Efficient Generation of Targeted and Transferable Adversarial Examples for Vision-Language Models via Diffusion Models

Qi Guo, Shanmin Pang, *Member, IEEE*, Xiaojun Jia, Yang Liu, *Senior Member, IEEE*, and Qing Guo, *Senior Member, IEEE*

***Abstract*—Adversarial attacks, particularly targeted transfer-based attacks, can be used to assess the adversarial robustness of large visual-language models (VLMs), allowing for a more thorough examination of potential security flaws before deployment. However, previous transfer-based adversarial attacks incur high costs due to high iteration counts and complex method structure. Furthermore, due to the unnaturalness of adversarial semantics, the generated adversarial examples have low transferability. These issues limit the utility of existing methods for assessing robustness. To address these issues, we propose AdvDiffVLM, which uses diffusion models to generate natural, unrestricted and targeted adversarial examples via score matching. Specifically, AdvDiffVLM uses Adaptive Ensemble Gradient Estimation (AEGE) to modify the score during the diffusion model's reverse generation process, ensuring that the produced adversarial examples have natural adversarial targeted semantics, which improves their transferability. Simultaneously, to improve the quality of adversarial examples, we use the GradCAM-guided Mask Generation (GCMG) to disperse adversarial semantics throughout the image rather than concentrating them in a single area. Finally, AdvDiffVLM embeds more target semantics into adversarial examples after multiple iterations. Experimental results show that our method generates adversarial examples 5x to 10x faster than state-of-the-art (SOTA) transfer-based adversarial attacks while maintaining higher quality adversarial examples. Furthermore, compared to previous transfer-based adversarial attacks, the adversarial examples generated by our method have better transferability. Notably, AdvDiffVLM can successfully attack a variety of commercial VLMs in a black-box environment, including GPT-4V. The code is available at https://github.com/gq-max/AdvDiffVLM**

***Index Terms*—Adversarial attack, visual language models, diffusion models, score matching.**

I. Introduction

LARGE VLMs have shown great success in tasks like image-to-text [1], [2], [3] and text-to-image generation [4], [5]. Particularly in image-to-text generation, users can use images to generate executable commands for robot control [6], which has potential applications in autonomous driving systems [7], [8], visual assistance systems [9], and content moderation systems [10]. However, VLMs are highly susceptible to adversarial attacks [11], [12], which can result in life and property safety issues [13], [14]. As a result, it is critical to evaluate the adversarial robustness [15], [16], [17], [18] of these VLMs before deployment.

The early research on assessing the adversarial robustness of VLMs concentrated on white-box and untargeted scenarios [19], [20], [21]. Black-box and targeted attacks can cause models to generate targeted responses without knowing the models' internal information, resulting in greater harm [22], [23]. Furthermore, targeted attacks on black-box models present more challenges than untargeted attacks [24], [25]. As a result, when assessing the adversarial robustness of VLMs, it is critical to consider more threatening and challenging black-box and targeted attacks [16]. AttackVLM [16] is the first work to explore the adversarial robustness of VLMs in both black-box and targeted scenarios using query attacks with transfer-based priors. However, due to the large number of queries required and the complex method structure, this method is inefficient, which reduces its validity and suitability for a comprehensive assessment of the limitations of VLMs. Another attack method that can be used in black-box and targeted scenarios is the transfer-based attack [26], [27], [28], [29]. However, this type of attack method is slow to generate adversarial examples due to its complex structure and numerous iterations. Furthermore, because it adds unnatural adversarial semantics, the transferability of adversarial examples is poor. Unrestricted adversarial examples [30], [31], [32], [33] can incorporate more natural adversarial targeted semantics into the image, thereby

Received 22 July 2024; revised 20 November 2024; accepted 9 December 2024. Date of publication 23 December 2024; date of current version 21 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 61972312 and Grant 62376212; in part by the National Research Foundation, Singapore, and the Cyber Security Agency through its National Cybersecurity Research and Development Programme under Grant NCRP25-P04-TAICeN; in part by the National Research Foundation, Singapore, and Infocomm Media Development Authority through its Trust Tech Funding Initiative; in part by the National Research Foundation, Singapore, and Defence Science Organization (DSO) National Laboratories through the Artificial Intelligence (AI) Singapore Programme (AISG) under Award AISG2-GC-2023-008; and in part by the Career Development Fund (CDF) of Agency for Science, Technology and Research (A*STAR) under Grant C233312028. The associate editor coordinating the review of this article and approving it for publication was Prof. Yanjiao Chen. *(Corresponding authors: Shanmin Pang; Xiaojun Jia.)*

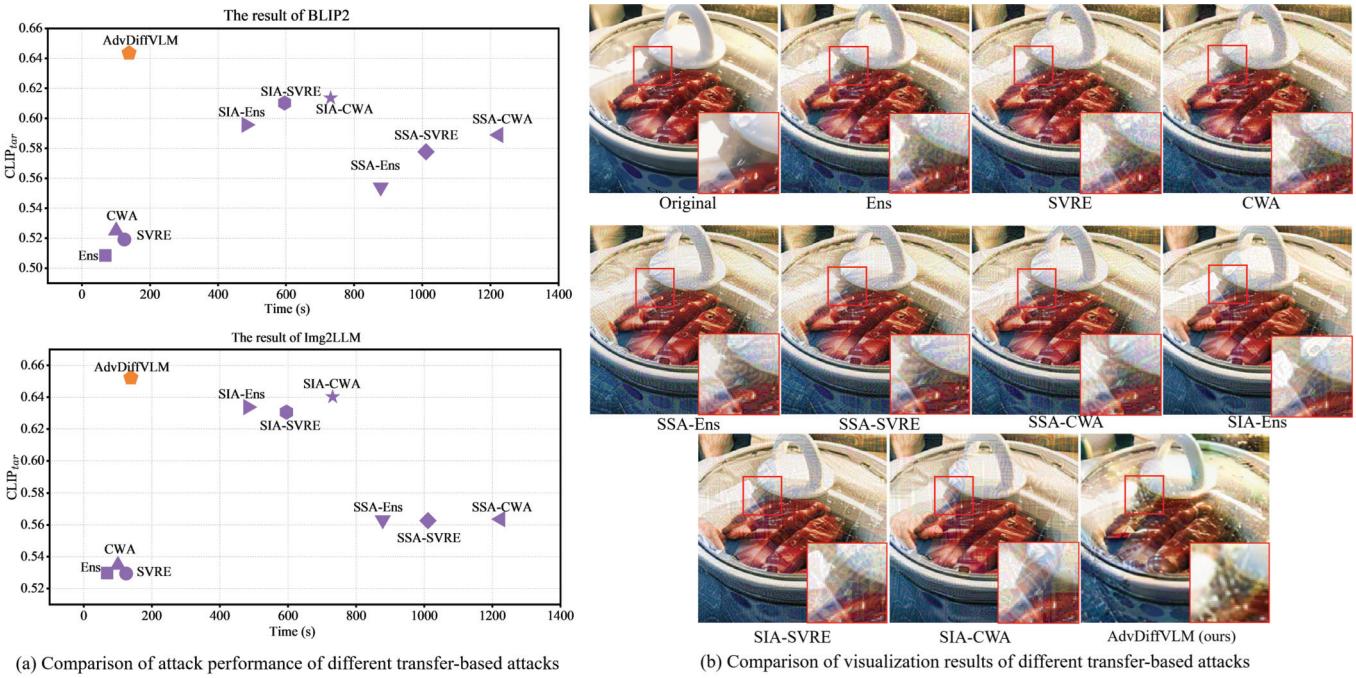
Qi Guo is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China. He was with the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: gq19990314@stu.xjtu.edu.cn).

Shanmin Pang is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: pangsm@xjtu.edu.cn).

Xiaojun Jia and Yang Liu are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: jiaxiaojunqaq@gmail.com; yangliu@ntu.edu.sg).

Qing Guo is with the Institute of High Performance Computing (IHPC) and the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: tsingqguo@ieee.org).

Digital Object Identifier 10.1109/TIFS.2024.3518072



(a) Comparison of attack performance of different transfer-based attacks

(b) Comparison of visualization results of different transfer-based attacks

Fig. 1. Comparison of different transfer-based attacks and our method on VLMs. (a) Comparison of attack performance. We select BLIP2 [2] and Img2LLM [35] as the representation models of VLMs. We select existing transfer-based attacks in conjunction with AttackVLM [16] as comparison methods, including Ens [36], SVRE [27], CWA [26], SSA [37] and SIA [28]. We report the CLIP_{tar} score, which is the similarity between the response generated by the input images. (b) Comparison of image quality. We enlarge the local area of the adversarial examples to enhance visual effects. It is evident that adversarial examples generated by transfer-based attacks exhibit notable noise. Our method has better visual effects. Magnify images for improved contrast.

improving the image quality and transferability of adversarial examples. For example, AdvDiffuser [32] incorporates PGD [34] into the reverse process of the diffusion model to generate targeted adversarial examples with better transferability against classification models. However, applying PGD to the latent image in the reverse process is not suitable for the more difficult task of attacking VLMs. At the same time, performing PGD on each step of the reverse process incurs high costs.

In this paper, we propose AdvDiffVLM, an efficient framework that leverages diffusion models to generate natural, unrestricted, and targeted adversarial examples through score matching. Score matching, initially proposed by Hyvärinen and Dayan [38], is a computationally simple probability density estimation method. It was later introduced into the field of image generation by Song and Ermon [39], demonstrating its ability to guide image generation toward specific target semantics by modifying the score function. Furthermore, Song et al. [40] combined score matching with a diffusion model, significantly enhancing image quality. Inspired by these developments, we investigate the use of score matching to effectively and efficiently attack VLMs, aiming to embed richer adversarial target semantics compared to existing methods like AdvDiffuser [32]. Specifically, we derive a score generation theory tailored for VLM attacks and propose the AEGE based on this theoretical foundation. Furthermore, to improve the naturalness of the outputs, we propose the GMGC module, which effectively distributes adversarial target semantics across the examples. This prevents the concentration of adversarial features in specific regions, thereby improving overall image quality. In addition, we embed more target semantics into adversarial examples through multiple iterations, further enhancing the visual quality of the generated outputs. As demonstrated in Figure 1, AdvDiffVLM outperforms existing attack methods by generating targeted adversarial examples more efficiently while achieving superior transferability. Moreover, the generated adversarial examples exhibit enhanced naturalness, establishing AdvDiffVLM as a more effective tool for evaluating the adversarial robustness of VLMs.

We summarize our contributions as follows:

- We explore existing adversarial attack techniques against VLMs and conduct research on more realistic and challenging scenarios, specifically focusing on targeted and transferable attacks. Furthermore, we propose the AdvDiffVLM framework to efficiently generate targeted and transferable adversarial examples for VLMs.
- We present a score calculation method that embeds adversarial target semantics into the diffusion model, supported by theoretical analysis. Additionally, we propose an adaptive ensemble method to better estimate the score. Building on this theoretical foundation and adaptive ensemble method, we propose the AEGE that combines the generated gradient with score matching, embedding adversarial target semantics naturally in the inverse generation process of the diffusion model.
- We apply an innovative use of GradCAM and propose the GCMG. In contrast to traditional applications, our method allows modifications to be made across the entire image while minimizing alterations to key areas, thus balancing attack capability with image quality.

- Extensive experiments show that our method generates targeted adversarial examples faster than SOTA adversarial attack methods in attacking VLMs, and the generated adversarial examples exhibit better transferability. In addition, our research identifies vulnerabilities in both open-source and commercial VLMs, offering insights toward developing more robust and trustworthy VLMs.

II. Related Work

A. Visual-Language Models (VLMs)

Large language models (LLMs) [41], [42], [43] have demonstrated great success in a variety of language-related tasks. The knowledge contained within these powerful LLMs has aided the development of VLMs. There are several strategies and models for bridging the gap between text and visual modalities [44], [45]. Some studies [2], [46] extract visual information from learned queries and combine it with LLMs to enhance image-based text generation. Models like LLaVA [3] and MiniGPT-4 [47] learn simple projection layers to align visual encoder features with LLM text embeddings. Some works [5] train VLMs from scratch, which promotes better alignment of visual and textual modalities. In this paper, we focus on the adversarial robustness of these VLMs, with the goal of discovering security vulnerabilities and encouraging the development of more robust and trustworthy VLMs.

B. Adversarial Attacks in VLMs

Adversarial attacks are classified as white-box or black-box attacks based on adversary knowledge, as well as targeted or untargeted attacks based on adversary objectives [48], [49], [50]. Studies have investigated the robustness of VLMs, focusing on adversarial challenges in visual question answering [51] and image captioning [19]. However, most studies focus on traditional CNN-RNN-based models, which make assumptions about white-box access or untargeted goals, limiting their applicability in real-world scenarios. Recently, AttackVLM [16] implemented both transfer-based and query-based attacks on large open-source VLMs with black-box access and targeted goals. Nonetheless, this method is time-consuming due to its reliance on numerous VLM queries. In addition, [52] studied the adversarial robustness of VLMs using ensemble transfer-based attacks, assuming untargeted goals. In this paper, we investigate the adversarial robustness of VLMs against targeted transfer-based attacks. Initially, we evaluate VLM's robustness against current SOTA transfer-based attacks in conjunction with AttackVLM. We then examine the limitations of current methods and implement targeted improvements, culminating in the proposal of AdvDiffVLM.

