaks. That is, by perturbing multiple copies of each prompt, we rely on the fact that on average, we are likely to flip characters in the adversarially-generated portion of the prompt.

Based on the above two insights, here we give the formal definition of SmoothVLM in Definition 3 and include the details about the algorithm in Algorithm 1.

3.5 Probability Guarantee of SmoothVLM

To understand the robustness of SmoothVLM against VPI, we also provided a thorough analysis of the defense success probability (DSP) $\text{DSP}([I; \emptyset])$. Here we give the result of DSP in Proposition 4. Detailed computation process is included in Appendix A.



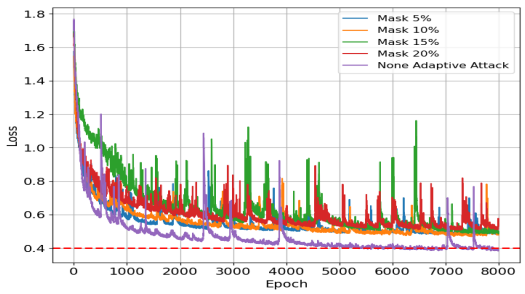


Figure 3: **Validation of q-instability on EOT Attack.** The left figure plots the ASR of EOT adversarial examples w/wo $q\%$ pixels masked. The red dashed line at the ASR of 100% denotes that all the original samples are successfully attacked. "Mask EOT ASR" means that after we get adversarial examples with EOT, we further mask $q\%$ pixels as our defense. For the right subplot, we plot the training loss with 8000 epochs (requiring ∼50 mins on one A100), "mask $q\%$" means we mask $q\%$ in EOT attack process, "none adaptive attack" means normal patch attack. The dotted red line in the right figure indicates the required loss for a successful adversarial optimization, *i.e.*, loss=0.4. **The two figures demonstrate that EOT is extremely hard to optimize and subject to our identified q-instability as well.**

Algorithm 1 SmoothVLM

Require: Visual Prompt I
Input: Number of Samples N, Perturbation Rate p
for $j = 1 \dots N$ **do**
  $I_j \leftarrow \text{RandomPerturbation}(I, p)$
  $R_j \leftarrow \text{VLM}([I_j; \emptyset])$
end for
$A \leftarrow$ **function** MAJORITYVOTE$(R_1, \dots, R_j)$
$j^* \sim \text{Unif}(\{j \in [N] \mid R_j = A\})$
return R_{j^*}

function MAJORITYVOTE$(R_1, \dots, R_N)$:
  return $\mathbb{I}\left[\frac{1}{N}\sum_{j=1}^{N} \text{VPI}(R_j) \geq \frac{1}{2}\right]$
end function

Definition 3. ***Definition 3 (SmoothVLM).*** *Let a visual prompt (image) I and a distribution $\mathbb{P}_p(I)$ over randomly masked copies of I be given. Let $I_1, ..., I_N$ be drawn i.i.d. from $\mathbb{P}_p(I)$*

$$A \doteq \mathbb{I}[\frac{1}{N}\sum_{j=1}^{N}(VPI \circ VLM)(I_j) > \frac{1}{2}] \tag{5}$$

and our SmoothVLM is defined as

$$SmoothVLM([I; \emptyset]) \doteq VLM([\mathbf{I}; \emptyset]) \tag{6}$$

where $\mathbf{I}$ *represents the image agrees with majority voting,* $(VPI \circ VLM)([\mathbf{I}; \emptyset]) = A$.

4 Evaluations

In this section, we conduct a comprehensive evaluation of our proposed SmoothVLM, which mainly show the results of two attack methods on `llava-1.5`. Specifically, we leverage `Vicuna-30B` as a proxy function for $VPI()$.

4.1 Injection Mitigation

We conduct extensive experiments on our SmoothVLM using two attack methods on `llava-1.5` and `miniGPT4` and mainly present the results for `llava-1.5` in this section. Specifically, we utilized 300 adversarial examples, all verified against the corresponding VLM model to ensure a successful attack. Figure 5 displays various values of the number of samples N and the perturbation percentage q. Generally, the attack success rate (ASR) significantly decreases as both q and N increase. It's observed that even with a minimal perturbation q=5%, increasing the number of samples N leads to a substantial drop in ASR. We also find that our certified defense performance converges with q=20%. We also demonstrate that the random perturbation with q=20% will not hurt the image understanding for VLMs. For the three methods, it is clear that the ASR of the *swap* method is significantly higher than others, consistent with the results in our q-instability analysis.

