Enhance Stealthiness and Transferability of Adversarial Attacks with Class Activation Mapping Ensemble Attack

Hui Xia*
Ocean University of China
xiahui@ouc.edu.cn

Rui Zhang
Ocean University of China
zhangrui0504@stu.ouc.edu.cn

Zi Kang
Ocean University of China
kangzi@stu.ouc.edu.cn

Shuliang Jiang
Ocean University of China
jiangshuliang@stu.ouc.edu.cn

Shuo Xu
Ocean University of China
xushuo@stu.ouc.edu.cn

***Abstract*—Although there has been extensive research on the transferability of adversarial attacks, existing methods for generating adversarial examples suffer from two significant drawbacks: poor stealthiness and low attack efficacy under low-round attacks. To address the above issues, we creatively propose an adversarial example generation method that ensembles the class activation maps of multiple models, called class activation mapping ensemble attack. We first use the class activation mapping method to discover the relationship between the decision of the Deep Neural Network and the image region. Then we calculate the class activation score for each pixel and use it as the weight for perturbation to enhance the stealthiness of adversarial examples and improve attack performance under low attack rounds. In the optimization process, we also ensemble class activation maps of multiple models to ensure the transferability of the adversarial attack algorithm. Experimental results show that our method generates adversarial examples with high perceptibility, transferability, attack performance under low-round attacks, and evasiveness. Specifically, when our attack capability is comparable to the most potent attack (VMIFGSM), our perceptibility is close to the best-performing attack (TPGD). For non-targeted attacks, our method outperforms the VMIFGSM by an average of 11.69% in attack capability against 13 target models and outperforms the TPGD by an average of 37.15%. For targeted attacks, our method achieves the fastest convergence, the most potent attack efficacy, and significantly outperforms the eight baseline methods in low-round attacks. Furthermore, our method can evade defenses and be used to assess the robustness of models[1].**

I. Introduction

Deep Neural Networks (DNNs) have achieved remarkable achievements in image classification [1]–[3]. They are playing an increasingly important role in many fields [4]. In autonomous driving, DNNs accurately recognize and understand road and traffic conditions by analyzing images and data from cameras and radar sensors. In the medical area [5]–[8], DNNs automatically identify and segment lesions in medical images, helping doctors diagnose and treat diseases more quickly and accurately. In the security field [9]–[12], DNNs achieve automatic alarm and tracking of security monitoring systems by recognizing and tracking objects such as faces and vehicles. In addition to the above application areas, DNNs have wide applications in many other fields. For example, DNNs are used for natural language processing for machine translation, sentiment analysis, and text generation tasks [13], [14]. In recommendation systems, DNNs can use to recommend products, music, movies, and other content. In industrial control, DNNs can use to predict machine failures and optimize production lines. However, as DNNs are increasingly applied in more fields, the threat of adversarial attacks is becoming more serious [15]–[20].

An adversarial attack is a malicious attack against machine learning models to deceive the model and cause it to make incorrect predictions. Adversarial attacks may have a serious impact on the security and privacy of a model. For example, when the adversarial attack is successfully executed against an image classification model, a harmless image may be incorrectly classified as a completely different object, leading to serious safety issues. In autonomous vehicles, attackers can deceive the vehicle's cameras by adding specific patterns and noise, resulting in traffic accidents. In facial recognition tasks, attackers may trick the system by adding invisible noise to human eyes, thus accessing sensitive personal information. The above examples demonstrate that we must consider the risk of adversarial attacks when developing and deploying DNNs.

Researchers have proposed many methods to defend against adversarial attacks [21], [22], including adversarial training [23], defensive regularization [24], and adversarial example detection [25], [26]. Adversarial training [27] is a commonly used method that enhances the robustness of DNNs by adding adversarial examples to the training data and alternating between adversarial and benign examples during

[1]Corresponding author is Xia Hui, e-mail: xiahui@ouc.edu.cn. Hui Xia and Rui Zhang contributed equally to this work. Our code is available at https://github.com/DreamyRainforest/Class_Activation_Mapping_Ensemble_Attack/tree/main

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.23164
www.ndss-symposium.org

training. Defensive regularization methods limit the mapping space between model input and output by introducing regularization terms into the loss function, thereby reducing the impact of adversarial examples. Adversarial example detection methods attempt to detect adversarial examples from input data to prevent them from entering the model. Although these methods can effectively improve the robustness of DNNs, adversarial attacks continue to evolve and pose significant challenges. Therefore, more efforts and exploration are still needed to research the security and robustness of DNNs.

Understanding the principles and methods of adversarial attacks, exploring possible attack methods, and mitigating potential risks are essential to enhance the security and robustness of DNNs. Various adversarial attack methods have been proposed, including optimization-based and gradient-based methods, which can achieve high success rates in the white-box setting [28]–[32]. However, in the black-box setting, the attack efficacy is lower due to the inability to access the target model's internal details. Some transferability-enhancing attack methods have been proposed to solve the above problem, including gradient optimization attack, input transformation attack, and model ensemble attack [33]–[38]. However, these methods add indiscriminate perturbations to all pixel locations in the image, resulting in generated adversarial examples with poor stealthiness and low attack capability for low-round attacks. In particular, compared to the two types of methods, gradient optimization attack and input transformation attack, the model ensemble attack is an efficient attack method and widely used to improve black-box attack performance. Therefore, in this work, we still focus on model ensemble attacks and attempt to perturb important locations to enhance the stealthiness and convergence speed of the adversarial examples' attack capability under low-round attacks.