Our method is most closely related to AttackVLM, as both aim to conduct adversarial attacks on VLMs. However, there are two notable differences. First, while AttackVLM generates adversarial examples by estimating gradients through black-box model outputs, our approach leverages the transferability of adversarial examples to effectively attack multiple VLMs. This makes our method more versatile for attacking diverse VLMs. Furthermore, AttackVLM relies on extensive black-box queries to estimate gradients, making its generation process time-intensive. In contrast, our method utilizes the diffusion model's generation process, enabling faster creation of adversarial examples.

Second, in terms of methodology, AttackVLM combines gradient-based attacks with black-box query techniques, using the gradient-based component to initialize black-box queries effectively. In contrast, our approach adopts an unconstrained adversarial example generation framework grounded in generative models. By integrating adaptive gradient estimation with score matching, we embed the adversarial example generation process directly within the diffusion model's workflow. To further enhance the quality of adversarial examples, we incorporate a GradCAM-guided masking technique, which refines the generated outputs.

C. Unrestricted Adversarial Examples

Researchers are increasingly interested in unrestricted adversarial examples, as the l_p norm distance fails to capture human perception [30], [31], [32], [33], [53], [54]. Some approaches use generative methods to create unrestricted adversarial examples. For example, [30] and [31] modify the latent representation of GANs to produce unrestricted adversarial examples. However, due to the limited interpretability of GANs, the generated adversarial examples are of poor quality. Diffusion models [55] are SOTA generative models based on likelihood and theoretical foundations, sampling data distribution with high fidelity and diversity. AdvDiffuser [32] incorporates the PGD [34] method into the diffusion model's reverse process, resulting in high-quality adversarial examples without restrictions. In this study, we explore using the diffusion model for generating unrestricted adversarial examples, focusing on modifying the score in the diffusion model's reverse process rather than adding noise to the latent image. We discuss the differences between our method and AdvDiffuser in Section IV-D.

III. Preliminaries

A. Diffusion Models

In this work, we use diffusion models [4], [55], [56] to generate unrestricted and targeted adversarial examples. In a nutshell, diffusion models learn a denoising process from $x_T \sim \mathcal{N}(x_T; 0, \mathbf{I})$ to recover the data $x_0 \sim q(x_0)$ with a Markov chain and mainly include two processes: forward process and reverse process. Forward process defines a fixed Markov chain. Noise is gradually added to the image x_0 over T time steps, producing a series of noisy images $\{x_1, x_2, \cdots, x_T\}$. Specifically, noise is added by $q(x_t \mid x_0) := \sqrt{\bar{\alpha}_t}x_0 + \epsilon_1\sqrt{1-\bar{\alpha}_t}, \epsilon_1 \sim \mathcal{N}(0,1)$, where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$ and β_t is a fixed variance to control the step sizes of the noise. The purpose of the reverse process is to gradually denoise from x_T to obtain a series of $\{\tilde{x}_{T-1}, \tilde{x}_{T-2}, \cdots, \tilde{x}_1\}$, and finally restore x_0. It learns the denoising process through a denoising model ε_θ, and the training objective is $\mathcal{L}_{simple} := E_{t\sim[1,T],\epsilon_1\sim\mathcal{N}(0,\mathbf{I})}\|\varepsilon_\theta(x_t, t) - \epsilon_1\|^2$.

B. Problem Settings

Then we give the problem setting of this paper. We denote the victim VLM model as f_ξ, and aim to induce f_ξ to output

the target response. This can be formalized as

$$\begin{aligned} &\max \ CS(g_\psi(f_\xi(\boldsymbol{x}_{\text{adv}}; \boldsymbol{c}_{\text{in}})), g_\psi(\boldsymbol{c}_{\text{tar}})) \\ &\text{s.t. } D(\boldsymbol{x}, \boldsymbol{x}_{\text{adv}}) \leq \epsilon \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ represents the original image, $\boldsymbol{x}_{\text{adv}}$ and $\boldsymbol{c}_{\text{tar}}$ respectively refer to adversarial example and adversarial target text, and $g_\psi(\cdot)$ denotes the CLIP text encoder. Moreover, $D(\boldsymbol{x}, \boldsymbol{x}_{\text{adv}}) \leq \epsilon$ places a bound on a distance metric, and $CS(\cdot,\cdot)$ refers to the cosine similarity metric. Finally, $\boldsymbol{c}_{\text{in}}$ denotes the input text.

Since f_ξ is a black-box model, we generate adversarial examples on the surrogate model ϕ_ψ and transfer them to f_ξ. In addition, inspired by [16], matching image-image features can lead to better results, we define the problem as,

$$\begin{aligned} &\max \ CS(\phi_\psi(\boldsymbol{x}_{\text{adv}}), \phi_\psi(\boldsymbol{x}_{\text{tar}})) \\ &\text{s.t. } D(\boldsymbol{x}, \boldsymbol{x}_{\text{adv}}) \leq \epsilon \end{aligned} \tag{2}$$

where $\boldsymbol{x}_{\text{tar}}$ represents the target image generated by $\boldsymbol{c}_{\text{tar}}$. We use stable diffusion [4] to implement the text-to-image generation. ϕ_ψ refers to CLIP image encoder. Our study is the most realistic and challenging attack scenarios, i.e., targeted and transfer scenarios.

C. Rethinking Transfer-Based Attacks

Transfer-based attacks can effectively solve Eq. 2. In this context, we assess the robustness of VLMs against current SOTA transfer-based attacks, in conjunction with AttackVLM. Specifically, we consider ensemble methods including Ens [36], SVRE [27] and CWA [26], data augmentation methods including SSA [37] and SIA [28], and combinations of both.[1] We primarily employ the simple ensemble version of data augmentation attacks, as relying on a single surrogate model tends to yield poor performance. For hyperparameter settings, in all attacks, the value range of adversarial example pixels is [0,1]. We set the perturbation budget as $\epsilon = 16/255$ under the ℓ_∞ norm. The number of iterations N_I for all attacks is set to 300. In addition, we use the MI-FGSM [36] method and set $\mu = 1$. Furthermore, for SVRE, internal step size $\beta_{inter} = 16/255/10$ and internal decay factor $\mu_2 = 1$. For CWA, CSE step size $\beta_{cse} = 16/255/15$ and inner gradient ascent step $r = 250$. For SSA, tuning factor $\rho = 0.5$, the number of spectrum transformations $N_t = 20$ and standard deviation $\sigma_s = 16/255$. For SSA, the number of the block $s = 3$ and the number of image for gradient calculation $N_{grad} = 20$.

The outcomes of these transfer-based attacks on VLMs are depicted in Figure 1. As illustrated, current transfer-based attacks face challenges such as slow adversarial example generation, noticeable noise within these examples, and limited transferability. The limitations of existing transfer-based attacks on VLMs are analyzed as follows: First, existing

[1]https://github.com/xiaosen-wang/SIT for SIA and https://github.com/thu-ml/Attack-Bard for others. we adapt the targeted attacks to untargeted attacks to better align with our scenario. Additionally, to ensure a fair comparison with our model, we modify the surrogate models by ensembling various CLIP visual encoders, including ResNet-50, ResNet-101, ViT-B/16, and ViT-B/32. Finally, we modify the loss function to cosine similarity loss to make it consistent with AttackVLM.

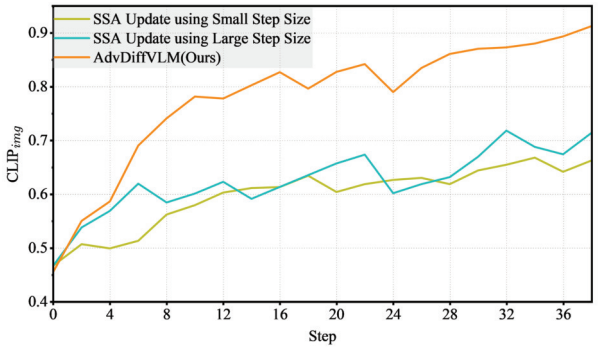


Fig. 2. The CLIP$_{img}$ score varies with the step sizes. Here, CLIP$_{img}$ is the similarity between the adversarial examples and the adversarial target images, which is calculated by the visual encoder of CLIP ViT-B/32. We choose SSA [37] as the representative of transfer-based attacks.

SOTA transfer-based attacks only access the original image during the optimization of Eq. 2. Consequently, they employ small steps and strategies like data augmentation to tentatively approach the optimal solution, necessitating numerous iterations and resulting in high attack costs. As shown in Figure 2, using a larger step size results in pronounced fluctuations during the optimization process. This issue may be mitigated by leveraging score, which provides insights into the data distribution. By offering score guidance towards solving Eq. 2, faster convergence is expected. Therefore score information can be considered in the design of new improved attack method. Second, existing transfer-based attacks introduce unnatural adversarial noises with limited transferability. Unrestricted adversarial examples can introduce more natural adversarial targeted semantics, increasing transferability. These imply that new transfer-based targeted attacks can consider unrestricted adversarial attacks.

IV. Methodology

The main framework of AdvDiffVLM is illustrated in Figure 3. First, we input the original image, $\boldsymbol{x}$, followed by the forward process $x_{t^*} \sim q(x_{t^*} \mid x_0)$ to obtain a noisy image, x_{t^*}. Subsequently, we apply the reverse denoising process. At each step, we first obtain the mask m from the GCMG module. This mask is then used to fuse the original noisy image, x_t, with the adversarial noisy image, $\tilde{x}_t$. Next, we apply the AEGE module to obtain gradient information, which we then use to calculate the score. Finally, we derive the next step of the noisy image based on the score matching method.

In the following, we begin by presenting the motivation behind our approach and a theoretical analysis of our method. This is followed by a comprehensive explanation of the proposed AdvDiffVLM framework. Lastly, we highlight the key distinctions between our method and AdvDiffuser, emphasizing their unique features and contributions.

A. Motivation and Theoretical Analysis

With the growing deployment of VLMs in critical applications such as autonomous driving and content moderation, ensuring their robustness against adversarial attacks has become essential for maintaining system security and reliability. While existing approaches have made notable progress

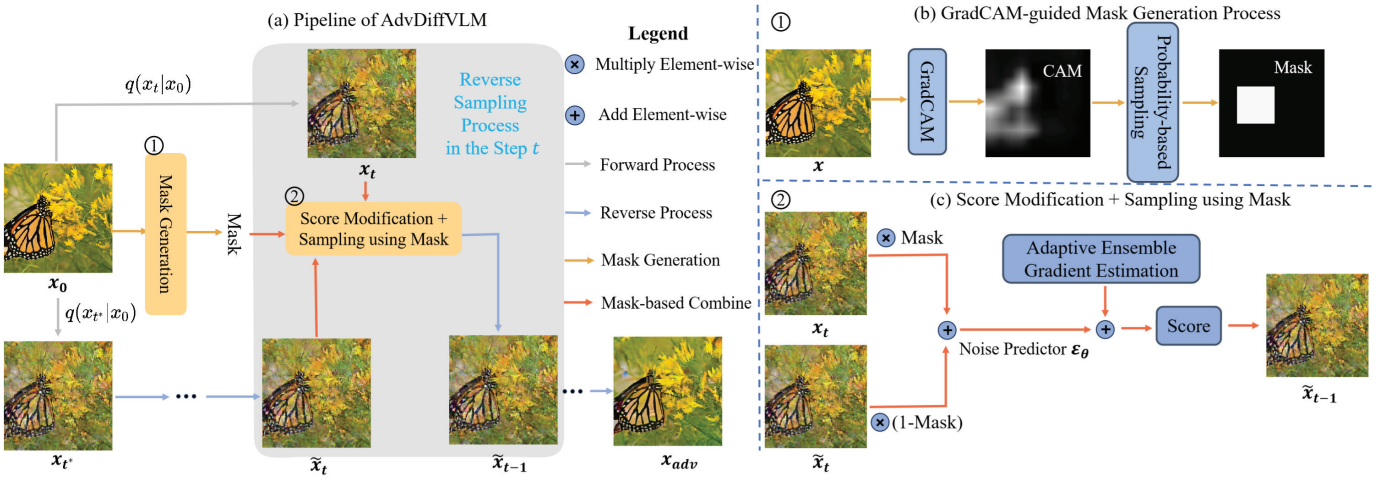


Fig. 3. The main framework of the AdvDiffVLM for efficiently generating transferable unrestricted adversarial examples. AdvDiffVLM mainly includes two components: AEGE and GCMG. Details are respectively described in Secs. IV-B and IV-C. Please refer to Section IV for specific symbol meanings.

in evaluating VLM robustness, they still face fundamental limitations in terms of efficiency and effectiveness. High computational overhead and limited transferability hinder the ability to comprehensively assess robustness across diverse VLMs. This challenge motivates our work to develop an efficient, high-quality, and transferable method for generating adversarial examples, thereby facilitating a more effective evaluation of VLM robustness. We achieve this by leveraging insights from diffusion models and score matching techniques.

Specifically, we focus on modeling adversarial attacks from a generative perspective, considering how to utilize the data distribution (score) of the generative model to produce natural, unrestricted and targeted adversarial examples. Additionally, as indicated in [57], learning to model the score function is equivalent to modeling the negative of the noise, suggesting that score matching and denoising are equivalent processes. Thus, our method derives from integrating diffusion models and score matching, positioning it as a novel approach for generating high-quality, unrestricted, transferable and targeted adversarial examples.