4.2 Visual Prompt Recovery

A successful defense in the context of VLM is to recover the semantics of the input visual prompt. In this section, we demonstrate our SmoothVLM does not hurt the image understanding of VLMs. We evaluate the similarity between the responses generated from the smoothed image and the original image to determine if our SmoothVLM can fully recover the adversarial example to its original state. We define and leverage distortion rate (DR) as a proxy to quantify the discrepancy between the responses to the original image and those after applying SmoothVLM. Here we use `Vicuna-30B` as a metric function to calculate the DR as more powerful tools like GPT4 will reject harmful contents. In Figure 6, a smaller value of q = 5% results in a higher DR, suggesting that lower perturbation levels are insufficient to fully eliminate the concealed harmful context within the visual prompt and fail to restore the original visual semantics. In contrast, although q=20% indicates a higher perturbation

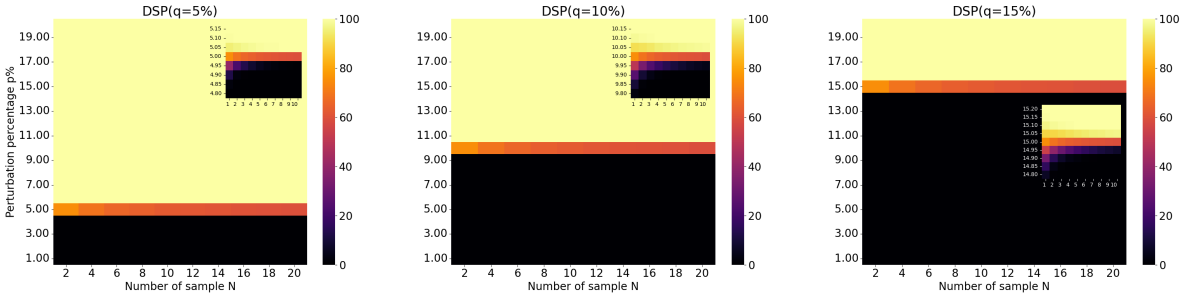


Figure 4: **Robustness Guarantee on Patched Visual Prompt Injection.** We plot the probability DSP($[I \oplus P; \emptyset]$) that SmoothVLM will consider attacks as a function of the number of samples N and the perturbation percentage q; warmer colors denote larger probabilities. From left to right, probabilities are calculated for ten distinct values of the instability parameter k from 2 to 20. Each subplot reveals the pattern: with the increase in both N and q, there is an increasing DSP.

Proposition 1. ***Proposition 4 (Defense Success Probability of SmoothVLM).*** *Assume that an adversarial patch $P \in [0,1]^{m \times n \times 3}$ for the visual prompt $I_{h \times w} \in [0,1]^{h \times w \times 3}$ is* ***visual q-unstable with probability error*** *ϵ. Recall that N is the number of randomly masked samples drawn i.i.d. and p is the perturbation percentage on the whole visual prompt. The DSP is derived as follows:*

$$DSP([I \oplus P; \emptyset]) = Pr[(VPI \circ SmoothVLM)([I \oplus P; \emptyset]) = 0] \tag{7}$$

$$= \sum_{t=\lceil N/2 \rceil}^{N} \binom{N}{t} \alpha^t (1-\alpha)^{N-t} \tag{8}$$

$$where \quad \alpha \geq (1 - \epsilon - \mu) \sum_{k=\lceil qmn \rceil}^{mn} \binom{mn}{k} p^k (1-p)^{mn-k} \tag{9}$$

level, we observe that the powerful reasoning ability of VLMs can still understand and narrate the visual prompt very well. Therefore, SmoothVLM achieves both good defensive performance and usefulness in real practice.

4.3 Efficiency

Smoothing-based certified defenses unavoidably increase the inference runtime (Xiang et al., 2022; Cohen et al., 2019). In this section, we demonstrate that our SmoothVLM is efficient from two perspectives. We first compare the efficiency of the attack method (JIP) with our defense strategy (SmoothVLM). The JIP method focuses on reducing the loss in the embedding space of the VLM, offering a more time-efficient approach compared to the VAE, which computes the loss over the entire VLM and is thus significantly more time-consuming. Even though we select the JIP method, the optimization of a single adversarial example using JIP still requires around **30** minutes using one A100 (`openai/clip-vit-base-patch32`). In contrast, with the usage of a large model, `Vicuna-30B`, our method takes less than 1 minute under $N = 10$, making our most resource-intensive defense approach more than 30x faster than the fastest attack method. As shown in Figure 5, the ASRs are consistently under 5% with $N = 10$, indicating that SmoothVLM successfully guards the models. On the other hand, our method indeed offers substantial efficiency improvements. Compared to the classic randomized smoothing method (Cohen et al., 2019), which necessitates 100,000 runs for a single input, our innovative smoothing approach requires only 10-20 runs per visual prompt. This represents a 10,000-fold increase in efficiency over state-of-the-art methods. Therefore, we believe SmoothVLM effectively balances efficiency and usefulness.