Severi *et al.* [39] utilized machine learning interpretability tools, such as Shapley Additive exPlanations, to generate malware samples that would be misclassified as benign samples by a classifier. Inspired by their work, we attempt to employ the Class Activation Mapping (CAM) method to generate adversarial examples. The CAM method can reveal the connection between the decisions of DNNs and the regions in an image, resulting in adversarial examples with solid attack capabilities. However, the targeted nature of CAM scores may result in adversarial examples applying only to specific target models, leading to poor transferability. To address this problem, we incorporate CAM scores as weights for adding perturbations to each pixel, enhancing the attack capabilities for low attack epochs and stealthiness. Additionally, we improve the transferability of adversarial examples by ensembling CAM scores from multiple models. Thus, we propose a class activation mapping ensemble attack. We validate our attack method on the ILSVRC 2012 validation set and compare it with ten baseline methods regarding perceptibility and attack ability for two attack modes (targeted and non-targeted). Experimental results show that our attack method generates adversarial examples with good stealthiness and has good attack and transferability abilities for 13 models, including AlexNet [40], VGG16 [41], EfficientNet_b0 [42], ResNet18/34/50 [43], WideResNet50/101 [44], Inception_v2 [45], MobileNet_v2 [46], ConvNeXt [47], ViT [48], RegNet [49]. Our contributions are summarized as follows:

- As far as we know, this is the first black-box adversarial attack method that considers both attack transferability and perturbation weighting simultaneously.

- We use a strategy different from traditional methods to improve the stealthiness and transferability of adversarial attack algorithms. Traditional methods directly integrate the outputs of multiple models into the objective function, while we incorporate the CAM into the search process for minimum perturbation. This method can avoid adding excessive perturbations in unimportant regions, thereby preserving the stealthiness of the adversarial examples. It can quickly change the decision region of benign images to enable the attacking algorithm to exhibit good attack performance with low attack rounds. We also ensemble the CAM of multiple models to ensure the transferability of adversarial examples, thereby improving the success rate and robustness of the attack.

- Experimental results show that our method produces adversarial examples with good perceptibility, attack ability, transferability, convergence, and evasiveness. Specifically, when our attack ability is comparable to the most vigorous VMIFGSM attack, our perceptibility is close to the best-performing TPGD. Compared to VMIFGSM, L_2 is reduced by 24.08, Low_fre by 13.91, SSIM by 0.04, and PSNR by 0.69 in our method. Under non-targeted attack mode, compared to the VMIFGSM, our method improves the average attack success rate against 13 target models by 11.69%, and compared to the TPGD, the average attack success rate is enhanced by 37.15%. Under targeted attack mode, our method converges fastest and has the most substantial attack ability, and the attack ability is significantly better than the eight baseline methods under low attack rounds. The attack success rate of our method has been at least 10% higher than that of VMIFGSM since the fourth round. Also, our method can bypass defense methods, making it helpful in evaluating the robustness of models.

In the following chapters, we will briefly introduce our research method's main content and contributions. Specifically, this paper will be divided into the following sections: Section II will review existing research related to adversarial attacks, including gradient-based attacks, input transformation attacks, and model ensemble attacks. Section III will briefly introduce seven adversarial attack methods to help readers understand our attack strategy. Section IV will elaborate on our research method, a class activation mapping ensemble attack for generating adversarial examples. Section V will introduce the datasets, experimental settings, and evaluation metrics used in this study, as well as the detailed implementation of our method. Also, we will present the experimental results and compare the performance of different methods regarding attack effectiveness, transferability, and evasiveness. Finally, in Section VI, we will summarize the contributions and limitations of our method and present future research directions and recommendations.

II. Related Work

Research on adversarial attacks can be roughly divided into five categories: gradient-based attacks, optimization-based attacks, score-based attacks, decision-based attacks, and transferable attacks. We focus on transferable attacks, and in this section, we introduce existing transferability-enhancing methods from three aspects: gradient-based attacks, input transformation attacks, and model ensemble attacks.

A. Gradient Optimization Attack

The gradient-based attack method is a standard adversarial attack method that utilizes the gradient information of the target model to generate adversarial examples. The most famous way among them is the Fast Gradient Sign Method (FGSM) [50], which obtains a perturbation vector by multiplying the gradient direction of each pixel with the perturbation and then adds it to the benign image to generate an adversarial example. Although FGSM has a high success rate in attacking a single model, it cannot guarantee the transferability of the attack effect to other models. Kurakin *et al.* [51] proposed the Basic Iterative Method (BIM) by setting multiple smaller steps to construct more accurate adversarial examples. Madry *et al.* [52] searched for perturbations within the L-norm ball range of the data point and proposed Projected Gradient Descent (PGD). Although PGD has good attack ability in white-box settings, it is prone to overfitting the target model in black-box settings, leading to poor transferability of adversarial examples. To enhance the transferability of adversarial examples, Dong *et al.* [53] proposed a Momentum Iterative Fast Gradient Sign Method (MIFGSM), which stabilizes the update direction during iterations and avoids poor local maxima. Lin *et al.* [54] introduced Nesterov Accelerated Gradient into gradient-based attacks by effectively looking ahead to improve the transferability of adversarial examples. Wang *et al.* [55] introduced variance tuning in gradient-based iterative attacks to enhance the transferability of attacks. Although these methods improve the transferability of adversarial examples, the effects are insignificant.

