Pattern Recognition 172 (2026) 112710

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Beyond deceptive flatness: Dual-order solution for strengthening adversarial transferability

Zhixuan Zhang [a], Pingyu Wang [b,*], Xingjian Zheng [c], Linbo Qing [a], Qi Liu [d]

[a] School of Cyber Science and Engineering, Sichuan University, Chengdu, 610207, Sichuan, China
[b] College of Electronics and Information Engineering, Sichuan University, Chengdu, Sichuan, 610065, China
[c] Company of Frost Drill Intellectual Software Pte. Ltd, Singapore
[d] School of Future Technology, South China University of Technology, Guangzhou, Guangdong, 511442, China

ARTICLE INFO

Keywords:
Adversarial transferability
Black-box attack
Inner-loop sampling
Adversarial flatness
Deep neural network

ABSTRACT

Transferable attacks generate adversarial examples on surrogate models to fool unknown victim models, posing real-world threats and growing research interest. Despite focusing on flat losses for transferable adversarial examples, recent studies still fall into suboptimal regions, especially the flat-yet-sharp areas, termed as deceptive flatness. In this paper, we introduce a novel black-box gradient-based transferable attack from a perspective of dual-order information. Specifically, we feasibly propose Adversarial Flatness (AF) to the deceptive flatness problem and a theoretical assurance for adversarial transferability. Based on this, using an efficient approximation of our objective, we instantiate our attack as Adversarial Flatness Attack (AFA), addressing the altered gradient sign issue. Additionally, to further improve the attack ability, we devise MonteCarlo Adversarial Sampling (MCAS) by enhancing the inner-loop sampling efficiency. Extensive results on ImageNet-compatible dataset demonstrate our superiority over six baselines by generating adversarial examples in flatter regions, boosting transferability across model architectures, input transformation attacks and the Baidu Cloud API.

1. Introduction

Despite the excellent pattern recognition capabilities of Deep Neural Networks (DNNs) in various tasks, they remain vulnerable to adversarial attacks, generating imperceptible perturbed adversarial examples that cause misclassifications. And, it represents a significant risk for DNN-based applications that are security sensitive [1,2]. Essentially, adversarial attacks aim to uncover DNNs' vulnerabilities and improve model resilience through defense strategies.

Depending on whether the attacker has full access to the target model's internal information, adversarial attacks can be categorized as white-box attacks and black-box attacks. The former [3,4], with complete access to the target model, often achieves remarkable Attack Success Rates (ASRs). The impracticality of these methods in real-world scenarios stems from the privacy and security of target models. On the other hand, the effectiveness of the latter is restricted, since they have much lower ASRs than white-box attacks. Recently, several techniques have been suggested to improve the inferior results in black-box scenarios, using gradient estimation [5], input transformation [6–10] and model ensembles [11]. While recent works [12,13] improve attack capability by minimizing the loss difference within a gradient neighborhood (zeroth-order flatness), they may overlook truly flat local regions when the neighborhood radius is too small. To address this problem, another study [14] investigates stable neighboring gradients (first-order flatness), but it still encounters the issue of flat-yet-sharp regions, termed deceptive flatness, as illustrated in Fig. 1. Notably, the flat-yet-sharp region typically arises from the unstable training process of the selected surrogate models, which may be attributed to the mathematical properties of the loss function itself (such as saturation regions). This identified misleading area could lead to increased computational costs and overfitting of the surrogate model. This indicates that these techniques may not be adequate for effectively targeting other black-box models.

In this study, to eliminate the impact of deceptive flatness on adversarial transferability, we propose a novel gradient-based black-box transferable attack technique from a perspective of dual-order information. First, based on the deceptive flatness issue we have identified, we propose an Adversarial Flatness (AF) to effectively integrate global zero-order gradient information to compensate for the limitations of first-order flatness. And, we provide a theoretical guarantee for its adversarial transferability. Subsequently, through the utilization of an efficient estimation of our objective, we are able to flexibly implement our method as the Adversarial Flatness Attack (AFA). Meanwhile, the AFA

* Corresponding author.
E-mail addresses: zhangzhixuan77@gmail.com (Z. Zhang), wangpingyu@scu.edu.cn (P. Wang), xingjian1972zheng@gmail.com (X. Zheng), qing_lb@scu.edu.cn (L. Qing), drliuqi@scut.edu.cn (Q. Liu).

https://doi.org/10.1016/j.patcog.2025.112710
Received 9 June 2025; Received in revised form 20 September 2025; Accepted 1 November 2025
Available online 8 November 2025

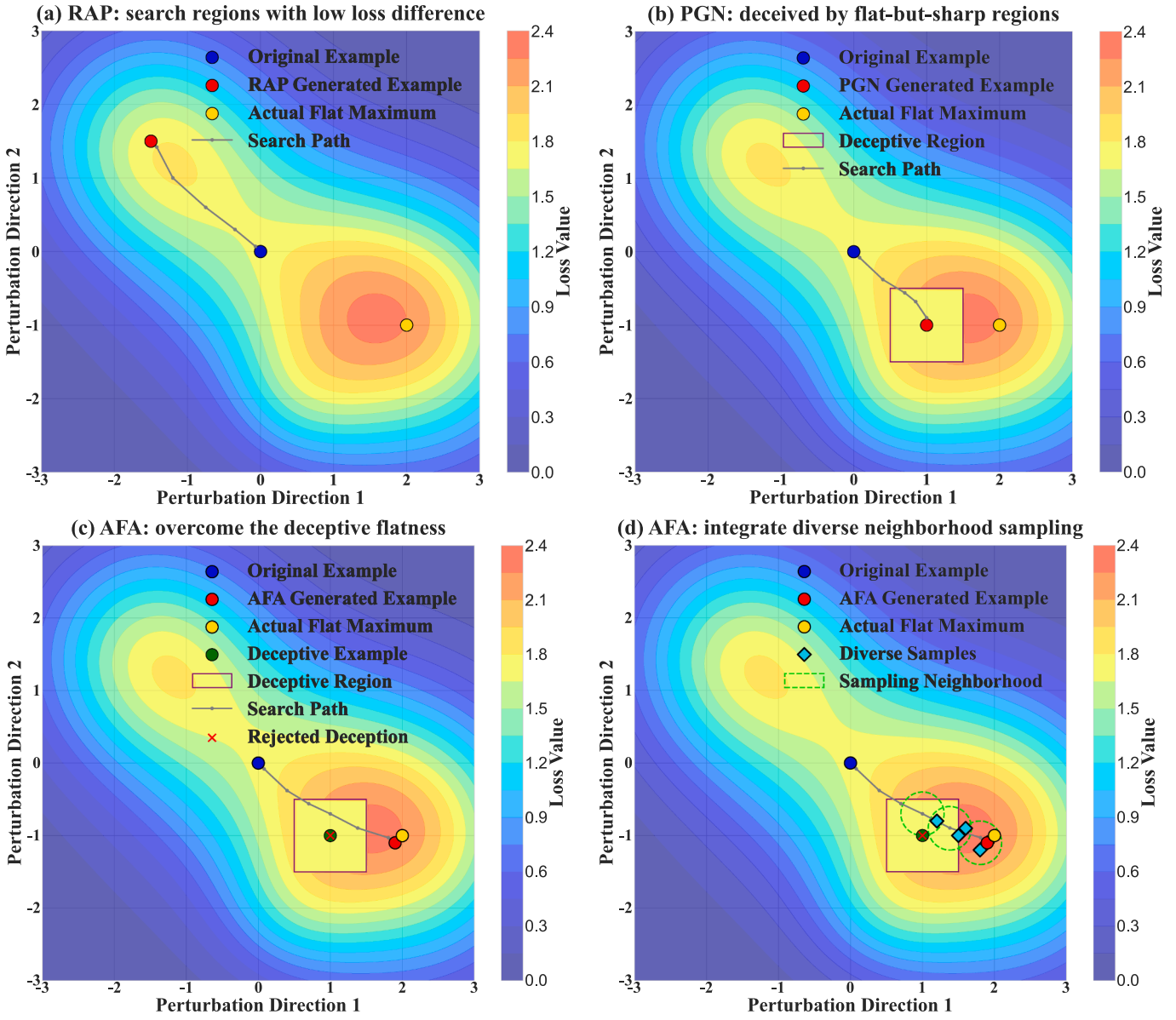

Fig. 1. An example showing: (a)(b) the limitations of single-order flatness methods, and (c)(d) the improvements from our dual-order gradient fusion and diversified sampling strategy.

tackles the changed gradient sign that may occur during the iterative process. Additionally, to further strengthen our attack method, we also design MonteCarlo Adversarial Sampling (MCAS) by diversifying the inner sampling. Finally, extensive experiments on ImageNet-compatible dataset against six representative baselines demonstrate the superiority of the proposed method in various model settings. Additionally, the validation on the Baidu Cloud API also verifies our performance. In particular, following the integration of various input transformation attacks, our approach not only significantly enhances their effectiveness but also surpasses other baseline methods.

Formally, our main contributions are summarized as follows:

- Motivated by the issues of deceptive flatness in adversarial attacks, we first propose Adversarial Flatness (AF) as a feasible solution and provide a theoretical demonstration of its assurance for adversarial transferability.
- To reduce the complexity of implementation, we employ an efficient approximation of the attack objective and instantiate our method as Adversarial Flatness Attack (AFA), meanwhile solving the problem of the altered gradient sign.
- To improve the inner-loop sampling efficiency and further enhance adversarial transferability, we design MonteCarlo Adversarial Sampling (MCAS) to augment the variety of inner sampling.
- The extensive experiments on ImageNet-compatible dataset exhibit our superior performance compared to six representative baselines. Then, the results on the Baidu Cloud API also validate the effectiveness of our method. Additionally, together with integrating various input transformation attacks, our method not only boosts their effectiveness significantly but also surpasses other baselines.

The remaining sections of the paper are structured as follows. Section 2 provides background information and discusses related works. The proposed method is presented in Section 3. In Section 4, the evaluation results demonstrate that our method surpasses the competitors. Finally, conclusions and discussions are drawn in Section 5.

2. Related work

2.1. Preliminary

The objective of adversarial attacks is to generate an adversarial example, leading to misclassifications of the target model. Given an input example x, any model $\boldsymbol{M}$, true label y^{gt}, loss function $\mathcal{L}$ (i.e., Cross-Entropy Loss) and adversarial loss $\mathcal{L}^{adv}$, the adversarial example x^{adv} in white-box untargeted setting can be expressed as:

$$\begin{aligned} x_{adv} &= x + \min_{\|\sigma\|_p \le \eta} \left(\mathcal{L}^{adv}(\boldsymbol{M}(x+\sigma), y^{gt})\right) \\ &= x + \min_{\|\sigma\|_p \le \eta} \left(-\mathcal{L}(\boldsymbol{M}(x+\sigma), y^{gt})\right), \end{aligned} \tag{1}$$

where σ is the adversarial perturbation, $\|.\|_p$ calculates p norm and η means perturbation constraint. It should be noted that our work focuses on the untargeted transferable adversarial attack. Additionally, adversarial attacks can be classified into digital and physical attacks based on their mode of implementation. Physical attacks have advanced in certain critical areas, such as face recognition. For example, a recent method [15] projects a light adversarial mask onto the face to deceive the recognition system. However, this paper focuses on digital attacks.

2.2. Gradient-based attacks

In the early stage, Goodfellow et al. [16] propose Fast Gradient Sign Method (FGSM) by adding perturbation along the ascent direction of loss gradient $\bigtriangledown_x \mathcal{L}(\boldsymbol{M}(x), y^{gt})$ and generating adversarial examples as follows:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\bigtriangledown_x \mathcal{L}(\boldsymbol{M}(x), y^{gt})). \tag{2}$$

By applying the concept of Stochastic Gradient Descent (SGD) with momentum, Dong et al. [11] propose Momentum Iterative Fast Gradient Sign Method (MI-FGSM) to deviate from poorer local optima and improve the adversarial transferability. Then, Wang et al. [17] reduce gradient variance between the adversarial example's gradient and the average gradient of its neighbors from the previous iteration, which is Variance-tuning Momentum Iterative Fast Gradient Sign Method (VMI-FGSM). Wang et al. [18] introduce the Global-momentum-Initialization Momentum Iterative method (GI-MI), providing global momentum knowledge to mitigate gradient elimination. Wang et al. [19] design the Frequency-Guided Sample Relevance Attack (FGSRA) to avoid the sharpness of decision boundaries in model-sensitive regions. Ren et al. [20] propose the Multiple Monotonic Diversified Integrated Gradients (MuMoDIG) attack by focusing on the integration path of Integral Gradient and refining this path. In addition, inspired by flat local domain generalization, Qin et al. [12] introduce Reverse Adversarial Perturbation (RAP), which first minimizes $\mathcal{L}$ near x_{adv} before maximizing $\mathcal{L}$, equivalent to minimizing the maximum difference of $\mathcal{L}^{adv}$ on x_{adv} and its vicinity. Similarly, Qiu et al. [13] propose Neighborhood Conditional Sampling (NCS), to achieve the maximization of $\mathcal{L}(x_{adv})$ and the minimization of $\mathcal{L}$ on any neighborhood of x_{adv}. Differently, Ge et al. [14] devise Penalize Gradient Norm (PGN) to limit the gradient norm of $\mathcal{L}$ and aim to identify the plateau local maxima.

Currently, there is no solution to the issue of deceptive flatness, nor is there any theoretical evidence concerning the effect of such a solution on adversarial transferability. Our approach not only tackles this problem but also proposes a novel gradient-based black-box attack method that incorporates dual-order flatness optimization.

2.3. Input transformation-based attacks

Xie et al. [6] present Diverse Input Method (DIM), which involves applying random resizing and padding to inputs in every iteration, improving the transferable attack. Dong et al. [8] convolve the gradients of the original images with a fixed Gaussian kernel to approximate the average gradient of a set of translated images, namely Translation-Invariant Method (TIM). Inspired by the scale-invariant property in DNNs, Lin et al. [7] propose Scale-Invariant Method (SIM), computing the mean gradient across images scaled by $1/2^i$. Building on Mixup [21], Wang et al. [9] develop Admix, which modifies images by blending the initial input with small segments of images from other classes. Long et al. [10] design Spectrum Simulation Attack (SSA) to enhance transferability by altering the input image in the frequency domain. Similarly, Qian et al. [22] present Mixed-Frequency Inputs (MFI) to obtain a more stable gradient direction by aggregating the high-frequency components.

Although input transformation-based attacks help cut down on computational expenses and enhance transferability, their effectiveness is still constrained and frequently insufficient. Indeed, the input transformation-based attacks above can be combined with current gradient-based attacks. As a result, they can improve transferability and act as a strong basis for assessing the efficacy of our suggested approach.

2.4. Adversarial defense strategies

To mitigate the harmful effects of adversarial examples and improve the robustness of the model, numerous studies have developed various defense strategies. Tramèr et al. [23] enhance the robustness of the model by using adversarial examples generated from ensemble models. Zhou et al. [24] propose a class fairness-based adversarial training method, which uses the most misleading category as feedback to construct targeted adversarial training data. Wu et al. [25] devise Adversarial Weight Perturbation (AWP) to explicitly regularize the flatness of weight loss landscape, forming a double-perturbation mechanism in the adversarial training framework. Random Smoothing (RS), as proposed by Cohen et al. [26], treats adversarial perturbations as regular noise and enhances robustness by incorporating random elements. To reduce error amplification in traditional denoisers, Liao et al. [27] construct High-level-representation Guided Denoiser (HGD), which utilizes the output-layer loss between the original and adversarial images. Xie et al. [28] design a network for Feature Denoising (FD) to suppress features in semantically irrelevant regions. Naseer et al. [29] present Neural Representation Purifier (NRP) in a self-supervised fashion to eliminate adversarial perturbation. Jia et al. [30] introduce a compression-reconstruction network that removes adversarial perturbations but preserves nearby pixel correlations. Guo et al. [31] employ JPEG compression on input images to remove adversarial perturbations before classification. Xu et al. [32] propose feature squeezing through pixel color Bit-depth Reduction (Bit-Red) to detect adversarial examples. Xie et al. [33] suggest using random resizing and random padding (R&P) to reduce the effects of adversarial examples. NIPS-r3 is similar to NRP [29]. Zhou et al. [34] defend against adversarial examples by modeling adversarial noise based on the learned transition relationship between adversarial and true labels.