Formally, we want to obtain distribution meeting the condition that the adversarial example has target semantic information during the reverse generation process

$$p(x_{t-1}|x_t, f_\xi(\boldsymbol{x}_{\text{adv}}; \boldsymbol{c}_{\text{in}}) = \boldsymbol{c}_{\text{tar}}) \tag{3}$$

where x_t represents the latent image of the diffusion model. Next, we start from the perspective of score matching [40] and consider the score $\nabla \log p(x_{t-1}|x_t, \boldsymbol{c}_{\text{tar}})$ of this distribution, where ∇ is the abbreviation for ∇_{x_t}. According to Bayes theorem,

$$\begin{aligned}
&\nabla \log p\left(x_{t-1} \mid x_t, \boldsymbol{c}_{\text{tar}}\right) \\
&= \nabla \log \left(\frac{p\left(\boldsymbol{c}_{\text{tar}} \mid x_{t-1}, x_t\right) \cdot p\left(x_{t-1} \mid x_t\right)}{p\left(\boldsymbol{c}_{\text{tar}} \mid x_t\right)} \right) \\
&= \nabla \log p\left(\boldsymbol{c}_{\text{tar}} \mid x_{t-1}, x_t\right) + \nabla \log p\left(x_{t-1} \mid x_t\right) \\
&\quad - \nabla \log p\left(\boldsymbol{c}_{\text{tar}} \mid x_t\right) \\
&= \nabla \log p\left(\boldsymbol{c}_{\text{tar}} \mid x_{t-1}\right) + \nabla \log p\left(x_t \mid x_{t-1}, \boldsymbol{c}_{\text{tar}}\right) \\
&\quad - \nabla \log p\left(x_t \mid x_{t-1}\right) + \nabla \log p\left(x_{t-1} \mid x_t\right) \\
&\quad - \nabla \log p\left(\boldsymbol{c}_{\text{tar}} \mid x_t\right) \\
&= \nabla \log p\left(x_t \mid x_{t-1}, \boldsymbol{c}_{\text{tar}}\right) - \nabla \log p\left(x_t \mid x_{t-1}\right) \\
&\quad + \nabla \log p\left(x_{t-1} \mid x_t\right) - \nabla \log p\left(\boldsymbol{c}_{\text{tar}} \mid x_t\right)
\end{aligned} \tag{4}$$

$p(x_t \mid x_{t-1}, c_{tar})$ and $p(x_t \mid x_{t-1})$ respectively denote the add noise process with target text and the add noise process devoid of target semantics. From an intuitive standpoint, whether target text is present or not, the forward noise addition process follows a Gaussian distribution, and the added noise remains consistent, indicating that the gradient solely depends on x_t. The difference between x_t without target text and x_t with target text is minimal, as constraints are employed to ensure minimal variation of the adversarial example from the original image. Therefore, $\nabla \log p(x_t \mid x_{t-1}, \boldsymbol{c}_{\text{tar}})$ and $\nabla \log p(x_t \mid x_{t-1})$ are approximately equal. So the final score is $\nabla \log p(x_{t-1} \mid x_t) - \nabla \log p(\boldsymbol{c}_{\text{tar}} \mid x_t)$.

Because score matching and denoising are equivalent processes, that is, $\nabla \log p(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(x_t)$. Therefore we can get score $(\nabla \log p(x_{t-1}|x_t, \boldsymbol{c}_{\text{tar}}))$,

$$\text{score} = -\left(\frac{\varepsilon_\theta\left(x_t\right)}{\sqrt{1-\bar{\alpha}_t}} + \nabla \log p_{f_\xi}\left(c_{\text{tar}} \mid x_t\right) \right) \tag{5}$$

where ε_θ is denoising model, and $\bar{\alpha}_t$ is the hyperparameter.

Eq. 5 demonstrates that the score of $p(x_{t-1}|x_t, \boldsymbol{c}_{\text{tar}})$ can be derived by incorporating gradient information into the inverse process of the diffusion model. Consequently, adversarial semantics can be incrementally embedded into adversarial examples based on the principle of score matching.

B. Adaptive Ensemble Gradient Estimation (AEGE)

Since f_ξ is a black-box model and cannot obtain gradient information, we use surrogate model to estimate $\nabla \log p_{f_\xi}(c_{\text{tar}} \mid x_t)$. As a scalable method for learning joint representations between text and images, CLIP [58] can leverage pre-trained CLIP models to establish a bridge between images and text. Therefore we use the CLIP model as the surrogate model to estimate the gradient.

Specifically, we first add noise to the original image $\boldsymbol{x}$ by t^* steps through the forward process $q(x_{t^*}|x_0)$ to obtain x_{t^*}, where $x_0 = \boldsymbol{x}$. Then, at each step of reverse process, we change score:

$$\text{score} = -\left(\frac{1}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\tilde{x}_t) + s\nabla_{\tilde{x}_t}(CS(\phi_\psi(\tilde{x}_t), \phi_\psi(\boldsymbol{x}_{\text{tar}})))\right) \quad (6)$$

where s is the adversarial gradient scale used to control the degree of score change and $\tilde{x}_t$ is the latent image in the reverse process.

We find that gradient estimation using only a single surrogate model is inaccurate. Therefore, we consider using a set of surrogate models $\{\phi_\psi^i\}_{i=1}^{N_m}$ to better estimate the gradient. Specifically, we make the following improvements to Eq. 6:

$$\text{score} = -\left(\frac{\varepsilon_\theta(\tilde{x}_t)}{\sqrt{1-\bar{\alpha}_t}} + s\nabla_{\tilde{x}_t}\left(w_i\sum_{i=1}^{N_m} CS(\phi_\psi^i(\tilde{x}_t), \phi_\psi^i(\boldsymbol{x}_{\text{tar}}))\right)\right) \quad (7)$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_{N_m})$ represents the weight of cosine loss of different models.

Since different images have different sensitivities to surrogate models, only using simple ensemble cannot obtain optimal solution. Inspired by [59], we propose a new adaptive ensemble method, and obtain $\mathbf{w}$ in Eq. 7 in the following way:

$$w_i(t) = \frac{\sum_{j=1}^{N_m}\exp(\tau\mathcal{L}_j(t+1)/\mathcal{L}_j(t+2))}{N_m\exp(\tau\mathcal{L}_i(t+1)/\mathcal{L}_i(t+2))} \quad (8)$$

where τ refers to the temperature. A larger τ makes all weights close to 1. $\mathcal{L}_i = CS(\phi_\psi^i(\tilde{x}_t), \phi_\psi^i(\boldsymbol{x}_{\text{tar}}))$. We initialize $\{w_i(t^*)\}_{i=1}^{N_m}$ and $\{w_i(t^*-1)\}_{i=1}^{N_m}$ to 1. Through Eq. 8, we reduce the weight of surrogate models with fast-changing losses to ensure that gradient estimations of different surrogate models are updated simultaneously.

Figure 4 presents the detailed visualization results of AEGE. Specifically, both the target image and the current adversarial example are independently input into N_m visual encoders to obtain N_m cosine similarity values. Here, $\hat{x}_t$ represents the current adversarial example derived from the mask, as described in Sec. IV-C. The gradient, denoted as grad, is then computed by applying weights $\mathbf{w}$ to these values. Finally, this weighted result is combined with the noise prediction value to obtain the final score.

Finally, we set the perturbation threshold δ, and then clip the adversarial gradient to ensure the naturalness of the synthesized adversarial examples.

C. GradCAM-Guided Mask Generation (GCMG)

We detailed AEGE earlier but observed that relying solely on it generates obvious adversarial features in specific areas, leading to poor visual effects. To balance visual quality and attack capabilities, we propose GCMG, which uses a mask to combine the forward noisy image x_t and the generated image $\tilde{x}_t$. This combination distributes adversarial semantics across the image, enhancing the natural visual quality of adversarial examples.

First, we utilize GradCAM [60] to derive the class activation map CAM of $\boldsymbol{x}$ with respect to ground-truth label $\boldsymbol{y}$. CAM assists in identifying important and non-important areas in the image. Subsequently, we clip the CAM values to the range $[0.3, 0.7]$ and normalize them to obtain the probability matrix P. We sample according to the P to obtain the coordinate (x, y), and then set the k × k area around (x, y) to be 1 and remain other areas to obtain mask m. Here, m has the same shape as $\tilde{x}_t$. This approach disperses more adversarial features in non-important areas and less in important areas of adversarial examples, improving the natural visual effect of adversarial examples.

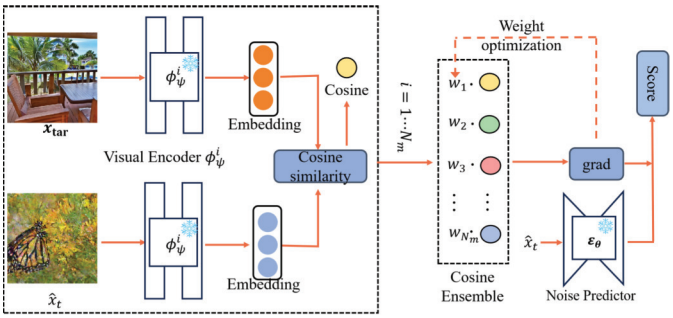


Fig. 4. The pipeline of the AEGE.

At each step t, we combine x_t and $\tilde{x}_t$ as following:

$$\hat{x}_t = \text{m} \odot x_t + (1 - \text{m}) \odot \tilde{x}_t \quad (9)$$

where m denotes the final mask, $\odot$ refers to Hadamard Product. Afterwards, we can obtain new score by integrating $\varepsilon_\theta(\hat{x}_t)$ with the estimated gradient and then use $\tilde{x}_{t-1} = -\sqrt{1-\bar{\alpha}_t} \times \text{score}$ for sampling.

Finally, we take the generated adversarial example as x_0, and iterate N times to embed more target semantics into it. We provide a complete algorithmic overview of AdvDiffVLM in Algorithm 1.

D. Differences From AdvDiffuser

Both our method and AdvDiffuser [32] produce unrestricted adversarial examples using the diffusion model. Here, we discuss the distinctions between them, highlighting our contributions.

1) Tasks of Varying Difficulty Levels: AdvDiffuser is oriented towards classification models, while our research targets the more intricate Vision-Language Models (VLMs). Initially, within the realm of classification tasks, each image is associated with a singular label. Conversely, in the image-to-text tasks, images may be linked to numerous text descriptions. When faced with an attack targeting a singular description, VLMs have the capability to generate an alternate description, thereby neutralizing the attack's effectiveness. As a result, our task presents a greater challenge.

2) Different Theoretical Foundations and Implementation Methods: AdvDiffuser utilizes PGD [34] to introduce high-frequency adversarial noise, while our method employs score matching to incorporate target semantics. These theoretical distinctions lead to differences in implementation: Without considering the mask, AdvDiffuser operates on the latent image $\tilde{x}_t$, adding adversarial noise directly to $\tilde{x}_t$. In contrast,

Algorithm 1 The Overall Algorithm of AdvDiffVLM

Input: Original image $\boldsymbol{x}$, N_m surrogate models ϕ_θ^i, adversarial guidance scale s, reverse generation process timestep t^*, mask area size k, perturbation threshold δ, temperature τ, adversarial target image $\boldsymbol{x}_{\text{tar}}$, Number of iterations N.
Output: adversarial example $\boldsymbol{x}_{\text{adv}}$.

Initialize $\{w_i\}_{i=1}^{N_m} = 1$, CAM, $x_0 = \boldsymbol{x}$;
Sample $x_{t^*} \sim q(x_{t^*}|x_0)$, let $\tilde{x}_{t^*} = \bar{x}_{t^*} = x_{t^*}$;
for $n \leftarrow 1, \cdots, N$ **do**
  for $t \leftarrow t^*, \cdots, 1$ **do**
    Get mask m according to CAM ; // Mask generation;
    $x_t \sim q(x_t|x_0)$;
    $\hat{x}_t = \mathrm{m} \odot x_t + (1 - \mathrm{m}) \odot \tilde{x}_t$; // Mask-based combination;
    $w_i = \frac{\sum_{j=1}^{N_m} \exp(\tau \mathcal{L}_j(t+1)/\mathcal{L}_j(t+2))}{N_m \exp(\tau \mathcal{L}_i(t+1)/\mathcal{L}_i(t+2))}$; // Weight optimization;
    $g = \nabla_{\hat{x}_t}(w_i \sum_{i=1}^{N_m} CS(\phi_\psi^i(\hat{x}_t), \phi_\psi^i(\boldsymbol{x}_{\text{tar}})))$; // Ensemble gradient estimation;
    $g = clip(g, -\delta, \delta)$;
    score $= -(\varepsilon_\theta(\hat{x}_t)/\sqrt{1-\bar{\alpha}_t} + s \cdot g)$; // Score calculation;
    $\tilde{x}_{t-1} = (\hat{x}_t + (1-\alpha_t) \cdot \text{score})/\sqrt{\alpha_t}$; // Score matching;
  end
end
Return $\boldsymbol{x}_{\text{adv}} = \tilde{x}_0$

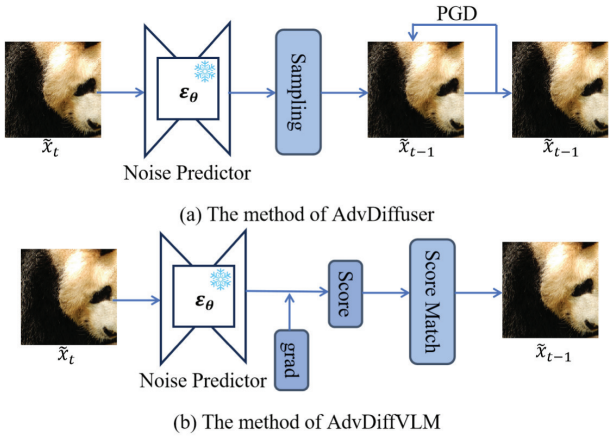


Fig. 5. Different theoretical foundations and implementation methods between AdvDiffuser and our method. Where "Sampling" refers to $\tilde{x}_{t-1} = (\tilde{x}_t - (1-\alpha_t) \cdot \varepsilon_\theta(\tilde{x}_t, t)/\sqrt{1-\bar{\alpha}_t})/\sqrt{\alpha_t}$ and "Score Match" refers to $\tilde{x}_{t-1} = (\tilde{x}_t + (1-\alpha_t) \cdot \text{score})/\sqrt{\alpha_t}$.

our method modifies the predicted noise $\varepsilon_\theta(\tilde{x}_t)$, obtaining a score that encodes adversarial target semantics. For clarity, we include a visual comparison, as shown in Figure 5. Furthermore, our approach obviates the need for initiating with Gaussian noise, initially introducing noise to $\boldsymbol{x}$ through t^* steps, followed by the application of adversarial gradient to modify score, thereby facilitating more efficient generation of adversarial examples.