B. Input Transform Attacks

The input transformation-based attack method enhances the attack's transferability by utilizing the image's invariance. This type of attack method mainly includes methods such as rotation, translation, and scaling to transform the image. Xie *et al.* [56] proposed the Diverse Input Method (DIFGSM), which increases the transferability of adversarial examples by using input diversity such as randomly adjusting size and padding to create different input patterns. Dong *et al.* [53] generated adversarial examples by optimizing perturbations on a set of translated images and proposed the Translation-Invariant Attack Method (TIFGSM). Lin *et al.* [54] demonstrated the scale invariance of deep learning models and proposed the Scale-Invariant Method (SINIFGSM) to improve the transferability of adversarial examples by optimizing adversarial perturbations on scaled copies of the input image. Wang *et al.* [57] proposed the Admix attack, which computes the gradients on the input image and mixes them with a small portion of each additional module image while using the original label of the input to generate more transferable adversarial examples. Although these methods improve the transferability of adversarial examples to some extent, the improvement is insignificant.

C. Model Ensemble Attacks

The model ensemble attack improves the transferability of the attack by attacking multiple models. This method can effectively bypass the defense mechanisms of different models, improving the success rate and transferability of the attack. Liu *et al.* [58] first proposed ensemble attacks by averaging the predictions (probability) of multiple models and using existing adversarial attack methods (such as FGSM and PGD) to improve the transferability of adversarial examples. Dong *et al.* [56] proposed two variants of multi-model ensemble attacks, which generate adversarial examples by integrating multiple models' logit outputs and losses to improve the transferability of adversarial examples. However, these methods uniformly integrate all models' outputs and use stochastic gradient descent for optimization can easily get trapped in local optima. Therefore, Xiong *et al.* [59] proposed a Stochastic Variance-Reduced Ensemble attack (SVRE), but the efficiency of generating adversarial examples is slow, and the attack capability of adversarial examples is not significant.

Although existing methods have shown some improvement in the transferability of adversarial examples, the effect is insignificant. Moreover, these methods indiscriminately add perturbations to all pixel positions in the image, resulting in poor perceptual quality and low attack capability at low iteration counts. In summary, the current methods could be improved in improving adversarial examples' transferability and stealthiness.

Inspired by the above approach, we are still committed to using model ensemble attacks to enhance the transferability of adversarial examples. However, unlike the previous method, we do not ensemble multiple target models' logit outputs, predictions, and losses, but rather the CAMs of numerous models. By integrating the CAMs of various models, we can ensure the transferability of adversarial examples and improve the attack capability of adversarial examples at low attack rounds while improving the stealthiness of adversarial examples to some extent.

III. TECHNICAL BACKGROUND

We briefly introduce seven adversarial attack methods: FGSM, PGD, TPGD, NIFGSM, MIFGSM, VMIFGSM, and SVRE. This is to help readers understand our attack strategy.

A. Fast Gradient Sign Method (FGSM)

The FGSM [50] is a standard adversarial attack that can deceive DNNs into producing incorrect outputs. FGSM can be described as follows: assuming that $\mathbf{x}$ is the benign image, $\mathbf{y}$ is the true label, and $\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)$ is the loss function, θ is the model parameters. The process of generating an adversarial example with FGSM is as follows:

$$\mathbf{x}' = \mathbf{x} + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)) \tag{1}$$

where $\mathbf{x}'$ is the perturbed image, α is the magnitude of the perturbation, $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)$ is the gradient of the loss function

$\mathcal{L}(\mathbf{x}, \mathbf{y}; \theta)$ with respect to the input image, $\text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta))$ is the sign of the gradient element-wise. This method is computationally efficient because it only requires one forward and backward propagation to generate an adversarial example. However, the choice of the perturbation magnitude may affect the attack effectiveness, and it needs to be selected experimentally or by other methods.

B. Projected Gradient Descent (PGD)

The main idea of PGD [52] is to iteratively add a certain amount of perturbation within a specific range to the input image to cause misclassification. The optimization objective is:

$$\min_{\mathbf{x}'} \{\mathcal{L}(\mathbf{F}(\mathbf{x}'), \mathbf{y}; \theta)\} \quad \text{s.t.,} \quad \|\mathbf{x}' - \mathbf{x}\|_\infty \leq \epsilon \tag{2}$$

$\mathbf{F}$ is the target model, ϵ is the maximum perturbation range, $\|\cdot\|_p$ represents the L_p norm. To solve the optimization problem, PGD iteratively updates the perturbation using the gradient descent method, i.e.,

$$\mathbf{x}_{t+1} = \Pi_{\mathbf{x}+\epsilon}(\mathbf{x}_t + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}_t} \mathcal{L}(\mathbf{F}(\mathbf{x}_t), \mathbf{y}; \theta))) \tag{3}$$

where $\mathbf{y}$ is the true label, $\mathbf{x}_t$ is the sample obtained after the t-th iteration, α is the learning rate, $\text{sign}(\cdot)$ represents the sign function, and $\Pi_{\mathbf{x}+\epsilon}(\cdot)$ represents the projection operation. Overall, PGD is a powerful and flexible iterative adversarial attack algorithm that gradually approaches the optimal solution by updating the perturbation multiple times, generating more effective adversarial examples.

