Diffusion-based Cumulative Adversarial Purification for Vision Language Models

Anonymous authors
Paper under double-blind review

Abstract

Vision Language Models (VLMs) have shown remarkable capabilities in multimodal understanding, yet their susceptibility to adversarial perturbations poses a significant threat to their reliability in real-world applications. Despite often being imperceptible to humans, these perturbations can drastically alter model outputs, leading to erroneous interpretations and decisions. This paper introduces DiffCAP, a novel diffusion-based purification strategy that can effectively neutralize adversarial corruptions in VLMs. We theoretically establish a certified recovery region in the forward diffusion process and meanwhile quantify the convergence rate of semantic variation with respect to VLMs. These findings manifest that adversarial effects monotonically fade as diffusion unfolds. Guided by this principle, DiffCAP leverages noise injection with a similarity threshold of VLM embeddings as an adaptive criterion, before reverse diffusion restores a clean and reliable representation for VLM inference. Through extensive experiments across six datasets with three VLMs under varying attack strengths in three task scenarios, we show that DiffCAP consistently outperforms existing defense techniques by a substantial margin. Notably, DiffCAP significantly reduces both hyperparameter tuning complexity and the required diffusion time, thereby accelerating the denoising process. Equipped with theorems and empirical support, DiffCAP provides a robust and practical solution for securely deploying VLMs in adversarial environments.

1 Introduction

Vision language models (VLMs) have exhibited impressive performance in a diverse range of multimodal understanding tasks (Radford et al., 2021; Lu et al., 2019; Jia et al., 2021; Alayrac et al., 2022), empowering numerous real-world applications such as image-grounded text generation (e.g., image captioning and visual question answering) (Li et al., 2020; Mokady et al., 2021; Li et al., 2022b) and zero-shot classification (Radford et al., 2021; Zhai et al., 2022; Alayrac et al., 2022; Zhai et al., 2023). However, their inherent susceptibility to adversarial perturbations presents a critical challenge (Zhao et al., 2023; Qi et al., 2024; Zhang et al., 2022). These perturbations are usually designed to be imperceptible to humans, but when added to natural images, can deceive models into making incorrect predictions, severely undermining their reliability and effectiveness (Goodfellow et al., 2014; Madry et al., 2017; Carlini & Wagner, 2017). Adversarial vulnerability is especially concerning as malicious actors may exploit these ML systems to spread misinformation or fraudulent activities (Wu et al., 2024a), highlighting the urgent need for robust defensive strategies (Jin et al., 2024; Liu et al., 2024a).

To mitigate this threat, significant research efforts have focused on adversarial defenses specifically designed for VLMs (Liu et al., 2024a; Weng et al., 2025). A dominant direction in this field is adversarial training, which fine-tunes models using adversarially perturbed data to enhance robustness. For instance, recent approaches such as RobustCLIP (Schlarmann et al., 2024) have leveraged supervised adversarial fine-tuning to fortify VLMs against specific attack types. Although effective within their training scope, these methods exhibit significant limitations, particularly poor generalization to novel, unseen attacks (Dolatabadi et al., 2022; Laidlaw et al., 2020) and substantial computational overhead associated with continuous retraining and fine-tuning procedures (Wong et al., 2020; Andriushchenko & Flammarion, 2020).

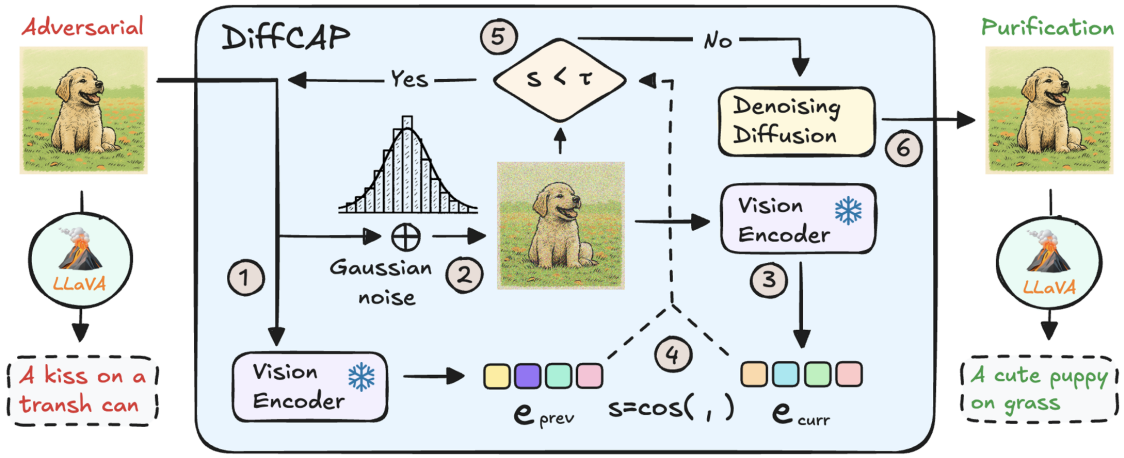

Figure 1: Overview of the DiffCAP Pipeline: An adversarial input is cumulatively processed through steps 1 to 5; if a stopping condition is not satisfied at step 5, the process restarts from step 1, otherwise the purification output is generated at step 6. See Alg. 1 for details.

In contrast, adversarial purification (Yoon et al., 2021; Nie et al., 2022) emerges as a promising alternative that does not require the expensive adversarial fine-tuning of models. Techniques like DiffPure (Nie et al., 2022) have demonstrated the feasibility of purification approaches by directly removing adversarial perturbations from input data using generative models, such as diffusion models (Song et al., 2020a;b; Yang et al., 2023; Song & Ermon, 2019). These methods maintain a high generalization capability to unseen attacks without necessitating modifications to the underlying VLMs, thus preserving original model performance on benign inputs. Despite these advantages, current generative model-based purification techniques suffer from substantial slowdown at inference time, hindering their practical deployment, particularly for real-time scenarios involving large VLMs.

To address these critical shortcomings, we propose DiffCAP, abbreviated for *Diffusion-based Cumulative Adversarial Purification*, the first adversarial purification strategy specifically designed for VLMs. DiffCAP leverages a novel mechanism that dynamically identifies the minimal necessary diffusion time, effectively balancing purification efficacy and computational efficiency. Our method cumulatively injects random Gaussian noise into adversarially perturbed images until the embeddings of two consecutively noised images converge to a predefined similarity threshold, indicating the potential neutralization of adversarial perturbations. A diffusion model subsequently denoises this stabilized image, enabling the recovery of a clean and interpretable image for VLM inference.

Contribution and novelty. We summarize as following points:

- We formulate a certified recovery region during the forward diffusion process, with the VLM acting as a zero-shot classifier (Thm. 1). Furthermore, we quantify the convergence rate of VLM-encoded semantic between adjacent forward diffusion steps (Thm. 2). These two theorems reveal that adversarial perturbations can be counteracted after sufficient diffusion steps, with the embedding change diminishing monotonically as diffusion progresses.
- DiffCAP's per-input minimized diffusion time resolves a key limitation of previous diffusion-based purification works, including DiffPure: their reliance on a fixed diffusion time, which is suboptimal for inputs of varying difficulty and requires hyperparameter tuning.
- We conduct comprehensive experiments across three popular VLMs, six diverse datasets, multiple perturbation strengths, and various multimodal tasks, including image captioning, visual question answering, and zero-shot classification. The results demonstrate that DiffCAP consistently outperforms existing state-of-the-art defenses by a considerable margin.

2 Related work

Perturbation-based attacks. Perturbation-based attacks cause models to make incorrect predictions by introducing small, often imperceptible, alterations to the input data (Chakraborty et al., 2021; Huang et al., 2017; Chakraborty et al., 2018). These attacks commonly leverage gradient-based methods to find the most vulnerable parts of an input, then craft perturbations to maximize the model's loss. A classic illustration is adversarial image attacks, where minute pixel modifications, invisible to humans, can trick a model into misclassifying an image (Goodfellow et al., 2014; Madry et al., 2017).

Adversarial training. Adversarial training employs optimization techniques to bolster model robustness and safety alignment. TeCoA (Mao et al., 2022) applies supervised adversarial fine-tuning on ImageNet, while FARE (Schlarmann et al., 2024) leverages an unsupervised adversarial fine-tuning approach using an embedding loss on PGD-perturbed (Madry et al., 2017) inputs to enhance CLIP vision encoder robustness and zero-shot performance. Addressing challenges in such methods, Hossain et al. (Hossain & Imteaj, 2024b) developed Sim-CLIP, which integrates a Siamese architecture with a cosine similarity loss to align clean and perturbed representations, incorporating a stop-gradient mechanism for efficient training without negative samples. This was further extended by Sim-CLIP+ (Hossain & Imteaj, 2024a), which tailors the cosine similarity loss and stop-gradient mechanism to defend VLMs against advanced optimization-based jailbreak attacks, preventing symmetric loss collapse while maintaining computational efficiency.

Adversarial purification. Adversarial purification techniques (Shi et al., 2021; Yoon et al., 2021; Fu et al., 2025) offer a distinct defense paradigm by employing generative models to sanitize images from adversarial perturbations (Samangouei et al., 2018; Hill et al., 2021). The main advantage is its "plug-in" simplicity to address new threats without the need to retrain vision models—since the adversarial images being sanitized independently of attack specifics and the vision models. However, this adaptability used to be constrained by weaker performance compared to adversarial training methods (Croce & Hein, 2020). The vulnerability becomes especially clear when faced with adaptive attackers who have knowledge of the defense system (Athalye et al., 2018a; Tramer et al., 2020), a problem generally rooted in the inherent weaknesses of the previous generative models, such as GAN (Goodfellow et al., 2020). The rise of diffusion models (Song et al., 2020b), known for their generative capabilities, high sample diversity and inherent stochasticity, signals a promising direction for mitigating these persistent issues. DiffPure (Nie et al., 2022) proposed to use the diffusion model for adversarial purification. CLIPure (Zhang et al., 2025) operates directly in the CLIP latent space, correcting embeddings of adversarial examples for downstream tasks. However, previous works did not discuss the optimal number of steps for forward noise-injection. DiffPure (Nie et al., 2022) and CLIPure (Zhang et al., 2025) both use a fixed diffusion time, which is inflexible for adversarial inputs of diverse hardness. We propose a threshold-based stopping criterion in DiffCAP, and therefore reduces the number of noise-injection steps and improves performance by a significant margin.

3 Preliminaries

This section provides the background and preliminary definitions of vision encoders, including their use for both CLIP and Vision LLMs, as well as continuous-time diffusion models.