4.4 Compatibility

First, our SmoothVLM framework is agnostic to the specific VLM used. Our theoretical results generalize to all existing VLMs with slim modifications of hyperparameters. Besides, we further demonstrate the compatibility of our defense method under dual-patch attacks. Especially, we assume a stronger threat where attacks can put two adversarial patches and the total area of the patches remains 112×112. we implement two kinds of dual patch attacks (JIP, VAE) and report the defense performance of *mask*, *swap*, and *replace* on 300 adversarial attack examples. As shown in In Figure 7, the experimental data indicate a consistent trend where the ASR decreases as the number of sam-

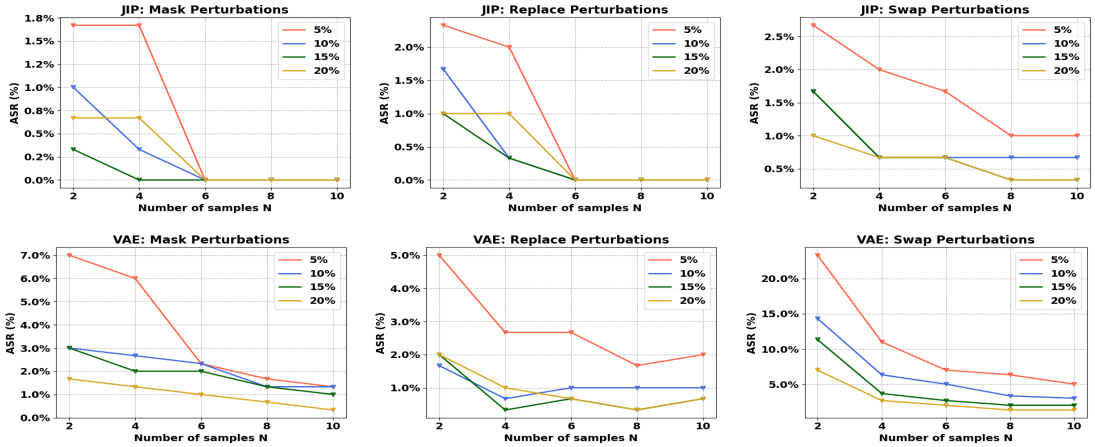


Figure 5: **Injection Mitigation Effectiveness of SmoothVLM.** We plot the ASR of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$.

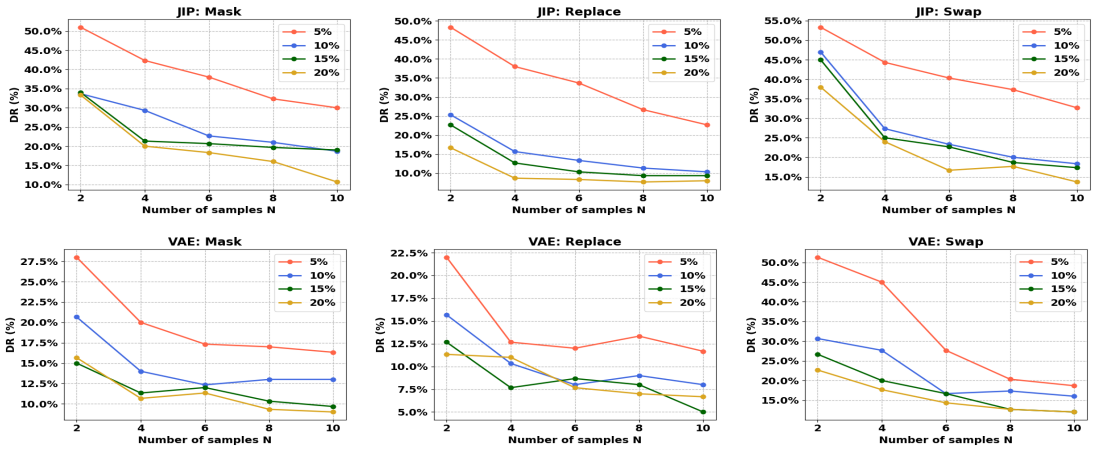


Figure 6: **Visual Prompt Recovery.** We plot the distortion rate of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$.