C. Theoretically Grounded Approach (TPGD)

The author discusses the trade-off between adversarial robustness and accuracy in DNNs [60]. The authors propose a theoretically grounded approach that balances the trade-off by modifying the training data. Specifically, the authors define a loss function that combines adversarial robustness and accuracy, called TPGD. The form of the loss function is:

$$\mathcal{L}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim D}[\max_{\delta \in \mathcal{S}} \mathcal{L}_{CE}(\mathbf{F}(\mathbf{x}+\delta), \mathbf{y}) - \xi R(\delta)] \tag{4}$$

D is the distribution of the training data, $\mathcal{L}_{CE}$ is the cross-entropy loss function, $\mathcal{S}$ is the set of adversarial examples, $R(\delta)$ is the robustness evaluation of adversarial perturbation δ, ξ is a balancing factor. The meaning of this loss function is to maximize the cross-entropy loss function for correctly predicted examples among all adversarial examples, while simultaneously minimizing the robustness evaluation of the adversarial examples to achieve a balance between adversarial robustness and accuracy. The authors prove that this loss function can be minimized under certain conditions and propose an optimization algorithm to solve this minimization problem.

D. Nesterov Accelerated Gradient Fast Gradient Sign Method (NIFGSM)

The authors propose an adversarial attack method based on Nesterov accelerated gradient and scale invariance, called NIFGSM [54]. Mathematically, this method can be represented by the following equation:

$$\mathbf{x}' = \mathbf{x} + \alpha \cdot \text{sign}(g_t) \tag{5}$$

where g_t is the gradient computed using Nesterov accelerated gradient method. Specifically, this method uses the Nesterov Accelerated Gradient Method to calculate the adversarial gradient quickly and uses scale invariance to ensure the stability of the attack effect. The Nesterov Accelerated Gradient Method is a momentum-based optimization algorithm that can accelerate the convergence speed of gradient descent. Its mathematical expression is as follows:

$$\mathbf{v}_{t+1} = \mu \cdot \mathbf{v}_t - \eta \cdot \nabla_{\mathbf{x}} \mathbf{F}(\mathbf{x}_t + \mu \cdot \mathbf{v}_t) \tag{6}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1} \tag{7}$$

where μ is the momentum parameter, η is the learning rate.

Scale invariance means that the attack effect should remain stable for different scales of input data. Scale invariance is achieved by dividing the perturbation into two parts: a global scaling and a local perturbation. Specifically, the attack perturbation can be represented as:

$$d = \alpha \cdot \text{sign}(g_t) \odot \frac{\|\mathbf{x}\|_2}{\|g_t\|_2} \tag{8}$$

where $\odot$ represents element-wise multiplication, $\|\mathbf{x}\|_2$ and $\|g_t\|_2$ represent the L_2 norms of the input data and the adversarial gradient. Combining Nesterov's accelerated gradient method and scale invariance achieves good performance in both adversarial attack effectiveness and computational efficiency.

E. Moment Iterative Fast Gradient Sign Method (MIFGSM)

The MIFGSM [53] is a commonly targeted adversarial attack algorithm against DNNs, based on the FGSM and the Momentum Gradient Descent algorithm. The basic idea of MIFGSM is to add a momentum term to FGSM to speed up the attack and increase the attack's success rate. Its mathematical expression is:

$$\begin{aligned} \mathbf{x}_{t+1} &= clip(\mathbf{x}_t + \alpha \cdot \text{sign}(g_t)) \\ g_t &= \mu \cdot g_{t-1} + \frac{\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_t, \mathbf{y}; \theta)}{||\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_t, \mathbf{y}; \theta)||_1} \end{aligned} \tag{9}$$

where μ is the momentum factor used to control the smoothness of the perturbation, the clip function is used to truncate $\mathbf{x}_t$ to ensure it is within the range of $[\mathbf{x} - \epsilon, \mathbf{x} + \epsilon]$. MIFGSM algorithm introduces a momentum term, enabling the attacker to find the path to a successful attack faster and avoid getting stuck in local minima. Additionally, MIFGSM can increase the attack success rate and amplitude by increasing the number of epochs, but it also increases attack time and cost.