3.1 Vision encoder

CLIP. Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) consists of a vision encoder $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^m$ and a text encoder $\boldsymbol{\psi} : \mathbb{R}^{d'} \to \mathbb{R}^m$. The vision encoder and the text tokenizer have the same embedding dimension. We define a zero-shot classifier h. For a K-class task, text prompts $\boldsymbol{t}_k$, such as "A photo of <class k>", are generated for $k = 1, \ldots, K$. The classifier h, determined by $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, calculates logits for an input image $\boldsymbol{x}$ via the cosine similarity between the image embedding and each prompt embedding:

$$h_k(\boldsymbol{x}) = \cos(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\psi}(\boldsymbol{t}_k)) = \left\langle \frac{\boldsymbol{\phi}(\boldsymbol{x})}{\|\boldsymbol{\phi}(\boldsymbol{x})\|_2}, \frac{\boldsymbol{\psi}(\boldsymbol{t}_k)}{\|\boldsymbol{\psi}(\boldsymbol{t}_k)\|_2} \right\rangle. \tag{1}$$

Vision LLMs. Vision LLMs, such as LLaVA (Liu et al., 2024c; 2023a;b) and MiniGPT (Zhu et al., 2023), consist of a vision encoder $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^m$, a text tokenizer, and a language model. The vision encoder and the text tokenizer have the same embedding dimension. When feeding the VLM with an image and the instruction, a vision encoder transforms this image to hidden embeddings of dimension m. The text tokenizer first tokenizes the instruction into tokens, and then looks up the embedding matrix to get its m-dimensional embedding. The image embedding and the instruction embedding are concatenated and then fed into a language model to generate a description. The vision encoder can be the vision encoder in CLIP (Radford et al., 2021), and the instruction prompt is usually like `"Describe this image in detail"`.

3.2 Diffusion model

In this section, we briefly introduce the continuous-time diffusion models (Song et al., 2020b). Let $p(\boldsymbol{x})$ represent the underlying, unknown distribution for data points $\boldsymbol{x} \in \mathbb{R}^d$. The core idea of diffusion models is to progressively transform samples from $p(\boldsymbol{x})$ into Gaussian noise. Formally, the transformation can be formulated by the forward diffusion process $\{\boldsymbol{x}(t)\}_{t\in[0,1]}$, which is governed by the following stochastic differential equation (SDE) in the time interval $[0, 1]$:

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\mathrm{d}t + g(t)\mathrm{d}\boldsymbol{w}(t), \tag{2}$$

where $\boldsymbol{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ stands for the drift function, $g : \mathbb{R} \to \mathbb{R}$ is the diffusion function, and $\boldsymbol{w}(t) \in \mathbb{R}^d$ denotes the Brownian motion. Note that the diffusion process starts with $\boldsymbol{x}(0)$ drawn from the underlying data distribution $p(\boldsymbol{x})$.

The distribution of $\boldsymbol{x}(t)$ at any time t is $p_t(\boldsymbol{x})$, with the initial data distribution being $p_0(\boldsymbol{x}) = p(\boldsymbol{x})$. The functions $\boldsymbol{f}(\boldsymbol{x}, t)$ and $g(t)$ are chosen carefully so that as t approaches 1, the distribution $p_1(\boldsymbol{x})$ closely resembles a standard d-dimensional Gaussian distribution, $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$. We follow the Variance Preserving (VP) SDE (Song et al., 2020b), where the drift and diffusion coefficients are $\boldsymbol{f}(\boldsymbol{x}, t) = -\frac{1}{2}\beta(t)\boldsymbol{x}$ and $g(t) = \sqrt{\beta(t)}$ respectively. Here, $\beta(t)$ is a function that controls the noise level over time.

To generate new samples, one must reverse the diffusion process. This is achieved by solving a corresponding reverse-time SDE of Eq. (2):

$$\mathrm{d}\hat{\boldsymbol{x}} = \left[\boldsymbol{f}(\hat{\boldsymbol{x}}, t) - g(t)^2 \nabla_{\hat{\boldsymbol{x}}} \log p_t(\hat{\boldsymbol{x}})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\boldsymbol{w}}. \tag{3}$$

The generation process starts with drawing an initial sample $\hat{\boldsymbol{x}}(1)$ from the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$. Then, by integrating this SDE from $t = 1$ down to $t = 0$, the noisy sample $\hat{\boldsymbol{x}}(t)$ is progressively denoised. The goal is for the final output $\hat{\boldsymbol{x}}(0)$ to be a sample from the original data distribution $p_0(\boldsymbol{x})$. However, the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ in Eq. (3) is usually intractable. In practice, we use a neural network, denoted by $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ and parameterized by $\boldsymbol{\theta}$, to approximate the score function (Song et al., 2020b; Kingma et al., 2021).

4 Methodology

In this section, we introduce DiffCAP, a purification mechanism that leverages forward diffusion dynamics and semantic stability to remove adversarial perturbations in VLMs. When we pass an adversarial image $\boldsymbol{x}_{\text{adv}}$ into the diffusion model, $\boldsymbol{x}(t)$ follows a forward diffusion process governed by the VP-SDE:

$$\mathrm{d}\boldsymbol{x}(t) = -\tfrac{1}{2}\beta(t)\boldsymbol{x}(t)\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{w}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_{\text{adv}}, \tag{4}$$

$\beta(t) > 0$ is a smooth noise schedule, and $\boldsymbol{w}(t)$ is a standard Wiener process. This process has a closed-form solution at time t given by:

$$\boldsymbol{x}(t) = \sqrt{\alpha(t)}\,\boldsymbol{x}_{\text{adv}} + \sqrt{1 - \alpha(t)}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d), \tag{5}$$

where $\alpha(t) = \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)$. Our method builds upon the insight from randomized smoothing (Cohen et al., 2019): adding Gaussian noise to adversarial inputs can recover the original predictions with high probability. We need the following basic assumptions to facilitate our analysis.

Algorithm 1 DiffCAP: Diffusion-based Cumulative Adversarial Purification

Require: Adversarially perturbed input image $\boldsymbol{x}_{\text{adv}}$; An image encoder $\boldsymbol{\phi}(\cdot)$ (e.g., from VLM); Similarity threshold τ; Pretrained diffusion denoiser $D(\cdot)$; Maximum number of forward diffusion steps T.

Ensure: Purified image $\boldsymbol{x}_{\text{clean}}$.

Initialize step counter $t \leftarrow 0$.
Set initial image $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_{\text{adv}}$.
Calculate initial embedding $\boldsymbol{e}_{\text{prev}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_0)$.
while $t < 1$ **do** ▷ Iteratively inject noise and check stability
  Inject noise into the images based on Eq. (5) to obtain $\boldsymbol{x}_{t+1/T}$.
  Calculate current embedding $\boldsymbol{e}_{\text{curr}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_{t+1/T})$.
  if $\cos(\boldsymbol{e}_{\text{curr}}, \boldsymbol{e}_{\text{prev}}) \geq \tau$ **then** ▷ Check if embeddings have stabilized
    break ▷ Exit loop, stabilization reached
  end if
  Update previous embedding: $\boldsymbol{e}_{\text{prev}} \leftarrow \boldsymbol{e}_{\text{curr}}$, update $t \leftarrow t + 1/T$.
end while
Set the stabilized (but potentially noisy) image $\boldsymbol{x}_{\text{stable}} \leftarrow \boldsymbol{x}_t$.
Denoise the stabilized image using the diffusion model: $\boldsymbol{x}_{\text{clean}} \leftarrow D(\boldsymbol{x}_{\text{stable}})$.
return $\boldsymbol{x}_{\text{clean}}$.

Assumption 1 (Scale Invariance)**.** *We assume that the classifier h (defined in Eq. (1)) is scale-invariant: for any scalar $\lambda > 0$ and any input $\boldsymbol{x} \in \mathbb{R}^d$, we have $h(\lambda \boldsymbol{x}) = h(\boldsymbol{x})$.*

Remark 1. *Asm. 1 is standard in deep learning theory and practically justified for models as we can always scale the input (Zhang et al., 2020; Wu et al., 2024b).*

We now present our main result that establishes a certified recovery region under forward diffusion for the adversarial image. The proof can be found at Sec. B.1.

Theorem 1 (Certified recovery region under forward diffusion)**.** *Let $h : \mathbb{R}^d \to [K]$ be the classifier. Let $\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\epsilon}_{adv}$ be an adversarial example with perturbation $\boldsymbol{\epsilon}_{adv}$, and let $\boldsymbol{x}(t)$ be the solution to the forward diffusion process defined in Eq. (5) with a linear noise schedule $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$. Suppose there exists $\underline{p_1}, \overline{p_2}, k_1$ such that for all $t \in [0, 1]$*

$$\mathbb{P}(h(\boldsymbol{x} + \boldsymbol{\epsilon}'(t)) = k_1) \geq \underline{p_1} > \overline{p_2} \geq \max_{k \neq k_1} \mathbb{P}(h(\boldsymbol{x} + \boldsymbol{\epsilon}'(t)) = k), \tag{6}$$

where $\boldsymbol{\epsilon}'(t) \sim \mathcal{N}(\mathbf{0}, \frac{1-\alpha(t)}{\alpha(t)} \boldsymbol{I}_d)$. Define

$$t_{\min} = \frac{2M}{\sqrt{\beta_{\min}^2 + 2(\beta_{\max} - \beta_{\min})M} + \beta_{\min}}, \quad \text{where } M := \log\left(1 + \left(\frac{2\|\boldsymbol{\epsilon}_{adv}\|_2}{\boldsymbol{\phi}^{-1}(\underline{p_1}) - \boldsymbol{\phi}^{-1}(\overline{p_2})}\right)^2\right).$$

Then, under Asm. 1, when $\beta_{\min} \geq M$, for all $t_{\min} \leq t \leq 1$, we have $\arg\max_{k \in [K]} \mathbb{P}(h(\boldsymbol{x}(t)) = k) = k_1$, i.e., the adversarial example is classified as its original label k_1 after sufficient forward diffusion.

Remark 2. *Thm. 1 indicates that an adversarially perturbed image with added noise will eventually be classified correctly. Then, we can expect to use the diffusion model to remove the added noise to obtain the clean image.*

Upon this point, we start to analyze the dynamics in the VLM embedding space during the forward diffusion process. In Thm. 1, we prove a certified recovery region, indicating the local smoothness of $\boldsymbol{\phi}(\boldsymbol{x}(t))$ for $t \geq t_{\min}$. Therefore, we pose a local Lipschitz assumption after time $t_{\min}$:

Assumption 2. *We assume $\boldsymbol{\phi}$ is $L-$Lipschitz for $t \geq t_{\min}$*

$$\|\boldsymbol{\phi}(\boldsymbol{x}(t)) - \boldsymbol{\phi}(\boldsymbol{x}(t'))\|_2 \leq L\|\boldsymbol{x}(t) - \boldsymbol{x}(t')\|_2 \quad \forall t, t' \geq t_{\min}.$$

Algorithm 2 Adaptive Similarity Threshold (τ) Calculation

Require: The dataset of clean-adversarial image pairs $D_{\text{pairs}} = \{(\boldsymbol{x}_{\text{clean}}, \boldsymbol{x}_{\text{adv}})\}$; The embedding function $\boldsymbol{\phi}(\cdot)$. Maximum number of forward diffusion steps T.

Ensure: The similarity threshold τ.