3) Distinct Schemes of GradCAM Utilization: The GradCAM mask utilized by AdvDiffuser leads to restricted modification of crucial image areas, rendering it inadequate for image-based attacks. Addressing this issue, we introduc the GCMG. Contrary to utilizing GradCAM results directly as a mask, we employ them as a directive to generate the mask further. This not only guarantees a likelihood of modification across all image areas but also secures minimal alteration of significant areas, striking a balance between image quality and attack ability.

TABLE I
THE DETAILS OF VICTIM VLMS, INCLUDE CODE AND CONFIGURATION

Models	Code	Version
Unidifusser	https://github.com/thu-ml/unidiffuser	/
BLIP2	https://github.com/salesforce/LAVIS	(blip2_opt, pretrain_opt2.7b)
MiniGPT-4	https://github.com/Vision-CAIR/MiniGPT-4	(Vicuna 7B)
LLaVA	https://github.com/haotian-liu/LLaVA	(Vicuna, llava-v1.5-7b)
Img2LLM	https://github.com/salesforce/LAVIS	(img2prompt_vqa, base)

V. EXPERIMENTS

A. Experimental Setup

1) Datasets and Victim VLMs: Following [52], we use NeurIPS'17 adversarial competition dataset, compatible with ImageNet, for all the experiments. In addition, we select 1,000 text descriptions from the captions of the MS-COCO dataset as our adversarial target texts and then use Stable Diffusion [4] to generate 1,000 adversarial targeted images. For the victim VLMs, SOTA open-source models are evaluated, including Unidiffuser [5], BLIP2 [2], MiniGPT-4 [47], LLaVA [3] and Img2LLM [35]. The details are shown in Table I. Among them, Unidiffuser is a gray-box model, and the others are black-box models.

2) Baselines: We compare with AdvDiffuser [32] and other SOTA transfer-based attackers described in Section III-C. Since AdvDiffuser is used for classification models, we use cosine similarity loss instead of classification loss for adversarial attacks on VLMs. For a fair comparison, we implement the ensemble version of AdvDiffuser, including simple ensemble and adaptive ensemble, which are denoted as AdvDiffuser$_{ens}$, AdvDiffuser$_{adaptive}$ respectively. For hyperparameters (in AdvDiffuser), we choose $T = 200, \sigma = 0.4, I = 25$.

3) Evaluation Metrics: Following [16], we adopt CLIP score between the generated responses from victim models and predefined targeted texts, as computed by ViT-B/32 text encoder, refered as CLIP$_{tar}$. We adopt the method of calculating the attack success rate (ASR) in [52], positing that an attack is deemed successful solely if the image description includes the target semantic main object. In order to measure the quality of adversarial examples and the perceptibility of applied perturbations, we use four evaluation metrics: SSIM [61], FID [62], LPIPS [63] and BRISQUE [64].

4) Implementation Details: Since our adversarial diffusion sampling does not require additional training to the original diffusion model, we use the pre-trained diffusion model in our experiment. We adapt LDM [4] with DDIM sampler [56] (using T =200 diffusion steps) and select the version trained on ImageNet. Additionally, we use four versions of CLIP [58], namely ResNet-50, ResNet-101, ViT-B/16, and ViT-B/32, each trained on 400 million unpublished image-text pairs. For other

TABLE II

COMPARISON WITH EXISTING SOTA ATTACK METHODS, WHERE THE BEST RESULT IS BOLDED. WE ALSO REPORT THE STANDARD DEVIATION OF THE RESULTS. NOTE THAT WE USE FOUR VERSIONS OF THE CLIP VISUAL ENCODER, INCLUDING RESNET50, RESNET101, VIT-B/16 AND VIT-B/32, AS SURROGATE MODELS. SINCE UNIDIFFUSER USES VIT-B/32 AS THE VISUAL ENCODER, IT IS A GRAY BOX SCENARIO, WHICH WE INDICATE WITH *. IN ADDITION, WE PROVIDE THE AVERAGE TIME (S) FOR EACH STRATEGY TO CRAFTA SINGLE x_{adv}. THE SHADED PARTS REPRESENT OUR PROPOSED METHOD

	Unidiffuser*		BLIP2		MiniGPT-4		LLaVA		Img2LLM		Time(s)
	$CLIP_{tar}$ ↑	ASR ↑	$CLIP_{tar}$ ↑	ASR ↑	$CLIP_{tar}$ ↑	ASR ↑	$CLIP_{tar}$ ↑	ASR ↑	$CLIP_{tar}$ ↑	ASR ↑	
Original	$0.4770_{\pm 0.0017}$	$0.0\%_{\pm 0.00\%}$	$0.4931_{\pm 0.0027}$	$0.0\%_{\pm 0.00\%}$	$0.4902_{\pm 0.0030}$	$0.0\%_{\pm 0.00\%}$	$0.5190_{\pm 0.0052}$	$0.0\%_{\pm 0.00\%}$	$0.5288_{\pm 0.0039}$	$0.0\%_{\pm 0.00\%}$	/
Ens	$0.7353_{\pm 0.0012}$	$99.1\%_{\pm 0.14\%}$	$0.5085_{\pm 0.0019}$	$0.9\%_{\pm 0.11\%}$	$0.4980_{\pm 0.0035}$	$1.8\%_{\pm 0.14\%}$	$0.5366_{\pm 0.0052}$	$3.5\%_{\pm 0.20\%}$	$0.5297_{\pm 0.0027}$	$4.5\%_{\pm 0.22\%}$	69
SVRE	$0.7231_{\pm 0.0020}$	$100.0\%_{\pm 0.00\%}$	$0.5190_{\pm 0.0023}$	$2.4\%_{\pm 0.15\%}$	$0.5107_{\pm 0.0029}$	$2.2\%_{\pm 0.12\%}$	$0.5385_{\pm 0.0049}$	$4.6\%_{\pm 0.18\%}$	$0.5292_{\pm 0.0035}$	$3.8\%_{\pm 0.17\%}$	125
CWA	$0.7568_{\pm 0.0016}$	$100.0\%_{\pm 0.00\%}$	$0.5249_{\pm 0.0022}$	$5.2\%_{\pm 0.27\%}$	$0.5211_{\pm 0.0033}$	$3.8\%_{\pm 0.20\%}$	$0.5493_{\pm 0.0057}$	$7.1\%_{\pm 0.26\%}$	$0.5346_{\pm 0.0042}$	$5.4\%_{\pm 0.04\%}$	101
SSA-Ens	$0.7275_{\pm 0.0031}$	$100.0\%_{\pm 0.00\%}$	$0.5539_{\pm 0.0050}$	$9.2\%_{\pm 0.42\%}$	$0.5175_{\pm 0.0052}$	$10.1\%_{\pm 0.33\%}$	$0.6098_{\pm 0.0056}$	$37.5\%_{\pm 0.49\%}$	$0.5629_{\pm 0.0050}$	$19.6\%_{\pm 0.31\%}$	879
SSA-SVRE	$0.7217_{\pm 0.0039}$	$100.0\%_{\pm 0.00\%}$	$0.5776_{\pm 0.0046}$	$18.7\%_{\pm 0.40\%}$	$0.5395_{\pm 0.0056}$	$16.5\%_{\pm 0.47\%}$	$0.6005_{\pm 0.0063}$	$40.2\%_{\pm 0.57\%}$	$0.5625_{\pm 0.0067}$	$18.4\%_{\pm 0.39\%}$	1012
SSA-CWA	$0.7485_{\pm 0.0024}$	$100.0\%_{\pm 0.00\%}$	$0.5888_{\pm 0.0041}$	$23.3\%_{\pm 0.57\%}$	$0.5407_{\pm 0.0057}$	$20.6\%_{\pm 0.45\%}$	$0.6152_{\pm 0.0070}$	$40.7\%_{\pm 0.52\%}$	$0.5634_{\pm 0.0061}$	$20.4\%_{\pm 0.33\%}$	1225
SIA-Ens	$0.7377_{\pm 0.0058}$	$100.0\%_{\pm 0.00\%}$	$0.5956_{\pm 0.0074}$	$49.6\%_{\pm 1.40\%}$	$0.5605_{\pm 0.0064}$	$40.4\%_{\pm 1.06\%}$	$0.7158_{\pm 0.0085}$	$84.7\%_{\pm 1.80\%}$	$0.6337_{\pm 0.0073}$	$27.0\%_{\pm 1.27\%}$	483
SIA-SVRE	$0.7302_{\pm 0.0066}$	$100.0\%_{\pm 0.00\%}$	$0.6102_{\pm 0.0068}$	$50.1\%_{\pm 0.79\%}$	$0.5782_{\pm 0.0080}$	$46.4\%_{\pm 1.17\%}$	$0.7122_{\pm 0.0079}$	$88.3\%_{\pm 1.73\%}$	$0.6305_{\pm 0.0087}$	$35.4\%_{\pm 1.54\%}$	596
SIA-CWA	$0.7498_{\pm 0.0053}$	$100.0\%_{\pm 0.00\%}$	$0.6135_{\pm 0.0085}$	$51.8\%_{\pm 1.18\%}$	$0.5810_{\pm 0.0064}$	$47.8\%_{\pm 1.24\%}$	$0.7194_{\pm 0.0086}$	$89.5\%_{\pm 2.95\%}$	$0.6401_{\pm 0.0078}$	$40.6\%_{\pm 1.30\%}$	732
$AdvDiffuser_{ens}$	$0.6774_{\pm 0.0037}$	$86.7\%_{\pm 1.93\%}$	$0.5396_{\pm 0.0034}$	$8.6\%_{\pm 0.26\%}$	$0.5371_{\pm 0.0041}$	$8.2\%_{\pm 0.37\%}$	$0.5507_{\pm 0.0071}$	$25.3\%_{\pm 0.37\%}$	$0.5395_{\pm 0.0063}$	$11.5\%_{\pm 0.25\%}$	574
$AdvDiffuser_{adaptive}$	$0.6932_{\pm 0.0029}$	$88.9\%_{\pm 1.75\%}$	$0.5424_{\pm 0.0062}$	$10.4\%_{\pm 0.35\%}$	$0.5391_{\pm 0.0046}$	$9.6\%_{\pm 0.30\%}$	$0.5595_{\pm 0.0064}$	$27.4\%_{\pm 0.41\%}$	$0.5502_{\pm 0.0060}$	$14.8\%_{\pm 0.32\%}$	602
AdvDiffVLM	$0.7502_{\pm 0.0072}$	$100.0\%_{\pm 0.00\%}$	$\mathbf{0.6435}_{\pm 0.0101}$	$\mathbf{66.7\%}_{\pm 1.86\%}$	$\mathbf{0.6145}_{\pm 0.0096}$	$\mathbf{58.6\%}_{\pm 2.07\%}$	$\mathbf{0.7206}_{\pm 0.0113}$	$\mathbf{91.2\%}_{\pm 2.35\%}$	$\mathbf{0.6521}_{\pm 0.0107}$	$\mathbf{43.8\%}_{\pm 1.92\%}$	139

hyperparameters, we use $s = 35, \delta = 0.0025, t^* = 0.2, \mathrm{k} = 8, \tau = 2$ and $N = 10$. All the experiments are conducted on a Tesla A100 GPU with 40GB memory.

B. Main Experiments

In this subsection, we evaluate the effectiveness of our method in targeted and transferable scenarios. Specifically, we first quantitatively compare the transferability of our approach against baseline methods on both open-source and commercial VLMs. Following this, we present qualitative results of our method applied to these VLMs. Lastly, we analyze the model's complexity and demonstrate its practical efficiency.