Usually, assessing the attack capability through the lens of defense strategies can be seen as a more reliable gauge from the viewpoint of opponents, suggesting a higher level of difficulty. Hence, in order to thoroughly showcase the efficacy of our method, we will undertake a comparative analysis between our approach and other baseline methods under the defense strategies mentioned earlier.

2.5. Flat local domain generalization

Currently, domain generalization has been a more challenging problem than domain adaptation. Among the recent works, Foret et al. [35] present Sharpness-Aware Minimization (SAM) to improve model generalization by simultaneously minimizing the loss and sharpness, and SAM can seek parameters in regions that consistently yield low loss values. Zhang et al. leverage first-order optimization [36] or combine them with zeroth-order techniques [37] to enhance the domain generalization capability.

Notably, domain generalization and transferable attacks are two separate tasks. As discussed in [7,17], adversarial transferability can be

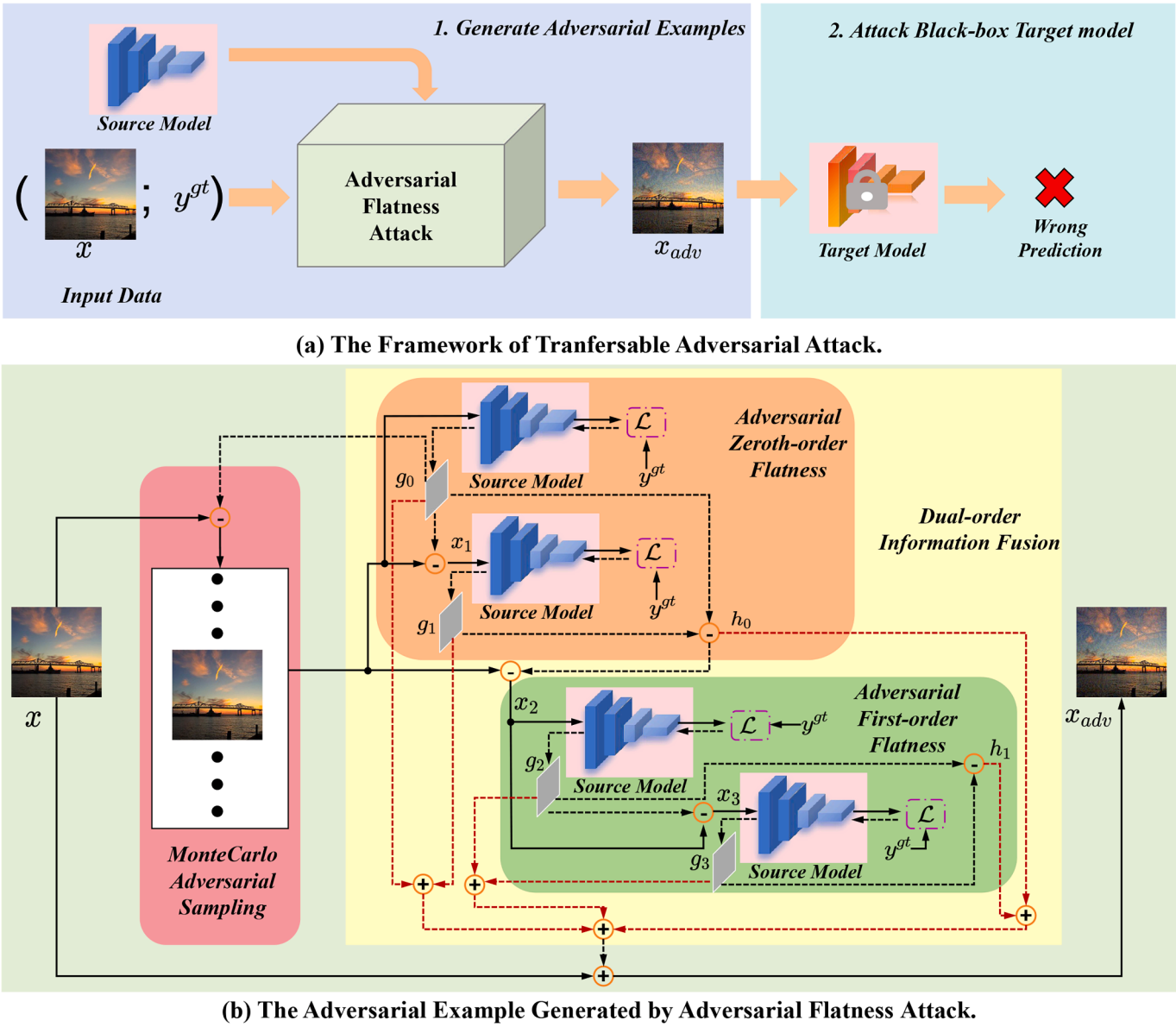

Fig. 2. (a) The process of the transferable adversarial attack. (b) The overview of generating the adversarial example by our proposed Adversarial Flatness Attack, corresponding to stage 1 in (a). In (b), our attack method consists of MonteCarlo Adversarial Sampling and Dual-order Information Fusion. The dashed red line denotes information fusion flow. The dashed black line indicates the backward gradient from the white-box source model. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

analogized to model generalization, where the optimization of perturbations is likened to model training, and the implementation of transferable attacks on other black-box target models corresponds to testing procedures. Thus, our work can be naturally inspired by domain generalization.

3. Methodology

In this section, we first discuss the motivation of deceptive flatness, then introduce our dual-order solution with theoretical guarantees. We efficiently approximate the objective function and propose the Adversarial Flatness Attack (AFA) for black-box attacks. Finally, we present Monte Carlo Adversarial Sampling (MCAS) to further improve transferability. An overview is provided in Fig. 2.

3.1. Motivation

Inspired by domain generalization [35–37], the recent works such as RAP [12] and PGN [14] are representative works that are respectively built upon low-loss differences and stable gradients around adversarial examples. It is worth noting that NCS [13] bears resemblance to RAP [12]. Subsequently, we formally elaborate on the mechanisms underlying [12,14] as follows:

Definition 1. *(Adversarial Zeroth-order Flatness). For any radius $\xi > 0$, the adversarial zeroth-order flatness $\Psi_0(x_{adv})$ of $\mathcal{L}_{src}^{adv}$ at a point x_{adv} is formulated as*

$$\Psi_0(x_{adv}) = \max_{x'_{adv} \in \mathcal{B}_\xi(x_{adv})} \left(\mathcal{L}_{src}^{adv}(x'_{adv}) - \mathcal{L}_{src}^{adv}(x_{adv})\right), x_{adv} \in \mathcal{B}_\epsilon(x), \quad (3)$$

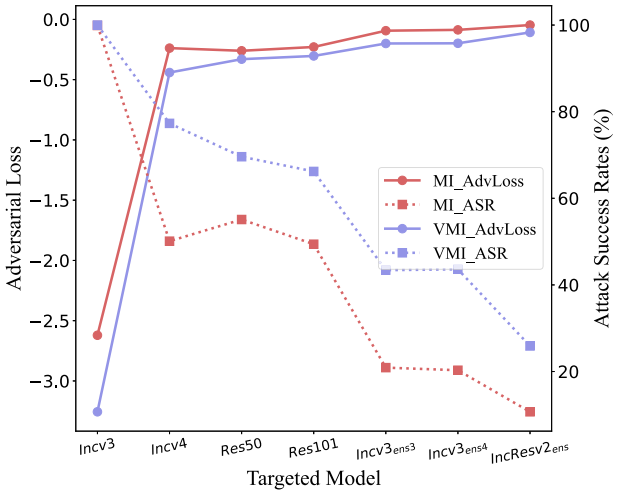

Fig. 3. The attack success rates (%) and adversarial loss of MI and VMI on seven black-box models. The adversarial examples are generated on Inc-v3. The adversarial losses of MI-FGSM and VMI-FGSM are individually represented as MI_AdvLoss and VMI_AdvLoss. The attack success rates of MI-FGSM and VMI-FGSM are denoted as MI_ASR and VMI_ASR respectively.

where x is the original example of adversarial example x_{adv}, ϵ is the upper bound of the magnitude of perturbation on x, $\mathcal{L}_{src}^{adv}$ is adversarial loss of x_{adv} on source model $\boldsymbol{M}_{src}$, and ξ represents the upper bound of uniform sampling centered at adversarial example x_{adv}. $\mathcal{B}_\epsilon(x)$ indicates the neighborhood on point x at radius of ϵ. Similarly, $\mathcal{B}_\xi(x_{adv})$ indicates the neighborhood on point x_{adv} at radius of ξ.

Definition 2. *(Adversarial First-order Flatness). For any radius $\xi > 0$, the adversarial first-order flatness $\Psi_1(x_{adv})$ of $\mathcal{L}_{src}^{adv}$ at a point x_{adv} is formulated as*

$$\Psi_1(x_{adv}) = \xi \cdot \max_{x'_{adv} \in \mathcal{B}_\xi(x_{adv})} \left\| \nabla \mathcal{L}_{src}^{adv}(x'_{adv}) \right\|_2, x_{adv} \in \mathcal{B}_\epsilon(x), \tag{4}$$

where $\nabla \mathcal{L}_{src}^{adv}(x'_{adv})$ is the gradient of $\mathcal{L}_{src}^{adv}(x'_{adv})$ w.r.t. x'_{adv}.

For simplified expression, we shorten Adversarial Zeroth-order Flatness and Adversarial First-order Flatness as AZF and AFF respectively.

Combined with the above definitions, we further analyze the limitations of [12,14]. On the one hand, RAP [12] assumes that an adversarial example will have a strong transferability if the loss difference within its vicinity is sufficiently small, falling under the category of AZF in Definition 1. However, as shown in Fig. 1 (a), this method is prone to becoming trapped in a local suboptimal region if the search steps or radius are excessively small. On the other hand, PGN [14] addresses this issue by generating adversarial examples in areas of the loss function with consistent gradients, known as AFF in Definition 2. Whereas, if there is a flat-yet-sharp area, a localized flat loss surface surrounding sharp losses near the current adversarial example, the AFF-based method (such as PGN [14]) is prone to being fooled by this region, as illustrated in Fig. 1 (b). Here, we refer to this shortcoming as **deceptive flatness**, leading to overfitting the surrogate model and degradation of adversarial transferability. In other words, we think the gradient flatness, which can get rid of the flat-yet-sharp areas, is important to adversarial transferability across diverse target models.

To this end, this paper focuses on addressing the issue of deceptive flatness at the gradient level and proposes a novel attack method to improve the effectiveness of attacks against other black-box models.

3.2. Dual-order solution

3.2.1. Adversarial flatness

As shown in Definition 1, in searching for an adversarial example in [12], AZF focuses more on the global loss information in its neighborhood. Meanwhile, AFF in Definition 2 highlights the change of loss gradients in the neighborhood of the adversarial example. More precisely, AFF emphasizes the local information to some extent. Naturally, we suppose that fusion of the gradients information from AZF to AFF, can provide a global guidance to AFF, escaping from deceptive flatness, as expected in Fig. 1 (c). Theoretically, these two fused gradient information, namely dual-order information fusion, can also help reduce overfitting of the adversarial examples into the surrogate model caused by deceptive flatness. However, there is currently a lack of literature exploring the dual-order information fusion for addressing deceptive flatness and its role in adversarial transferability.

Although adversarial transferability is analogous to model generalization, there remains a lack of concrete evidence demonstrating the connection between domain generalization and adversarial transferability. Therefore, a rigorous description is first given as follows:

Assumption 1. Given white-box surrogate model $\boldsymbol{M}_{src}$, a series of black-box models $\boldsymbol{M}_{tar1}, \boldsymbol{M}_{tar2}, \ldots, \boldsymbol{M}_{tarn}$, and adversarial loss of x_{adv} on model $\boldsymbol{M}_k$ represented as $\mathcal{L}_k^{adv} = \mathcal{L}_k^{adv}(x_{adv}) = \mathcal{L}^{adv}(x_{adv}, y^{gt}; \boldsymbol{M}_k)$, if $\mathcal{L}_{src}^{adv} \le \mathcal{L}_{tar1}^{adv} \le \mathcal{L}_{tar2}^{adv} \le \ldots \le \mathcal{L}_{tarn}^{adv}$, the attack transferability expressed as ASRs across surrogate and target models tends to satisfy the following inequality, i.e., $ASR_{src} \ge ASR_{tar1} \ge ASR_{tar2} \ge \ldots \ge ASR_{tarn}$.

In Fig. 3, when transferring adversarial example x_{adv} generated on white-box Inc-v3 to other black-box targets, the higher ASRs correspond to the lower adversarial loss value. Such a remarkable phenomenon strongly supports Assumption 1. This is the first instance of showing the numerical similarity between untargeted adversarial attacks and domain generalization, with a lower loss value usually indicating improved accuracy. Then, based on this observation and inspired by [37] in domain generalization, we can feasibly propose our dual-order solution to deceptive flatness in our work, called Adversarial Flatness (AF), as below,

$$\Psi^{AF}(x_{adv}) = \beta_f \Psi_0(x_{adv}) + (1 - \beta_f)\Psi_1(x_{adv}), \tag{5}$$

where β_f is the flatness balanced coefficient. After joining the basic adversarial objective as Eq. (1), we derive our main objective as below.

$$\mathcal{L}_{all}^{adv} = \mathcal{L}_{src}^{adv} + \lambda_f \Psi^{AF}(x_{adv}), \tag{6}$$

where λ_f is the flatness item coefficient. Then, we explore the role of AF theoretically in adversarial transferability.

3.2.2. Theoretical guarantees to adversarial transferability

Given any adversarial example x_{adv}, the maximum radius of uniform sampling $\mathcal{U}$ centered at adversarial example ξ and step size α, we suppose that Eq. (6) has a second-order gradient at least. According to Definitions 1 and 2, $\vartheta \sim \mathcal{U}(-\xi, \xi)$, such that the adversarial loss on the surrogate model in the vicinity of x_{adv} and AF can be related as below:

$$\mathcal{L}_{src}^{adv}(x_{adv} + \vartheta) \le \mathcal{L}_{src}^{adv}(x_{adv}) + \Psi^{AF}(x_{adv}). \tag{7}$$

The detailed proof is provided in A.

Finally, after integrating Eq. (7) and the generalization boundary theory in theorem 4.2 [38], we obtain the expanded inequality between AF and adversarial transferability:

$$\begin{aligned} &\mathbb{E}_{x'_{adv} \sim \mathcal{B}_\xi(x_{adv})}[\mathcal{L}_{tar}^{adv}(x'_{adv})] \le \mathcal{L}_{src}^{adv}(x_{adv}) + \Psi^{AF}(x_{adv}) + BD_{upper}, \\ &BD_{upper} = \sup \|\mathcal{L}_{tar}(x^1, x^2) - \mathcal{L}_{src}(x^1, x^2)\|, x^1, x^2 \in \mathcal{B}_\xi(x_{adv}), \end{aligned} \tag{8}$$

where the term $\sup \|\cdot\|$ represents the upper bound of the discrepancy distance [39] between adversarial losses of x_{adv} on the source model $\boldsymbol{M}_{src}$ and the target model $\boldsymbol{M}_{tar}$, mimicking the training domain and the unknown test domain in domain generalization. Similarly, $\mathcal{B}_\xi(x_{adv})$ indicates the neighborhood on point x_{adv} at radius of ξ. The detailed proof is provided in Appendix B. Clearly, the AF item $\Psi^{AF}(x_{adv})$ (the second term on the right side of the inequality) leads to better adversarial transferability on the black-box model $\boldsymbol{M}_{tar}$. To fully demonstrate the rationale of the proposed AF, we also conduct an additional visual experiment at Appendix E.