F. Variance-based Moment Iterative Fast Gradient Sign Method (VMIFGSM)

VMIFGSM [55] is an improved targeted attack algorithm for DNNs. It introduces variance information based on the MIFGSM algorithm to better explore the model's gradient information and accelerate the attack process. The mathematical

expression of the VMIFGSM algorithm is as follows:

$$\begin{aligned} \mathbf{x}_{t+1}^{'} &= \mathbf{x}_{t}^{'} + \alpha \cdot sign(g_{t+1}) \\ \hat{g}_{t+1} &= \nabla_{\mathbf{x}'} \mathcal{L}(\mathbf{x}_{t}^{'}, \mathbf{y}; \theta) \\ g_{t+1} &= \mu \cdot g_t + \frac{\hat{g}_{t+1} + v_t}{||\hat{g}_{t+1} + v_t||_1} \\ v_{t+1} &= V(\mathbf{x}_{t}^{'}) \end{aligned} \tag{10}$$

where v_{t+1} represents the gradient variance of the *N* samples, $V(\cdot)$ is the approximate of gradient variance, i.e., $V(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\mathbf{x}^i} L(\mathbf{x}^i, \mathbf{y}) - \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y})$, *N* represents the number of sampled examples. The dynamic step size setting can help attackers better explore the gradient information, accelerate the attack process, and improve the attack success rate.

G. Stochastic Variance Reduced Ensemble Attack (SVRE)

The authors propose an adversarial attack method based on an ensemble with random variance reduction to improve the transferability of adversarial examples, called SVRE [59]. This method includes three key components: random perturbation, variance reduction, and ensemble. Specifically, random perturbation is used to increase the diversity of attack samples, variance reduction is used to reduce the impact of noise and improve the efficiency and stability of the attack, and the ensemble combines the prediction results of multiple models to improve the success rate and transferability of the attack. The goal of this method is to minimize a loss function on the attack samples, which consists of the distance between the output of the original model and the adversarial model and a regularization term.

To improve attack efficiency, the authors use a stochastic gradient descent-based optimization method with a mini-batch technique to randomly select a subset of data for an update. In addition, a new ensemble strategy, weighted averaging, is introduced to balance the contributions of different models. Finally, experiments on multiple datasets show that compared with other adversarial attack methods, this method has a higher attack success rate and better transferability.

IV. Methodology

This section overviews the proposed method from problem definition and class activation mapping ensemble attack. Firstly, the problem is defined by presenting the objective functions for targeted and non-targeted attacks, which introduces the problem that needs to be addressed. Secondly, the class activation mapping is introduced to enhance the method's attack performance and integrate the CAM into finding the minimum perturbation that avoids adding excessive perturbations in unimportant regions, thus ensuring the stealthiness of adversarial examples.

A. Problem Definition

Adversarial attacks can be viewed as an optimization problem. The goal is to deceive a DNN by applying small perturbations to the input image, resulting in a significantly different output than the original. Specifically, adversarial attacks can be defined as follows: given a target model $\mathbf{F}$ with parameters θ and an input image $\mathbf{x}$, find a small perturbation δ such that the output of $\mathbf{x}$ is significantly different from the output of $\mathbf{x}^{'}, \mathbf{x}^{'} = \mathbf{x} + \delta$, while the magnitude of the perturbation δ is small enough to be imperceptible to humans. Where ϵ is the perturbation constraint. This can be formulated as the following constrained minimization problem:

$$\begin{aligned} &\min_{\delta} d(\mathbf{x}^{'}, \mathbf{x}) \\ &s.t., \mathbf{F}(\mathbf{x}; \theta) \neq \mathbf{F}(\mathbf{x}^{'}; \theta) \end{aligned} \tag{11}$$

where $d(\mathbf{x}^{'}, \mathbf{x})$ represents the distance between $\mathbf{x}$ and $\mathbf{x}^{'}$, which can be Euclidean distance, L_p norm distance, etc.

For the non-targeted attack, the purpose of a non-targeted attack is to deceive a DNN by causing it to predict an incorrect label by maximizing the prediction error. The objective function for a non-targeted attack can be formulated as the following optimization problem:

$$\mathbf{x}^{'} = \underset{||\mathbf{x}-\mathbf{x}'||<\epsilon}{\arg\max}\ \mathcal{L}(\mathbf{x}^{'}, \mathbf{y}; \theta) \tag{12}$$

For the targeted attack, the goal of the attack is to find the closest $\mathbf{x}^{'}$ to the input image $\mathbf{x}$, such that the perturbed image $\mathbf{x}^{'}$ is classified by the model $\mathbf{F}$ as the targeted label $\mathbf{y}^{'}$. Assuming the input image $\mathbf{x}$, the specified targeted label $\mathbf{y}^{'}$, and the model's classification function $\mathbf{F}$, the targeted attack can be formulated as the following optimization problem:

Algorithm 1 Class Activation Mapping Ensemble Attack

Require: Target model $\mathbf{F}$, loss function $\mathcal{L}$, input image $\mathbf{x}$, true label $\mathbf{y}$, perturbation constraint ϵ, attack epochs *T*, decay factor μ.
Ensure: Adversarial example $\mathbf{x}^{'}$.

```
α = ε/T;
g_0 = 0;
v_0 = 0;
x_0 = x;
for t = 0 → T − 1 do
    Compute gradient with Eq. (17);
    Update gradient with Eq. (16);
    Update x'_{t+1} with Eq. (15);
end for
x' = x'_T;
Return x'.
```

$$\mathbf{x}^{'} = \underset{||\mathbf{x}-\mathbf{x}'||<\epsilon}{\arg\max}\ \mathcal{L}(\mathbf{x}^{'}, \mathbf{y}^{'}; \theta) \tag{13}$$