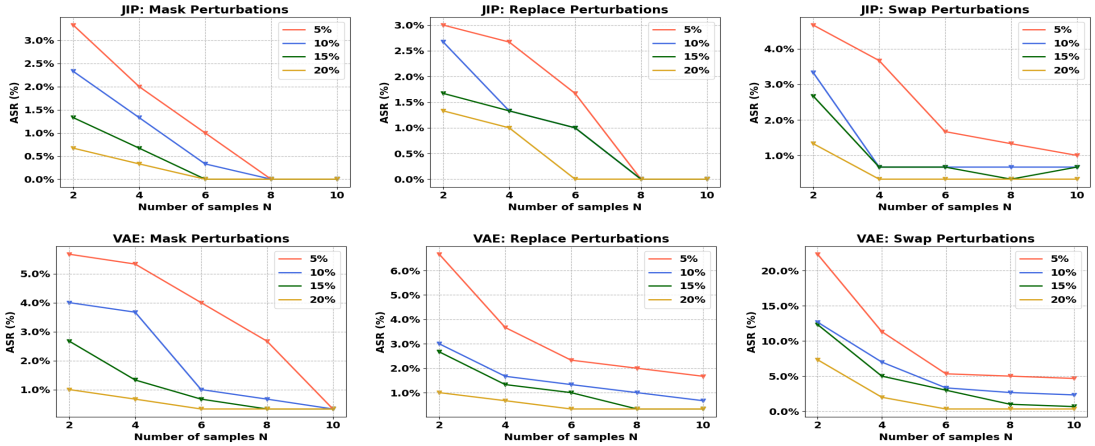


Figure 7: **Dual-Patched Injection Mitigation Effectiveness of SmoothVLM.** We plot the ASR of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$.

ples increases for all perturbation strategies: *mask*, *swap*, and *replace*, across different perturbation intensities.

5 Conclusion

In conclusion, we have presented SmoothVLM, a certifiable defense mechanism that effectively addresses the patched visual prompt injectors in vision-language models. SmoothVLM significantly reduces the success rate of attacks on two leading VLMs under 5%, while achieving up to 95% context recovery of the benign images, demonstrating a balance between security, usability, and efficiency.

6 Limitations

We acknowledge certain limitations within our SmoothVLM. Despite our efforts to fortify it using the expectation over transformation (EOT) approach as adaptive attacks, our defense mechanism primarily addresses patch-based visual prompt injections and remains vulnerable to ℓ_p based adversarial attacks. The reason is that we found adaptive formulations of the ℓ_p based adversaries are extremely challenging to tackle. Therefore, there is also a potential risk that our SmoothVLM may fail under stronger attacks beyond our threat model. We envision our study as an initial step toward establishing certified robustness in VLMs, laying a foundation for future research to build upon.

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*.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2018. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

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

Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. 2017. Adversarial patch.

Amirhosein Chahe, Chenan Wang, Abhishek Jeyapratap, Kaidi Xu, and Lifeng Zhou. 2023. Dynamic adversarial attacks on autonomous driving systems. *arXiv preprint arXiv:2312.06701*.

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*.

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

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. 2023. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, (6):84–90.

Minjong Lee and Dongwoo Kim. 2023. Robust evaluation of diffusion-based adversarial purification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 134–144.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

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

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023b. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*.

Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436. IEEE.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Mengdi Wang, and Prateek Mittal. 2023. Visual adversarial examples jailbreak large language models. *arXiv preprint arXiv:2306.13213*.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. 2023. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. *Preprint*, arXiv:2307.14539.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2024. Optimization-based prompt injection attack to llm-as-a-judge. *arXiv preprint arXiv:2403.17710*.

Lukas Strack, Futa Waseda, Huy H Nguyen, Yinqiang Zheng, and Isao Echizen. 2023. Defending against physical adversarial patch attacks on infrared human detection. *arXiv preprint arXiv:2309.15519*.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. 2023. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, pages 36246–36263. PMLR.

Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. 2022. {PatchCleanser}: Certifiably robust defense against adversarial patches for any image classifier. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2065–2082.

Chong Xiang, Tong Wu, Sihui Dai, Jonathan Petit, Suman Jana, and Prateek Mittal. 2024. Patchcure: Improving certifiable robustness, model utility, and computation efficiency of adversarial patch defenses. *Preprint*, arXiv:2310.13076.

Chaowei Xiao, Zhongzhu Chen, Kun Jin, Jiongxiao Wang, Weili Nie, Mingyan Liu, Anima Anandkumar, Bo Li, and Dawn Song. 2022. Densepure: Understanding diffusion models towards adversarial robustness. *arXiv preprint arXiv:2211.00322*.