Initialize step counter $t \leftarrow 0$.
for all pair $(\boldsymbol{x}_{\text{clean}}, \boldsymbol{x}_{\text{adv}}) \in D_{\text{pairs}}$ **do**
  Set initial image $\boldsymbol{x}_{0,\text{clean}} \leftarrow \boldsymbol{x}_{\text{clean}}$, $\boldsymbol{x}_{0,\text{adv}} \leftarrow \boldsymbol{x}_{\text{adv}}$.
  Calculate initial embedding $\boldsymbol{e}_{\text{prev,clean}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_{0,\text{clean}}), \boldsymbol{e}_{\text{prev,adv}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_{0,\text{adv}})$.
end for
while $t < 1$ **do** ▷ Iteratively inject noise and check stability
  Initialize total similarity set $S_{\text{clean}} \leftarrow [], S_{\text{adv}} \leftarrow []$.
  for all pair $(\boldsymbol{x}_{\text{clean}}, \boldsymbol{x}_{\text{adv}}) \in D_{\text{pairs}}$ **do**
    Inject noise into the images based on Eq. (5) to obtain $\boldsymbol{x}_{t+1/T,\text{clean}}, \boldsymbol{x}_{t+1/T,\text{adv}}$.
    Calculate current embedding $\boldsymbol{e}_{\text{curr,clean}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_{t+1/T,\text{clean}}), \boldsymbol{e}_{\text{curr,adv}} \leftarrow \boldsymbol{\phi}(\boldsymbol{x}_{t+1/T,\text{adv}})$.
    Calculate similarity $s_{\text{clean}} \leftarrow \cos(\boldsymbol{e}_{\text{curr,clean}}, \boldsymbol{e}_{\text{prev,clean}}), s_{\text{adv}} \leftarrow \cos(\boldsymbol{e}_{\text{curr,adv}}, \boldsymbol{e}_{\text{prev,adv}})$.
    Add the similarity score to the set $S_{\text{clean}} \leftarrow S_{\text{clean}} \cup \{s_{\text{clean}}\}$, $S_{\text{adv}} \leftarrow S_{\text{adv}} \cup \{s_{\text{adv}}\}$.
  end for
  if S_{adv} and S_{clean} are from the same underlying distribution **then**
    break
  end if
  for all pair $(\boldsymbol{x}_{\text{clean}}, \boldsymbol{x}_{\text{adv}}) \in D_{\text{pairs}}$ **do**
    Update previous embedding: $\boldsymbol{e}_{\text{prev,clean}} \leftarrow \boldsymbol{e}_{\text{curr,clean}}, \boldsymbol{e}_{\text{prev,adv}} \leftarrow \boldsymbol{e}_{\text{curr,adv}}$.
  end for
  Update $t \leftarrow t + 1/T$.
end while
return $\tau \leftarrow \text{mean}(S_{\text{clean}})$.

Lemma 1. *Under the same setting as in Thm. 1 and Asm. 2, let $\boldsymbol{x}(t)$ be defined as in Eq. (5). Then for any $t_1, t_2 \in [0, 1]$, as $t_1, t_2 \to 1$, we have: $\mathbb{E}\left[\|\boldsymbol{\phi}(\boldsymbol{x}(t_1)) - \boldsymbol{\phi}(\boldsymbol{x}(t_2))\|_2\right] \to 0$.*

The proof is deferred to Sec. B.2. Lemma 1 shows that embeddings converge during forward diffusion, which motivates our stopping criterion based on similarity of VLM latent representations. We quantify this convergence rate in the following theorem:

Theorem 2. *Let $\boldsymbol{x}(t)$ be defined as in Eq. (5). Then for small $\delta > 0$,*

$$\mathbb{E}\left[\|\boldsymbol{\phi}(\boldsymbol{x}(t)) - \boldsymbol{\phi}(\boldsymbol{x}(t+\delta))\|_2\right] = O\left(L \cdot \delta \cdot \beta(t)\sqrt{\frac{\alpha(t)}{1-\alpha(t)}}\right), \text{ for } t \in [t_{\min}, 1-\delta], \tag{7}$$

where $\alpha(t) = \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)$. Moreover, by using a common linear noise schedule $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$ with $\beta_{\min} > 0$ and $\beta_{\max} > \beta_{\min}$. Then $\beta(t) \cdot \sqrt{\frac{\alpha(t)}{1-\alpha(t)}}$ is strictly decreasing for all $t \in [t_{\min}, 1)$.

The proof is deferred to Sec. B.3. Thm. 2 quantifies the semantic change between adjacent forward diffusion steps. The bound in Eq. (7) decreases as $t \to 1$ under the common linear noise schedule. Consequently, the expected semantic change between adjacent forward diffusion steps diminishes as the process approaches terminal time.

Our derivations inspire a simple yet powerful strategy—inject Gaussian noise until the semantic embedding stabilizes. We call this algorithm DiffCAP, summarized in Alg. 1. The cumulative diffusion continues until the cosine similarity between consecutive VLM embeddings exceeds a threshold τ. A pretrained diffusion model is then applied in reverse to recover a clean image from the stabilized noisy input.

We describe how to select the threshold τ in Alg. 2. The core idea is to iteratively inject noise into both clean and adversarial images and track the cosine similarity between the embeddings across consecutive steps of noise injection. This is done for a collection of image pairs to reduce randomness. The process continues until the set of similarity scores for clean images and the set for their adversarial counterparts are statistically indistinguishable (i.e., likely from the same underlying distribution). At this point, the algorithm determines that the noise has reached a level where the embeddings' stability is comparable for both types of images. The mean of such a score set delivers the final threshold τ.

Extension to general VLM tasks. Our theoretical analysis naturally extends to VLMs through two complementary perspectives:

- The classifier h maps $\mathbb{R}^d \to [K]$. This naturally captures CLIP-based zero-shot classification, where the final prediction is a Softmax over cosine similarities between the image embedding and text prompts. For generative VLMs like LLaVA, this formulation holds under standard VQA setups. By adopting a structured prompting strategy (e.g., `"Classify this image into categories [1],...,[K]"`), the mapping becomes a function from $x \in \mathbb{R}^d$ to a probability distribution over tokens representing the K class indices. Thus, the theoretical guarantees for $h(x)$ directly apply to the VLM's decision-making process in discriminative tasks.
- Essentially, our theoretical contribution is not limited to the final output label but is rooted in the stability of the vision encoder's embedding space. VLMs generate text sequences conditioned on the image embedding $\phi(x)$. Thm. 2 proves the convergence and stability of this semantic embedding $\phi(x(t))$ during the diffusion process. Since the VLM's generation is a function of this embedding, establishing a certified recovery region serves as a necessary condition for robust generation, whether the downstream task is classification, captioning, or VQA.

5 Experiments

5.1 Settings

Models, datasets & metrics. We evaluate DiffCAP across three vision-language tasks: image captioning (IC), visual question answering (VQA), and zero-shot classification (ZSC). For IC and VQA, we adopt two large VLMs—OpenFlamingo (OF) (Awadalla et al., 2023) with 9B parameters and LLaVA-1.5 (Liu et al., 2024b) with 7B parameters. For ZSC, we utilize CLIP (Radford et al., 2021) with 88M parameters as the backbone model. Our experiments are conducted on standard benchmarks: COCO (Lin et al., 2014) and Flickr30k (Plummer et al., 2015) for IC, VQAv2 (Goyal et al., 2017) and TextVQA (Singh et al., 2019) for VQA, and CalTech101 (Li et al., 2022a) and ImageNet-R(endition) (Hendrycks et al., 2021) for ZSC. For both adversarial and clean evaluation, we randomly sample 500 images for IC and VQA, while $1,000$ images are chosen for ZSC. We report Consensus-based Image Description Evaluation (CIDEr) (Vedantam et al., 2015) score for IC, VQA accuracy (Antol et al., 2015) for VQA, and top-1 accuracy for ZSC.

Attacks. For IC and VQA, we adopt a two-stage attack pipeline following (Schlarmann & Hein, 2023). In the first stage, we apply 100-step Auto-PGD (APGD) attacks (Croce & Hein, 2020) in half-precision using multiple ground-truth captions or answers as supervision. Samples that fall below a predefined performance threshold are excluded from further attacks. In the second stage, we conduct stronger single-precision APGD attacks on the remaining samples. This progressive strategy maximizes adversarial impact while remaining computationally efficient. For ZSC, we follow the AutoAttack framework, employing APGD with cross-entropy loss and targeted Difference of Logits Ratio (DLR) loss with 100 iterations, respectively. The implementation of adaptive attack is articulated in Sec. C.1.

Baselines. We compare DiffCAP with two categories of adversarial defense methods. The first category includes adversarially fine-tuned vision encoders. Since both OF and LLaVA adopt CLIP as their vision backbone, we replace their CLIP vision encoder with two robust variants: TeCoA (Mao et al., 2022) and FARE (Schlarmann et al., 2024). TeCoA applies supervised adversarial training, while FARE employs an unsupervised loss. The second category includes purification methods: JPEG-DL (Salamah et al., 2024),

Table 1: CIDEr score of two VLMs in IC task on two datasets with clean images and adversarial perturbations of two sizes under different defenses. 2 and 4 with TeCoA and FARE suggest the version that is fine-tuned by $\ell_\infty^{2/255}$ and $\ell_\infty^{4/255}$ bounded adversarial examples, respectively. The best result is in **bold** and the runner-up is underlined.

Defense	OF-9B						LLaVA 1.5-7B					
	COCO			Flickr30k			COCO			Flickr30k		
	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$
No defense	79.7	1.5	1.1	60.1	0.7	0.4	115.5	4.0	3.1	77.5	1.6	1.0
TeCoA^2 (Mao et al., 2022)	73.5	31.6	21.2	49.5	14.1	9.5	98.4	44.2	30.3	57.1	23.2	15.3
FARE^2 (Schlarmann et al., 2024)	79.1	34.2	19.5	57.7	16.4	8.9	109.9	53.6	31.0	71.1	29.5	17.5
TeCoA^4 (Mao et al., 2022)	66.9	28.5	21.6	40.9	12.0	10.3	88.3	50.9	35.3	48.6	27.9	19.5
FARE^4 (Schlarmann et al., 2024)	74.1	30.9	22.8	51.4	15.7	10.5	102.4	57.1	40.9	61.6	31.4	22.8
JPEG-DL (Salamah et al., 2024)	78.2	66.1	43.9	58.8	47.6	30.7	113.3	106.4	77.2	74.8	69.6	47.9
DiffPure (Nie et al., 2022)	74.9	73.4	72.3	49.8	49.2	50.3	106.5	108.4	105.0	65.5	66.4	63.2
CLIPure (Zhang et al., 2025)	80.8	6.6	5.3	**59.3**	4.7	3.5	115.1	4.9	3.4	**76.9**	2.1	1.5
DiffCAP	**81.4**	**79.3**	**78.4**	55.6	**56.7**	**57.2**	**120.4**	**119.6**	**116.9**	75.0	**72.7**	**72.1**

the trainable JPEG compression layer to remove adversarial perturbations; DiffPure (Nie et al., 2022), the first approach leveraging the diffusion process to recover the clean image in the pixel space; CLIPure (Zhang et al., 2025), the latest method that operates directly in the CLIP latent space.

Hyperparameters. The algorithms are implemented through PyTorch, and main experiments are conducted on an NVIDIA A100 40G GPU. By default, we use the ViT-B/32 CLIP vision encoder to ensure computational efficiency. For the forward diffusion, we schedule noise with $\beta_{\min} = 0.1$, $\beta_{\max} = 20$, and a fixed step size of 0.01. In the reverse generation, we employ guided diffusion with a step size of 0.015. We take advantage of the pre-trained diffusion model from (Dhariwal & Nichol, 2021). Following Alg. 2, we determine the threshold $\tau = 0.96$ on subsets comprising 100 random clean-adversarial image pairs from the datasets mentioned above.