To improve the stealthiness of the adversarial example, attackers often need to add some constraints, such as limiting the size of the perturbation, to ensure that humans do not observe the perturbation. Different adversarial attack algorithms use other loss functions and constraints. For example, gradient-based adversarial attack algorithms such as the FGSM and PGD use cross-entropy loss and norm constraints to maximize the error and limit the perturbation size. However, their attack capability is defined by gradient information. On the other hand, evolutionary algorithms-based adversarial attack algorithms use the objective function to guide the search process to find the most aggressive perturbation, but their computational costs are often high. Although the aforementioned methods

have made great progress in improving the perception of adversarial examples, the adversarial examples generated by the above method have relatively weak attack ability in a black-box setting.

Researchers have proposed various methods to enhance the performance and transferability of adversarial attack algorithms. One commonly used way is integrating multiple models to improve the model's robustness and attack capability. For example, various models can be trained simultaneously in adversarial training, and then their outputs are integrated to obtain stronger robustness. In adversarial attacks, the model ensemble attack method can enhance attack capability. This method produces adversarial examples by weighted averaging the outputs of multiple models, thereby improving attack performance. Although these methods have some improvement in attack transferability, they all have the drawbacks of poor low iteration attack capability, slow convergence speed, and poor stealthiness.

The CAM can discover the relationship between a DNN's decision and the image region and visualize the model's output based on its feature map, thereby providing attackers with more information to construct adversarial examples. Therefore, we use the CAM to create adversarial examples, which can effectively enhance the attack capability of adversarial attack methods. We still focus on model ensemble attacks to improve the transferability of adversarial examples. However, unlike existing research on model ensemble attack methods (such as integrating multiple models' logit outputs, prediction probabilities, and losses), we solve the problem of poor transferability by integrating multiple models' CAMs to construct adversarial examples.

B. Class Activation Mapping Ensemble Attack

Overview: In the black-box attack setting, attackers can only access the input and output of the target model without obtaining its internal structure and parameter information. Therefore, we need to construct substitute models to generate adversarial examples to deceive targeted DNNs. We first train substitute models (gradient substitute model and CAMs substitute model) and use the CAM method to calculate the class activation score of each pixel, which is used as the weight of the perturbation to ensure the stealthiness of the adversarial examples and enhance the attack performance under low attack epochs. In the optimization process, we integrate the CAMs [61] of multiple CAM-based substitute models to ensure the transferability of the adversarial attack algorithm. Compared to traditional methods, using CAM scores as perturbation weights can effectively avoid adding excessive perturbations in unimportant regions, thereby ensuring the stealthiness of the adversarial examples. Meanwhile, this method can add perturbations in a targeted manner and quickly change the decision region of benign images, enabling the adversarial attack method to exhibit good attack performance even under low iteration times.

Substitution model: When selecting a substitute model, various factors, such as the complexity of the model, the similarity to the target model, and the availability of training data, need to be considered. Generally, the more similar the substitute model is to the target model, the more effective the generated adversarial examples will be. Based on prior knowledge, we choose a gradient substitute model, which is used to determine the sign direction of perturbation, and select the model with larger heatmap regions as the CAMs substitute model, which calculates the weights of the perturbations.

Non-targeted attack: Given the target model $\mathbf{F}$ and the input image $\mathbf{x}$, find a perturbed image $\mathbf{x}^{'}$ such that the output of $\mathbf{x}$ and the output of $\mathbf{x}^{'}$ differ significantly, while the perturbation in $\mathbf{x}^{'}$ is small enough not to draw human attention, and without specifying a specific deceived output. Specifically, the non-targeted attack can be represented as the following minimization-constrained problem:

$$\mathbf{x}^{'} = \underset{||\mathbf{x}-\mathbf{x}^{'}||<\epsilon}{\arg\max}\ \mathcal{L}(\mathbf{x}^{'}, \mathbf{y}; \theta) + ||\tilde{x} - \mathbf{x}^{'}||_2 \tag{14}$$

where $\tilde{x}$ is the maximum of ensemble class activation maps,

$$\tilde{x} = \max\{x^1_{cam}, x^2_{cam}, \cdots, x^n_{cam}\}$$

x_{cam} is the score of the class activation mapping, n is the number of CAM-based substitute models.