1) Quantitative Results On Open Source VLMs: To validate the effectiveness of AdvDiffVLM, we quantitatively evaluate the transferability of adversarial examples generated by AdvDiffVLM and baseline methods on various open source VLMs. As shown in Table II, all methods demonstrate favorable attack results in gray box scenarios. In the transfer attack scenario, our method yields the best results. For example, on BLIP2, our method improves $CLIP_{tar}$ and ASR by 0.0200 and 10.9%, respectively, when compared to SIA-CWA. Furthermore, our method generates adversarial examples much faster than baselines. Specifically, when compared to AdvDiffuser, SIA and SSA methods, our method generates adversarial examples 5x to 10x faster. Experimental results show that our method generates adversarial examples with better transferability at a faster rate, demonstrating its superiority. To ensure statistical significance, we repeat each experiment three times and report the standard deviation.

Additionally, it has been observed that AdvDiffuser exhibits suboptimal performance in challenging attack scenarios, particularly against VLMs. This is attributed to its direct application of GradCAM as the mask, which restricts the modifiable area for adversarial examples in demanding tasks, thereby diminishing attack effectiveness. Simultaneously, AdvDiffuser employs high-frequency adversarial noise to alter semantics. This adversarial noise, being inherently fragile, is significantly mitigated during the diffusion model's reverse process, further diminishing its attack potential on complex tasks. These observations validate the advantages of our GCMG and score matching idea.

TABLE III

THE RESULT OF ATTACKING COMMERCIAL VLMS. WE REPORT ASR AND PROVIDE THE AVERAGE TIME (S) FOR EACH STRATEGY TO CRAFT A SINGLE x_{adv}. THE BEST RESULT IS BOLDED

	GPT-4V	Gemini	Copilot	ERNIE Bot	Time(s)
No attack	0%	0%	0%	0%	/
SIA-CWA	35%	12%	25%	50%	732
AdvdiffVLM	**37%**	**17%**	**26%**	**58%**	**139**

2) Quantitative Results On Commercial VLMs: We conduct a quantitative evaluation of commercial VLMs such as OpenAI's GPT-4V,[2] Google's Gemini,[3] Microsoft's Copilot,[4] and Baidu's ERNIE Bot.[5] We choose SIA-CWA to represent baselines and ASR as an evaluation metric. We choose 100 images from the NeurIPS'17 adversarial competition dataset and 100 text descriptions from the MS-COCO dataset as target texts. Table III presents the experimental results. Our method outperforms SIA-CWA in terms of attack success rate, demonstrating its superior transferability.

3) Qualitative Results On Open Source VLMs: We then present visualizations depicting the outcomes of our method's attacks on open source VLMs, as illustrated in Figure 6. Considering the image caption task, we focus on two models: Unidiffuser and BLIP2. Considering the VQA task, we focus on MiniGPT-4, LLaVA and Img2LLM. In the case of MiniGPT-4, the input text is configured as "What is the image showing?". For LLaVA, the input text is set to "What is the main contain of this image?", and the prefix "The main contain is" is omitted in the output. For Img2LLM, the input text is configured as "What is the content of this image?". Our method demonstrates the capability to effectively induce both gray-box and black-box VLMs to produce adversarial

[2]https://chat.openai.com/
[3]https://gemini.google.com/
[4]https://copilot.microsoft.com/
[5]https://yiyan.baidu.com/

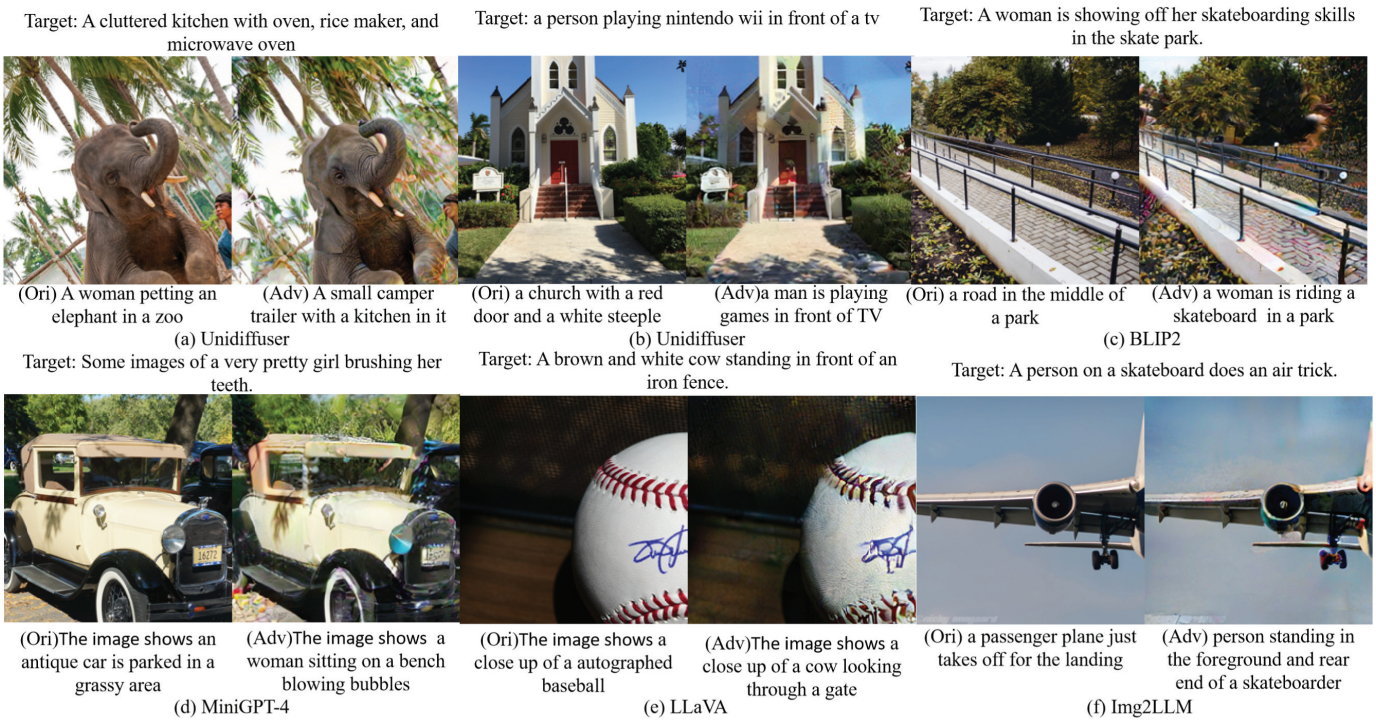


Fig. 6. Visualization of the attack results of our method on various open-source VLMs. We show the adversarial target text above the image, and display the image caption results of original image and adversarial example below the image.

target semantics. For example, in the case of LLaVA's attack, we define the adversarial target text as "A cake that has various gelatins in it." LLaVA generate the response "The main contain is a close-up view of a partially eaten cake with chocolate and white frosting." as the target output, while the original image's content is described as "The main contain is a bird, specifically a seagull, walking on the beach near the water.".

In addition, we visualize the outputs of various victim models with the same adversarial example, as shown in Figure 7. The visualization demonstrates that the adversarial example successfully induces all victim models to produce the target semantics.

4) Qualitative Results On Commercial VLMs: We finally show screenshots of successful attacks on various commercial VLMs image description tasks, including Google's Gemini, Microsoft's Copilot, Baidu's ERNIE Bot, and OpenAI's GPT-4V, as shown in Figure 9. These models are large-scale visual language models deployed commercially, and their model configurations and training datasets have not been made public. Moreoever, compared with open source VLMs, these models are equipped with more complex defense mechanisms, making them more difficult to attack. However, as shown in Figure 9, our method successfully induces these commercial VLMs to generate target responses. For example, in GPT-4V, we define the adversarial target text as "a kid is doing a skateboard trick down some stairs." GPT-4V generates the response "The main content of this image is a skateboarder performing a trick on a skateboard ramp...", while the semantics of the original image is "A bird standing on a branch." Moreover, our method is also applicable to various languages. For example, we use English to generate adversarial examples but successfully attack ERNIE Bot, which operates in Chinese.

5) Computational Resource Analysis: We compare the computational complexity of competing methods with our approach. Let N_s and N_d respectively denote the total number of parameters of the ensemble surrogate models and the diffusion model. Since N_s corresponds to the parameters of the ensemble models, it follows that $N_d \ll N_s$. In terms of space complexity, our method and AdvDiffuser both exhibit space complexity of $\mathbf{O}(N_d+N_s)$, whereas SSA exhibits $\mathbf{O}(N_s)$. Since SIA processes N_t images in parallel, its space complexity is $\mathbf{O}(N_t \cdot N_s)$. Thus, the space complexities of our method, AdvDiffuser, and SSA are comparable and notably lower than SIA. For time complexity, all methods exhibit linear time complexity, depending on the number of iterations and the duration of each iteration. AdvDiffuser and SSA involve internal loops within each iteration, resulting in longer iteration durations. SIA processes multiple images in parallel, which slightly increases iteration durations. In contrast, our method not only achieves the shortest iteration duration but also requires fewer iterations, making it the most time-efficient among the compared methods.

To further demonstrate the time efficiency of our method, we conduct the convergence analysis. We select CWA, SSA, and SIA as comparison methods. The experimental results are depicted in Figure 8. Our method converges to a flat trend when $N_I = 200$ ($N = 10$), whereas other methods achieve convergence at $N_I = 300$. Moreover, our method achieves the same attack effect with fewer iterations.

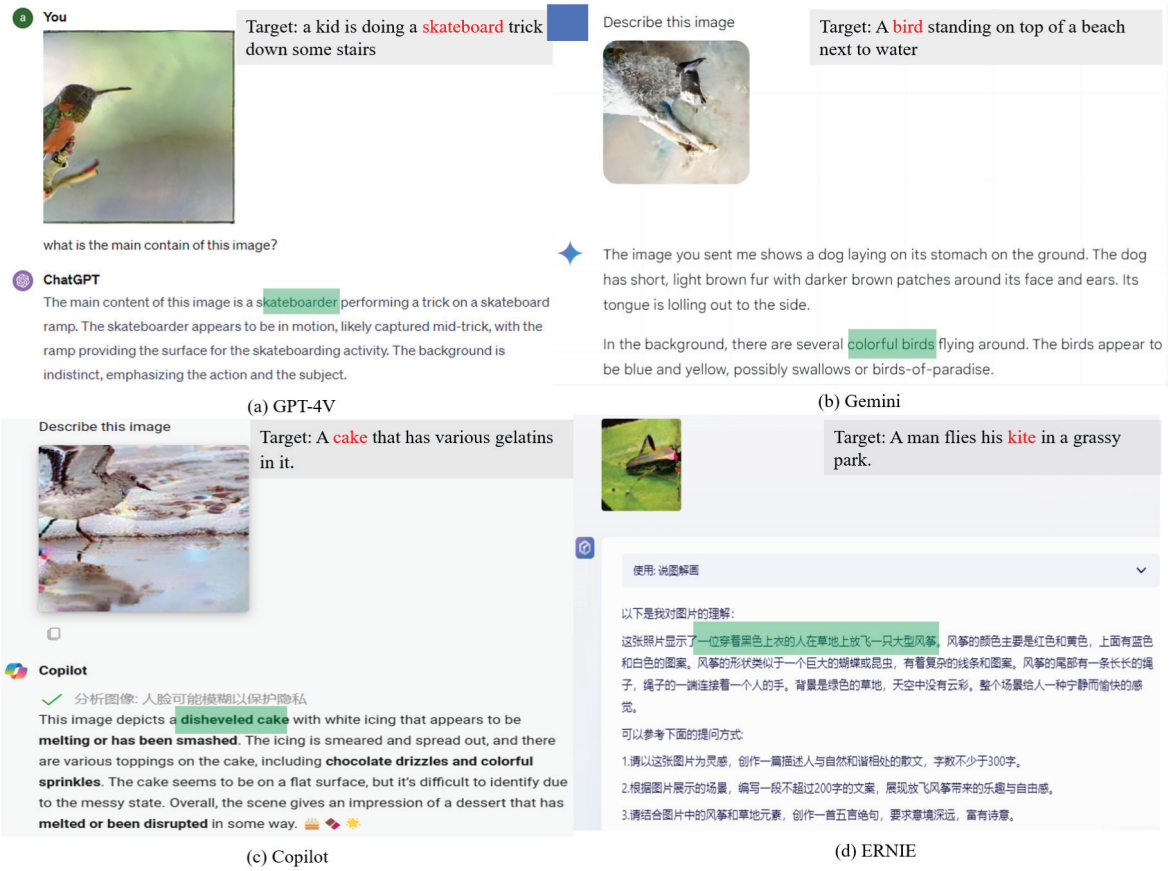


Fig. 7. Visualization results of using the same adversarial example to attack various victim models. Where the top one is the original image, and the bottom one is the adversarial example. The target semantics is marked in green.