5.2 Result analysis

Image captioning. As shown in Tab. 1, VLMs are highly vulnerable to adversarial perturbations: even 2/255 attacks can reduce CIDEr scores close to zero. Adversarial training methods (TeCoA and FARE) provide moderate robustness improvements. However, their effectiveness drops significantly under 4/255 attacks. Notably, TeCoA and FARE also degrade the clean performance, especially on Flickr30k. JPEG-DL shows better robustness than TeCoA and FARE, but remains sensitive to perturbation strength. DiffPure substantially improves robustness, lifting performance under both $\ell_\infty^{2/255}$ and $\ell_\infty^{4/255}$ attacks to levels comparable with clean conditions. DiffCAP consistently outperforms all baselines across both VLMs and datasets. For instance, DiffCAP improves CIDEr scores by over 10% with OF on Flickr30k and with LLaVA on COCO compared to DiffPure. Furthermore, DiffCAP maintains or even improves clean performance, demonstrating strong fidelity preservation. Lastly, CLIPure performs poorly in this task. Its limited effectiveness likely stems from token-level misalignment: purifying only the `[CLS]` token embedding fails to influence generation-related latent tokens, which dominate the captioning process.

Visual question answering. The results in Tab. 2 mirror the trends observed in Tab. 1. DiffCAP consistently delivers the strongest performance across all attack settings compared to all baselines in both datasets and VLMs. The improvement is particularly notable with LLaVA on TextVQA, where DiffCAP surpasses DiffPure by over 30%. Remarkably, on the TextVQA dataset, DiffPure performs worse than JPEG-DL, indicating that visual reasoning tasks depend more heavily on fine-grained visual features, which are susceptible to over-smoothing or distortion during purification. This underscores the importance of preserving semantic fidelity when applying generative models for purification. DiffCAP addresses this by dynamically calculating the minimal diffusion time required to remove adversarial noise for each image, thereby achieving a better trade-off between robustness and feature integrity for multi-hop reasoning tasks.

Table 2: VQA accuracy (%) of two VLMs in VQA task on two datasets with clean images and adversarial perturbations of two sizes under different defenses. 2 and 4 with TeCoA and FARE suggest the version that is fine-tuned by $\ell_\infty^{2/255}$ and $\ell_\infty^{4/255}$ bounded adversarial examples, respectively. The best result is in **bold** and the runner-up is underlined.

Defense	OF-9B						LLaVA 1.5-7B					
	TextVQA			VQAv2			TextVQA			VQAv2		
	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$
No defense	23.8	0.0	0.0	48.5	1.8	0.0	37.1	0.5	0.0	74.5	2.9	0.0
TeCoA^2 (Mao et al., 2022)	16.6	3.5	2.1	46.2	23.5	20.5	24.1	12.1	8.8	66.9	33.8	21.8
FARE^2 (Schlarmann et al., 2024)	21.6	4.1	1.9	47.0	24.0	17.2	31.9	14.7	9.1	71.7	34.9	23.0
TeCoA^4 (Mao et al., 2022)	15.4	2.1	1.8	44.8	23.6	21.3	20.7	12.6	9.3	63.2	41.0	31.7
FARE^4 (Schlarmann et al., 2024)	18.6	3.4	2.9	46.1	23.6	21.0	27.6	15.8	10.9	68.3	40.7	30.5
JPEG-DL (Salamah et al., 2024)	**23.4**	15.9	13.1	46.8	39.5	32.4	34.6	27.2	21.1	68.8	60.8	45.8
DiffPure (Nie et al., 2022)	13.6	13.2	13.5	45.1	43.5	43.6	20.9	22.0	22.2	67.3	65.8	66.0
CLIPure (Zhang et al., 2025)	20.5	6.8	8.8	**47.3**	18.8	17.5	**36.1**	2.1	1.4	**73.3**	4.6	2.1
DiffCAP	18.6	**16.2**	**16.7**	46.3	**45.4**	**45.3**	28.3	**29.0**	**28.9**	70.3	**69.1**	**68.5**

Table 3: Evaluation for OF-9B and LLaVA 1.5-7B in IC and VQA tasks on four datasets under clean and adversarial ($\ell_\infty^{8/255}$) conditions, with and without (w/o) DiffCAP defense against adaptive attacks.

	Dataset	Clean		APGD		BPDA		BPDA + EOT	
		w/o	with	w/o	with	w/o	with	w/o	with
OF	COCO	90.1	92.4	4.7	91.1	27.1	79.9	29.2	83.4
	Flicker30k	63.9	62.7	4.9	60.5	19.1	50.6	15.5	56.5
	TextVQA	23.1	18.6	0.6	17.6	7.1	18.2	2.3	16.0
	VQAv2	46.2	47.1	8.3	44.6	24.0	44.5	18.0	39.5
LLaVA1.5	COCO	125.9	122.2	11.3	123.4	21.9	115.9	19.5	114.9
	Flicker30k	81.7	78.0	8.5	76.2	20.9	73.4	18.0	74.6
	TextVQA	36.9	25.1	7.4	22.7	8.8	24.3	8.7	21.7
	VQAv2	74.3	69.9	23.4	67.5	25.7	65.4	27.1	66.9

Table 4: Top-1 accuracy (%) in ZSC task. We use different CLIP vision encoders for DiffCAP. Numbers in parenthesis denote parameters in M.

	Encoder	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$
CalTech101	RN50 (102)	82.8	82.2	82.5
	RN101 (123)	83.2	83.1	81.2
	ViT-B/32 (88)	82.6	81.7	80.9
	ViT-B/16 (149)	83.0	82.3	80.9
	ViT-L/14 (304)	82.1	82.4	81.5
ImageNet-R	RN50 (102)	84.2	82.9	80.8
	RN101 (123)	86.7	85.5	84.1
	ViT-B/32 (88)	87.2	84.4	81.1
	ViT-B/16 (149)	84.7	85.0	82.2
	ViT-L/14 (304)	85.5	83.2	81.4

Adaptive attack. The above *gray-box* setting assumes the adversary can access the gradients of the model but has no knowledge of the defense pipeline. We also evaluate DiffCAP in a *white-box* setting, where the adversary has full knowledge of the deployed defense mechanism. Tab. 3 presents the detailed evaluations for IC and VQA tasks across various attack configurations, comparing performance with and without DiffCAP defense. Even under an increased attack budget ($\ell_\infty^{8/255}$), DiffCAP maintains high fidelity on clean inputs, with only an average performance drop of 3.3 points. Under APGD (Croce & Hein, 2020) attacks, it successfully restores the performance of VLMs on different datasets to levels closely matching their clean baselines, showing only a modest average degradation of 4.8 points.

In scenarios where the adversary bypasses gradient obfuscation through backward pass differentiable approximation (BPDA) (Athalye et al., 2018a) and simulates stochasticity via expectation over transformations (EOT) (Athalye et al., 2018b), DiffCAP continues to demonstrate strong resilience. The best-case performance degradation relative to clean conditions is only 1.7 points (OF-VQAv2) by BPDA without EOT and 6.7 points (OF-COCO) with EOT. The corresponding worst-case performance reductions are observed as 13.3 points (OF-Flickr30k) and 15.2 points (LLaVA-TextVQA), respectively. These results elucidate the inherent uncertainty of DiffCAP's image-adaptive diffusion step calculation, which determines the minimal purification for individual adversarial examples based on semantic convergence during the diffusion process. Such a dynamic strategy significantly prevents trivial gradient approximations and random regressions from

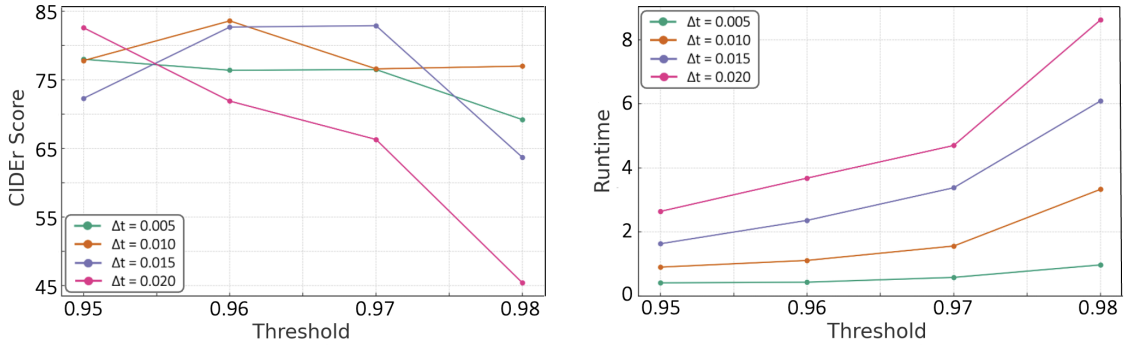

Figure 2: CIDEr score and running time (in seconds) per image with varying thresholds τ and diffusion step sizes (Δt) for DiffCAP. The evaluation is based on the IC task under $\ell_\infty^{2/255}$ attack.

Table 5: CLIP scores evaluating the DiffCAP against AttackVLM on MiniGPT 4-13B.

Setting	RN50	RN101	ViT-B/32	ViT-B/16	ViT-L/14	**Avg.**
Clean (Baseline)	0.383	0.619	0.375	0.361	0.356	0.419
MF-ii (No Defense)	0.689	0.784	0.772	0.732	0.674	0.730
MF-ii (DiffCAP)	**0.470**	**0.505**	**0.481**	**0.459**	**0.411**	**0.465**
MF-ii + MF-tt (No Defense)	0.756	0.752	0.787	0.772	0.750	0.763
MF-ii + MF-tt (DiffCAP)	**0.405**	**0.528**	**0.439**	**0.422**	**0.380**	**0.435**

circumventing its defense, enhancing adversarial robustness against adaptive attacks of prohibitively high time complexity.

Ablation study. We conduct systematic ablation studies to validate the utility of DiffCAP. Tab. 4 presents results obtained by replacing the vision encoder in DiffCAP with different CLIP backbones. The results illustrate that DiffCAP is largely insensitive to the choice of vision encoder, maintaining robustness across all variants. To validate the effectiveness of the adaptive similarity threshold calculation described in Alg. 2, we conduct an ablation study over different threshold values τ and diffusion step sizes Δt. Fig. 2 displays the results by OF on the COCO dataset. We observe that setting the threshold to 0.96 achieves the best overall robustness in terms of CIDEr score across a range of step sizes. A higher threshold generally leads to more diffusion steps, increasing time for reverse diffusion sampling. In practice, we find that a step size of 0.01 offers the best trade-off between performance and computational cost.

Other attack and model. We consummate with experiments on MiniGPT 4-13B (Zhu et al., 2023). We evaluated DiffCAP with default hyperparameters against the AttackVLM (Zhao et al., 2023), which proposed more advanced transfer-based and query-based adversarial attacks, specifically designed to mislead VLMs into generating specific target captions, rather than plain untargeted degradation. We strictly follow their evaluation protocol and attack configurations. The clean images are sampled from the ImageNet1K (Deng et al., 2009) dataset and a target text is randomly selected from the COCO captions for each clean image. We report the CLIP score between the generated responses of input images and predefined targeted texts, as computed by various CLIP text encoders and their average. The prompt is fixed as `"what is the content of this image?"`. Pretrained CLIP encoders (ViT-B/32) are used as *surrogate* models for attacks.