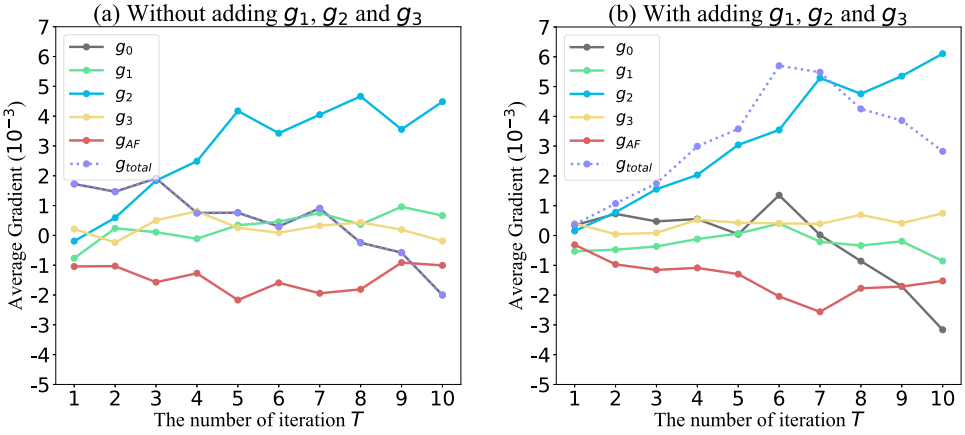

Fig. 4. Illustrations of the average gradients of our approximated objective over T iterations (a) without and (b) with adding g_1, g_2 and g_3. The dashed line representing g_{total} is our focus.

3.3. Adversarial flatness attack

Since our goal is to achieve a transferable untargeted attack, we can clarify our objective in Eq. (6) as follows.

$$\max_{x_{adv}\in B_\epsilon(x)} [\mathcal{L}\left(x_{adv}, y^{gt}; \boldsymbol{M}_{src}\right) - \lambda_f \Psi^{AF}(x_{adv})], \tag{9}$$

where $\Psi^{AF}(x_{adv})$ includes the first-order gradient of ℓ_{src}^{tar}. In addition, $\Psi^{AF}(x_{adv})$ is computationally expensive when directly deriving from the first-order gradient to the second-order gradient in generating the adversarial example x_{adv}. Hence, we approximate the second-order gradient using the first-order gradients. And similar to [14], we also take the search step $\triangle x = \alpha \cdot \frac{-\nabla_x \mathcal{L}(x, y^{gt}; \boldsymbol{M}_{src})}{\|\nabla_x \mathcal{L}(x, y^{gt}; \boldsymbol{M}_{src})\|_1}$ within the neighborhood size ξ, where $\|\cdot\|$ is the L_1 norm. Then the gradient of Eq. (9) is approximated as follows. Several gradient approximation methods exist [40], and we have selected the Finite Difference Method because of its simplicity. Additional experiments comparing our method with those based on other gradient approximation techniques are presented in Appendix F.

3.3.1. The approximation of gradients on $\Psi_0(x_{adv})$ and $\Psi_1(x_{adv})$

Let $x_0 = x_{adv}$, $x_1 = x_0 + \triangle x_0 = x_0 - \alpha \cdot \frac{\nabla_{x_0} \mathcal{L}(x_0, y^{gt}; \boldsymbol{M}_{src})}{\|\nabla_{x_0} \mathcal{L}(x_0, y^{gt}; \boldsymbol{M}_{src})\|_1}$. We can approximate the gradient of $\Psi_0(x_{adv})$ as follows:

$$\begin{aligned}
&\nabla\Psi_0(x_{adv}) = h_0 \approx g_1' - g_0' = -(g_1 - g_0), \text{ where} \\
&g_0' = \nabla_{x_0} \mathcal{L}_{src}^{adv}(x_0) = -g_0 = -\nabla_{x_0} \mathcal{L}(x_0, y^{gt}; \boldsymbol{M}_{src}), \\
&g_1' = \nabla_{x_1} \mathcal{L}_{src}^{adv}(x_1) = -g_1 = -\nabla_{x_1} \mathcal{L}(x_1, y^{gt}; \boldsymbol{M}_{src}).
\end{aligned} \tag{10}$$

Here we use the smaller search radius α than ξ, so we reformulate gradient of $\Psi_1(x_{adv})$ as the following procedure:

$$\begin{aligned}
&\nabla\Psi_1(x_{adv}) = h_1 \approx -(g_3 - g_2), \text{where} \\
&g_2 = \nabla_{x_2} \mathcal{L}(x_2, y^{gt}; \boldsymbol{M}_{src}), \\
&g_3 = \nabla_{x_3} \mathcal{L}(x_3, y^{gt}; \boldsymbol{M}_{src}), \\
&x_2 = x_0 - \alpha \cdot \frac{g_1 - g_0}{\|g_1 - g_0\|}, \\
&x_3 = x_2 - \alpha \cdot \frac{g_2}{\|g_2\|}.
\end{aligned} \tag{11}$$

The detailed proof is provided in Appendix C.

Next, after joining Eqs. (10) and (11), our objective in Eq. (9) can be rewritten as:

$$\max_{x_{adv}\in B_\epsilon(x)} [g_0 + \lambda_f(\beta_f(g_1 - g_0) + (1 - \beta_f)(g_3 - g_2))]. \tag{12}$$

The detailed proof is provided in Appendix D.

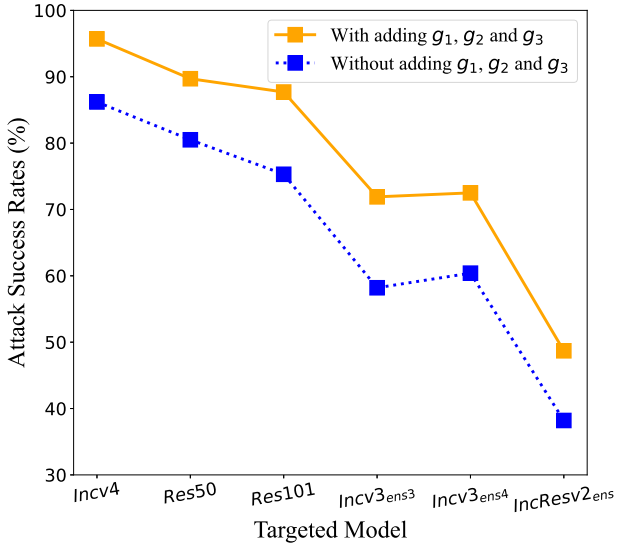

Fig. 5. The attack success rates (%) of our AFA without or with adding g_1, g_2 and g_3 on six black-box models. The adversarial examples are generated on Inc-v3. Incv3$_{ens3}$, Incv3$_{ens4}$ and IncResv2$_{ens}$ are Inception-v3 and Inception-ResNet-v2 using ensemble adversarial training [23].

3.3.2. Problem of altered gradient sign

However, our objective in Eq. (12) represented as g_{total} considers both g_0 and the other gradients of the three neighborhoods (x_1, x_2 and x_3), which means an intricate optimization process. Besides, it may lead to the unexpected direction alteration with $g_{total} < 0$ after $T > 7$, as illustrated in Fig. 4 (a). Since our objective is to find the adversarial example in the direction of gradient accent, this phenomenon can lead to a significant decline in transferable attack effectiveness. Moreover, based on our intuition, we can optimize g_1, g_2 and g_3 along the ascent direction simultaneously, as evidenced in Fig. 4(b). Formally, we update Eq. (12) explicitly for our objective as follows:

$$\max_{x_{adv}\in B_\epsilon(x)} [g_0 + \lambda_f(\beta_f(g_1 - g_0) + (1 - \beta_f)(g_3 - g_2)) + g_1 + g_2 + g_3]. \tag{13}$$

In reality, it can be easily integrated with iterative attacks, aligned with existing works [12–14,17]. Here, based on Eq. (13) and MI-FGSM [11], we formulate our Adversarial Flatness Attack (AFA). Hence, Algorithm 1 contains all the specifics of our AFA. This subsection mainly reflects on the rows except for the fourth, seventh and sixteenth.

In Fig. 5, it is evident that our AFA based on the enhanced objective in Eq. (13) can achieve much better attack performance, compared to

that based only on Eq. (12). It once confirms our assumption that the altered gradient sign of our objective can reduce the effectiveness of our attack, especially when the optimization process includes different gradients from multiple neighborhood examples. By explicitly maximizing the gradients in the vicinity of adversarial examples, the issue of the objective's gradient sign switching can be effectively reduced, preventing the decay of attack ability. Also, it can be shown in Fig. 2(b).

Algorithm 1 Adversarial flatness attack.

Input: A clean image x with ground-truth label y^{gt}; the loss function $\mathcal{L}$ with source model $\boldsymbol{M}_{src}$; the magnitude of perturbation ϵ; the maximum iterations number T; the momentum decay η; the sampling number N; the upper bound of neighborhood sampling ξ; the radius of MonteCarlo Adversarial Sampling γ_{MCAS}; the momentum decay of MonteCarlo Adversarial Sampling η_{MCAS}; the flatness balanced coefficient β_f; the flatness item coefficient λ_f.

Output: An adversarial example x_{adv}.

$m_0 = 0$, $x^0_{adv} = x$, $\alpha = \frac{\epsilon}{T}$
for $t \leftarrow 0$ **to** T-1 **do**
  $\overline{g} = 0$
  $\overline{g_s} = 0$
  for $i \leftarrow 0$ **to** N-1 **do**
    Randomly sample $x' = x + \vartheta$, $\vartheta \sim \mathcal{U}(-\xi, \xi)$
    MonteCarlo Adversarial Sampling $x_0 = x' - \gamma_{MCAS} \cdot \text{sign}(\overline{g_s})$
    Calculate the gradient $g_0 = \bigtriangledown_{x_0} \mathcal{L}(x_0)$
    Update sample $x_1 = x_0 - \alpha \cdot \frac{g_0}{\|g_0\|_1}$
    Calculate the gradient $g_1 = \bigtriangledown_{x_1} \mathcal{L}(x_1)$
    Update sample $x_2 = x_0 - \alpha \cdot \frac{g_1 - g_0}{\|g_1 - g_0\|_1}$
    Calculate the gradient $g_2 = \bigtriangledown_{x_2} \mathcal{L}(x_2)$
    Update sample $x_3 = x_2 - \alpha \cdot \frac{g_2}{\|g_2\|_1}$
    Calculate the gradient $g_3 = \bigtriangledown_{x_3} \mathcal{L}(x_3)$
    Accumulate the gradient $\overline{g} += \frac{g_0 + \lambda_f(\beta_f(g_1 - g_0) + (1-\beta_f)(g_3 - g_2)) + g_1 + g_2 + g_3}{N}$
    Update MCAS momentum $\overline{g_s} = \eta_{MCAS} \cdot \overline{g_s} - g_0$
  end
  $m_{g+1} = \eta \cdot m_t + \frac{\overline{g}}{\|\overline{g}\|_1}$
  $x^{t+1}_{adv} = \Pi_{\mathcal{B}_\epsilon(x)}[x^t_{adv} + \alpha \cdot \text{sign}(m_{g+1})]$
end
$x_{adv} = x^T_{adv}$

3.4. MonteCarlo adversarial sampling

As mentioned before, the sampling technique for the attack effectiveness is crucial. Naturally, we expect it to efficiently enhance our AFA, as shown in Fig. 1(d)H.1(c). Nevertheless, the above works largely overlook this aspect, where they merely use a uniform sampling around x^t_{adv}.

Recently, Qiu et. al[13]. have for the first time explained that diversifying the surrounding points centered on x_{adv} can help perturbation optimization avoid suboptimal outcomes. They suggest utilizing the gradient information of the previous two sampling processes. Rather than concentrating on increasing diversity in sampling, we prioritize fixing its two weaknesses. Increasing the iteration number T and the inner sampling number N will require more hardware memory and computational resources, which is not feasible. However, conditional sampling only avoids repeating the same sampling as the previous two iterations, without ensuring dissimilarity with the previous ones.

Based on MonteCarlo estimation in reinforcement learning [41], we introduce MonteCarlo Adversarial Sampling (MCAS) to broaden the surroundings of x^t_{adv}, which strategically eliminates the most recent sampling information (gradients) in a momentum manner, as shown in Fig. 2

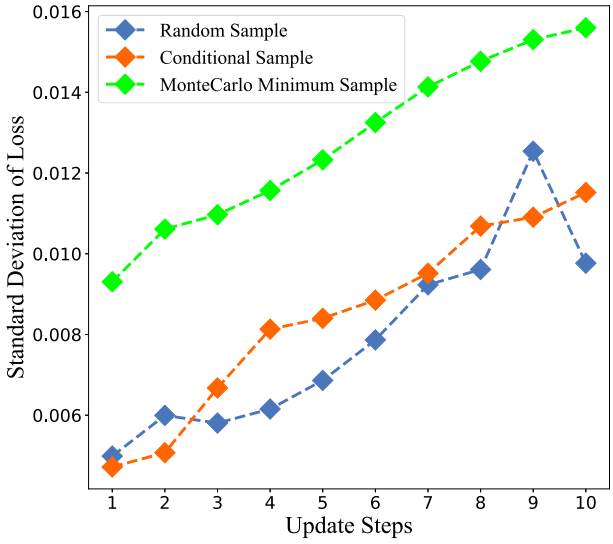

Fig. 6. Comparison of average loss standard deviation for every iteration t. The update steps are equivalent to the maximum iteration $T = 10$.

(b). Mathematically, the proposed MCAS is formulated as:

$$x_0 = x^t_{adv} - \gamma_{MCAS} \cdot \text{sign}(\overline{g_s}), \text{where } \overline{g_s} = \eta_{MCAS} \cdot \overline{g_s} - \bigtriangledown_{x_0} \mathcal{L}(x_0), \tag{14}$$

where $\overline{g_s} = 0$ when $t = 0$. Besides, γ_{MCAS} and η_{MCAS} indicate the radius and the momentum decay of MCAS, respectively. To showcase the diversity of our MCAS compared to conditional sampling and random sampling, we calculate the average loss standard deviation for each iteration t. To initially validate the effectiveness of MCAS, we conduct a toy experiment, where random sampling, conditional sampling and our MCAS are individually equipped with MI-FGSM [11]. As shown in Fig. 6, among ten update steps, the toy example demonstrates that our MCAS can achieve a higher average standard deviation of loss. It indicates our method can produce more diverse samples in the vicinity of x^t_{adv}. Additionally, when generating an adversarial example, our MCAS requires 412 MB less memory than NCS and has an inference time 0.4s shorter than NCS. Furthermore, a more detailed discussion of our method is provided in Appendix M. Finally, our proposed MCAS is allocated to the fourth, seventh and sixteenth rows as specified in Algorithm 1.

4. Experiments

Datasets. Our validation is conducted on ImageNet-compatible dataset from the NIPS 2017 adversarial competition, commonly used in earlier studies. It includes 1000 images sized $299 \times 299 \times 3$, along with ground-truth labels and target labels for targeted attacks. It notes that our AFA focuses on transferable untargeted attacks, disregarding the target label in the dataset.