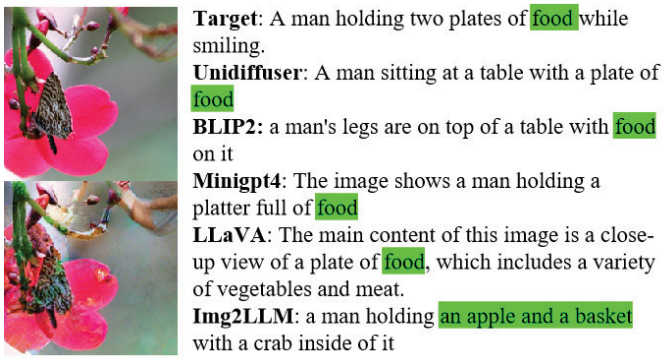


Fig. 8. Convergence analysis of different methods, with the horizontal axis denoting the number of iterations N_I.

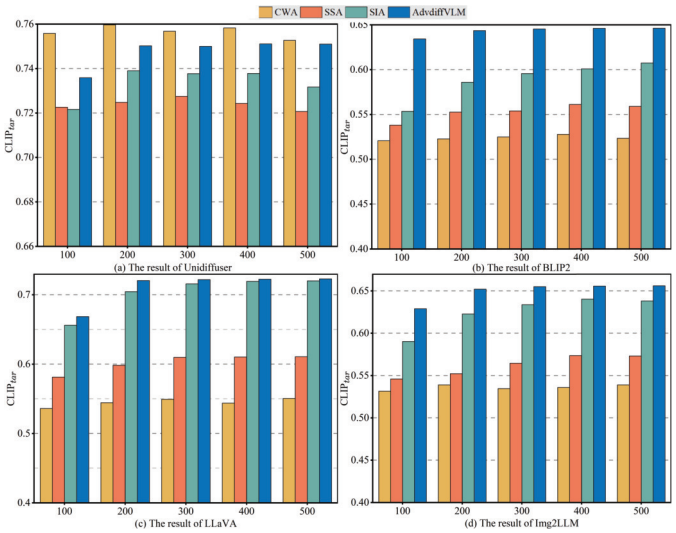


Fig. 9. Screenshots of successful attacks against various commercial VLMs API's image description. We give the adversarial target text on the right side of the image. In addition, we mark the main objects of the adversarial target in red and the main objects in the API's response in green.

The comprehensive experiments and analyses above demonstrate that our method delivers superior attack performance on both open-source and commercial VLMs. Moreover, it offers notable advantages in terms of resource efficiency and faster convergence speed.

C. More Experiments

1) Experiment With a Single Surrogate Model: All experiments in the previous subsection use the ensemble surrogate models to improve the transferability of adversarial examples. To further illustrate the effect of the single surrogate model, we conduct the comparative experiment and select ViT-B/32 as the surrogate model. The experimental results are shown in Table IV. As shown, our method outperforms SSA and

TABLE IV
COMPARATIVE RESULTS USING THE SINGLE SURROGATE MODEL

Method	Unidiffuser	BLIP2	LLaVA	Img2LLM
SSA-Ens	0.7356	0.5024	0.5522	0.5363
SIA-Ens	0.7473	0.5330	0.5923	0.5393
AdvDiffuser$_{adaptive}$	0.6930	0.5011	0.5238	0.5297
AdvDiffVLM	0.6982	0.5279	0.5793	0.5402

AdvDiffuser but is slightly less effective than SIA. This is because the score estimated by the single surrogate model deviates significantly from the true score. Furthermore, the transferability of a single surrogate model is significantly lower compared to that of the ensemble surrogate models.

2) Against Adversarial Defense Models: Our method achieves superior attack performance on both open source and commercial VLMs. In recent years, various adversarial defense methods have been proposed to mitigate the threat of adversarial examples. Defense methods can be broadly categorized into adversarial training and data preprocessing. Due to the high time and resource costs and instability of adversarial training [18], it has not been applied to VLM defense. In contrast, data preprocessing is model-independent and highly adaptable, making it a popular defense strategy across various models. To demonstrate the effectiveness of our method in resisting data preprocessing attacks, we conduct extensive experiments on Bit Reduction [65], STL [66], JPEG Compression [67], DISCO [68], JPEG+DISCO, and DiffPure [69]. The data preprocessing techniques we used are grouped into three categories: introducing randomness, denoising, and data reconstruction. Bit Reduction introduces randomness by modifying image bits; JPEG Compression performs denoising through blurring operations; and other methods involve reconstructing adversarial examples using various reconstruction networks and techniques. We report the CLIP$_{tar}$ metric. At the same time, we report the CLIP$_{tar}$ reduction results, which more accurately reflect the ability of adversarial examples to resist defense methods. The experimental results are shown in Table V. It can be observed that, for all defense methods, both CLIP$_{tar}$ and CLIP$_{tar}$ reduction results of our methods outperform the baselines. This demonstrates the superiority of our method against defense methods compared to baselines.

To better evaluate the resistance of our method against adversarial defense methods, we further in detail show the results of the SOTA defense method, namely DiffPure, in Table VI. It can be found that our method outperforms baselines in both gray-box and black-box settings. For example, on Unidiffuser, for CLIP$_{tar}$ score, our method is 0.0895 higher than SIA-CWA. On BLIP2, for CLIP$_{tar}$ score, our method is 0.0428 higher than SIA-CWA. Furthermore, in all cases, the attack success rate of our methods is higher than the baselines. These experimental results demonstrate that our method outperforms baselines in evading the DiffPure defense method.

We can break the SOTA defense method Diffpure with an attack success rate of more than 10% in a completely black-box scenario, exposing the flaws in current defense methods and raising new security concerns for designing more robust deep learning models.

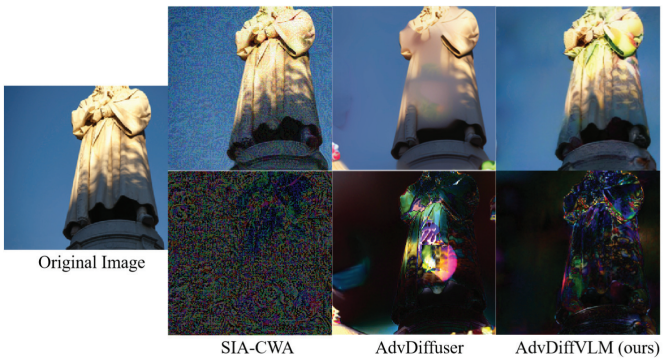


Fig. 10. Visualization of adversarial perturbations generated by different attack methods. Note that the first row represents adversarial examples, and the second row represents adversarial perturbations. We choose SIA-CWA and AdvDiffuser$_{adaptive}$ as representatives of baselines. We amplify the perturbation values for better visualization.

3) Image Quality Comparison: The image quality of adversarial examples is also particularly important. Adversarial examples with poor image quality can be easily detected. We further evaluate the image quality of the generated adversarial examples using four evaluation metrics: SSIM, FID, LPIPS, and BRISQUE. As shown in Table VII, compared to baselines, the adversarial examples generated by our method exhibit higher image quality. Specifically, our results are significantly better than the baselines in terms of SSIM, LPIPS, and FID evaluation metrics. For the BRISQUE metric, AdvDiffuser outperforms our method. This is because BRISQUE is a reference-free image quality assessment algorithm and is sensitive to blur, noise, color change, etc. As shown in Figure 10, the adversarial examples generated by AdvDiffuser lack obvious abnormalities in these elements, so its results are marginally better than our method. However, as shown in Figure 10, the perturbation introduced by our method is semantic, while AdvDiffuser significantly alters the non-salient area, resulting in poor visual effects. This shows that the adversarial examples generated by AdvDiffuser are unsuitable for more complex scenarios, such as attacking VLMs. In addition, it can be seen that the adversarial examples generated by the transfer-based methods exhibit significant noise, indicating that our method has obvious superiority in terms of image quality.

D. Ablation Experiments

To further understand the effectiveness of AdvDiffVLM, we discuss the role of each module. We set $N = 1$ to more conveniently discuss the impact of each module. We consider three cases, including using only a single ViT-B/32 to calculate the loss, using a simple ensemble strategy, and not using the GCMG module, named Single, Ens, and w/o mask respectively.

1) Is AEGE Module Beneficial for Boosting the Attack Capability?: We first explore whether the AEGE module could help improve the transferability and robustness of adversarial examples. We divide the AEGE module into two

TABLE V

COMPARISON RESULTS OF DEFENSE EXPERIMENTS WITH SOTA METHOD SIA. WE USE CLIP_{TAR} EVALUATION METRIC AND REPORT THE REDUCTION RESULTS OF CLIP_{TAR} WHERE THE BEST RESULT IS BOLDED. OTHERWISE, THE PARENTHESES REPRESENT THE HYPERPARAMETERS (IN THEIR PAPER)

Defense models	Attack methods	Unidiffuser	BLIP2	MiniGPT-4	LLaVA	Img2LLM
Bit Reduction (4)	SIA-Ens	$0.7204_{\downarrow 0.0173}$	$0.5602_{\downarrow 0.0454}$	$0.5273_{\downarrow 0.0432}$	$0.7034_{\downarrow 0.0124}$	$0.6284_{\downarrow 0.0053}$
	SIA-CWA	$0.7281_{\downarrow 0.0217}$	$0.5798_{\downarrow 0.0435}$	$0.5442_{\downarrow 0.0468}$	$0.7063_{\downarrow 0.0131}$	$0.6375_{\downarrow 0.0026}$
	AdvDiffVLM	$\mathbf{0.7397_{\downarrow 0.0105}}$	$\mathbf{0.6320_{\downarrow 0.0115}}$	$\mathbf{0.6261_{\downarrow 0.0084}}$	$\mathbf{0.7168_{\downarrow 0.0038}}$	$\mathbf{0.6501_{\downarrow 0.0020}}$
STL (k=64, s=8, λ=0.2)	SIA-Ens	$0.7192_{\downarrow 0.0185}$	$0.5571_{\downarrow 0.0485}$	$0.5192_{\downarrow 0.0513}$	$0.6968_{\downarrow 0.0190}$	$0.6230_{\downarrow 0.0107}$
	SIA-CWA	$0.7233_{\downarrow 0.0265}$	$0.5733_{\downarrow 0.0500}$	$0.5385_{\downarrow 0.0525}$	$0.7001_{\downarrow 0.0193}$	$0.6314_{\downarrow 0.0087}$
	AdvDiffVLM	$\mathbf{0.7329_{\downarrow 0.0173}}$	$\mathbf{0.6267_{\downarrow 0.0168}}$	$\mathbf{0.5997_{\downarrow 0.0148}}$	$\mathbf{0.7145_{\downarrow 0.0061}}$	$\mathbf{0.6471_{\downarrow 0.0050}}$
JPEG Compression (p=50)	SIA-Ens	$0.6734_{\downarrow 0.0642}$	$0.5345_{\downarrow 0.0711}$	$0.5002_{\downarrow 0.0703}$	$0.6542_{\downarrow 0.0616}$	$0.6020_{\downarrow 0.0317}$
	SIA-CWA	$0.6801_{\downarrow 0.0697}$	$0.5525_{\downarrow 0.0708}$	$0.5273_{\downarrow 0.0637}$	$0.6550_{\downarrow 0.0644}$	$0.6088_{\downarrow 0.0313}$
	AdvDiffVLM	$\mathbf{0.6896_{\downarrow 0.0606}}$	$\mathbf{0.6218_{\downarrow 0.0217}}$	$\mathbf{0.5865_{\downarrow 0.0380}}$	$\mathbf{0.6983_{\downarrow 0.0223}}$	$\mathbf{0.6354_{\downarrow 0.0167}}$
DISCO (s=3, k=5)	SIA-Ens	$0.6087_{\downarrow 0.1290}$	$0.5134_{\downarrow 0.0922}$	$0.4986_{\downarrow 0.0719}$	$0.6274_{\downarrow 0.0884}$	$0.5771_{\downarrow 0.0566}$
	SIA-CWA	$0.6114_{\downarrow 0.1384}$	$0.5290_{\downarrow 0.0943}$	$0.5114_{\downarrow 0.0796}$	$0.6331_{\downarrow 0.0863}$	$0.5842_{\downarrow 0.0559}$
	AdvDiffVLM	$\mathbf{0.6215_{\downarrow 0.1287}}$	$\mathbf{0.5892_{\downarrow 0.0543}}$	$\mathbf{0.5727_{\downarrow 0.0418}}$	$\mathbf{0.6728_{\downarrow 0.0478}}$	$\mathbf{0.6093_{\downarrow 0.0428}}$
DISCO+JPEG	SIA-Ens	$0.5642_{\downarrow 0.1735}$	$0.5025_{\downarrow 0.1031}$	$0.4878_{\downarrow 0.0827}$	$0.6067_{\downarrow 0.1091}$	$0.5681_{\downarrow 0.0656}$
	SIA-CWA	$0.5735_{\downarrow 0.1763}$	$0.5176_{\downarrow 0.1057}$	$0.5074_{\downarrow 0.0836}$	$0.6106_{\downarrow 0.1088}$	$0.5692_{\downarrow 0.0709}$
	AdvDiffVLM	$\mathbf{0.5924_{\downarrow 0.1578}}$	$\mathbf{0.5859_{\downarrow 0.0576}}$	$\mathbf{0.5650_{\downarrow 0.0495}}$	$\mathbf{0.6724_{\downarrow 0.0482}}$	$\mathbf{0.6081_{\downarrow 0.0440}}$
DiffPure (t* = 0.15)	SIA-Ens	$0.4921_{\downarrow 0.2456}$	$0.5048_{\downarrow 0.1008}$	$0.4919_{\downarrow 0.0786}$	$0.5356_{\downarrow 0.1802}$	$0.5372_{\downarrow 0.0965}$
	SIA-CWA	$0.4942_{\downarrow 0.2556}$	$0.5099_{\downarrow 0.1136}$	$0.5025_{\downarrow 0.0885}$	$0.5360_{\downarrow 0.1835}$	$0.5388_{\downarrow 0.1013}$
	AdvDiffVLM	$\mathbf{0.5837_{\downarrow 0.1665}}$	$\mathbf{0.5527_{\downarrow 0.0908}}$	$\mathbf{0.5506_{\downarrow 0.0639}}$	$\mathbf{0.5857_{\downarrow 0.1349}}$	$\mathbf{0.5711_{\downarrow 0.0810}}$