The first term is used to improve the attack capability of adversarial examples, while the second term is used to improve their transferability among different models. We use the gradient variance optimization method to optimize the objective function. However, differently from the method in [53], we incorporate CAMs [61] as the weight of the perturbation to further enhance the optimization effect. Specifically, we multiply the CAM with the perturbation to obtain a new perturbation vector. Then we use the sign of the gradient variance of the gradient substitute model as the direction for updating the perturbation, thereby improving the adversarial attack ability of the adversarial examples. That is,

$$\mathbf{x}^{'}_{t+1} = \mathbf{x}^{'}_t + (\lambda \cdot \alpha) \cdot (w \cdot \mathcal{M}_c) \cdot sign(g_{t+1}) \tag{15}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\hat{g}_{t+1} + v_t}{||\hat{g}_{t+1} + v_t||_1} \tag{16}$$

$$\hat{g}_{t+1} = \nabla_{\mathbf{x}^{'}_t} \mathcal{L}(\mathbf{x}^{'}, \mathbf{y}; \theta) + ||\tilde{x} - \mathbf{x}^{'}||_2 \tag{17}$$

$$v_{t+1} = \frac{1}{N} \sum_{i=1}^{N} \nabla_{\mathbf{x}^i} \mathcal{L}(\mathbf{x}^i, \mathbf{y}; \theta) - \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta) \tag{18}$$

where λ represents the perturbation step size factor, w represents the perturbation magnitude factor used to constrain the magnitude of perturbation, $\mathcal{M}_c$ represents the perturbation weight of pixels, i.e., the score of class activation map of CAMs substitute model,

$$\mathcal{M}_c(a,b) = \max\{m^1_c(a,b), m^2_c(a,b), \cdots, m^n_c(a,b)\} \tag{19}$$

$$m_c(a,b) = \sum_{k=1}^{K} (\alpha^k_c \mathbf{F}^{lk}(a,b)) \tag{20}$$

where c represents the class of interest in addition to the current class,

$$\alpha^k_c = \sum_{a,b} \left(\frac{\mathbf{F}^{lk}(a,b)}{\sum_{a,b} \partial \mathbf{F}^{lk}(a,b)} \frac{\partial S_c(\mathbf{F}^l)}{\partial \mathbf{F}^{lk}(a,b)} \mathbf{F}^{lk}(a,b)\right) \tag{21}$$

where l represents the index of the target layer, $\mathbf{F}^l$ represents the response of the target layer, $\mathbf{F}^{lk}$ represents the response of

TABLE I: Non-targeted Attack: Perceptive measure (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Target model: ResNet50, $\epsilon = 16/255$, $\alpha = 1/255$, $\lambda = 0.75$, $w = 2$).

Metric \ Attack	TPGD [60]	PGD [52]	TIFGSM [53]	DIFGSM [56]	NIFGSM [54]	MIFGSM [53]	SINIFGSM [54]	VMIFGSM [55]	VNIFGSM [55]	OUR
PSNR	**31.54**	29.60	30.75	29.60	29.32	29.39	29.20	29.41	29.31	**30.10**
MSE	**0.0008**	0.0012	0.0009	0.0012	0.0012	0.0012	0.0013	0.0012	0.0012	**0.0011**
L_2	**120.02**	175.88	139.00	175.58	185.30	182.62	190.08	182.64	186.75	**158.56**
L_∞	**0.2751**	0.2813	0.2765	0.2828	0.278	0.2848	0.2816	0.2828	0.2842	**0.2795**
Low_fre	**47.27**	69.90	72.98	70.91	78.79	78.27	82.14	82.16	83.45	**68.25**
SSIM	**0.8900**	0.8100	0.8900	0.8100	0.7900	0.8000	0.7900	0.8100	0.8022	**0.8500**
AASR	37%	49%	53%	60%	62%	63%	69%	**71%**	**71%**	**71%**

the *k*th feature map in layer *l*, $\mathbf{F}^{lk}(a,b)$ represents the response at position (*a*, *b*) in feature map $\mathbf{F}^{lk}$, and $S_c(\mathbf{F}^l)$ represents the score of the interested class,

$$S_c(\mathbf{F}^l) = \sum_{k=1}^{K} \sum_{a,b} \left(\frac{\partial S_c(\mathbf{F}^l)}{\partial \mathbf{F}^{lk}(a,b)} \mathbf{F}^{lk}(a,b)\right) + \mathbf{\Phi}(\mathbf{F}^l) \tag{22}$$

$$\mathbf{\Phi}(\mathbf{F}^l) = \sum\nolimits_{t=l+1}^{L} \sum\nolimits_{j} \frac{\partial S_c(\mathbf{F}^l)}{\partial \mu_j^t} b_j^t \tag{23}$$

where μ_j^t represents the unit in the *t*-th layer, and b_j^t represents the offset of the unit in the *t*-th layer. The pseudo-code for the class activation mapping ensemble attack is shown in Algorithm 1.

Targeted attack: Give the input image $\mathbf{x}$, the target label $\mathbf{y}^{'}$, and the model $\mathbf{F}$, the objective function for targeted attack can be derived as follows:

$$\mathbf{x}^{'} = \underset{||\mathbf{x}-\mathbf{x}^{'}||<\epsilon}{\arg\max} -\mathcal{L}(\mathbf{x}^{'}, \mathbf{y}^{'}; \theta) - ||\tilde{x}_{tar} - \mathbf{x}^{'}||_2 \tag{24}$$

where $\tilde{x}_{tar}$ is the integrated class activation map of the target class,

$$\tilde{x}_{tar} = \max\{x_{tar\_cam}^1, x_{tar\_cam}^2, \cdots, x_{tar\_cam}^n\}$$