Models. To showcase the efficacy of our proposed methods, we evaluate the attack performance on both normally trained models and robust models. For normally trained models, we use seven classical Convolutional Neural Networks (CNNs), including Inception-v3 (Inc-v3), Inception-v4 (Inc-v4), Inception-ResNet-v2 (IncRes-v2), ResNet-50 (Res-50), ResNet-101 (Res-101), DenseNet-121 (Dense-121) and VGG-19bn (VGG-19) [42]. Aside from the CNNs above, we also test on models with more diverse architectures: MobileNet-v2 (MobileNet) [42], PASNet-5-Large (PASNet-L) [43], ViT-Based/16 (ViT-B/16), PiT-S, Swin-T, MLP-mixer and ResMLP [44]. For robust models, we employ methods such as adversarial training, random smoothing, active defense and more. Precisely, we take Inc-v3*, Inc-v3$_{ens}$, Inc-v4$_{ens}$ and IncRes-v2$_{ens}$ [23] as adversarially trained models. Other defense strategies under attack

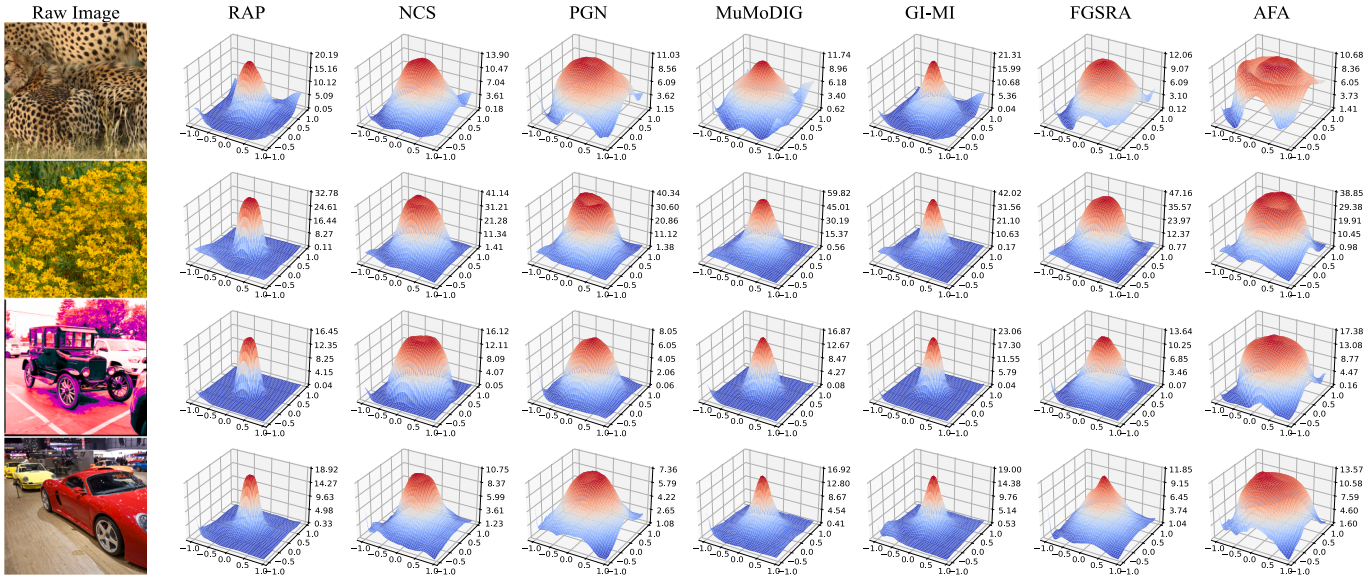

Fig. 7. Visualization of loss surfaces along two random directions for two randomly sampled adversarial examples on the surrogate Inc-v3. The center of each 2D graph corresponds to the adversarial example generated by different attack methods. The x and y axes represent the random noises added to x_{adv} twice in succession. The z axis indicates the loss value.

contain RS [26], HGD [27], FD [28], NRP [29], ComDefend [30], JPEG [31], Bit-Red [32], R&P [33] and NIPS-r3 [29].

Baselines. Our baselines consist of six gradient-based iterative adversarial attacks, including GI-MI [18], FGSRA [19], MuMoDIG [20], RAP [12], PGN [14] and NCS [13]. Additionally, we test the effectiveness of our AFA by incorporating various input transformations such as DIM [6], TIM [8], SIM [7], Admix [9], and SSA [10].

Implementation details. Our method is applied using PyTorch on a NVIDIA GeForce GTX 3090 GPUs. We typically choose a maximum perturbation as $\epsilon = 16.0/255$ with iterations $T = 10$, a step size of $\alpha = \epsilon/T$ and a momentum decay of $\eta = 1.0$. For GI-MI, we set the number of sampled examples $N = 20$ and the pre-attack epochs $P = 5$. For RAP, we set the step size $\alpha = 2.0/255$, the number of iterations $K = 400$, the inner iteration number $T = 10$, the late-start $K_{LS} = 100$, the size of neighborhoods $\epsilon_n = 16.0/255$. It should be noted that RAP in this paper is equipped with momentum decay for a fair comparison. For PGN, we set the number of sampling $N = 20$, the balanced coefficient $\delta = 0.5$, and the upper bound of $\xi = 3.0 \times \epsilon$. For NCS, we set the upper bound of neighborhood sampling $\xi = 2 \times \epsilon$, the number of sampling $N = 20$, the upper bound of sub-regions $\gamma = 0.15 \times \epsilon$, and the balanced coefficient $\lambda = \alpha/T$. For MuMoDIG, the epoch T and the inner interpolation times N_i are set to 10 and 20, respectively, while the transformation number N_{trans} is set to 0 to ensure a fair comparison. It should be noted that the baselines are initially excluded from all integrated transformation methods, in accordance with our approach, especially for FGSRA and MuMoDIG. For our proposed AFA, we set the maximum radius of uniform sampling centered at adversarial example $\xi = 3 \times \epsilon$, the number of sampling $N = 20$, the radius of MonteCarlo Adversarial Sampling $\gamma_{MCAS} = 0.15 \times \epsilon$, the momentum decay of MonteCarlo Adversarial Sampling $\eta_{MCAS} = 0.9$, the flatness balanced coefficient $\beta_f = 0.5$ and the flatness item coefficient $\lambda_f = \alpha \times \beta_f$.

Evaluation metrics. Given that our proposed AFA targets transferable attacks, we use the fool rate to measure the success of the attacks. More specifically, the fooling rate indicates the proportion of adversarial examples that successfully deceive the target model compared to all the adversarial examples created.

Table 1
The untargeted attack success rates of various gradient-based attacks in the normal model setting. * indicates the results on the white-box model. **BOLD** indicates the best.

Model	Attack	Inc-v3	Inc-v4	Res-50	Res-101	InRes-v2	VGG-19	Dense-121
Inc-v3	RAP	99.90*	80.60	84.50	76.70	80.10	82.30	69.50
	NCS	**100.0***	82.50	77.80	73.80	79.80	81.00	77.60
	PGN	**100.0***	90.50	85.60	81.60	89.60	87.20	87.30
	MuMoDIG	**100.0***	78.30	74.00	71.80	76.10	76.10	74.50
	GI-MI	**100.0***	66.40	67.50	59.70	64.30	67.60	56.30
	FGSRA	**100.0***	87.00	80.70	77.20	85.90	84.00	82.80
	AFA	**100.0***	**95.20**	**89.90**	**87.70**	**95.40**	**91.50**	**91.90**
Inc-v4	RAP	78.00	99.80*	80.90	74.80	70.60	80.80	63.50
	NCS	86.20	99.50*	78.40	75.40	81.80	84.50	78.60
	PGN	91.50	99.40*	85.80	82.10	87.80	90.40	87.00
	MuMoDIG	85.30	99.90*	76.90	74.30	81.00	81.70	77.00
	GI-MI	71.40	**100.0***	67.00	60.80	60.00	72.50	56.90
	FGSRA	88.40	99.30*	82.80	79.50	85.20	86.00	83.50
	AFA	**95.90**	**100.0***	**91.80**	**89.80**	**93.70**	**95.70**	**92.30**
Res-50	RAP	73.70	87.00	99.90*	97.90	62.90	87.90	76.70
	NCS	84.20	82.70	**100.0***	99.50	76.90	94.60	91.90
	PGN	85.20	83.20	**100.0***	99.20	78.10	95.00	92.90
	MuMoDIG	79.20	74.60	**100.0***	99.00	65.40	90.20	82.70
	GI-MI	60.60	55.10	**100.0***	99.00	45.10	81.30	65.80
	FGSRA	87.00	83.60	**100.0***	99.40	78.20	95.00	93.10
	AFA	**89.50**	**88.20**	**100.0***	**99.90**	**82.60**	**97.90**	**95.10**
Res-101	RAP	77.00	69.50	98.80	**100.0***	64.10	87.70	79.10
	NCS	85.10	82.60	99.40	**100.0***	77.20	93.80	91.40
	PGN	87.30	83.00	99.50	**100.0***	80.00	94.90	93.70
	MuMoDIG	80.20	74.90	99.60	**100.0***	67.90	89.20	85.00
	GI-MI	63.80	56.20	99.60	**100.0***	47.30	78.00	68.20
	FGSRA	86.80	83.70	99.40	**100.0***	79.60	94.90	93.50
	AFA	**89.40**	**87.60**	**99.80**	**100.0***	**83.80**	**96.20**	**95.80**

4.1. Visualization of loss surfaces for adversarial example

To validate the efficacy of our proposed AFA in identifying adversarial examples within a flat maxima region, we compare the loss surface maps of adversarial examples generated by different baselines on the

Table 2
The untargeted attack success rates of various gradient-based attacks in the diverse model architectures setting. **BOLD** indicates the best.

Model	Attack	MobilNet	PASNet-L	ViT-B/16	PiT-S	MLP-mixer	ResMLP
Inc-v3	RAP	89.10	71.70	23.70	26.60	50.30	43.30
	NCS	79.20	73.30	39.40	39.10	56.30	54.10
	PGN	85.90	84.00	50.70	51.20	63.30	66.30
	MuMoDIG	79.00	70.60	30.70	30.90	53.80	47.40
	GI-MI	76.50	58.40	20.40	22.10	46.30	35.60
	FGSRA	82.90	78.90	44.80	47.00	60.10	60.30
	AFA	**91.10**	**87.40**	**56.70**	**59.20**	**65.30**	**72.30**
Inc-v4	RAP	89.90	69.70	22.50	25.20	48.70	36.20
	NCS	80.10	79.30	43.10	50.00	57.60	55.70
	PGN	86.30	88.30	54.70	54.90	62.30	66.00
	MuMoDIG	82.00	77.50	31.80	38.30	51.50	47.90
	GI-MI	76.30	64.30	20.80	23.80	44.10	33.10
	FGSRA	82.90	83.30	50.00	54.70	59.50	61.10
	AFA	**92.20**	**92.00**	**62.80**	**68.20**	**67.00**	**74.10**
Res-50	RAP	95.20	70.70	25.20	31.80	49.20	43.70
	NCS	97.00	84.30	40.90	45.20	54.90	59.20
	PGN	96.80	88.70	47.70	45.10	62.20	65.20
	MuMoDIG	96.20	73.40	21.90	29.10	44.80	39.60
	GI-MI	92.50	59.90	17.20	19.20	41.70	31.10
	FGSRA	97.50	88.00	45.00	45.60	59.70	65.20
	AFA	**98.20**	**91.10**	**48.70**	**49.90**	**62.30**	**68.70**
Res-101	RAP	96.00	74.00	28.30	31.70	52.60	47.30
	NCS	96.20	83.60	44.10	46.00	57.00	61.40
	PGN	97.00	88.30	50.80	48.30	62.20	68.40
	MuMoDIG	95.40	72.00	26.50	30.40	47.30	42.50
	GI-MI	90.30	59.70	19.60	20.40	43.40	31.30
	FGSRA	97.40	86.80	49.50	47.70	60.00	66.90
	AFA	**98.10**	**89.50**	**55.10**	**52.40**	**64.30**	**70.30**

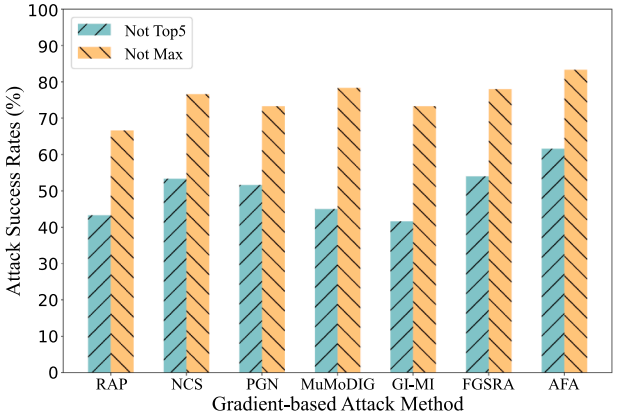

Fig. 8. Comparison of the average untargeted attack success rates between various gradient-based attacks on the Baidu Cloud API. The white-box source model is Inc-v3.

surrogate Inc-v3 model. Each 2D graph corresponds to an adversarial example, with the adversarial example located in the middle. Four images are randomly selected from the dataset and their loss surfaces are compared in Fig. 7, with each row displaying the visualization of one image across all baselines. As shown in Fig. 7, it is clear that our AFA method helps adversarial examples reach flatter peaks compared to all baseline methods. It demonstrates that our approach is able to generate adversarial examples within the smooth maximum region. AFA demonstrates stronger transferability compared to these baselines due to the presence of adversarial examples in flatter local maxima, a fact further confirmed in subsequent experiments.

4.2. Attack a single model

Under normally trained models. Initially, we compared the ASR of our fully automated aircraft with baselines from seven normally trained models. Specifically, we choose one of four models (such as Inc-v3, Inc-v4, Res-50, and Res-101) to serve as the white-box surrogate model, while the remaining models trained normally (six CNNs and six advanced models with various architectures) are considered the black-box target models. After creating adversarial examples on the substitute model, they can be used to test how well adversarial attacks transfer to the black-box target models. As illustrated in Table 1, our approach consistently achieves better ASR performance compared to other methods. Our AFA clearly shows much better transferable attack performance than all other comparison models. Besides, as shown in Table 2, some different model architectures (especially ViT-B/16, PiT-S and MLP-mixer) might significantly degrade the transferable ASRs compared to CNN-like architectures.

It can be attributed to the shift of attention areas in these models, which has been demonstrated in [8] and can usually be alleviated by the input transformation strategies such as [6–8]. These input transformation strategies typically increase the diversity of the backward gradients and reduce the likelihood that the generated perturbations will overfit to high-frequency information and the current surrogate model. Whereas, our suggested approach consistently outperforms the last four different architectures. In general, our AFA method demonstrates a greater level of versatility in its ability to handle adversarial attacks on different architectures. For the page limitation, the experiments of generating adversarial examples on transformers (ViT-B/16, Swin-T) are also provided at G.

Under models equipped with defense strategies. By using improved robustness obtained from adversarial training, we confirm the effectiveness of our method on four different models equipped with various defense strategies, including adversarial techniques (namely, Inc-v3*, Inc-v3$_{ens}$, Inc-v4$_{ens}$, and IncRes-v2$_{ens}$) and other defense models like random smoothing, active defenses and so on. Moreover, the white-box surrogate models share similarities with normally trained models, whereas these models with defense methods are considered black-box target models. As shown in Table 3, the ASRs of all methods on target models consistently decrease. Nevertheless, our method outperforms others in terms of effectiveness. Our method effectively preserves the high level of attack transferability in the scenario of adversarially trained models. Besides, as illustrated in Table 4, various defense models have different levels of resilience when facing adversarial attacks. The substantial decreases in ASR for RS and NIPS-r3 across all methods are especially notable. However, our suggested method consistently shows impressive deceptive abilities even in the face of strong defenses. For the page limitation, the validation on transformers-based surrogates (ViT-B/16, Swin-T) is also shown at H.

4.3. Transfer to the commercial API

To further demonstrate the broad applicability of our method, we select the Baidu Cloud system as a target. Besides, it is a commercial API capable of recognizing over 100,000 categories and thus exhibits strong representativeness. In practice, we randomly choose 60 images from the datasets and set Inc-v3 as the surrogate. Additionally, we focus not only on the max prediction score, but also on the top five predictions. In other words, we set the successful attack conditions as the commonly used wrong prediction at the maximum prediction score (simplified as “Not Max”), and the more restrictive wrong classification within the top five predictions (simplified as “Not Top5”). As shown in Fig. 8, our method achieves at least 5 % higher ASRs compared to other baselines under both conditions. Besides, we also display two groups of recognition results from the Baidu Cloud system. It is suggested that our method can more successfully deceive the Baidu Cloud system in Fig. 9, while these images with added perturbations can still be classified correctly by human eyes. These results have verified our superiority preliminarily.