TABLE VI

DEFENSE RESULTS WITH DIFFPURE. THE SETTING ARE THE SAME AS TABLE II EXCEPT THE ADVERSARIAL EXAMPLES ARE PURIFIED BY DIFFPURE. IN THIS TABLE, CLIP_{TAR} EVALUATES THE SIMILARITY BETWEEN THE RESULTS OF PURIFIED EXAMPLES AND THE TARGET TEXTS

	Unidiffuser*		BLIP2		MiniGPT-4		LLaVA		Img2LLM	
	$\text{CLIP}_{tar}\uparrow$	ASR $\uparrow$	$\text{CLIP}_{tar}\uparrow$	ASR $\uparrow$	$\text{CLIP}_{tar}\uparrow$	ASR $\uparrow$	$\text{CLIP}_{tar}\uparrow$	ASR $\uparrow$	$\text{CLIP}_{tar}\uparrow$	ASR $\uparrow$
Original	0.4802	0.0%	0.4924	0.0%	0.4831	0.0%	0.5253	0.0%	0.5302	0.0%
Ens	0.4833	0.0%	0.4929	0.0%	0.4840	0.0%	0.5263	0.0%	0.5332	0.0%
SVRE	0.4846	0.7%	0.4953	0.0%	0.4852	0.0%	0.5264	0.0%	0.5312	0.0%
CWA	0.4873	2.1%	0.4973	0.0%	0.4901	1.0%	0.5272	0.8%	0.5307	0.0%
SSA-Ens	0.4914	0.9%	0.5024	0.0%	0.4916	0.0%	0.5280	1.2%	0.5322	0.0%
SSA-SVRE	0.4899	2.1%	0.4984	0.2%	0.4918	0.0%	0.5273	1.2%	0.5356	0.0%
SSA-CWA	0.4868	2.5%	0.4997	0.0%	0.4997	0.0%	0.5283	2.8%	0.5367	0.7%
SIA-Ens	0.4921	3.7%	0.5048	1.2%	0.4919	1.1%	0.5356	2.5%	0.5372	1.6%
SIA-SVRE	0.4930	3.9%	0.5012	1.8%	0.5011	1.6%	0.5349	4.2%	0.5380	2.5%
SIA-CWA	0.4942	5.8%	0.5099	2.6%	0.5025	2.2%	0.5360	4.0%	0.5388	1.5%
AdvDiffuser_{ens}	0.4920	4.2%	0.4933	2.6%	0.4906	2.4%	0.5325	3.7%	0.5310	2.7%
$\text{AdvDiffuser}_{adaptive}$	0.4922	4.5%	0.5001	3.2%	0.5001	3.2%	0.5336	3.4%	0.5325	2.8%
AdvDiffVLM	**0.5837**	**22.4%**	**0.5527**	**10.2%**	**0.5506**	**12.6%**	**0.5857**	**18.0%**	**0.5711**	**10.5%**

TABLE VII

QUALITY COMPARISON OF ADVERSARIAL EXAMPLES UNDER FOUR EVALUATION METRICS. THE BEST RESULT IS BOLDED

Method	SSIM $\uparrow$	LPIPS $\downarrow$	FID $\downarrow$	BRISQUE $\downarrow$
SSA-Ens	0.6687	0.3320	110.5	66.89
SSA-SVRE	0.6610	0.3325	112.6	70.05
SSA-CWA	0.6545	0.3673	123.4	67.67
SIA-Ens	0.6925	0.2990	117.3	55.61
SIA-SVRE	0.6920	0.3042	120.0	57.42
SIA-CWA	0.6892	0.3306	125.3	56.02
AdvDiffuser_{ens}	0.6520	0.3074	115.5	**14.61**
$\text{AdvDiffuser}_{adaptive}$	0.6471	0.3096	126.7	15.32
AdvDiffVLM	**0.6992**	**0.2930**	**107.4**	32.96

approaches, Single and Ens, and maintain all other conditions constant. The results are shown in Figure 11(a) and (b). It is observable that the ensemble method exhibits better performance in transferability and robustness compared to the single loss method. Furthermore, the performance of the adaptive ensemble method is enhanced compared to the basic ensemble method. The experimental results demonstrate that the AEGE module enhances the transferability and robustness of adversarial examples.

2) Does GCMG Module Help Trade-Off Image Quality and Attack Capability?: Next, we explore the role of the GCMG module in balancing image quality and transferability. We compare this with the w/o mask method, and the results are presented in Figure 11. As shown in Figure 11(a) and (b), the use of the GCMG module results in a slight decrease in the transferability and robustness of the adversarial examples. However, as shown in Figure 11(c), the absence of the GCMG module leads to the adversarial examples exhibiting obvious target features, and the use of the GCMG module enhances

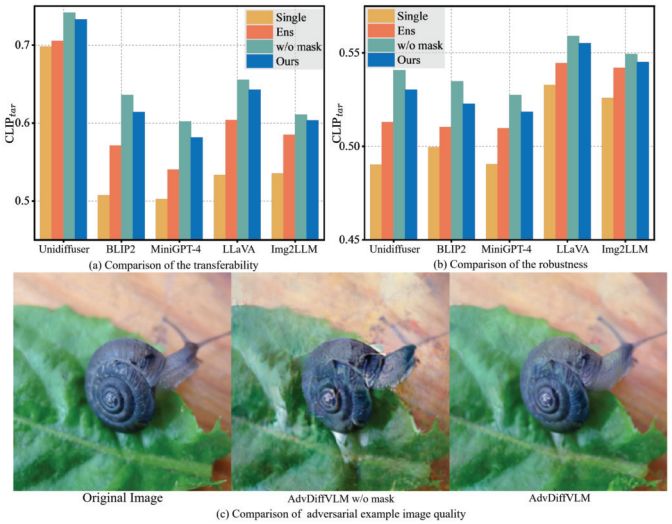


Fig. 11. Comparison results of different ablation methods. Here, "Single" means using a single ViT-B/32 to calculate the loss, "Ens" means using the simple ensemble strategy, and "w/o mask" means not using GCMG module.

TABLE VIII

COMPARISON OF IMAGE QUALITY OF ADVERSARIAL EXAMPLES BEFORE AND AFTER USING THE GCMG MODULE. THE BEST RESULT IS BOLDED

Method	SSIM ↑	LPIPS ↓	FID ↓	BRISQUE ↓
w/o mask	0.7129	0.2687	111.9	16.92
Ours	**0.7188**	**0.2358**	**96.1**	**16.80**

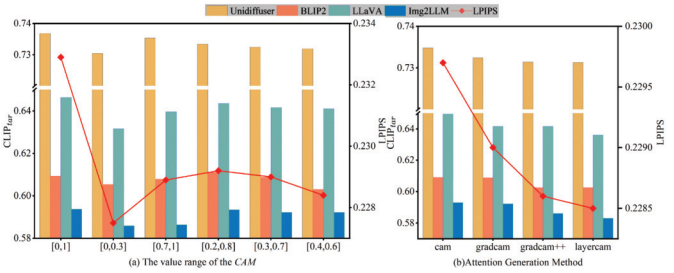


Fig. 12. Ablation study of the different configurations of obtaining m. We adopt the CLIP$_{tar}$ and LPIPS scores to show the impact of transferability and image quality with four VLMs.

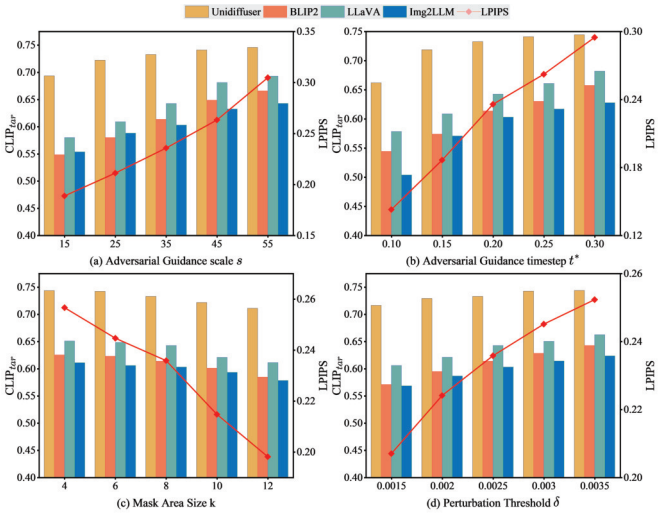


Fig. 13. The impact of inner loop hyperparameters. We adopt the CLIP$_{tar}$ and LPIPS scores to show the impact of transferability and image quality with four VLMs. A higher CLIP$_{tar}$ value indicates better performance, whereas a lower LPIPS value signifies better results. We only vary one of the hyperparameters at a time, and then fix the other three hyperparameters to the preset values shown in Section V-A. Note: the results of CLIP$_{tar}$ are presented using bar graphs, while LPIPS results are depicted using dot-line graphs.

the visual quality of the adversarial example. In addition, Table VIII further shows that the GCMG module can improve the visual quality of adversarial examples. The experimental results demonstrate that GCMG effectively balances the visual quality and attack capability of the adversarial examples.

Then we perform ablation experiments on different configurations of GCMG. Firstly, we conduct experiments on the effects of the value range of the *CAM*. The *CAM* value range is used to balance image quality with attack capability. Cropping the lower boundary increases the probability of selecting non-critical areas, while cropping the upper boundary reduces the probability of selecting critical areas, enhancing the overall quality of adversarial examples. To determine an optimal range, we test several intervals: [0,1], [0,0.3], [0.7,1], [0.2,0.8], [0.3,0.7], and [0.4,0.6]. The results, shown in Figure 12 (a), indicate that adjusting the *CAM* value range allows for a trade-off between attack capability and image quality. Based on these results, we select the range [0.3,0.7] to achieve an optimal balance between quality and attack effectiveness. Then we conduct experiments on the effects of different attention generation methods. In the field of adversarial attacks, GradCAM is a commonly used method for generating masks, as demonstrated in [32], [70] and [71]. To further illustrate the role of various attention mechanisms in mask generation, we compare CAM [72], GradCAM [60], GradCAM++ [73], and LayerCAM [74], with results shown in Figure 12 (b). Our findings indicate that differences in attack and transfer capabilities among adversarial examples generated by these mechanisms are minimal, with variations in CLIP and LPIPS values within 0.001. Moreover, we observe a trade-off between attack capability and image quality. After weighing both factors and for consistency with previous studies, we opted to use GradCAM for mask generation in this paper.

E. Hyperparameter Studies

In this subsection, we explore the impact of hyperparameters, including inner loop hyperparameters s, t^*, k, and δ and outer loop hyperparameter N.

1) The Impacts of Inner Loop Hyperparameters: We first discuss the impacts of inner loop hyperparameters (including the s, t^*, k, and δ). We set $N = 1$ and conduct tests on Unidiffuser, BLIP2, LLaVA and Img2LLM. The experimental results are shown in Figure 13. It is evident that all parameters influence the trade-off between transferability and image quality. Increasing values for parameters s, t^*, and δ enhance transferability but diminish the visual quality of adversarial examples. This is because larger values for these parameters result in a greater perturbation, allowing for the embedding of more adversarial semantics into the image. Conversely,

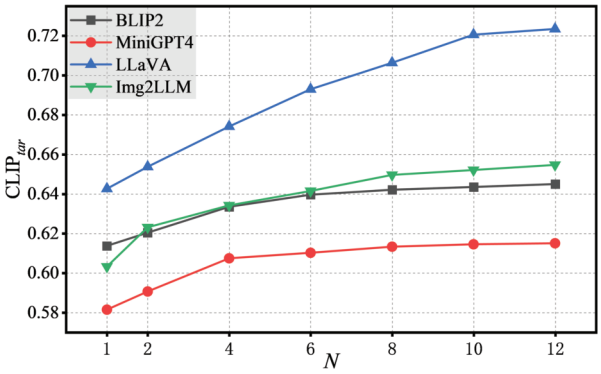


Fig. 14. Transferability of adversarial examples on various black-box VLMs as N changes from 1 to 12.

increasing the value of k produces adversarial examples with improved visual effects but reduces transferability. The reason is that larger values of k result in a larger generated mask, making it more challenging to modify the important areas in the image. To achieve an optimal trade-off between transferability and image quality, we empirically select $s = 35, t^* = 0.2, \mathrm{k} = 8$ and $\delta = 0.0025$.