As shown in Tab. 5, the 'No Defense' settings yield high CLIP scores, indicating that the VLM was successfully manipulated into generating the target text. However, DiffCAP maintains its effectiveness under MF-ii (image-to-image transfer) and MF-ii + MF-tt (joint image-text query) attacks, reducing the CLIP scores comparable to, and in some cases lower than, the clean baseline. This verify the generalizability of DiffCAP for larger VLM architectures and stronger attack strategies.

Warning: The first example contains adversarially generated text that may be considered offensive.

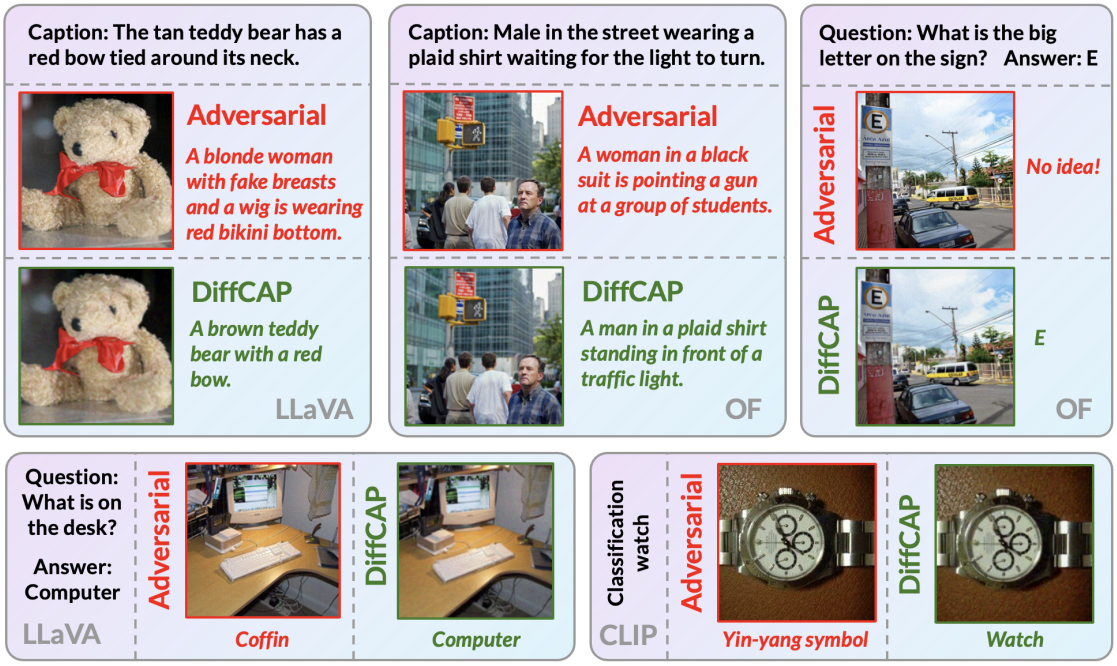

Figure 4: Adversarial examples and their DiffCAP purified outcomes under different tasks. Ground-truth labels are shown in black text. VLMs used for inference are shown in gray text.

Efficiency. For the $\ell_\infty^{2/255}$ attack on the COCO dataset with OF-9B, DiffCAP demonstrates substantial efficiency gain over DiffPure in IC task. After a one-time calibration of threshold τ (~ 3.4 seconds) by Alg. 2, the average purification time is only ~ 1.1 seconds per image for DiffCAP, in contrast to ~ 2.3 seconds for DiffPure. The embedding extraction and the Gaussian noise injection consume only ~ 6 and ~ 4 milliseconds per iteration, respectively. Since the reverse denoising dominates the runtime, the additional overhead from embedding comparisons hardly offsets the runtime savings achieved through the reduced $\sim 2/3$ diffusion steps over DiffPure (illustrated by Fig. 3), ensuring DiffCAP's practicality for real-time deployment scenarios.

Further discussion. In Sec. C, we supplement additional experimental results on the ZSC task, $\ell_\infty^{16/255}$ attacks, robustness to visual hallucination (Li et al., 2023) and jailbreaking (Qi et al., 2024), perceptual quality, ℓ_2 attacks, end-to-end runtime and memory comparisons, and under-/over-purification. In Sec. D, we further attach a systematic analysis of DiffCAP's critical hyperparameters, including the threshold sensitivity (across calibration subsets, domain shifts, task transformations, and CLIP variants), the step size control (on robustness-fidelity trade-off), and the noise scheduler choice.

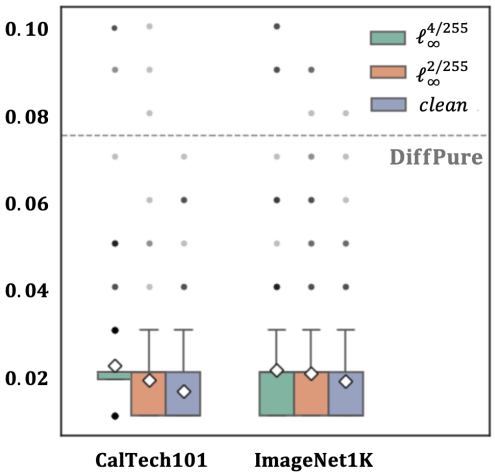

Figure 3: The box plot of DiffCAP's diffusion time t (y-axis) before exiting noise injection loop. The dashline ($t = 0.075$) is the noise injection time tuned on ℓ_∞ attack reported by DiffPure paper. DiffCAP requires significantly smaller diffusion time than DiffPure. The dots mark outliers and rhombuses mark mean values.

Fig. 4 showcases the purification consequence of DiffCAP in IC, VQA, and ZSC scenarios. As a generative adversarial purification method, it introduces no noticeable degradation in fidelity. DiffCAP prominently

mitigates the tension between robustness, efficiency, and image quality, establishing a new state-of-the-art among both purification- and training-based defenses for VLMs.

6 Conclusion

In conclusion, this paper proposes DiffCAP, an efficient and theoretically inspired defense strategy for VLMs, supported by a certified recovery region and descending semantic change in forward diffusion. By leveraging cumulative Gaussian noise injection and a VLM embedding similarity-based stopping criterion, DiffCAP dynamically identifies the minimal purification steps required before denoising, substantially reducing computational overhead while maintaining high fidelity. DiffCAP consistently outperforms existing defenses under heterogeneous attacks across manifold tasks, VLMs, and datasets empirically.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018a.

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

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

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

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI TIT*, pp. 25–45, 2021.

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

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

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

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Hadi M Dolatabadi, Sarah Erfani, and Christopher Leckie. ℓ_∞-robustness and beyond: Unleashing efficient adversarial training. In *European Conference on Computer Vision*, pp. 467–483. Springer, 2022.

Jia Fu, Xiao Zhang, Sepideh Pashami, Fatemeh Rahimian, and Anders Holst. Diffpad: Denoising diffusion-based adversarial patch decontamination. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6602–6611. IEEE, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144, 2020.

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

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8340–8349, 2021.

Mitch Hill, Jonathan Craig Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *International Conference on Learning Representations*, 2021.

Md Zarif Hossain and Ahmed Imteaj. Securing vision-language models with a robust encoder against jailbreak and adversarial attacks. *arXiv preprint arXiv:2409.07353*, 2024a.

Md Zarif Hossain and Ahmed Imteaj. Sim-clip: Unsupervised siamese adversarial fine-tuning for robust and semantically-rich vision-language models. *arXiv preprint arXiv:2407.14971*, 2024b.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Haibo Jin, Leyang Hu, Xinuo Li, Peiyan Zhang, Chonghan Chen, Jun Zhuang, and Haohan Wang. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *arXiv preprint arXiv:2407.01599*, 2024.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022a.

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

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pp. 121–137. Springer, 2020.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Yu Cheng, and Wei Hu. A survey of attacks on large vision-language models: Resources, advances, and future trends. *arXiv preprint arXiv:2407.07403*, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023b.

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024c. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. *arXiv preprint arXiv:2212.07016*, 2022.

Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

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

OpenAI. Chatgpt-4o, 2024. URL `https://openai.com`. Large Language Model.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ahmed H Salamah, Kaixiang Zheng, Yiwen Liu, and En-Hui Yang. Jpeg inspired deep learning. *arXiv preprint arXiv:2410.07081*, 2024.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

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

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

Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33, 2020.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

Fenghua Weng, Yue Xu, Chengyan Fu, and Wenjie Wang. Mmj-bench: A comprehensive study on jailbreak attacks and defenses for vision language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 27689–27697, 2025.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Xiyang Wu, Ruiqi Xian, Tianrui Guan, Jing Liang, Souradip Chakraborty, Fuxiao Liu, Brian M Sadler, Dinesh Manocha, and Amrit Bedi. On the safety concerns of deploying llms/vlms in robotics: Highlighting the risks and vulnerabilities. In *First Vision and Language for Autonomous Driving and Robotics Workshop*, 2024a.

Yongtao Wu, Fanghui Liu, Carl-Johann Simon-Gabriel, Grigorios Chrysos, and Volkan Cevher. Robust NAS under adversarial training: benchmark, theory, and beyond. In *The Twelfth International Conference on Learning Representations*, 2024b. URL `https://openreview.net/forum?id=cdUpf6t6LZ`.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, pp. 12062–12072. PMLR, 2021.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18123–18133, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Jiaming Zhang, Qi Yi, and Jitao Sang. Towards adversarial attack on vision-language pre-training models. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 5005–5013, 2022.

Mingkun Zhang, Keping Bi, Wei Chen, Jiafeng Guo, and Xueqi Cheng. Clipure: Purification in latent space via clip for adversarially robust zero-shot classification. *arXiv preprint arXiv:2502.18176*, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Yi Zhang, Orestis Plevrakis, Simon S Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. *Advances in Neural Information Processing Systems*, 33:679–688, 2020.

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

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

Contents of the Appendix

We organize the appendix as follows:

- In Sec. A, we declare the LLM usage, limitations, reproducibility, and border impact of this paper.
- In Sec. B, we provide complete proofs for Thm. 1 and 2 and Lemma 1.
- In Sec. C, we supply more implementation details and experiment results. We further explore the potential of applying DiffCAP to boarder safety alignment challenges.
- In Sec. D, we attach a comprehensive analysis of the key hyperparameters of DiffCAP.

A Author statements

The LLM usage. We only use the ChatGPT-4o (OpenAI, 2024) to rectify writing errors paragraph by paragraph as `"Please only revise the necessary part of the following paragraph if there are any incorrect grammar, unclear syntax, or unacademic expression: [our paragraph]"`.

Future work. While our method is highly effective in the vision modality, an important limitation is its current reliance on image-based diffusion models. Extending the cumulative purification framework to text or multimodal diffusion processes remains an open direction, potentially broadening its applicability to adversarial text modifications or joint vision-text threats.

Reproducibility. We provide the clear assumptions and a complete proof of the proposed theorems and lemmas in Sec. 4 and Sec. B, respectively. All used datasets/VLMs, pretrained diffusion models, and applied adversarial attacks in this work are publicly available. Our code will also be released upon the acceptance. In the meantime, the relevant experimental setups are fully described in paper Sec. 5.1 and Sec. C.1, together with the step-by-step algorithm flow Alg. 1 and Alg. 2, ensuring that the results can be manually reproduced.