Jiawei Zhang, Zhongzhu Chen, Huan Zhang, Chaowei Xiao, and Bo Li. 2023. {DiffSmooth}: Certifiably robust learning via diffusion models and local smoothing. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 4787–4804.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232.

Dawei Zhou, Tongliang Liu, Bo Han, Nannan Wang, Chunlei Peng, and Xinbo Gao. 2021. Towards defending against adversarial examples via attack-invariant features. In *International conference on machine learning*, pages 12835–12845. PMLR.

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*.

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

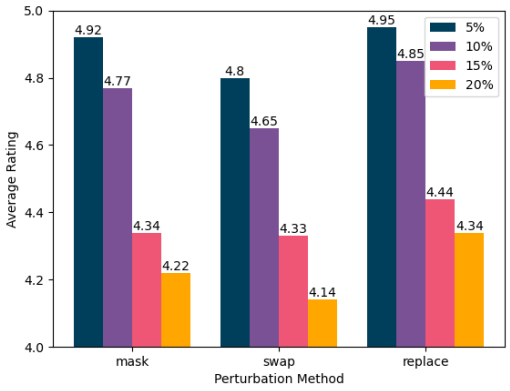


Figure 8: We show the average rating of our 300 examples dataset with `GPT-3.5-Turbo` metric.

A Proof of Proposition 4

Below is the complete proof of Proposition 4.

B Experiments

We elaborate on more experimental details and results in this section.

B.1 Adversarial Examples

In this section, we introduce how we obtained 300 adversarial examples. To ensure the reproducibility and generalizability of the experiment, we use images from ImageNet. We first selected 10 images from each of a random 100 categories of ImageNet to form a dataset of 1,000 images. Then, we use two attack algorithms, VAE and JIP, to create adversarial examples on this dataset. We fixe the experimental settings for both attack algorithms and attacked each image in the dataset. Finally, we took the intersection of all the images that were successfully attacked by both methods to form a set of 300 images, which ensure fairness in the evaluation.

B.2 Injection Mitigation

In this section, we present more results about the Injection Mitigation using VLM miniGPT4 in Figure 9 and InstructBLIP in Figure 10. In comparison with the `llava-1.5` VLM as shown in Figure 5, we observed that SmoothVLM consistently achieves a greater reduction in ASR with the `llava-1.5` model under the same q and N settings. This suggests that adversarial examples which are successful in attacking the `miniGPT4` are less likely to be thwarted when subjected to masking defenses in the `llava-1.5` model. This observation indicates that the masking defense is more effective in `llava-1.5` than in `miniGPT4`.

B.3 Visual Prompt Recovery

We present results about the Visual Prompt Recovery using VLM miniGPT4 in Figure 11 and InstructBLIP in Figure 12. Comparing with Figure 6, we observe a consistent distortion rate trend across different VLMs; that is, as N increases, the distortion rate gradually decreases, and a greater amount of perturbation contributes to the restoration of the align image information. In contrast, as shown in Figure 9, when the proportion of perturbed pixels is too small and N is relatively low, the distortion rate is significantly higher than the ASR, indicating that the current level of perturbation, although sufficient to mask the attack, is inadequate for recovering the original image information. Therefore, to ensure both a low ASR and a minimal distortion rate, it is necessary to employ larger perturbations and a higher N.

B.4 GPT metric

In this section, we address concerns regarding the perturbation of benign visual prompts. We contend that within our defense framework, the impact of perturbations is minimal compared to the potential harm from adversarial information. To substantiate this claim, we employed `llava-1.5` to evaluate both clean images and images subjected to 5%, 10%, 15%, and 20% perturbations. Subsequently, we used `GPT-3.5-Turbo` to provide an objective metric by rating the similarity between responses from `llava-1.5` on clean and perturbed images. The ratings, ranging from 1 to 5, indicate that a score of 5 corresponds to identical responses, while a score of 1 denotes completely dissimilar responses. As illustrated in Figure 8, even with the 20% perturbation across three methods, under the GPT metric, we can still observe that the average rating is higher than 4.0, which demonstrates that perturbations have negligible effects on the outcomes for clean images.