2) The Impact of Outer Loop Hyperparameter: Next, we investigate the impact of the outer hyperparameter N on the transferability of adversarial examples. We conduct experiments on BLIP2, MiniGPT-4, LLaVA, and Img2LLM with $s = 35, t^* = 0.2, \mathrm{k} = 8$, and $\delta = 0.0025$. The experimental results are shown in Figure 14. It can be found that N improves the transferability of adversarial examples, but the improvement gradually fades. Specifically, the increase in transferability is limited after $N = 6, 6, 8, 10$ for BLIP2, MiniGPT-4, Img2LLM, and LLaVA. Given that increasing N increases the computational cost, we choose $N = 10$ to strike a balance between transferability and cost.

VI. Conclusion

In this work, we propose AdvDiffVLM, an unrestricted and targeted adversarial example generation method for VLMs. We design the AEGE based on the idea of score matching. It embeds the target semantics into adversarial examples, which can generate targeted adversarial examples that exhibit enhanced transferability in a more efficient manner. To balance adversarial example quality and attack effectiveness, we propose the GCMG module. Additionally, we enhance the embedding of target semantics into adversarial examples through multiple iterations. Extensive experiments show that our method generates targeted adversarial examples 5x to 10x times faster than baseline methods while achieving superior transferability.

Impact Statements

Our research mainly aims to discover vulnerabilities in open-source large VLMs and commercial VLMs such as GPT-4V, providing insights for developing more robust and trustworthy VLMs. However, our attack methods can be abused to evade actual deployed commercial systems, causing potential negative social impacts. For example, criminals may use our methods to cause GPT-4V APIs to output target responses, causing serious harm.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Cyber Security Agency of Singapore, and Infocomm Media Development Authority.

References

[1] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 12888–12900.

[2] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2023, pp. 1–13.

[3] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, Jan. 2023, pp. 34892–34916. [Online]. Available: https://openreview.net/forum?id=w0H2xGHlkw

[4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10684–10695.

[5] F. Bao et al., "One transformer fits all distributions in multi-modal diffusion at scale," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2023, pp. 1692–1717.

[6] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "ChatGPT for robotics: Design principles and model abilities," *IEEE Access*, vol. 12, pp. 55682–55696, 2024.

[7] G. Liao, J. Li, and X. Ye, "VLM2Scene: Self-supervised image-text-LiDAR learning with foundation models for autonomous driving scene understanding," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 4, pp. 3351–3359.

[8] C. Cui et al., "A survey on multimodal large language models for autonomous driving," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2024, pp. 958–979.

[9] P. Nagesh, B. Prabha, S. B. Gole, G. S. N. Rao, and N. V. Ramana, "Visual assistance for visually impaired people using image caption and text to speech," in *Proc. AIP Conf.*, 2024, vol. 2512, no. 1, pp. 20–37.

[10] W. Wang et al., "An image is worth a thousand toxic words: A metamorphic testing framework for content moderation software," in *Proc. 38th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Sep. 2023, pp. 1339–1351.

[11] D. Han, X. Jia, Y. Bai, J. Gu, Y. Liu, and X. Cao, "OT-attack: Enhancing adversarial transferability of vision-language models via optimal transport optimization," 2023, *arXiv:2312.04403*.

[12] S. Gao, X. Jia, X. Ren, I. Tsang, and Q. Guo, "Boosting transferability in vision-language attacks via diversification along the intersection region of adversarial trajectory," 2024, *arXiv:2403.12445*.

[13] X. Gu et al., "Agent smith: A single image can jailbreak one million multimodal LLM agents exponentially fast," 2024, *arXiv:2402.08567*.

[14] J. Zheng, C. Lin, J. Sun, Z. Zhao, Q. Li, and C. Shen, "Physical 3D adversarial attacks against monocular depth estimation in autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 24452–24461.

[15] M. T. West et al., "Towards quantum enhanced adversarial robustness in machine learning," *Nature Mach. Intell.*, vol. 5, no. 6, pp. 581–589, May 2023.

[16] Y. Zhao et al., "On evaluating adversarial robustness of large vision-language models," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, Jan. 2023, pp. 54111–54138.

[17] X. Jia et al., "Revisiting and exploring efficient fast adversarial training via LAW: Lipschitz regularization and auto weight averaging," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8125–8139, 2024.

[18] X. Jia et al., "Improving fast adversarial training with prior-guided knowledge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 9, pp. 6367–6383, Sep. 2024.

[19] N. Aafaq, N. Akhtar, W. Liu, M. Shah, and A. Mian, "Language model agnostic gray-box adversarial attack on image captioning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 626–638, 2023.

[20] Y. Xu et al., "Exact adversarial attack to image captioning via structured output learning with latent variables," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4130–4139.

[21] R. Lapid and M. Sipper, "I see dead people: Gray-box adversarial attack on image-to-text models," in *Proc. Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discovery Databases*, Jan. 2023, pp. 1–13.

[22] A. E. Baia, V. Poggioni, and A. Cavallaro, "Black-box attacks on image activity prediction and its natural language explanations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3688–3697.

[23] L. Zhu, T. Wang, J. Li, Z. Zhang, J. Shen, and X. Wang, "Efficient query-based black-box attack against cross-modal hashing retrieval," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–25, Jul. 2023.

[24] P. N. Williams and K. Li, "Black-box sparse adversarial attack via multi-objective optimisation CVPR proceedings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12291–12301.

[25] H. Zhu, X. Sui, Y. Ren, Y. Jia, and L. Zhang, "Boosting transferability of targeted adversarial examples with non-robust feature alignment," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120248.

[26] H. Chen, Y. Zhang, Y. Dong, and J. Zhu, "Rethinking model ensemble in transfer-based adversarial attacks," in *Proc. 12th Int. Conf. Learn. Represent.*, Jan. 2023, pp. 1–27. [Online]. Available: https://openreview.net/forum?id=AcJrSoArlh

[27] Y. Xiong, J. Lin, M. Zhang, J. E. Hopcroft, and K. He, "Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 14983–14992.

[28] X. Wang, Z. Zhang, and J. Zhang, "Structure invariant transformation for better adversarial transferability," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4584–4596.

[29] Y. Wang et al., "Boosting the transferability of adversarial attacks with frequency-aware perturbation," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 6293–6304, 2024.

[30] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Jan. 2018, pp. 8312–8323.

[31] Z. Zhao, D. Dua, and S. Singh, "Generating natural adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=H1BLjgZCb

[32] X. Chen, X. Gao, J. Zhao, K. Ye, and C.-Z. Xu, "AdvDiffuser: Natural adversarial example synthesis with diffusion models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4539–4549.

[33] A. S. Shamsabadi, R. Sanchez-Matilla, and A. Cavallaro, "ColorFool: Semantic adversarial colorization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nov. 2020, pp. 1151–1160.

[34] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2017, pp. 1–23. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[35] J. Guo et al., "From images to textual prompts: Zero-shot visual question answering with frozen large language models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10867–10877.

[36] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[37] Y. Long et al., "Frequency domain model augmentation for adversarial attack," in *Proc. Eur. Conf. Comput. Vis.*, vol. 13664. Cham, Switzerland: Springer, 2022, pp. 549–566.

[38] A. Hyvärinen, "Estimation of non-normalized statistical models by score matching," *J. Mach. Learn. Res.*, vol. 6, no. 24, pp. 695–709, Dec. 2005.

[39] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019, pp. 11918–11930.

[40] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2020, *arXiv:2011.13456*.

[41] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[42] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[43] W.-L. Chiang. (2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 With 90% ChatGPT Quality*. Accessed: Apr. 14, 2023. [Online]. Available: https://vicuna.lmsys.org

[44] S. Yin et al., "A survey on multimodal large language models," 2023, *arXiv:2306.13549*.

[45] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu, "Multimodal large language models: A survey," in *Proc. IEEE Int. Conf. Big Data (BigData)*, Dec. 2023, pp. 2247–2256.

[46] J.-B. Alayrac et al., "Flamingo: A visual language model for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 23716–23736.

[47] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing vision-language understanding with advanced large language models," 2023, *arXiv:2304.10592*.

[48] J. C. Costa, T. Roxo, H. Proença, and P. R. M. Inácio, "How deep learning sees the world: A survey on adversarial attacks & defenses," *IEEE Access*, vol. 12, pp. 61113–61136, 2024.

[49] S. Han, C. Lin, C. Shen, Q. Wang, and X. Guan, "Interpreting adversarial examples in deep learning: A review," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–38, Jul. 2023.

[50] J. Gu et al., "A survey on transferability of adversarial examples across deep neural networks," 2023, *arXiv:2310.17626*.

[51] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, "Fooling vision and language models despite localization and attention mechanism," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4951–4961.

[52] Y. Dong et al., "How robust is Google's bard to adversarial image attacks?" 2023, *arXiv:2309.11751*.

[53] Q. Li, Q. Hu, H. Fan, C. Lin, C. Shen, and L. Wu, "Attention-SA: Exploiting model-approximated data semantics for adversarial attack," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 8673–8684, 2024.

[54] D.-T. Peng, J. Dong, M. Zhang, J. Yang, and Z. Wang, "CSFAdv: Critical semantic fusion guided least-effort adversarial example attacks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5940–5955, 2024.

[55] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[56] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021, pp. 1–20. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[57] C. Luo, "Understanding diffusion models: A unified perspective," 2022, *arXiv:2208.11970*.

[58] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[59] Z. Cai, Y. Tan, and M. S. Asif, "Ensemble-based blackbox attacks on dense prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 4045–4055.

[60] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[61] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[62] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6629–6640.

[63] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[64] A. Mittal, A. K. Moorthy, and A. C. Bovik, "Blind/referenceless image spatial quality evaluator," in *Proc. 45th Asilomar Conf. Signals, Syst. Comput. (ASILOMAR)*, Nov. 2011, pp. 723–727.

[65] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*.

[66] B. Sun, N.-H. Tsai, F. Liu, R. Yu, and H. Su, "Adversarial defense by stratified convolutional sparse coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11447–11456.

[67] G. Karolina Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of JPG compression on adversarial images," 2016, *arXiv:1608.00853*.

[68] C.-H. Ho and N. Vasconcelos, "DISCO: Adversarial defense with local implicit functions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 23818–23837.

[69] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 1–23.

[70] Y. Zhu and Y. Jiang, "A non-global disturbance targeted adversarial example algorithm combined with C&W and grad-cam," *Neural Comput. Appl.*, vol. 35, no. 29, pp. 21633–21644, Oct. 2023.

[71] M. Yoshida, H. Namura, and M. Okuda, "Adversarial examples for image cropping: Gradient-based and Bayesian-optimized approaches for effective adversarial attack," *IEEE Access*, vol. 12, pp. 86541–86552, 2024.
[72] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[73] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.
[74] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "LayerCAM: Exploring hierarchical class activation maps for localization," *IEEE Trans. Image Process.*, vol. 30, pp. 5875–5888, 2021.

Qi Guo received the B.E. degree from Xi'an Jiaotong University, Xi'an, China, in 2021, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and AI security.

Shanmin Pang (Member, IEEE) received the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2015. From 2017 to 2018, he was a Visiting Scholar with The University of Virginia. He is currently an Associate Professor with the School of Software Engineering, Xi'an Jiaotong University. His research interests include computer vision, machine learning, and human-machine hybrid intelligence. He was a recipient of the Best Application Paper Award at ACCV 2012.

Xiaojun Jia received the Ph.D. degree from the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, and the School of Cyber Security, University of Chinese Academy of Sciences, Beijing. He is currently a Research Fellow with the Cyber Security Research Centre @ NTU, Nanyang Technological University, Singapore. His research interests include computer vision, deep learning, and adversarial machine learning.

Yang Liu (Senior Member, IEEE) received the Bachelor of Computing (Hons.) and Ph.D. degrees from the National University of Singapore (NUS) in 2005 and 2010, respectively. He started his post-doctoral work at NUS and MIT. In 2012, he joined Nanyang Technological University (NTU), where he is currently a Full Professor and the Director of the Cybersecurity Laboratory. He specializes in software engineering, cybersecurity, and artificial intelligence. He has more than 400 publications in top-tier conferences and journals. His research interests include the theory and practical usage of program analysis, data analysis, and AI to evaluate the design and implementation of software for high assurance and security. He has received several prestigious awards, including the MSRA Fellowship, the TRF Fellowship, the Nanyang Assistant Professor, the Tan Chin Tuan Fellowship, the Nanyang Research Award in 2019, an ACM Distinguished Speaker, the NRF Investigators, and 15 best paper awards and one most influence system award in top software engineering conferences, such as ASE, FSE, and ICSE.

Qing Guo (Senior Member, IEEE) received the Ph.D. degree in computer application technology from the School of Computer Science and Technology, Tianjin University, China. He is currently a Senior Research Scientist and a Principal Investigator (PI) with the Center for Frontier AI Research (CFAR), A*STAR, Singapore. He is also an Adjunct Assistant Professor with the National University of Singapore (NUS), Singapore, and a Senior PC Member of AAAI. His research interests include computer vision, AI security, adversarial attacks, and robustness. He was a recipient of the Best Platinum Paper Award from ICME in 2018, the ACM Tianjin Outstanding Doctoral Dissertation Award in 2020, third place in the AISG Trusted Media Challenge, won the Best Paper Award at the ECCV 2022 AROW workshop, and the AISG Robust AI Challenge Grant 2023. He serves as the Area Chair for ICML, ICLR, and IJCAI.