Social impact. With the swift application of VLMs, the risk of adversarial attacks has become a critical concern. This paper proposes DiffCAP, an adversarial purification method that can improve robustness without retraining the model, which may enlarge its usability in application scenarios, such as autonomous driving based on VLMs. While maintaining a remarkable defense effect, this method greatly reduces the diffusion steps and hyperparameter adjustments, which promotes the safe and fast implementation for defending pre-trained large models. However, any single defense mechanism may fail in the presence of new attacks, so maintaining a diverse and regularly tested defense strategy is essential.

B Theoretical proofs

B.1 Proof of Thm. 1

Proof of Thm. 1. Expanding the forward diffusion solution from Eq. (5), we get:

$$\boldsymbol{x}(t) = \sqrt{\alpha(t)}\,\boldsymbol{x}_{\text{adv}} + \sqrt{1-\alpha(t)}\,\boldsymbol{\epsilon} = \sqrt{\alpha(t)}\left[\boldsymbol{x} + \frac{\sqrt{1-a(t)}}{\sqrt{a(t)}}\boldsymbol{\epsilon} + \boldsymbol{\epsilon}_{\text{adv}}\right], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I).$$

Given Asm. 1 and the property of Gaussian distribution, we can analyze the classifier output as follows by absorbing the scaling factor $\sqrt{a(t)}$ and introduce $\boldsymbol{\epsilon}'$:

$$h(\boldsymbol{x}(t)) = h(\boldsymbol{x} + \boldsymbol{\epsilon}' + \boldsymbol{\epsilon}_{\text{adv}}), \quad \text{where } \boldsymbol{\epsilon}' \sim \mathcal{N}(0, \sigma(t)^2 I), \quad \sigma(t)^2 = \frac{1-\alpha(t)}{\alpha(t)}.$$

Applying the randomized smoothing bound from Theorem 1 of (Cohen et al., 2019), the classification is guaranteed to return class k_1 if:

$$\|\boldsymbol{\epsilon}_{\text{adv}}\|_2 < \frac{\sigma(t)}{2}\left(\Phi^{-1}(\underline{p_1}) - \Phi^{-1}(\overline{p_2})\right).$$

By re-organizing the term, we have:

$$\sigma(t) > \frac{2\|\boldsymbol{\epsilon}_{\text{adv}}\|_2}{\Phi^{-1}(\underline{p_1}) - \Phi^{-1}(\overline{p_2})}.$$

Since $\sigma(t)^2 = \frac{1-\alpha(t)}{\alpha(t)}$, this yields:

$$\frac{1-\alpha(t)}{\alpha(t)} > \left(\frac{2\|\boldsymbol{\epsilon}_{\text{adv}}\|_2}{\Phi^{-1}(\underline{p_1}) - \Phi^{-1}(\overline{p_2})}\right)^2.$$

By re-organizing the term, we obtain:

$$\alpha(t) < \frac{1}{1 + \left(\frac{2\|\boldsymbol{\epsilon}_{\text{adv}}\|_2}{\Phi^{-1}(\underline{p_1}) - \Phi^{-1}(\overline{p_2})}\right)^2}.$$

Since $\alpha(t) = \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right)$, and with linear schedule $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$, we compute:

$$\int_0^t \beta(s)\mathrm{d}s = \beta_{\min} t + \frac{1}{2}(\beta_{\max} - \beta_{\min})t^2.$$

Thus,

$$\alpha(t) = \exp\left(-\beta_{\min} t - \frac{1}{2}(\beta_{\max} - \beta_{\min})t^2\right).$$

Let $M := \log\left(1 + \left(\frac{2\|\boldsymbol{\epsilon}_{\text{adv}}\|_2}{\Phi^{-1}(\underline{p_1}) - \Phi^{-1}(\overline{p_2})}\right)^2\right)$. Then by setting:

$$\beta_{\min} t + \frac{1}{2}(\beta_{\max} - \beta_{\min})t^2 = M,$$

we obtain:

$$t = t_{\min} = \frac{2M}{\sqrt{\beta_{\min}^2 + 2(\beta_{\max} - \beta_{\min})M} + \beta_{\min}}.$$

Then, when $\beta_{\min} \geq M$, we have $t_{\min} \leq 1$, and for $t_{\min} \leq t \leq 1$, we have $\arg\max_{k\in[K]} \mathbb{P}(h(\boldsymbol{x}(t) = k) = k_1$. □

B.2 Proof of Lemma 1

Proof of Lemma 1. we have:

$$\mathbb{E}\left[\|\phi(\boldsymbol{x}(t_1)) - \phi(\boldsymbol{x}(t_2))\|_2\right] \leq L \cdot \mathbb{E}\left[\|\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2)\|_2\right] \tag{8}$$

$$\leq L \cdot \sqrt{\mathbb{E}\left[\|\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2)\|_2^2\right]}, \tag{9}$$

where Eq. (8) follows from the Lipschitz continuity of ϕ, Eq. (9) uses Jensen's inequality, i.e., $\mathbb{E}\left[\|Z\|\right] \leq \sqrt{\mathbb{E}\left[\|Z\|^2\right]}$ for any random vector Z. Next, to compute $\mathbb{E}\left[\|\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2)\|^2\right]$, use Eq. (5):

$$\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2) = \left(\sqrt{\alpha(t_1)} - \sqrt{\alpha(t_2)}\right)\boldsymbol{x}_{\text{adv}} + \left(\sqrt{1-\alpha(t_1)} - \sqrt{1-\alpha(t_2)}\right)\boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. Taking the squared norm and expectation:

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}(t_1) - \boldsymbol{x}(t_2)\|^2\right] &= \left(\sqrt{\alpha(t_1)} - \sqrt{\alpha(t_2)}\right)^2 \|\boldsymbol{x}_{\text{adv}}\|^2 + \left(\sqrt{1-\alpha(t_1)} - \sqrt{1-\alpha(t_2)}\right)^2 \cdot \mathbb{E}\left[\|\boldsymbol{\epsilon}\|^2\right] \\ &= \left(\sqrt{\alpha(t_1)} - \sqrt{\alpha(t_2)}\right)^2 \|\boldsymbol{x}_{\text{adv}}\|^2 + \left(\sqrt{1-\alpha(t_1)} - \sqrt{1-\alpha(t_2)}\right)^2 d.\end{aligned}$$

As $t_1, t_2 \to 1$, both terms vanish, so the expectation tends to zero. □

B.3 Proof of Thm. 2

Before proving Thm. 2, we need the following Lemma.

Lemma 2. *Let $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t$ be a linear noise schedule with $\beta_{\min} > 0$ and $\beta_{\max} > \beta_{\min}$. Define:*

$$\alpha(t) := \exp\left(-\int_0^t \beta(s)\,\mathrm{d}s\right), \quad \textit{and} \quad f(t) := \beta(t) \cdot \sqrt{\frac{\alpha(t)}{1-\alpha(t)}}.$$

Then $f(t)$ is strictly decreasing for all $t \in [0, 1)$.

Proof. We first analyze the function $f(t)$ by taking its logarithm:

$$\log f(t) = \log \beta(t) + \frac{1}{2}\log\left(\frac{\alpha(t)}{1-\alpha(t)}\right).$$

Differentiating and using the chain rule, we attain:

$$\begin{aligned}
\frac{d}{dt}\log f(t) &= \frac{\beta'(t)}{\beta(t)} + \frac{1}{2}\left(\frac{d}{dt}\log\alpha(t) - \frac{d}{dt}\log(1-\alpha(t))\right) \\
&= \frac{\beta'(t)}{\beta(t)} + \frac{1}{2}\cdot\alpha'(t)\left(\frac{1}{\alpha(t)} + \frac{1}{1-\alpha(t)}\right) \\
&= \frac{\beta'(t)}{\beta(t)} - \frac{1}{2}\cdot\frac{\beta(t)}{1-\alpha(t)},
\end{aligned} \tag{10}$$

where we use $\alpha'(t) = -\beta(t)\cdot\alpha(t)$. Let's define $g(t) := \frac{d}{dt}\log f(t)$. We now analyze the sign of $g(t)$. First, observe that $\beta'(t) = \beta_{\max} - \beta_{\min} > 0$ is constant, $\beta(t) \geq \beta_{\min} > 0$, so $\frac{\beta'(t)}{\beta(t)}$ is strictly decreasing in t. Also $\alpha(t)$ is strictly decreasing, so $1-\alpha(t)$ is strictly increasing, and hence $\frac{\beta(t)}{1-\alpha(t)}$ is strictly increasing. Therefore, $g(t)$ is strictly decreasing in t. To show that $g(t) < 0$ for all $t \in [0, 1)$, it suffices to show $g(t) < 0$ near $t = 0$. When $t \approx 0$, we take:

$$\beta(t) \approx \beta_{\min}, \quad \int_0^t \beta(s)\,\mathrm{d}s \approx \beta_{\min}t, \quad \alpha(t) = \exp(-\beta_{\min}t) \approx 1 - \beta_{\min}t + o(t).$$

Thus,

$$\frac{\beta(t)}{1-\alpha(t)} \approx \frac{\beta_{\min}}{\beta_{\min}t} = \frac{1}{t}, \quad \text{which diverges as } t \to 0.$$

Given the result:

$$\frac{\beta'(t)}{\beta(t)} \approx \frac{\beta_{\max} - \beta_{\min}}{\beta_{\min}},$$

which is a finite number, thus, $g(t) \to -\infty$ as $t \to 0$. Since $g(t)$ is strictly decreasing, it follows that $g(t) < 0$ for all $t \in [0, 1)$. This implies that $\log f(t)$ is strictly decreasing, and thus $f(t)$ is strictly decreasing as well. □

Now we are ready to present the proof of Thm. 2.