C Discussion

Drawing on our unique insights, this study is informed by the methodologies of randomized smoothing (Cohen et al., 2019) and its successor, SmoothLLM (Robey et al., 2023). However, as outlined in § 3.1, visual prompts differ markedly

Proposition 2. *Proof.* *In **Proposition 4**, we want to compute the probability* $Pr[(\textit{VPI} \circ \textit{SmoothVLM})([I \oplus P; \emptyset]) = 0]$. *Base on Definition 3, we have*

$$(\textit{VPI} \circ \textit{SmoothVLM})([I \oplus P; \emptyset]) = (\textit{VPI} \circ \textit{VLM})([\mathbf{I} \oplus \mathbf{P}; \emptyset]) \tag{10}$$

$$= \mathbb{I}\Big[\frac{1}{N}\sum_{j=1}^{N}(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) > \frac{1}{2}\Big] \tag{11}$$

where $I_j \oplus P_j$ *for* $j \in [N]$ *are drawn i.i.d. from* $\mathbb{P}_p(I \oplus P)$. *Thus, we can compute the probability with the following equalities:*

$$Pr[(\textit{VPI} \circ \textit{SmoothVLM})([I \oplus P; \emptyset]) = 0] \tag{12}$$

$$= Pr\Big[\frac{1}{N}\sum_{j=1}^{N}(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) > \frac{1}{2}\Big] \tag{13}$$

$$= Pr\Big[(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0 \textit{ for at least } \lceil N/2 \rceil \textit{ of the indices } j \in [N]\Big] \tag{14}$$

$$= \sum_{t=\lceil N/2 \rceil}^{N} Pr\Big[(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0 \textit{ for exactly } t \textit{ of the indices } j \in [N]\Big] \tag{15}$$

To make a precise computation, here we denote α *as the probability that a randomly drawn* $I_j \oplus P_j \sim \mathbb{P}_p(I \oplus P)$ *leads to a successful defense, i.e.,*

$$\alpha \doteq Pr[(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0] \tag{16}$$

Then we can see the random variable t *follows the binomial distribution with parameters* N *and* α. *Based on the probability mass function of the binomial distribution, we can simply get the sum of the probability as the following equation:*

$$Pr[(\textit{VPI} \circ \textit{SmoothVLM})([I \oplus P; \emptyset]) = 0] = \sum_{t=\ulcorner N/2 \urcorner}^{N} \binom{N}{t} \alpha^t (1-\alpha)^{N-t} \tag{17}$$

To compute α, *we can decompose the probability based on whether* $\ell_0(P_j, P) \geq \ulcorner qmn \urcorner$. *Formally, we have:*

$$\alpha = Pr[(\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0] \tag{18}$$

$$= Pr[((\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0) | (\ell_0(P_j, P) \geq \ulcorner qmn \urcorner)] Pr[\ell_0(P_j, P) \geq \ulcorner qmn \urcorner] \tag{19}$$

$$+ Pr[((\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0) | (\ell_0(P_j, P) < \ulcorner qmn \urcorner)] Pr[\ell_0(P_j, P) < \ulcorner qmn \urcorner] \tag{20}$$

$$\geq Pr[((\textit{VPI} \circ \textit{VLM})(I_j \oplus P_j) = 0) | (\ell_0(P_j, P) \geq \ulcorner qmn \urcorner)] Pr[\ell_0(P_j, P) \geq \ulcorner qmn \urcorner] \tag{21}$$

Since the adversarial patch P *is visual* q*-unstable with probability error* ϵ, *based on our Assumption 2, we can know that* $Pr[(\textit{VPI} \circ \textit{VLM})([I_j \oplus P_j; \emptyset]) = 0] \geq 1 - \epsilon - \mu$, *if* $\ell_0(P_j, P) \geq \ulcorner qmn \urcorner$. *For* $Pr[\ell_0(P_j, P) \geq \ulcorner qmn \urcorner)]$, *since* $\ell_0(P_j, P)$, *the number of randomly masked pixels falling on the adversarial patch* P, *also follows the binomial distribution with parameters* mn *and* p, *we have* $Pr[\ell_0(P_j, P) \geq \ulcorner qmn \urcorner] = \sum_{k=\ulcorner qmn \urcorner}^{mn} \binom{mn}{k} p^k (1-p)^{mn-k}$. *Finally we can obtain:*

$$\alpha \geq (1 - \epsilon - \mu) \sum_{k=\ulcorner qmn \urcorner}^{mn} \binom{mn}{k} p^k (1-p)^{mn-k} \tag{22}$$