Table 3
The untargeted attack success rates of various gradient-based attacks in the adversarially trained model setting. **BOLD** indicates the best.

Attack	Model	Inc-v3*	Inc-v3$_{ens}$	Inc-v4$_{ens}$	InRes-v2$_{ens}$	Model	Inc-v3*	Inc-v3$_{ens}$	Inc-v4$_{ens}$	InRes-v2$_{ens}$
RAP		34.80	17.00	17.10	8.30		31.90	18.20	17.00	7.80
NCS		55.40	53.70	53.60	35.10		53.40	54.50	54.80	40.40
PGN		71.30	64.40	66.50	46.20		64.80	67.30	64.60	48.20
MuMoDIG	Inc-v3	44.50	40.70	39.90	20.60	Inc-v4	41.70	41.70	38.60	23.60
GI-MI		30.60	22.20	20.80	9.80		28.40	20.30	18.90	10.50
FGSRA		67.50	61.50	63.00	43.30		62.50	63.70	63.10	48.00
AFA		**75.70**	**73.00**	**73.20**	**50.90**		**73.20**	**75.10**	**75.30**	**57.10**
RAP		32.80	19.50	17.10	11.10		35.50	20.90	19.30	9.30
NCS		52.00	50.40	48.70	35.50		55.90	54.40	55.40	42.10
PGN		59.90	57.50	57.10	43.50		65.10	63.10	62.50	51.10
MuMoDIG	Res-50	30.40	24.90	23.40	12.50	Res-101	33.70	26.50	26.20	15.40
GI-MI		23.60	17.80	17.10	9.10		24.80	18.10	16.70	9.30
FGSRA		56.40	56.70	55.90	42.30		63.10	60.70	60.20	48.40
AFA		**61.10**	**59.30**	**59.90**	**44.70**		**67.20**	**63.70**	**63.90**	**51.80**

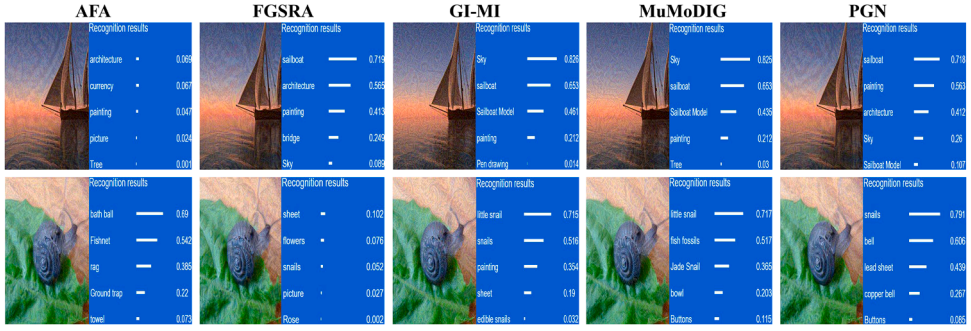

Fig. 9. Samples of the recognition results of adversarial examples generated by ours, FGSRA, GI-MI, MuMoDIG and PGN on the Baidu Cloud API. The white-box source model is Inc-v3.

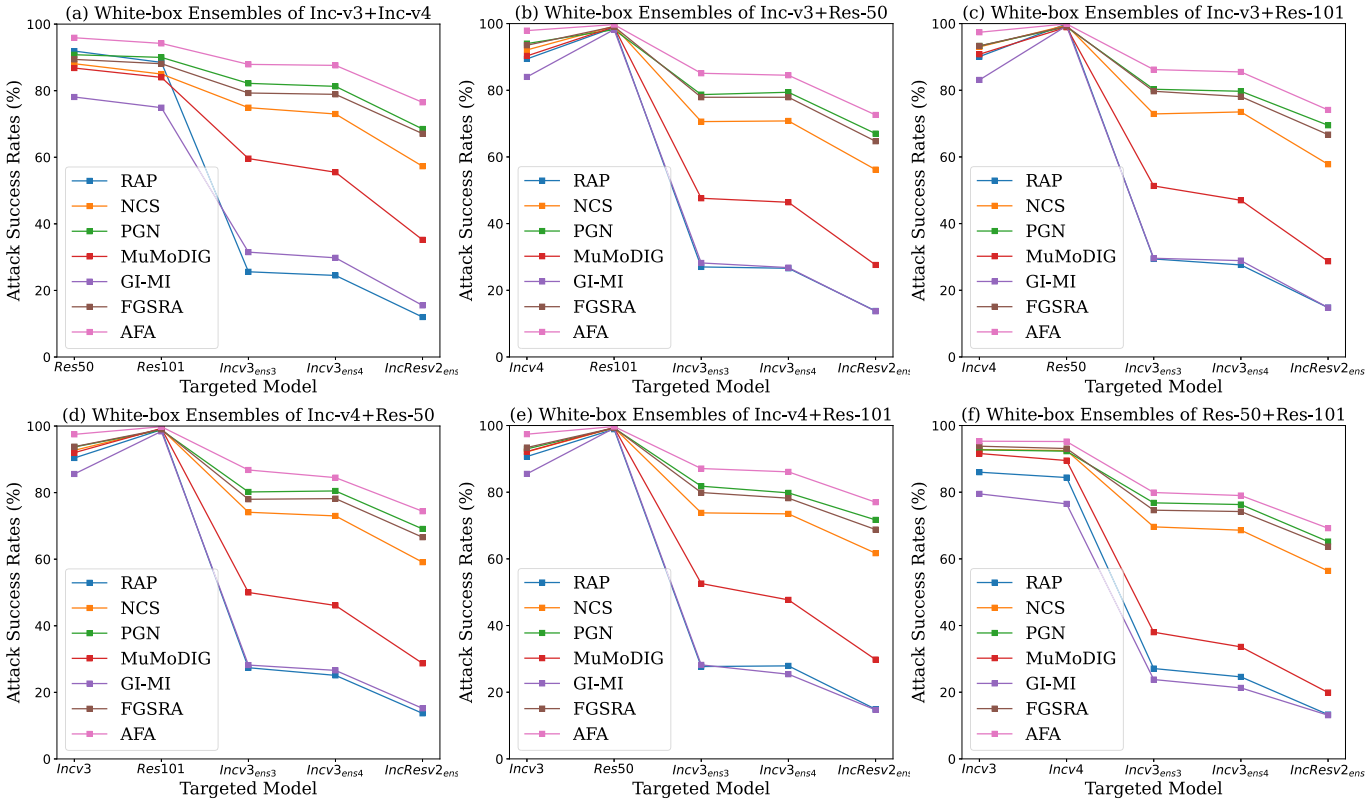

Fig. 10. Untargeted attack success rates between various gradient-based attacks in the ensemble models setting. The white-box source ensemble models are in the title. The black-box target models are on x axis.

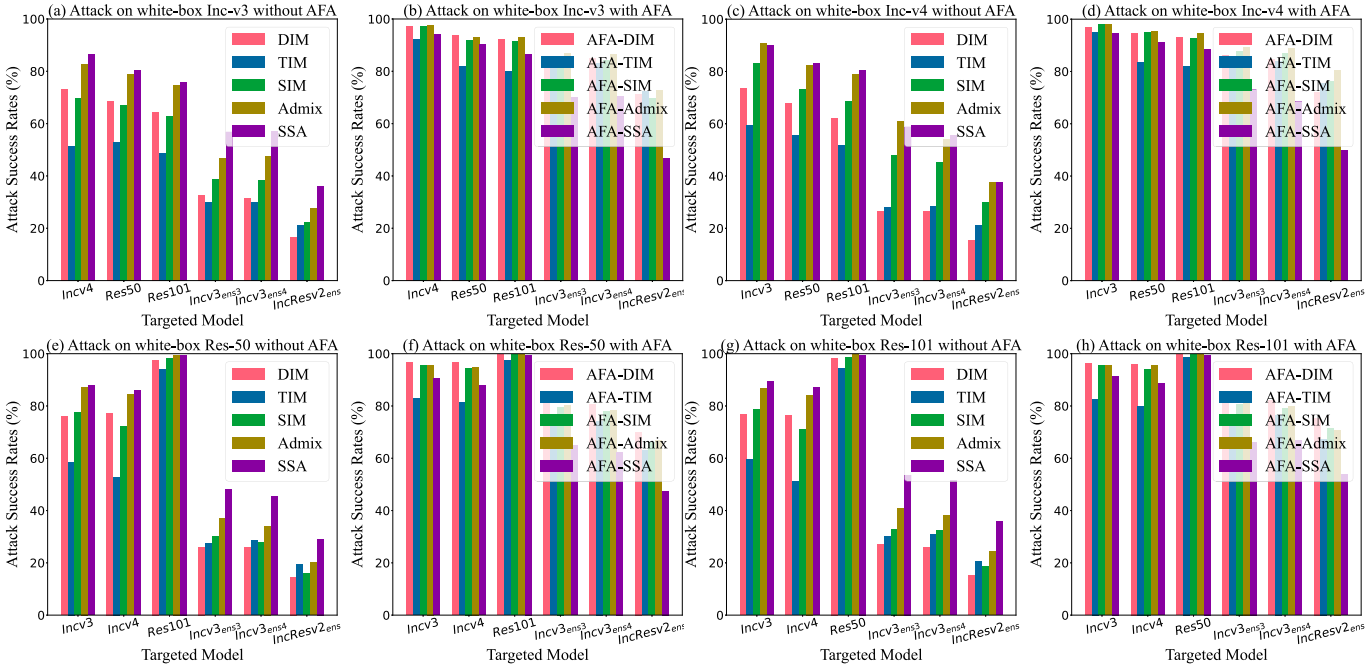

Fig. 11. Untargeted attack success rates between various gradient-based attacks under the augmentation setting without/with AFA. The white-box source model is in the title. The black-box target models are on x axis.

Table 4
The untargeted attack success rates of various gradient-based attacks in the defense model setting. **BOLD** indicates the best.

Model	Attack	RS	HGD	FD	NRP	ComDefend	JPEG	Bit-Red	R&P	NIPS-r3
Inc-v3	RAP	25.20	2.20	63.40	57.10	55.70	61.20	71.40	77.40	12.90
	NCS	31.30	32.30	72.70	66.90	66.80	71.20	75.00	75.70	41.50
	PGN	42.20	39.00	82.50	72.00	77.90	81.40	83.50	86.00	54.10
	MuMoDIG	23.50	18.80	69.60	61.80	64.30	64.50	70.00	73.40	27.40
	GI-MI	20.50	4.30	54.50	57.00	49.80	48.80	56.00	57.60	14.70
	FGSRA	39.50	39.40	78.80	68.50	74.10	77.20	79.60	81.00	50.90
	AFA	**44.00**	**43.10**	**86.70**	**74.80**	**82.40**	**87.60**	**89.80**	**91.30**	**59.80**
Inc-v4	RAP	19.00	5.10	48.50	58.00	40.00	43.30	53.90	77.90	12.80
	NCS	32.70	37.70	72.10	67.10	67.50	70.10	74.60	79.00	45.40
	PGN	41.50	39.00	81.50	75.20	75.40	80.70	82.70	85.40	56.20
	MuMoDIG	25.20	21.80	66.90	62.70	61.70	64.80	70.10	77.00	29.90
	GI-MI	21.80	3.30	51.40	57.80	45.70	47.70	54.60	57.80	13.20
	FGSRA	38.50	42.00	77.40	70.70	72.40	75.70	79.10	82.60	53.40
	AFA	**45.30**	**48.20**	**88.40**	**77.80**	**81.70**	**87.10**	**89.40**	**91.90**	**63.70**
Res-50	RAP	29.30	5.20	77.10	81.90	64.60	89.20	94.40	66.70	13.90
	NCS	39.60	63.10	91.20	89.40	81.80	98.50	99.60	76.90	44.60
	PGN	47.70	63.30	92.30	93.70	85.20	98.00	98.70	79.20	54.30
	MuMoDIG	23.40	36.30	79.50	79.50	64.70	91.30	97.30	67.90	18.40
	GI-MI	21.60	11.70	63.80	76.10	47.80	81.70	92.40	47.40	11.70
	FGSRA	45.50	62.80	92.60	91.90	85.30	99.00	99.70	77.80	52.20
	AFA	**49.30**	**67.80**	**94.40**	**94.90**	**87.60**	**99.40**	**99.90**	**83.40**	**56.30**
Res-101	RAP	30.70	14.20	95.60	79.50	82.90	75.40	82.40	67.40	15.60
	NCS	43.30	73.50	98.60	88.40	97.00	90.30	92.80	75.70	49.70
	PGN	54.80	71.30	98.50	91.50	97.60	92.10	94.10	77.90	58.20
	MuMoDIG	24.60	44.50	98.30	79.20	90.40	75.40	82.40	68.90	20.20
	GI-MI	22.20	13.10	94.20	74.00	73.90	58.70	69.00	48.90	12.70
	FGSRA	51.10	72.20	99.40	89.90	97.70	92.20	94.00	78.70	55.90
	AFA	**55.20**	**78.70**	**99.90**	**92.80**	**98.50**	**94.60**	**96.00**	**92.30**	**61.60**

4.4. Attack an ensemble of models

Furthermore, we assess the effectiveness of our AFA in an ensemble-model setting, in addition to testing on an individual model. In this section, we use two models as the white-box surrogate [11], creating an ensemble by averaging the logit outputs of both models. The adversaries are generated by randomly combining two models from four normally trained models: Inc-v3, Inc-v4, Res-50 and Res-101. All the ensemble models are assigned equal weights and we test the performance of transferability on both two normally trained models and three adversarially trained models. The results presented in Fig. 10 indicate that our AFA method consistently achieves the highest attack success rates among the six ensemble settings. Our AFA achieves significantly more stable ASRs compared to previous gradient-based attacks. Furthermore, our method demonstrates remarkable improvements even against adversarially trained models. These results validate the effectiveness of integrating both adversarial zeroth-order flatness and adversarial first-order flatness in enhancing the transferability of adversarial attacks, as opposed to solely optimizing either one or using only adversarial first-order flatness.

4.5. Combined with input transformation attacks

Currently, input transformation attacks have shown substantial compatibility with each other. Given that our AFA method generates adversarial examples utilizing gradient-level information, it naturally complements these input transformation-based methods to boost the adversarial transferability. To further validate the effectiveness of our AFA, we integrate it into various input transformations such as DIM, TIM, SIM, Admix and SSA. Subsequently, we generate adversarial examples individually on Inc-v3, Inc-v4, Res-50 and Res-101 models. Finally, we evaluate the transferability of these adversarial examples on six black-box models, including three normally trained models and three adversarially trained models. It should be mentioned that we compare the ASRs of input transformation-based attacks with and without our AFA, as well as calculate the ASRs of input transformation-based attacks utilizing all baselines. The experimental results can be observed in Fig. 11. When integrated with our gradient-based approach, it can greatly improve the transferability of adversarial attacks relying on input transformations towards black-box target models. Moreover, our approach exhibits significant enhancements in attack performance when integrated with these input transformation-based techniques. For instance, input transformation-based methods alone yield an average ASRs below 35 % on IncRes-v2$_{ens}$ in white-box Res-50. Nonetheless, when combined with

Table 5
Ablation study conducted under normal models, adversarially trained models and diverse architectures. The white-box source models are IncRes-v2 and Inc-v3 respectively. The ✓and ✗symbols indicate our method with and without the corresponding component respectively.

Model	AF	MCAS	Res-50	Res-101	Inc-v3$_{ens}$	IncRes-v2$_{ens}$	ViT-B/16	PiT-S
IncRes-v2	✗	✗	79.50	74.40	69.40	62.30	49.90	52.40
	✓	✗	90.40	88.10	80.20	71.30	60.10	67.40
	✓	✓	**91.90**	**90.40**	**81.60**	**71.60**	**60.80**	**68.10**
Inc-v3	✗	✗	79.90	75.50	62.10	43.20	46.00	44.30
	✓	✗	87.20	87.20	70.50	48.60	54.30	57.40
	✓	✓	**90.10**	**88.60**	**71.10**	**50.20**	**55.40**	**58.70**

our AFA method, the average ASRs surpass 60 %, representing 20 % increase as depicted in Fig. 11. More results can be referred in Appendix I.