Proof of Thm. 2. Let $\Delta(t, \delta) := \boldsymbol{x}(t+\delta) - \boldsymbol{x}(t)$. From the closed-form solution of the VP-SDE,

$$\boldsymbol{x}(t) = \sqrt{\alpha(t)}\boldsymbol{x}_{\text{adv}} + \sqrt{1-\alpha(t)}\boldsymbol{\epsilon},$$

we have:

$$\Delta(t,\delta) = \left(\sqrt{\alpha(t+\delta)} - \sqrt{\alpha(t)}\right)\boldsymbol{x}_{\text{adv}} + \left(\sqrt{1-\alpha(t+\delta)} - \sqrt{1-\alpha(t)}\right)\boldsymbol{\epsilon}.$$

Let us define:

$$A := \sqrt{\alpha(t+\delta)} - \sqrt{\alpha(t)}, \quad B := \sqrt{1-\alpha(t+\delta)} - \sqrt{1-\alpha(t)}. \tag{11}$$

Then: $\Delta(t,\delta) = A\boldsymbol{x}_{\text{adv}} + B\boldsymbol{\epsilon}$. By the Lipschitz property of ϕ and the Cauchy-Schwarz inequality:

$$\mathbb{E}\left[\|\phi(\boldsymbol{x}(t+\delta)) - \phi(\boldsymbol{x}(t))\|_2\right] \leq L \cdot \mathbb{E}\left[\|\Delta(t,\delta)\|_2\right] \leq L \cdot \sqrt{\mathbb{E}\left[\|\Delta(t,\delta)\|_2^2\right]}, \tag{12}$$

where $L \in \{L_1, L_2\}$. We now compute this second moment:

$$\mathbb{E}\left[\|\Delta(t,\delta)\|_2^2\right] = \mathbb{E}\left[\|A\boldsymbol{x}_{\text{adv}} + B\boldsymbol{\epsilon}\|_2^2\right] = A^2\|\boldsymbol{x}_{\text{adv}}\|_2^2 + B^2\mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2^2\right].$$

Since $\boldsymbol{\epsilon}$ is standard Gaussian in $\mathbb{R}^d$, $\mathbb{E}[\|\boldsymbol{\epsilon}\|^2] = d$. Thus:

$$\mathbb{E}\left[\|\Delta(t,\delta)\|_2^2\right] = A^2\|\boldsymbol{x}_{\text{adv}}\|_2^2 + B^2 d.$$

We now perform Taylor expansion for A and B with respect to t. First note that:

$$\frac{d}{dt}\alpha(t) = -\beta(t)\alpha(t), \quad \frac{d}{dt}\sqrt{\alpha(t)} = -\frac{\beta(t)}{2}\sqrt{\alpha(t)}.$$

Plugging the above equation back into Eq. (11), we derive:

$$A = \sqrt{\alpha(t+\delta)} - \sqrt{\alpha(t)} = -\frac{\beta(t)}{2}\sqrt{\alpha(t)}\delta + o(\delta).$$

Similarly:

$$\frac{d}{dt}\sqrt{1-\alpha(t)} = \frac{\beta(t)\alpha(t)}{2\sqrt{1-\alpha(t)}}.$$

Plugging the above equation back into Eq. (11):

$$B = \sqrt{1-\alpha(t+\delta)} - \sqrt{1-\alpha(t)} = \frac{\beta(t)\alpha(t)}{2\sqrt{1-\alpha(t)}}\delta + o(\delta).$$

Take the square and sum them up:

$$A^2 + B^2 = \frac{\beta(t)^2\delta^2}{4}\left(\alpha(t) + \frac{\alpha(t)^2}{1-\alpha(t)}\right) + o(\delta^2) = \frac{\beta(t)^2\alpha(t)\delta^2}{4(1-\alpha(t))} + o(\delta^2).$$

Since $\|\boldsymbol{x}_{\text{adv}}\|^2 \leq d$, we gain:

$$\mathbb{E}\left[\|\Delta(t,\delta)\|_2^2\right] \leq \frac{d \cdot \beta(t)^2\alpha(t)}{4(1-\alpha(t))}\delta^2 + o(\delta^2). \tag{13}$$

Plugging Eq. (13) into Eq. (12), we arrive:

$$\mathbb{E}\left[\|\phi(\boldsymbol{x}(t+\delta)) - \phi(\boldsymbol{x}(t))\|_2\right] \leq L \cdot \sqrt{\mathbb{E}\left[\|\Delta(t,\delta)\|_2^2\right]} = O\left(L \cdot \delta \cdot \beta(t)\sqrt{\frac{\alpha(t)}{1-\alpha(t)}}\right).$$

Lastly, by Lemma 2, we prove that the bound on the right hand side decreases as t grows. □

C Additional experiments

C.1 More implementation details

All baseline methods are evaluated using their respective best-performing hyperparameters as reported in the original papers. The experiments in Sec. C.4 are conducted on an NVIDIA H100 GPU. We fix the threshold except for experiments in Sec. D. Unless otherwise specified, the remaining setups of the experiments in appendix are the same as those in the main text.

To keep the evaluation of adaptive attacks on large VLMs computationally tractable, we randomly choose 100 images per dataset. BPDA (Athalye et al., 2018a) attacks are run for 50 iterations. For the forward

Table 6: Top-1 accuracy (%) of CLIP in ZSC task with clean images and adversarial perturbations of two sizes under different defenses. 2 and 4 with TeCoA and FARE suggest the version that is fine-tuned by $\ell_\infty^{2/255}$ and $\ell_\infty^{4/255}$ bounded adversarial examples, respectively. The best result is in **bold** and the runner-up is underlined.

Defense	**CalTech101**			**ImageNet-R**		
	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$
No defense	83.3	0.0	0.0	87.9	0.0	0.0
TeCoA^2 (Mao et al., 2022)	80.7	70.2	57.4	80.1	58.8	36.7
FARE^2 (Schlarmann et al., 2024)	**84.8**	73.0	46.6	85.5	56.5	25.6
TeCoA^4 (Mao et al., 2022)	78.4	69.7	60.9	74.3	59.2	41.9
FARE^4 (Schlarmann et al., 2024)	84.7	76.7	64.1	80.2	61.6	40.6
JPEG-DL (Salamah et al., 2024)	83.9	68.5	33.4	**87.8**	48.5	16.4
DiffPure (Nie et al., 2022)	83.6	**83.2**	**83.1**	81.0	79.7	79.7
CLIPure (Zhang et al., 2025)	82.9	80.8	80.1	87.7	**85.4**	**84.6**
DiffCAP	82.6	81.7	80.9	87.2	84.4	81.1

Table 7: CIDEr scores with VLMs OF-9B and LLaVA 1.5-7B on datasets COCO and Flickr30k.

Defense	**OF-COCO**	**OF-Flickr30k**	**LLaVA-COCO**	**LLaVA-Flickr30k**
Clean	79.7	60.1	115.5	77.5
No defense	0.9	0.3	1.5	0.5
JPEG-DL	5.2	3.2	5.0	2.7
DiffPure	72.7	49.2	100.0	62.5
DiffCAP	**74.8**	**53.2**	**109.7**	**67.6**

pass, we execute $x' = \text{DiffCAP}(x)$ wrapped as a `torch.nn.Module` to serve as the first layer of the VLM. For the backward pass, we approximate the gradient of the DiffCAP$(\cdot)$ with respect to the input as the *identity matrix* ($\nabla_x \text{DiffCAP}(x) \approx I$), leveraging the "detach" trick in `PyTorch`. This implementation allows BPDA to fully exploit DiffCAP's knowledge, including the CLIP (ViT-B/32), the stopping rule, and the semantic similarity threshold. We integrate this BPDA module directly into the APGD framework. For EOT (Athalye et al., 2018b), we estimate the expected gradients by averaging the BPDA-derived gradients over three stochastic forward passes of the DiffCAP$(\cdot)$ for attack optimization. BPDA and EOT ensure the adaptive adversarial examples are generated with stable gradient flow and robust to the randomness from the diffusion sampling.

C.2 Zero-shot classification

From Tab. 6, we observe that JPEG-DL underperforms compared to TeCoA and FARE, particularly under stronger attacks. CLIPure recovers its defense effectiveness, as only the `[CLS]` token is involved in prediction and no generative decoding is required. On CalTech101, DiffCAP outperforms CLIPure under attacks, and on ImageNet-R, DiffCAP achieves higher robustness than DiffPure. These results confirm the generalizability of DiffCAP: it not only excels in complex vision-language tasks requiring rich semantics but also delivers stable performance on standard classification benchmarks with more sparse semantic demands.

C.3 Larger attack budget

We conducted additional experiments on the IC task using the larger budget of $\ell_\infty^{16/255}$. We excluded TeCoA, FARE, and CLIPure from this evaluation, since they exhibited limited performance even under smaller perturbations. The results are reported in Tab. 7. Unlike under $\ell_\infty^{2/255}$ and $\ell_\infty^{4/255}$ attacks, JPEG-DL collapses with negligible improvement. While DiffPure demonstrates that diffusion-based purification remains viable at

Table 8: Evaluation of DiffCAP in mitigating Hallucination and Jailbreaking of large VLMs.

	Hallucination with LLaVA 1.5-13B			**Jailbreaking with MiniGPT 4-13B**				
	adversarial	popular	random	any	identity	disinfo	crime	x-risk
Clean	82.7	84.3	85.0	16/40	3/11	6/13	6/13	1/3
Attack	N/A	N/A	N/A	24/40	6/11	7/13	12/13	2/3
DiffCAP	83.2	85.1	86.3	14/40	3/11	5/13	5/13	1/3

Table 9: Perceptual quality and semantic preservation. ↑ for higher is better, and ↓ for lower is better.

VLM-Dataset	**PSNR (dB) ↑**		**LPIPS ↓**		**CLIP Score ↑**	
	DiffPure	DiffCAP	DiffPure	DiffCAP	DiffPure	DiffCAP
OF-COCO	24.27	**29.03**	0.296	**0.160**	0.871	**0.930**
OF-Flickr30k	23.44	**28.21**	0.298	**0.151**	0.865	**0.930**
LLaVA-COCO	25.41	**29.76**	0.304	**0.200**	0.849	**0.893**
LLaVA-Flickr30k	24.59	**29.08**	0.306	**0.194**	0.837	**0.884**

this magnitude, it consistently underperforms DiffCAP by a substantial margin across different combinations of VLMs and datasets.

C.4 Hallucination and jailbreaking

Large VLMs tend to hallucinate objects that are not actually present in the image. POPE (Li et al., 2023) serves as a benchmark to formulate hallucination detection as a binary classification task. In Tab. 8, we report the F1-scores across three POPE categories using LLaVA 1.5-13B, with and without DiffCAP applied as image preprocessing. A consistent improvement is observed with DiffCAP. This suggests that DiffCAP, through Langevin dynamics, walks image features to semantically stable regions of the distribution. By suppressing high-frequency adversarial or spurious signals, DiffCAP becomes less sensitive to misleading cues and more robust against hallucination.

Large VLMs are also vulnerable to jailbreaking attacks on the visual modality (Carlini et al., 2024), where adversarially crafted images can induce harmful outputs in response to restricted prompts (e.g., `"How to make a bomb?"`). We apply the attack proposed by (Qi et al., 2024) to MiniGPT 4-13B and count the policy-violating outputs triggered by 40 harmful prompts spanning four categories. Even under a stronger perturbation budget ($\ell_\infty^{16/255}$), DiffCAP successfully restores the model's behavior to a level comparable to, or slightly better than, the clean condition. These findings reinforce the versatility of DiffCAP as a modular defense measure, readily adaptable to various VLMs and tasks requiring robustness guarantee. As jailbreaking attacks continue to evolve rapidly, benchmarking DiffCAP against such threats falls outside the scope of this work, but nonetheless marks a promising direction for future investigation.

C.5 Perceptual quality

We assessed the perceptual quality and semantic preservation of the purified images compare to their clean counterparts on the IC task under $\ell_\infty^{4/255}$ adversarial perturbations. We employed three standard metrics: PSNR (Peak Signal-to-Noise Ratio) measures pixel-level signal fidelity; LPIPS (Zhang et al., 2018) (Learned Perceptual Image Patch Similarity) measures perceptual distance using deep features, aligning closely with human visual perception; CLIP Score measures the semantic consistency.

The quantitative comparison between DiffCAP and DiffPure is presented in Tab. 9. DiffCAP significantly outperforms DiffPure across all three metrics with different VLMs and datasets. The higher PSNR and lower LPIPS indicate that DiffCAP preserves fine-grained visual details and reduces perceptual artifacts effectively.

Table 10: VQA accuracy (%) on the TextVQA dataset with OF-9B under different attack radii.

	clean	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	$\ell_\infty^{8/255}$	$\ell_2^{2/255}$	$\ell_2^{4/255}$	$\ell_2^{8/255}$
No defense	23.8	0.0	0.0	0.0	6.2	6.0	4.8
DiffCAP	18.6	16.2	16.7	16.7	16.5	16.6	16.2

Table 11: Runtime and memory comparison.