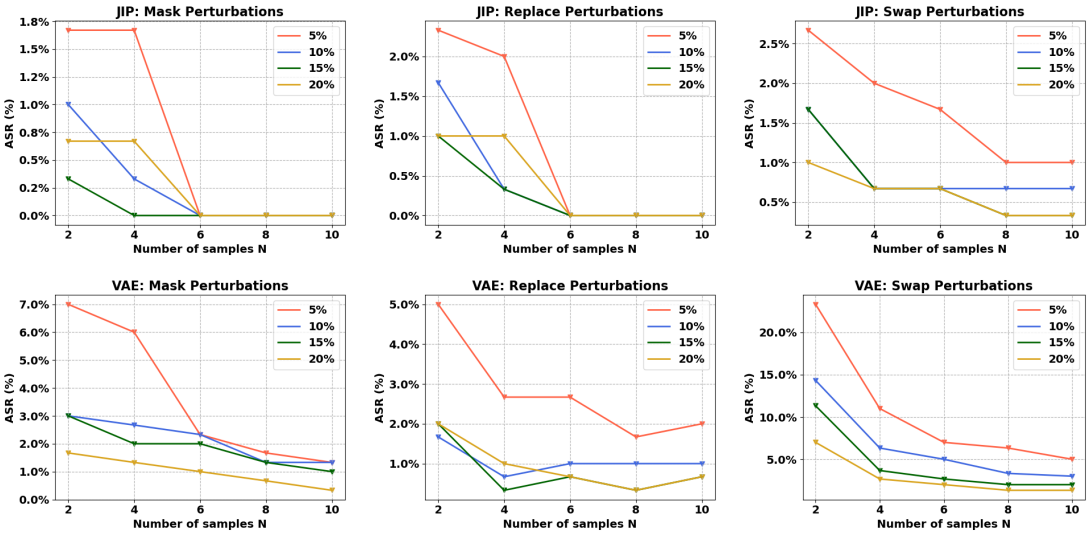


Figure 9: **SmoothVLM Injection Mitigation using miniGPT4.** We plot the ASRs of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$;

from textual prompts, prompting us to address several key distinctions from SmoothLLM. Primarily, our approach is characterized by a more rigorous formulation. In § 3.3, we present extensive experiments with adaptive attacks that substantiate the validity of our observations and assumptions. Furthermore, our model incorporates an error term, ϵ, enhancing the completeness of our proposition. Unlike SmoothLLM, which presupposes that attackers merely alter the suffix or prefix of a prompt, our framework, SmoothVLM, is designed to counteract any form of adversarial patches within reasonable sizes, thereby offering enhanced generalizability and performance. Although our current focus is on visual prompt injections, the theoretical foundation of our work could potentially be extended to encompass both textual and visual prompts.

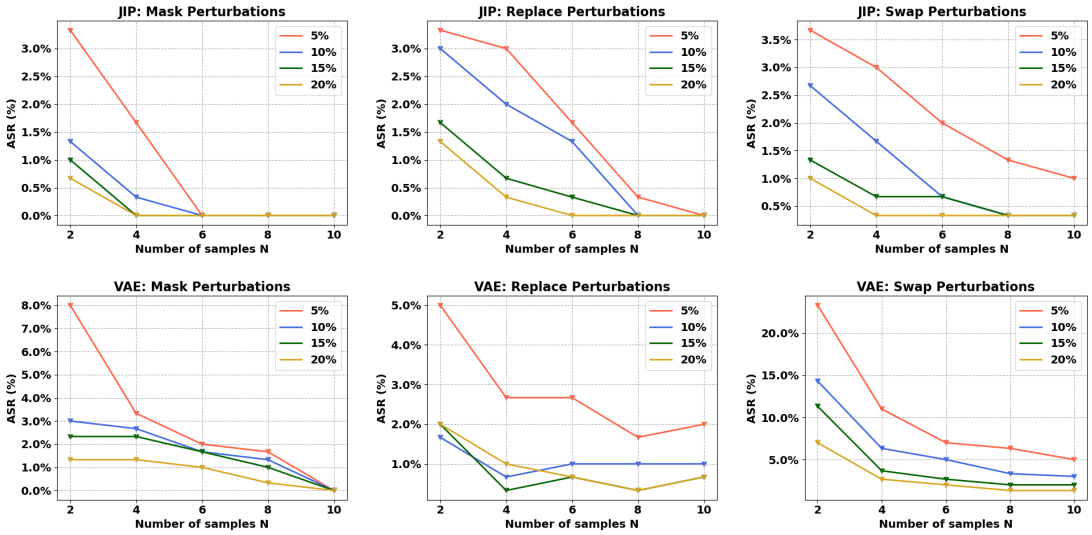


Figure 10: **SmoothVLM Injection Mitigation using InstructBLIP.** We plot the ASRs of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$;