4.6. Ablation study

Ablation for each component. To thoroughly examine how components affect our method, we perform ablation studies on various variants to target two normally trained models, two adversarially trained models and two diverse architectures using white-box surrogates Inc-v3 and IncRes-v2: (1) Without AF and MCAS, which equal to MI equipped with N inner samplings and is a basic framework of most relevant works. We treat the variant as a foundation to demonstrate the improvement of each component. (2) Utilizing AF but only sampling uniformly. (3) Both AF and MCAS are taken into consideration. As presented in Table 5, several noteworthy observations emerge when comparing the results of various versions. Together with the results of previous studies, it is clear that the basic MI method combined with N inner random sampling shows improved transferability compared to its traditional version, especially for models incorporating adversarially trained weights and different architectures. Given the surrogate model as Inc-v3, the average ASRs of MI with N inner samplings are 35 % larger than MI [11] in Fig. 3. This discovery confirms that our deliberate selection of the inner loop sampling design as it significantly mitigates the decline in attack effectiveness. Then when only integrating our proposed AF to generate adversarial examples on Inc-v3, our average ASRs can reach 4 %, 8 %, 14 %, 22 %, 28 % and 33 % more higher than PGN, FGSRA, NCS, MuMoDIG, RAP and GI-MI. It is evident to show the superiority and effectiveness of our proposed AF. Finally, equipped with our MCAS, compared to the version with only AF, it is clear that our new performance can be further enhanced, demonstrating our success in the new neighborhood sampling. To further demonstrate the effectiveness of our proposed components, we conduct the ablation study on the defense model setting in L. Finally, the ablation study results about more hyperparameters are provided in J.

5. Conclusion

In this paper, motivated by the deceptive flatness problem, we propose a novel black-box gradient-based adversarial attack method to improve adversarial transferability. First, we feasibly fuse the dual-order flatness to construct the solution to deceptive flatness, and theoretically prove its assurance of adversarial transferability. Subsequently, by efficiently approximating our optimization objective based on the proposed dual-order information fusion, we develop the Adversarial Flatness Attack (AFA) with addressing the gradient sign alteration, which achieves the best attack performance on ImageNet-compatible dataset under various attack and defense settings. Additionally, we introduce MonteCarlo Adversarial Sampling (MCAS) to diversify inner sampling and further bolster adversarial transferability, resulting in less memory and more diverse sampling. These findings indicate that the current advanced models are still vulnerable and highlight the importance of researching more robust defense methods.

While this approach has yielded significant results on commonly used benchmarks, it does have specific limitations. Our method is built on CNN-like surrogate models and lacks direct consideration of attention shifts between CNN and Transformer architectures. Although this issue can be mitigated by input transformation strategies, further exploration from the perspective of gradient flatness is required. Hence, our future goal is to match the attack effectiveness on Transformer-based models with that on CNNs.

Data and Code availability

Our key codes are released at: https://github.com/ZhixuanZhang77/AFA.

CRediT authorship contribution statement

Zhixuan Zhang: Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization; **Pingyu Wang:** Writing – review & editing, Validation, Methodology, Conceptualization; **Xingjian Zheng:** Writing – review & editing; **Linbo Qing:** Writing – review & editing; **Qi Liu:** Writing – review & editing.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relations that could have appeared to influence their work reported in this paper.

Acknowledgment

This work was supported by Science and Technology Projects of Xizang Autonomous Region (No. XZ202501ZY0064), the National Natural Science Foundation of China (No. 62301346 and No. 62202174), Sichuan Science and Technology Program (No. 2024NSFSC1424), Chengdu Technology Innovation Research and Development Project (No. 2024-YF05-00652-SN), Chengdu Major Technology Application Demonstration Project (No. 2023-YF09-00019-SN), the Fundamental Research Funds for the Central Universities (No. YJ202326 and No. 2025ZYGXZR053) and the GJYC program of Guangzhou (No. 2024D01J0081) and the ZJ program of Guangdong (No. 2023QN10X455).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2025.112710.

Appendix A. Proof of the relationships between the adversarial loss on the surrogate model in the vicinity of x_{adv} with AZF and AFF

Given any adversarial example x_{adv}, the maximum radius of uniform sampling $\mathcal{U}$ centered at adversarial example ξ and step size α, we suppose that Eq. (6) has the second-order gradient at least. On the one hand, according to the mean value theorem and Cauchy–Schwarz inequality, $\vartheta \sim \mathcal{U}(-\xi, \xi)$, there exists a constant $0 \le c \le 1$, such that the adversarial loss on the surrogate model in the vicinity of x_{adv} and AFF can be related as below:

$$\begin{aligned}\mathcal{L}_{src}^{adv}\left(x_{adv}+\vartheta\right) &= \mathcal{L}_{src}^{adv}\left(x_{adv}\right) + \left(\bigtriangledown \mathcal{L}_{src}^{adv}\left(x_{adv}+c\vartheta\right)\right)^{\top}\vartheta \\ &\le \mathcal{L}_{src}^{adv}\left(x_{adv}\right) + \left\|\bigtriangledown \mathcal{L}_{src}^{adv}\left(x_{adv}+c\vartheta\right)\right\|\left\|\vartheta\right\| \\ &\le \mathcal{L}_{src}^{adv}\left(x_{adv}\right) + \Psi_1\left(x_{adv}\right)\end{aligned} \tag{A.1}$$

On the other hand, as expressed in Definition 1, we can easily reason the relationship of the adversarial loss on the surrogate model in the vicinity of x_{adv} and AZF as follows:

$$\mathcal{L}_{src}^{adv}(x_{adv}+\vartheta) \le \mathcal{L}_{src}^{adv}(x_{adv}) + \Psi_0(x_{adv}) \tag{A.2}$$

Thus, by combining Eqs. (A.1) and (A.2), $0 \le \beta_f \le 1$, we can associate the adversarial loss on the surrogate model in the vicinity of x_{adv} with AF, deriving Eq. (7).

Appendix B. Proof of the relationship between AF and adversarial transferability

Subsequently, inspired by the generalization boundary theory in Theorem 4.2 [38], we further reformulate the adversarial transferability as below:

$$\begin{aligned}&\mathcal{L}_{tar}^{adv}(x_{adv}) \le \mathcal{L}_{src}^{adv}(x_{adv}) + BD_{upper},\\ &BD_{upper} = \sup \|\mathcal{L}_{tar}(x^1,x^2) - \mathcal{L}_{src}(x^1,x^2)\|, x^1, x^2 \in \mathcal{B}_\xi(x_{adv})\end{aligned} \tag{B.1}$$

where the term $\sup\|\cdot\|$ corresponds to the upper bound of the discrepancy distance [39], which measures the covariate shift between adversarial losses of x_{adv} on source model $\boldsymbol{M}_{src}$ and target model $\boldsymbol{M}_{tar}$, which are similar to the training domain and the unknown test domain in domain generalization respectively. Similarly, $\mathcal{B}_\xi(x_{adv})$ indicates the neighborhood on point x_{adv} at radius of ξ.

Finally, after integrating Eqs. (7) and (B.1), we obtain the expanded inequality between AF and adversarial transferability:

$$\begin{aligned}&\mathbb{E}_{x'_{adv}\sim\mathcal{B}_\xi(x_{adv})}[\mathcal{L}_{tar}^{adv}(x'_{adv})]\\ &\le \mathbb{E}_{x'_{adv}\sim\mathcal{B}_\xi(x_{adv})}[\mathcal{L}_{src}^{adv}(x'_{adv})] + BD_{upper}\\ &\le \mathcal{L}_{src}^{adv}(x_{adv}) + \Psi^{AF}(x_{adv}) + BD_{upper}\end{aligned} \tag{B.2}$$

We can obtain Eq. (8) by simplifying Eq. (B.2).

Appendix C. Proof of the gradient of Adversarial First-order Flatness

Based on Corollary 1 in [14], $\nabla_x\|\nabla_x\mathcal{L}_{src}^{adv}(x)\| = \frac{\nabla_{x'}\mathcal{L}_{src}^{adv}(x')|_{x'=x+\triangle x} - \nabla_x\mathcal{L}_{src}^{adv}(x)}{\alpha}$. Further, we can first drive x_2, $x_3 = x_2 + \triangle x_2$ and their corresponding gradients as follows:

$$\begin{aligned}&g'_2 = -g_2 = -\nabla_{x_2}\mathcal{L}(x_2, y^{gt}; \boldsymbol{M}_{src}), \text{ where}\\ &x_2 = x_0 + \alpha\cdot\frac{\nabla_{x_0}\|\nabla_{x_0}\mathcal{L}_{src}^{adv}(x_0)\|}{\|\nabla_{x_0}\|\nabla_{x_0}\mathcal{L}_{src}^{adv}(x_0)\|\|}\\ &\quad = x_0 - \alpha\cdot\frac{g_1-g_0}{\|g_1-g_0\|}\end{aligned} \tag{C.1}$$

$$\begin{aligned}&g'_3 = -g_3 = -\nabla_{x_3}\mathcal{L}(x_3, y^{gt}; \boldsymbol{M}_{src}), \text{ where}\\ &x_3 = x_2 + \triangle x_2\\ &\quad = x_2 - \alpha\cdot\frac{g_2}{\|g_2\|}\end{aligned} \tag{C.2}$$

Combine above Eqs. (C.1) and (C.2), gradient of $\Psi_1(x_{adv})$ can be approximated as below:

$$\begin{aligned}&\nabla\Psi_1(x_{adv}) = h_1 \approx \alpha\cdot\nabla\|\nabla\mathcal{L}_{src}^{adv}(x_2)\|\\ &= \alpha\cdot\frac{\nabla_{x_3}\mathcal{L}_{src}^{adv}(x_3) - \nabla_{x_2}\mathcal{L}_{src}^{adv}(x_2)}{\alpha}\\ &= g'_3 - g'_2\\ &= -(g_3 - g_2)\end{aligned} \tag{C.3}$$

To this end, we prove Eq. (11) as above all.

Appendix D. Proof of the gradient of the approximated objective function of AFA

After joining Eqs. (10) and (11), our AFA can be written as:

$$\begin{aligned}&\min_{x_{adv}\in\mathcal{B}_\epsilon(x)}[g'_0 + \lambda_f\Psi^{AF}(x_{adv})]\\ \Rightarrow &\min_{x_{adv}\in\mathcal{B}_\epsilon(x)}[-g_0 - \lambda_f(\beta_f(g_1-g_0) + (1-\beta_f)(g_3-g_2))]\\ \Rightarrow &\max_{x_{adv}\in\mathcal{B}_\epsilon(x)}[g_0 + \lambda_f(\beta_f(g_1-g_0) + (1-\beta_f)(g_3-g_2))]\end{aligned} \tag{D.1}$$

Therefore, we prove Eq. (12).

Appendix E. Visual experiment for the rationale of the combined Adversarial Flatness

In fact, relying solely on extensive abstract mathematical derivations is insufficient to intuitively uncover the rationale behind the combined Adversarial Flatness. Therefore, we present a visualization of loss surfaces for adversarial examples generated by our method solving the problem of altered gradient sign and three variants with only AZF, only AFF and only AF. As shown in Fig. E.1, it is obvious that the variant with the combined AF can achieve a much broader and flatter top plane, which is more beneficial to adversarial transferability. It verifies the rationale of the combined Adversarial Flatness again. Moreover, it is suggested that the last column of results also demonstrates the effectiveness of our solution to the problem of altered gradient sign.

Appendix F. Experiments based on different gradient approximation methods

As for the gradient approximation strategies used in our method, we mainly consider two aspects: precision and simplicity. It is easy to decide on the Finite Difference Method, especially the commonly used Forward Difference Method (FDM), which is also utilized in our method. However, there are other variants in the family of the Finite Difference Method, such as the Center Difference Method (CDM) and the Backward Difference Method (BDM). BDM is similar to FDM, although CDM is slightly more complex. We compare our method with its variants based on the two gradient approximation methods in terms of ASRs on different models. As shown in Table F.1, it is clear that the variety of gradient approximation methods significantly influences their effectiveness. Furthermore, the FDM-based implementation achieves superior performance compared to the other two implementations, which demonstrates the feasibility of the FDM in our methodm Table G.1.

Appendix G. Experiments based on transformer-based surrogates and attacking normally trained models

For comprehensively validating our performance on a single model, we also consider the advanced transformer-based models as the surrogate models. Because traditional CNNs differ from Transformers in terms of information extraction, it is essential to conduct these experiments to fully explore our superiority. Wherein, we set the surrogates as ViT-B/16 and Swin-T. And the experiments are divided into two sub-experiments: (1) attacking the normal models with CNN architectures and (2) attacking the models with diverse architectures. As for the target models, the former includes Inc-v3, Res-101, Dense-121 and VGG-19 while the latter employs MobilNet, PASNet-L, MLP-mixer and ResMLP. When comparing Table F.2 with Table 2, it is evident that when adversarial examples generated on Inc-v3 are used to attack ViT-B/16, the ASRs of all methods are significantly lower than those observed when attacking Inc-v3 from ViT-B/16. And, the same phenomenon also exists in Table F.3. In particular, for MLP-Mixer and ResMLP, attacks launched by ViT-B/16 and Swin-T achieve significantly higher average ASRs compared to those from CNNs. These results may derive from the difference in the information extraction process between CNNs and Transformers. However,

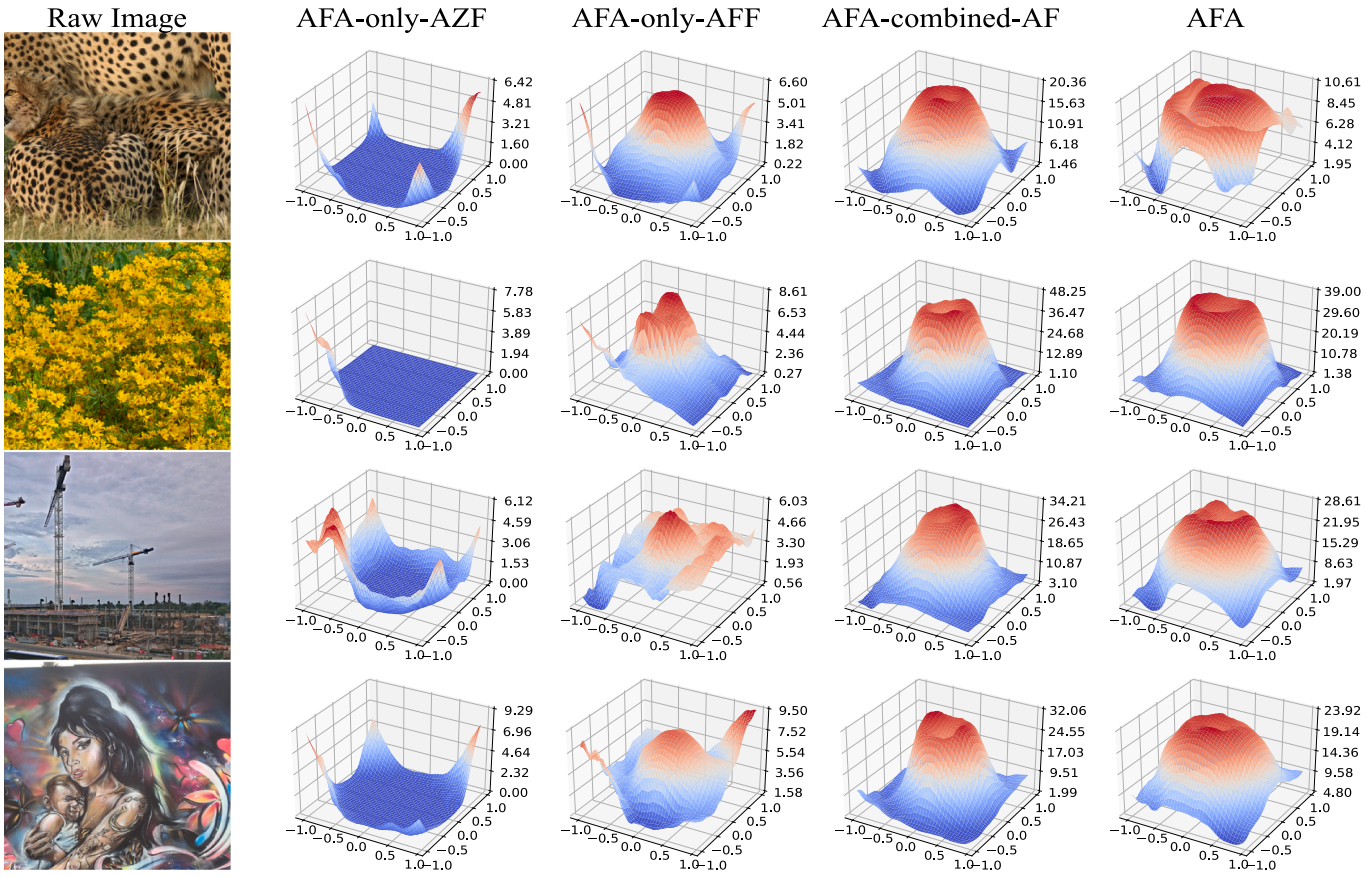

Fig. E.1. Visualization of loss surfaces along two random directions for two randomly sampled adversarial examples on the surrogate Inc-v3. The center of each 2D graph corresponds to the adversarial example generated by our method, and three variants with only AZF, only AFF and only AF. The x and y axes represent the random noises added to x_{adv} twice in succession. The z axis indicates the loss value.