Method	**Time per Image**	**Peak GPU Memory**
JPEG-DL	0.01 s	N/A (CPU)
DiffPure	2.3 s	2.7 GB
CLIPure	0.01 s	0.3 GB
DiffCAP	1.1 s	2.5 GB

The superior CLIP Scores further validate our method's high semantic integrity, establishing a new SOTA balance between robustness and faithfulness among purification methods.

C.6 Other discussion

ℓ_2 bounded threats. We perform a test of the VQA task, where Tab. 10 compares DiffCAP with no-defense baseline. The results demonstrate that DiffCAP maintains its effectiveness regardless of whether the adversarial perturbations follow ℓ_∞ or ℓ_2 norm bound.

End-to-end runtime and memory comparisons. We report the additional computational overhead introduced by the defense mechanisms, excluding the standard VLM inference portion. All measurements were performed with a batch size of 1, on the COCO with OF and $\ell_\infty^{2/255}$ adversarial examples.

For adversarially fine-tuned vision encoders (TeCoA and FARE), the computational burden is entirely from training phase. Once deployed, these methods incur zero additional inference latency and no extra memory overhead compared to the original VLMs, as they simply replace the weights of vision encoder. However, they require massive computational resources for training on perturbed data. For test-time defenses, we report the comparison of runtime and memory usage in the Tab. 11.

Under- and over-purification. We conduct a trial of IC task on the COCO dataset with LLaVA 1.5-7B. Tab. 12 records the reductions in CIDEr scores of DiffPure with two diffusion time compared to DiffCAP under three attack radii. DiffPure with a lower fixed diffusion time suffers from under-purification (insufficient robustness) against stronger attacks, while using a higher fixed diffusion time leads to over-purification (visual artifacts) against weaker attacks. This underscores the benefit of DiffCAP's design, where the dynamic diffusion mechanism determines the optimal purification extent for different attack vectors.

D Systematic analysis of hyperparameters

D.1 The variance of the threshold

The number of images in the calibration set for Alg. 2 has a faint impact on threshold calculation. When we vary the subset size of image pairs to 100, 200, 300, and use three different random seeds, the resulting τ remains stable at 0.958 ± 0.003.

D.2 The threshold for out-of-distribution (OOD)

Severe distribution shifts naturally degrade VLM performance compared to in-domain natural images. For example, VLMs can underperform simple Convolutional Neural Networks (CNNs) on datasets like

Table 12: The CIDEr score gains of DiffCAP over DiffPure. $\ell_\infty^{8/255}$ attack is with BPDA.

Diffusion time	$\ell_\infty^{2/255}$	$\ell_\infty^{4/255}$	$\ell_\infty^{8/255}$
0.020	5.2	5.9	18.2
0.075	11.2	11.9	11.7

Table 13: Top-1 accuracy (%) on 1,000 randomly selected MNIST test images under $\ell_\infty^{4/255}$ attack.

Clean	No defense	DiffCAP ($\tau = 0.96$)	DiffCAP ($\tau = 0.97$)
75.2	0.0	79.6	80.9

MNIST (Deng, 2012). However, our empirical results demonstrate that τ is highly robust, and Alg. 2 serves as an optimizer when representative data (e.g., medical or satellite image) are available. The calibration provides a minor performance boost typical of hyperparameter tuning but is not required to achieve SOTA defense.

We first deliver a validation on the MNIST dataset for ZSC task with CLIP. The results in Tab. 13 suggest that DiffCAP maintains strong performance even when the domain departs from natural, real-world imagery that characterizes most VLM application scenarios. While the threshold 0.96 is already robust for this non-natural, digital domain, using the threshold 0.97 computed on the MNIST via Alg. 2 brings a slight gain in performance.

To further analyze how semantic complexity affects the acquisition of τ, we conducted additional experiments on two datasets with distinct styles: ImageNet1K (colorful photos representing *rich* semantics) and ImageNet-S (Wang et al., 2019) (black and white outlines representing *sparse* semantics). We compressed resolutions and crafted long-tail categories to simulate the extreme OOD conditions.

Alg. 2 calibrates $\tau = 0.95$ for ImageNet1K and $\tau = 0.97$ for ImageNet-S. This aligns with our intuition: rich semantics are easier to stabilize, necessitating a slightly lower threshold, whereas sparse semantics reach the recovery region harder. The former contains redundant textural and color cues that facilitate faster feature reconstruction by the diffusion model, and the latter is more sensitive to noise interference. Despite these differences, the performance variance across the $\tau \in [0.95, 0.97]$ interval is inapparent. Even using a sub-optimal threshold, DiffCAP consistently outperforms the DiffPure.

The results in Tab. 14 corroborated that τ is not brittle to severe domain shifts in terms of both measurement and deployment. Users can safely tune within the $0.95 \sim 0.97$ "safety belt" without the risk of losing SOTA performance, although Alg. 2 allows for a quicker hyperparameter search.

D.3 The threshold for different tasks

The calculation of τ by Alg. 2 is based on the semantic stability of image embeddings and is therefore *task-agnostic*. Here we verify whether the calibrated threshold remains optimal across different downstream tasks. As shown in Fig. 2, for the IC task on COCO with OF-9B under $\ell_\infty^{2/255}$ attack, the calibrated $\tau = 0.96$ delivers the overall highest CIDEr scores across various diffusion step sizes Δt, reflecting the effectiveness of Alg. 2.

To demonstrate task transferability of τ, we conducted an analogous ablation study for the VQA task on VQAv2 with LLaVA 1.5-7B under $\ell_\infty^{4/255}$ attack. The Fig. 5 visualizes the quantitative results, which indicate that $\tau = 0.96$ achieves the highest accuracy at the default step size $\Delta t = 0.010$ while maintaining the efficiency. This is consistent with our observation in Fig. 2.

Table 14: Top-1 accuracy (%) of ZSC task with CLIP under $\ell_\infty^{4/255}$ attack on OOD datasets.

Setting	ImageNet1K	ImageNet-S(ketch)
Clean	75.5	58.5
No defense	0.0	0.1
DiffPure	63.5	50.1
DiffCAP ($\tau = 0.95$)	**69.2**	51.9
DiffCAP ($\tau = 0.96$)	67.6	52.5
DiffCAP ($\tau = 0.97$)	66.7	**53.1**

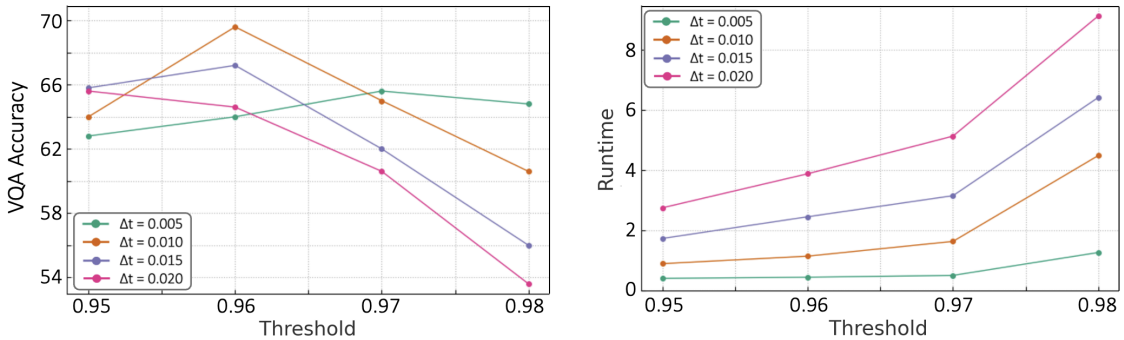

Figure 5: VQA Accuracy (%) and running time (in seconds) per image with varying thresholds τ and diffusion step sizes (Δt) for DiffCAP. The evaluation is based on the VQA task under $\ell_\infty^{4/255}$ attack.

D.4 The threshold for various CLIP

Alg. 2 employs a vision encoder to quantify semantic stability. To assess whether the encoder architecture biases the calibration of τ, we alternates Alg. 2 with several CLIP variants. The calibrated τ values are listed in Tab. 15, clustering around 0.96.

We posit that for natural images, unless the encoder architecture is fundamentally altered (e.g., deviating from the *contrastive learning* paradigm), the semantic threshold is primarily governed by the underlying data distribution rather than the specific vision backbone. This clue retrospectively explains the results in Tab. 4, where we revealed that $\tau = 0.96$ remains effective when the DiffCAP vision encoder is replaced by other CLIP variants. Overall, DiffCAP is largely insensitive to the vision backbone used for either calibration or purification.

D.5 The step size for robustness-fidelity

The trade-off is primarily influenced by the diffusion depth: deeper diffusion enhances robustness but risks losing information, while shallower diffusion preserves details but leaves residual adversarial noise. In DiffCAP, the diffusion depth is dynamically determined by the coupling between the Δt and τ.

Larger Δt "jumps farther" in the embedding space between steps. If combined with a high τ, i.e., a strict stability requirement, the forward diffusion may *overshoot* the recovery region, leading to redundant steps and over-purification. Smaller Δt induces finer granularity. If combined with a low τ, the stability condition can be triggered too early, causing *premature* stopping and under-purification.

Therefore, an optimal trade-off requires balancing Δt and τ, avoiding configurations where they are simultaneously too large or too small. Fig. 2 and Fig. 5 argue that, across different tasks, datasets, attack magnitudes, and VLMs, there exists a relatively secure range of Δt selection. With $\tau = 0.96$ fixed, $\Delta t \in [0.05, 0.15]$ consistently outperforms DiffPure, where $\Delta t = 0.10$ is set as a reliable and efficient default.

Table 15: The calibrated threshold across different vision backbones via Alg. 2.

	RN50	RN101	ViT-B/32	ViT-B/16	ViT-L/14	**Avg.**
τ	0.964	0.967	0.958	0.956	0.962	**0.961**

Table 16: Comparison of different scheduling strategies.

	Linear	**Cosine**	**Exponential Decay**
CIDEr Score for OF-COCO-$\ell_\infty^{2/255}$	79.3	79.2	79.2
VQA Acc. (%) for LLaVA-VQAv2-$\ell_\infty^{4/255}$	68.5	68.0	68.1

D.6 The justification of scheduler

We employ a linear noise schedule in DiffCAP for theoretical guarantee, alignment with diffusion dynamics, and empirical performance. Our theoretical derivations are explicitly formulated under the linear precondition. Regardless of whether the noise injection follows a linear or non-linear schedule, the cumulative noise will eventually push the image into the certified recovery region. With a sufficiently small step size ($\Delta t \approx 0.01$), the specific noise trajectory only marginally shifts the precise timestamp at which this region is entered but does not alter the fundamental semantic convergence behavior.

DiffCAP operates in *conjunction* with a pre-trained diffusion model for the reverse denoising step, where linear β schedule is adopted. To experimentally support DiffCAP is invariant to the moderate noise schedule variations, we compare our default Linear scheduler against a Cosine ($\Delta t' = \Delta t \cdot \cos(\frac{\pi i}{2N})$) and an Exponential Decay ($\Delta t' = \Delta t \cdot 0.9^i$) scheduler, where i denotes the step index and N refers to the total number of steps. As presented in Tab. 16, the alternative schedulers exhibit no material performance differences.