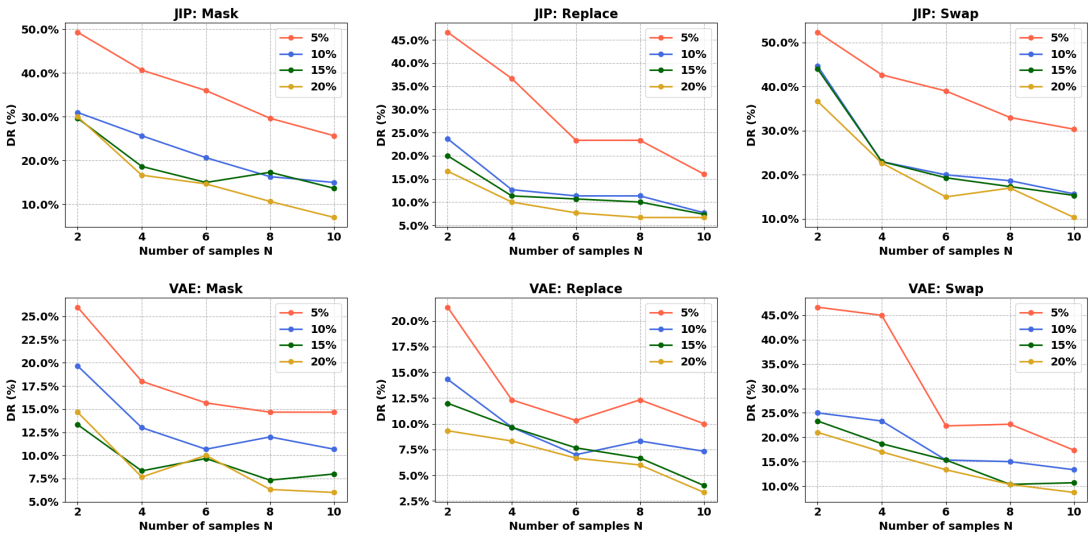


Figure 11: **Visual Prompt Recovery using miniGPT4.** We plot the Distortion Rate of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$;

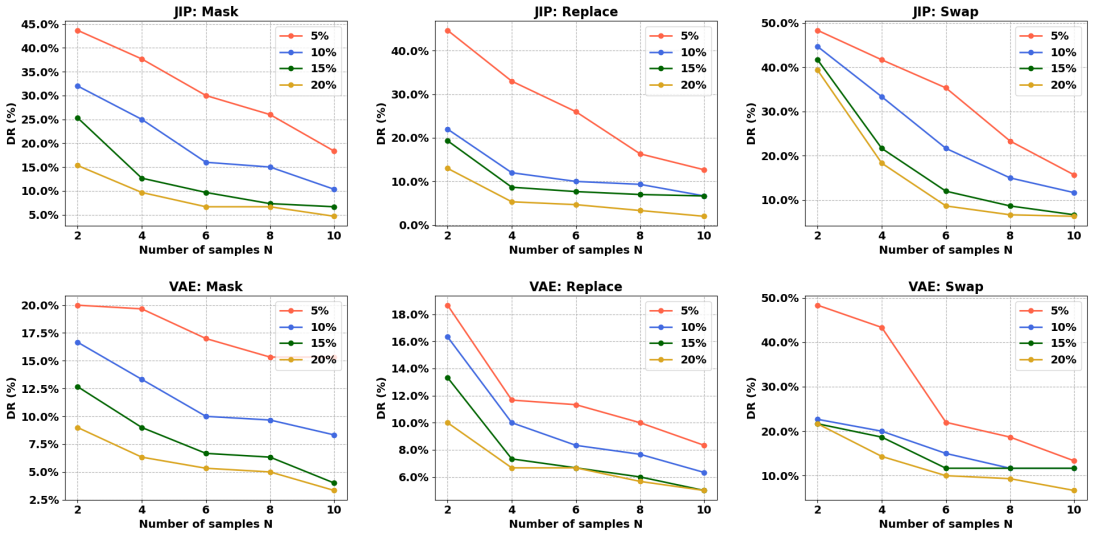


Figure 12: **Visual Prompt Recovery using InstructBLIP.** We plot the Distortion Rate of VLM patch attack JIP (top row) and VAE (bottom row) for various values of the perturbation percentage $q \in \{5, 10, 15, 20\}$ and the number of samples $N \in \{2, 4, 6, 8, 10\}$;

Figure 13: **VLM and SmoothVLM Responses to Patched Visual Prompt Injectors. Row 1:** Source images prepared for adversarial attacks alongside their aligned responses. **Row 2:** Target images containing adversarial attack information. **Row 3:** Images post-application of patch attacks with corresponding VLM responses. **Row 4:** Images following the application of SmoothVLM with their recovery responses.