Table F.1
The untargeted attack success rates based on different gradient approximation methods. **BOLD** indicates the best.

Model	Approx	Inc-v4	Dense-121	VGG-19	PASNet-L	MLP-mixer	ResMLP	Inc-v3$_{ens}$	Inc-v4$_{ens}$	InRes-v2$_{ens}$	NRP	JPEG	Bit-Red
Inc-v3	BDM	91.00	87.20	87.80	81.20	60.70	61.90	63.50	65.00	43.10	72.40	80.20	84.80
	CDM	94.20	90.00	90.80	85.80	63.30	67.50	68.90	69.50	46.50	73.80	84.50	88.00
	FDM	**95.20**	**91.90**	**91.50**	**87.40**	**65.30**	**72.30**	**73.00**	**73.20**	**50.90**	**74.80**	**87.60**	**89.80**
Swin-T	BDM	81.90	93.30	93.50	89.40	91.20	97.10	73.30	71.70	63.10	89.00	90.10	90.00
	CDM	81.70	93.10	93.70	88.90	90.80	96.90	72.10	71.30	61.50	89.30	90.00	89.70
	FDM	**85.70**	**94.30**	**94.70**	**89.90**	**93.20**	**97.60**	**75.40**	**74.50**	**65.30**	**90.80**	**90.60**	**91.10**

Table F.2
The untargeted attack success rates based on transformer surrogates among various gradient-based attacks in the normal model setting. **BOLD** indicates the best.

Model	Attack	Inc-v3	Res-101	Dense-121	VGG-19
ViT-B/16	RAP	73.50	82.90	84.00	83.80
	NCS	75.20	78.90	81.10	79.50
	PGN	79.80	84.40	85.80	84.60
	MuMoDIG	71.30	76.60	80.40	77.90
	GI-MI	66.60	75.60	77.80	80.00
	FGSRA	74.80	78.90	81.70	79.40
	AFA	**84.10**	**88.20**	**90.80**	**89.80**
Swin-T	RAP	53.80	67.70	69.80	77.30
	NCS	81.80	86.50	89.40	90.30
	PGN	85.20	89.70	91.10	91.90
	MuMoDIG	56.20	61.90	68.70	72.30
	GI-MI	49.50	60.70	64.60	72.40
	FGSRA	83.60	86.80	89.90	90.60
	AFA	**87.10**	**91.70**	**94.30**	**94.70**

Table F.3
The untargeted attack success rates based on transformer surrogates among various gradient-based attacks in the diverse model architectures setting. **BOLD** indicates the best.

Model	Attack	MobilNet	PASNet-L	MLP-mixer	ResMLP
ViT-B/16	RAP	90.20	82.30	92.30	97.10
	NCS	86.70	79.10	92.60	97.50
	PGN	89.60	84.30	94.00	98.10
	MuMoDIG	86.00	79.10	89.10	96.70
	GI-MI	87.60	75.50	89.80	96.70
	FGSRA	86.30	79.20	91.30	96.30
	AFA	**91.70**	**88.60**	**96.70**	**99.50**
Swin-T	RAP	85.00	69.30	71.40	72.50
	NCS	91.40	84.90	85.90	94.10
	PGN	93.80	88.40	91.00	95.20
	MuMoDIG	80.70	67.30	68.30	75.80
	GI-MI	81.60	63.00	67.20	69.30
	FGSRA	92.90	86.20	88.10	94.80
	AFA	**94.70**	**89.90**	**93.20**	**97.60**

Table G.1
The untargeted attack success rates based on transformer surrogates among various gradient-based attacks in the adversarially trained model setting. **BOLD** indicates the best.

Model	Attack	Inc-v3*	Inc-v3$_{ens}$	Inc-v4$_{ens}$	InRes-v2$_{ens}$
ViT-B/16	RAP	61.30	50.90	52.50	39.80
	NCS	65.40	62.90	62.80	56.10
	PGN	73.20	71.20	71.30	63.90
	MuMoDIG	63.40	58.10	59.10	50.30
	GI-MI	56.80	52.60	50.90	42.10
	FGSRA	67.90	66.50	66.20	58.70
	AFA	**79.30**	**75.40**	**75.70**	**69.40**
Swin-T	RAP	36.40	30.80	32.00	21.00
	NCS	66.10	66.60	57.90	62.00
	PGN	78.00	74.50	72.10	64.80
	MuMoDIG	39.30	34.10	34.20	25.70
	GI-MI	31.30	29.40	30.50	21.00
	FGSRA	75.40	71.10	71.20	61.90
	AFA	**80.00**	**75.40**	**74.50**	**65.30**

Table G.2
The untargeted attack success rates based on transformer surrogates among various gradient-based attacks in the defense model setting. **BOLD** indicates the best.

Model	Attack	HGD	NRP	JPEG	Bit-Red
ViT-B/16	RAP	48.40	75.60	79.30	78.40
	NCS	61.00	80.40	79.80	76.00
	PGN	69.20	85.20	84.90	81.70
	MuMoDIG	55.90	79.90	75.20	73.40
	GI-MI	48.90	76.50	73.10	72.30
	FGSRA	64.00	83.00	78.20	76.90
	AFA	**73.20**	**86.00**	**87.60**	**86.50**
Swin-T	RAP	20.90	80.10	64.90	66.40
	NCS	68.00	85.60	84.40	84.80
	PGN	75.30	89.00	89.60	89.60
	MuMoDIG	30.70	77.10	60.10	62.70
	GI-MI	23.20	76.50	56.80	60.70
	FGSRA	72.20	86.50	85.90	86.90
	AFA	**76.60**	**90.80**	**90.60**	**91.10**

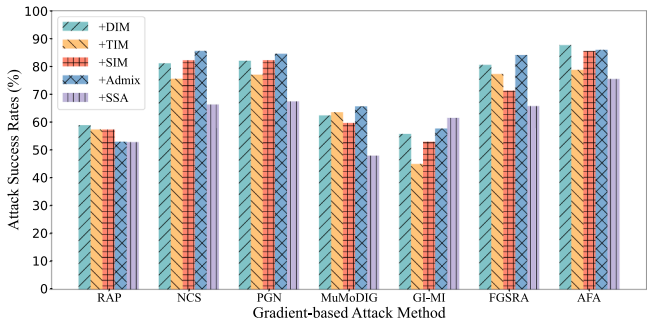

Fig. G.2. Comparison of the average untargeted attack success rates between various gradient-based attacks with the various augmentation-based settings. The white-box source model is Res-50.

our method still obtains the best performance in both Tables F.2 and F.3.

Appendix H. Experiments based on transformer-based surrogates and attacking defense models

Meanwhile, we also test our effectiveness on the transformer-based models for the defense models. Except for the defense strategies as adversarial training, HGD, NRP, JPEG and Bit-Red, the other settings are identical to Appendix G. We group this validation as two experiments: (1) aiming at the models equipped with the adversarial training strategy, including Inc-v3*, Inc-v3$_{ens}$, Inc-v4$_{ens}$ and InRes-v2$_{ens}$. Then, (2) attacking the models with the other defense strategies. In terms of attacking models with adversarial training, compared to Table 3, our method gains a 10.30 % increase in ASRs while maintaining the best effectiveness in Table G.1. In addition, in Table G.2, there exists a 2 % attack capability increment under the defense strategies compared with Table 4. And, our proposed method can still pose the state-of-the-art performance in Table G.2.

Appendix I. Additional experiments combined with input transformation attacks

We also display the results of the additional experiments combined with input transformation attacks. In Fig. G.2, it further shows that these input transformation-based methods can benefit more from our AFA than other baselines. Thereby, combined with current input transformation-based methods, our approach can further improve the success rate of transfer-based black-box attacks.

Appendix J. Ablation for the more hyperparameters

Furthermore, we conduct ablation experiments to analyze the impact of these hyperparameters, such as the flatness balanced coefficient β_f, the flatness item coefficient λ_f, the number of sampled examples N, the total iteration number T, the ablations towards the radius γ_{MCAS} and the momentum decay η_{MCAS} of MCAS. As default, they are set as 0.5, 0.0032, 20, 10, 0.0094 and 0.9 respectively. In this validation, we attempt different values of one parameter while keeping the other three parameters unchanged. As shown in Fig. H.1(c) and H.1(d), the attack ability of our proposed AFA is influenced by N and T more than the two other hyperparameters. As N and T increased, the transferability exhibits rapid improvement until $N = 13$ and $T = 7$, after which it gradually converges for normally trained models. Notably, when $N \geq 13$ and $T \geq 7$, a slight performance improvement can still be achieved by increasing the number of sampled examples and the number of iterations in our AFA method. Meanwhile, as depicted in Fig. H.1 (a), the best transferability to other models occurs at the intermediate value of the hyperparameter β_f, rather than at $\beta_f = 0$ or $\beta_f = 1$, which correspond to using only AFF or AZF independently. It confirms the efficacy of our dual-order information fusion. Further, Fig. H.1(b) suggests that the diverse range of the flatness item coefficient λ_f has varying effects on adversarial transferability. Finally, Fig. H.1(e) presents that the transferable ASRs among the adv-trained models might decrease with the radius γ_{MCAS} of MCAS larger than 0.0094. As shown in Fig. H.1(f), the smaller momentum decay η_{MCAS} of MCAS slightly affects the adversarial transferability, but $\eta_{MCAS} = 0.9$ is a suitable choice.

Appendix K. Ablation of the hyperparameters on robustness and applicability

Given the importance of robustness and applicability in a successful attack method, we conduct further ablation experiments on the transformer-based surrogate (Swin-T) for β_f, λ_f, T and ϵ as follows: (1) attacking defense models such as HGD, NRP, JPEG and Bit-Red. And (2) attack the Baidu Cloud API. In our experiment, robustness is demonstrated through the validation of the defense strategies. On the other hand, the applicability is proved through the substitution of the CNN-based surrogate with the transformer-based surrogate, and the validation on the commercial API. Notably, ϵ defaults to 0.0627, while the default values for the other hyperparameters can be found in Appendix J. First, as shown in Fig. K.1(a)–(c), it has been verified that our chosen hyperparameter values for β_f, λ_f and ϵ are feasible among the selected options for each hyperparameter, as they provide the best robustness to the four defense models. And, in Fig. K.1(d), it can be observed that the ASRs stabilize gradually when T exceeds 9, although better performance may still be possible. However, to balance computational cost and performance, $T = 10$ is also a suitable choice. Besides,

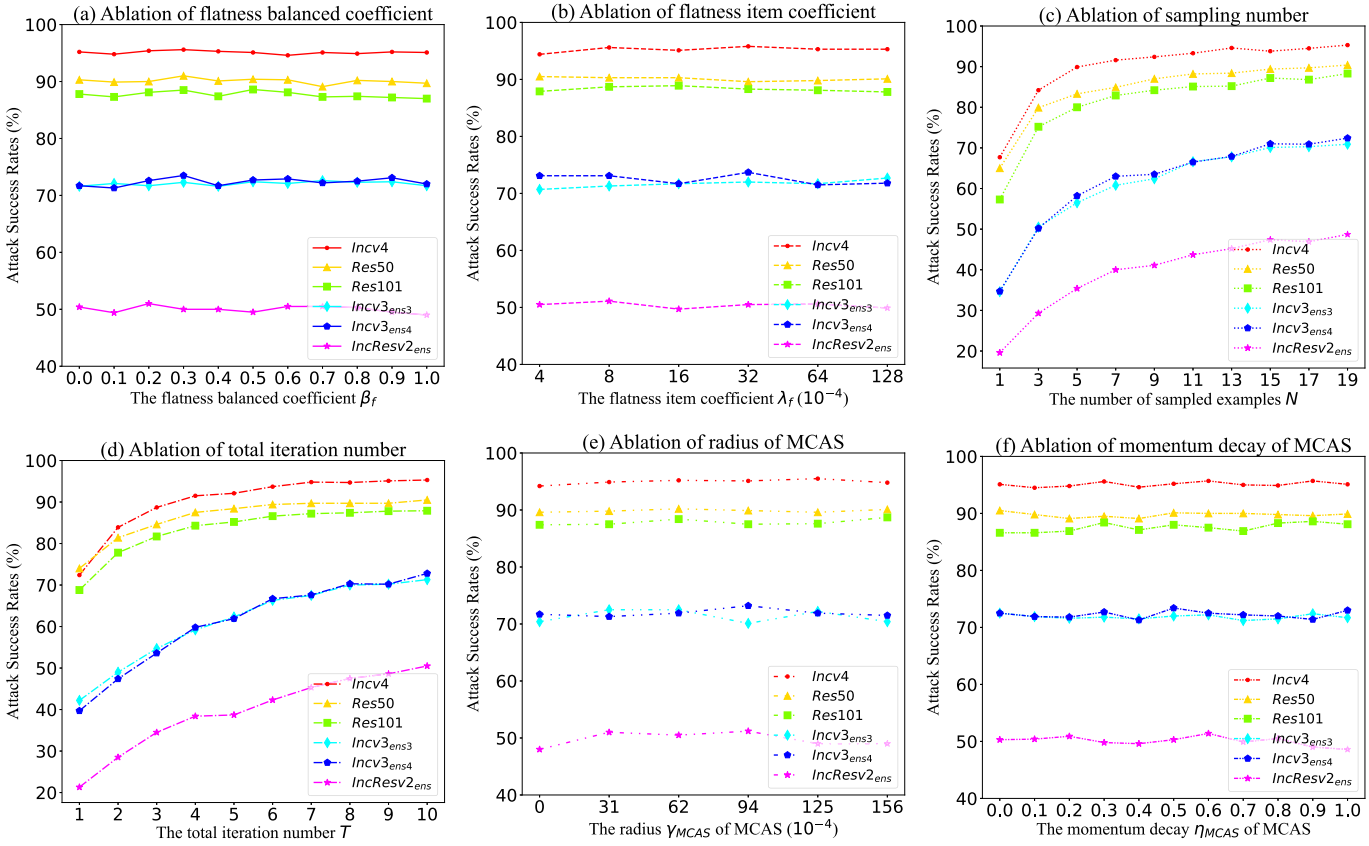

Fig. H.1. Untargeted attack success rates (%) in six black box models with different hyperparameter ablation settings. The adversarial examples are generated by our AFA on Inc-v3.

as displayed in Fig. K.2, when attacking the Baidu Cloud API, there are some meaningful observations: (1) Only when $\epsilon = 0.0627$ does our method successfully deceive the system into recognizing objects as anything other than "snails". (2) With β_f gets close to 0 or 1, the recognition results appear the classes close to "snails", such as "noctuidae" in $\beta_f = 0.2$ and "insects" in $\beta_f = 1.0$. It can verify the dual-order solution (AF) again. (3) The "snails" class and the similar class vanish when $\lambda_f \geq 0.0032$, corresponding to the results in Fig. K.1 (b). (4) Our method can thoughtfully confuse the system when $T \geq 9$. This section comprehensively explores hyperparameter ablation with respect to robustness and applicability.

Appendix L. Ablation of each component in the defense model setting

To sufficiently verify our proposed method, we take another step to carry out more ablation experiments of our AF and MCAS in the defense model setting. The surrogate model is set as Inc-v3, and the defense models maintain consistency with previous experiments. Besides, the three experimental variants are consistent with the previous ablation section. As shown in Table L.1, combined with our proposed AF, the inner-sampling MI method significantly shows stronger invasion. And compared with the results of Table 4, it is observed that our proposed AF can bring higher 3 %, 7 %, 13 %, 20 %, 25 % and 32 % ASRs than PGN, FGSRA, NCS, MuMoDIG, RAP and GI-MI on average. Therefore, it indicates that our proposed AF consistently maintains the best attack capability even in the defense model setting. Finally, integrated with our MCAS, our performance is further improved, which verifies our designed sampling method.

Table L.1
Ablation study conducted under the defense model setting. The white-box source model is Inc-v3. The ✓and ✗symbols indicate our method with and without the corresponding component respectively.

Model	AF	MCAS	RS	HGD	FD	NRP	ComDefend	JPEG	Bit-Red	R&P	NIPS-r3
Inc-v3	✗	✗	40.20	39.10	75.40	70.10	73.20	76.60	79.40	79.00	50.10
	✓	✗	43.50	42.20	86.20	74.00	81.30	87.20	88.80	90.50	58.80
	✓	✓	**44.20**	**43.00**	**87.10**	**74.90**	**82.40**	**87.40**	**89.60**	**91.60**	**59.30**

Table L.2
Quantity study of the average computational efficiency in our method. The white-box source models are Inc-v3, Res-101, ViT-B/16 and Swin-T. The ✓and ✗symbols indicate our method with and without the corresponding component respectively.

Model	AF	MCAS	Wall-Clock Time (s)	GPU Memory (MB)
Inc-v3	✗	✗	0.62	170.0
	✓	✗	2.34	182.8
	✓	✓	2.43	184.0
Res-101	✗	✗	0.99	320.2
	✓	✗	3.85	333.0
	✓	✓	4.05	334.2
ViT-B/16	✗	✗	1.25	212.4
	✓	✗	5.07	217.2
	✓	✓	5.18	219.2
Swin-T	✗	✗	0.65	168.2
	✓	✗	2.51	168.8
	✓	✓	2.67	169.4

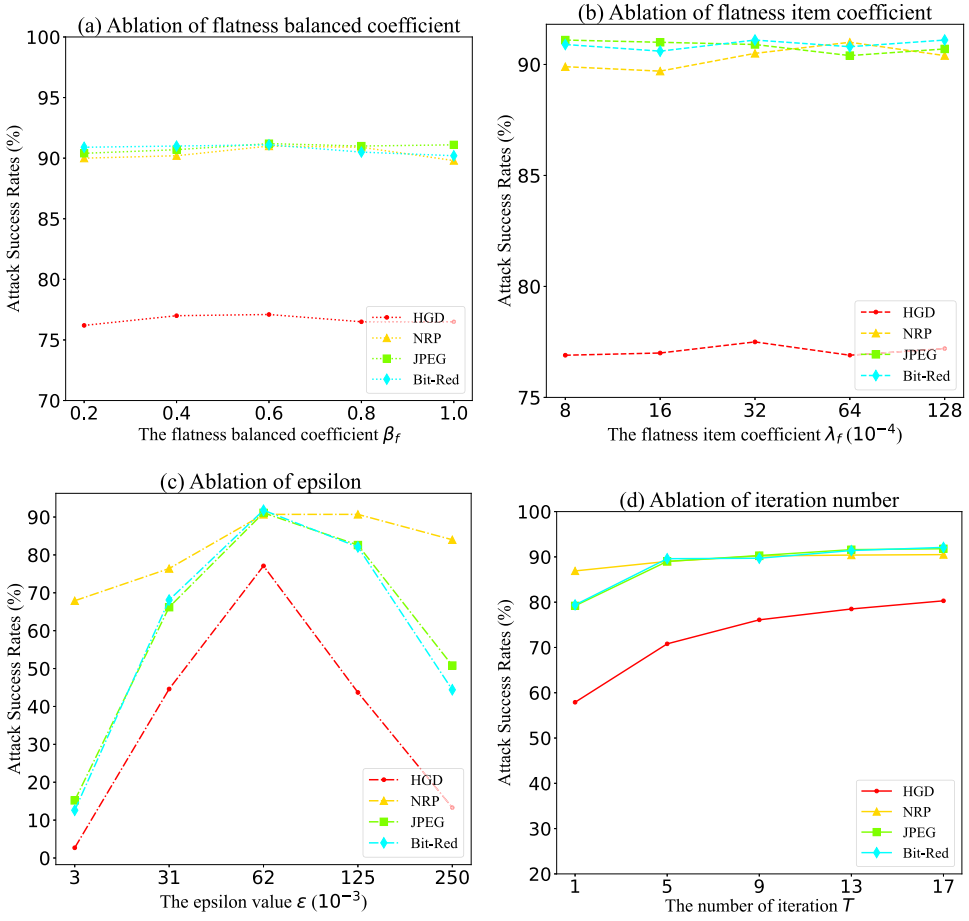

Fig. K.1. Untargeted attack success rates (%) under four defense models with different hyperparameter ablation settings. The adversarial examples are generated by our AFA on Swin-T.

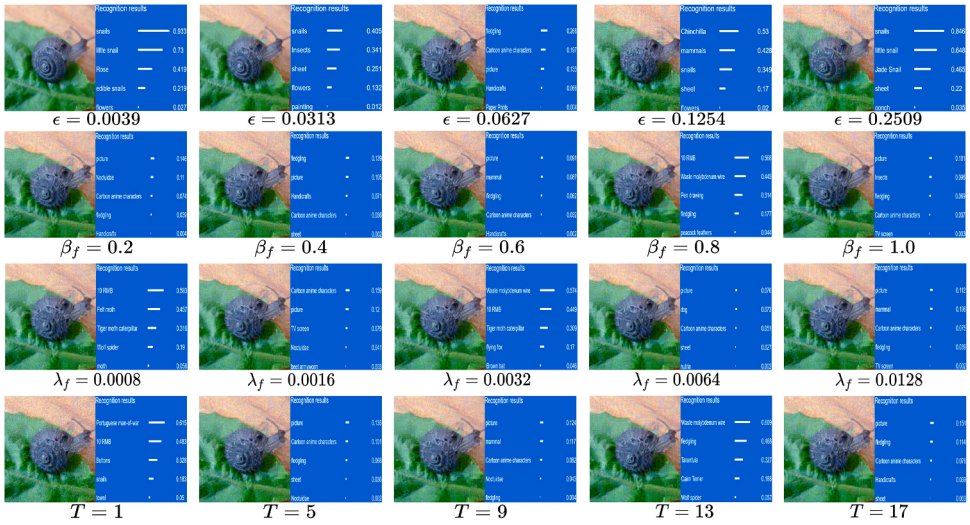

Fig. K.2. Samples of the recognition results of adversarial examples generated by AFA with different hyperparameter ablation settings on the Baidu Cloud API. The white-box source model is Swin-T.

Appendix M. Discussion on the computational efficiency of our method

As computational efficiency is important for practical applications, we design some inference validation on a GPU to quantify the wall-clock time and GPU memory. Then, we choose surrogate models with diverse architectures to set different scenarios. Specifically, in each scenario, we all test the computational efficiency for the variants with (1) a clean baseline (referred to in the previous ablation section), (2) only AF and (3) the additional MCAS. Due to the parallel computation in GPUs, we set the batch size as 10. Notably, the results are presented on average. As shown in Table L.2, it is clear that the inference time and memory usage

are closely linked to the number of parameters in the surrogate models, especially for AF. Additionally, our proposed MCAS is very lightweight, and the processing time is relatively short.

References

[1] Y. Lu, N. Liu, Y. Li, J. Chen, S. Velipasalar, Cross-task and time-aware adversarial attack framework for perception of autonomous driving, Pattern Recognit. 165 (2025) 111652.

[2] X. Liu, F. Shen, J. Zhao, C. Nie, RADAP: a robust and adaptive defense against diverse adversarial patches on face recognition, Pattern Recognit. 157 (2025) 110915.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, 2018.

[4] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, 2018, pp. 99–112.

[5] C. Hu, Z. He, X. Wu, Query-efficient black-box ensemble attack via dynamic surrogate weighting, Pattern Recognit. 161 (2025) 111263.

[6] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, A.L. Yuille, Improving transferability of adversarial examples with input diversity, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2730–2739.

[7] J. Lin, C. Song, K. He, L. Wang, J.E. Hopcroft, Nesterov accelerated gradient and scale invariance for adversarial attacks, in: International Conference on Learning Representations, 2020.

[8] Y. Dong, T. Pang, H. Su, J. Zhu, Evading defenses to transferable adversarial examples by translation-invariant attacks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4312–4321.

[9] X. Wang, X. He, J. Wang, K. He, Admix: enhancing the transferability of adversarial attacks, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 16138–16147.

[10] Y. Long, Q. Zhang, B. Zeng, L. Gao, X. Liu, J. Zhang, J. Song, Frequency domain model augmentation for adversarial attack, in: Proceedings of the European Conference on Computer Vision (ECCV), 13664, 2022, pp. 549–566.

[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9185–9193.

[12] Z. Qin, Y. Fan, Y. Liu, L. Shen, Y. Zhang, J. Wang, B. Wu, Boosting the transferability of adversarial attacks with reverse adversarial perturbation, in: Advances in Neural Information Processing Systems, 2022.

[13] C. Qiu, Y. Duan, L. Zhao, Q. Wang, Enhancing adversarial transferability through neighborhood conditional sampling, CoRR abs/2405.16181 (2024).

[14] Z. Ge, X. Wang, H. Liu, F. Shang, Y. Liu, Boosting adversarial transferability by achieving flat local maxima, in: Advances in Neural Information Processing Systems, 2023.

[15] Y. Liu, H. Wei, C. Jia, R. Xiao, W. Ruan, X. Wei, J.T. Zhou, Z. Wang, ProjAttacker: a configurable physical adversarial attack for face recognition via projector, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025, pp. 21248–21257.

[16] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015.

[17] X. Wang, K. He, Enhancing the transferability of adversarial attacks through variance tuning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 1924–1933.

[18] J. Wang, Z. Chen, K. Jiang, D. Yang, L. Hong, P. Guo, H. Guo, W. Zhang, Boosting the transferability of adversarial attacks with global momentum initialization, Expert Syst. Appl. 255 (2024) 124757.

[19] X. Wang, Z. Jin, Z. Zhu, J. Zhang, H. Chen, Improving adversarial transferability via frequency-guided sample relevance attack, in: Proceedings of the ACM International Conference on Information and Knowledge Management, 2024, pp. 2410–2419.

[20] Y. Ren, Z. Zhao, C. Lin, B. Yang, L. Zhou, Z. Liu, C. Shen, Improving integrated gradient-based transferable adversarial examples by refining the integration path, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 6731–6739.

[21] H. Zhang, M. Cissé, Y.N. Dauphin, D. Lopez-Paz, mixup: beyond Empirical risk minimization, in: International Conference on Learning Representations, 2018.

[22] Y. Qian, K. Chen, B. Wang, Z. Gu, S. Ji, W. Wang, Y. Zhang, Enhancing transferability of adversarial examples through mixed-frequency inputs, IEEE Trans. Inf. Forensics Secur. 19 (2024) 7633–7645.

[23] F. Tramèr, A. Kurakin, N. Papernot, I.J. Goodfellow, D. Boneh, P.D. McDaniel, Ensemble adversarial training: attacks and defenses, in: International Conference on Learning Representations, 2018.

[24] D. Zhou, N. Wang, T. Liu, X. Gao, Improving adversarial training from the perspective of class-Flipping distribution, IEEE Trans. Pattern Anal. Mach. Intell. 47 (6) (2025) 4330–4342.

[25] D. Wu, S. Xia, Y. Wang, Adversarial weight perturbation helps robust generalization, in: Annual Conference on Neural Information Processing Systems, 2020.

[26] J. Cohen, E. Rosenfeld, J.Z. Kolter, Certified adversarial robustness via randomized smoothing, in: International Conference on Machine Learning, 97, PMLR, 2019, pp. 1310–1320.

[27] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, J. Zhu, Defense against adversarial attacks using high-Level representation guided denoiser, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1778–1787.

[28] C. Xie, Y. Wu, L. van der Maaten, A.L. Yuille, K. He, Feature denoising for improving adversarial robustness, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 501–509.

[29] M. Naseer, S.H. Khan, M. Hayat, F.S. Khan, F. Porikli, A self-supervised approach for adversarial robustness, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 259–268.

[30] X. Jia, X. Wei, X. Cao, H. Foroosh, ComDefend: an efficient image compression model to defend adversarial examples, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6084–6092.

[31] C. Guo, M. Rana, M. Cissé, L. van der Maaten, Countering adversarial images using input transformations, in: International Conference on Learning Representations, 2018.

[32] W. Xu, D. Evans, Y. Qi, Feature squeezing: detecting adversarial examples in deep neural networks, in: Annual Network and Distributed System Security Symposium, 2018.

[33] C. Xie, J. Wang, Z. Zhang, Z. Ren, A.L. Yuille, Mitigating adversarial effects through randomization, in: International Conference on Learning Representations, 2018.

[34] D. Zhou, N. Wang, B. Han, T. Liu, X. Gao, Defending against adversarial examples via modeling adversarial noise, Int. J. Comput. Vis. 133 (9) (2025) 5920–5937.

[35] P. Foret, A. Kleiner, H. Mobahi, B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization, in: International Conference on Learning Representations, 2021.

[36] X. Zhang, R. Xu, H. Yu, H. Zou, P. Cui, Gradient norm aware minimization seeks first-Order flatness and improves generalization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 20247–20257.

[37] X. Zhang, R. Xu, H. Yu, Y. Dong, P. Tian, P. Cui, Flatness-aware minimization for domain generalization, in: IEEE/CVF International Conference on Computer Vision, IEEE, 2023, pp. 5166–5179.

[38] X. Zhang, Y. He, R. Xu, H. Yu, Z. Shen, P. Cui, NICO++: towards better benchmarking for domain generalization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 16036–16047.

[39] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation: learning bounds and algorithms, in: Conference on Learning Theory, 2009.

[40] J. Wang, A. Choromanska, A survey of optimization methods for training DL models: theoretical perspective on convergence and generalization, CoRR abs/2501.14458 (2025) 1–118.

[41] S.P. Singh, R.S. Sutton, Reinforcement learning with replacing eligibility traces, Mach. Learn. 22 (1–3) (1996) 123–158.

[42] E.K. Khayya, A.E. Oirrak, T. Datsi, A survey on RGB images classification using convolutional neural network (CNN) architectures: applications and challenges, in: 2024 International Conference on Circuit, Systems and Communication (ICCSC), 2024, pp. 1–8.

[43] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L. Li, L. Fei-Fei, A.L. Yuille, J. Huang, K. Murphy, Progressive neural architecture search, in: Proceedings of the European Conference on Computer Vision, 11205, Springer, 2018, pp. 19–35.

[44] B. Palanisamy, V. Hassija, A. Chatterjee, A. Mandal, D. Chakraborty, A. Pandey, G.S.S. Chalapathi, D. Kumar, Transformers for vision: a survey on innovative methods for computer vision, IEEE Access 13 (2025) 95496–95523.