x_{tar_cam} is the score of the targeted class activation mapping. It should be noted that

$$\mathbf{x}_{t+1}^{'} = \mathbf{x}_t^{'} + (\lambda \cdot \alpha) \cdot (w \cdot \mathcal{M}_{tc}) \cdot sign(g_{t+1})$$

$$\mathcal{M}_{tc}(a,b) = \max\{m_{tc}^1(a,b), m_{tc}^2(a,b), \cdots, m_{tc}^n(a,b)\}$$

m_{tc} is the activation map score for the specified class. Where

$$m_{tc}(a,b) = \sum_{k=1}^{K} (\alpha_{tc}^k \mathbf{F}^{lk}(a,b))$$

$$\alpha_{tc}^k = \sum_{a,b} \left(\frac{\mathbf{F}^{lk}(a,b)}{\sum_{a,b} \partial \mathbf{F}^{lk}(a,b)} \frac{\partial S_{tc}(\mathbf{F}^l)}{\partial \mathbf{F}^{lk}(a,b)} \mathbf{F}^{lk}(a,b)\right)$$

$$S_{tc}(\mathbf{F}^l) = \sum_{k=1}^{K} \sum_{a,b} \left(\frac{\partial S_{tc}(\mathbf{F}^l)}{\partial \mathbf{F}^{lk}(a,b)} \mathbf{F}^{lk}(a,b)\right) + \mathbf{\Phi}(\mathbf{F}^l)$$

$$\mathbf{\Phi}(\mathbf{F}^l) = \sum\nolimits_{t=l+1}^{L} \sum\nolimits_{j} \frac{\partial S_{tc}(\mathbf{F}^l)}{\partial \mu_j^t} b_j^t$$

V. Experiment

We validate our method mainly through five parts: experimental settings, perceptual evaluation, attack performance analysis, robustness analysis, and ablation studies. In the experimental settings, we introduce the dataset, evaluation metrics, baseline methods, and defense methods. In the perceptual evaluation section, we employ various visual perceptual metrics to analyze the perceptual differences between adversarial examples and benign images. The attack performance analysis section primarily assesses the attack capability of the proposed method in non-targeted and targeted attacks. In the robustness analysis section, we evaluate the effectiveness of the proposed attack against various defense methods. In the ablation studies section, we analyze the rationality of using CAMs as adversarial perturbation weights and investigate the impact of different modules and parameter values on the attack performance of our method.

A. Experimental Settings

Dataset: We validate the proposed method on the ILSVRC 2012 validation set and the data subsets provided in references [53]–[55] while ensuring that each developed model correctly classifies all selected test images.

Models: To differentiate the roles of different models, we categorize the models into the following three types: CAMs substitute models, which are employed to compute perturbation weights; gradient substitute models, which are used to calculate gradient signs when crafting adversarial examples; and target models, solely employ to evaluate the attack performance of the proposed method, without access to any information about the model. It is worth emphasizing that all the models we used, including CAMs substitute, gradient substitute, and target models, are pre-trained models specifically designed for the Imagenet dataset and provided within the PyTorch library. We employ WideResNet101 [44], Inception_v2 (Inception) [45], and ResNet34 [43] as CAMs substitute models, and ResNet50 [43] as the gradient substitute model. We also regard models such as AlexNet [40], VGG16 [41], EfficientNet_b0 (EfficientNet) [42], WideResNet50 [44], MobileNet_v2 (MobileNet) [46], ResNet18 [43], ConvNeXt [47], ViT [48], and RegNet [49] as target models.

Metrics: The perceptual metrics include PSNR, Mean Squared Error (MSE), SSIM, Low_fre, CIEDE2000, L_2 norm,

Benign image	DIFGSM	SINIFGSM	TPGD	PGD
				
Label: 131	Label: 133	Label: 728	Label: 134	Label: 133
L_2: 0	L_2: 210	L_2: 162	L_2: **149**	L_2: 210
SSIM: 1.00	SSIM: 0.8601	SSIM: 0.9070	SSIM: **0.9185**	SSIM: 0.8261
PSNR: INF	PSNR: 28.55	PSNR: 29.68	PSNR: **30.04**	PSNR: 28.56
ASR: ----	ASR: 1.00	ASR: 0.9750	ASR: 0.6333	ASR: 1.00
VMIFGSM	**VNIFGSM**	**NIFGSM**	**MIFGSM**	**OUR**
				
Label:133	Label: 133	Label: 133	Label:133	Label: 133
L_2: 162	L_2: 161	L_2: 161	L_2: 161	L_2: **154**
SSIM: 0.9072	SSIM: 0.9063	SSIM: 0.9055	SSIM: 0.9048	SSIM: **0.9132**
PSNR: 29.71	PSNR: 29.69	PSNR: 29.72	PSNR: 29.69	PSNR: **29.88**
ASR: 1.00	ASR: 1.000	ASR: 1.000	ASR: 1.00	ASR: 1.00

Fig. 1: Adversarial example generated by our method and baseline methods under non-targeted attack. L_2, SSIM, and PSNR represent perceptual metrics, and 'ASR' represents the success rate of attack against the target model (Target model: ResNet50, Attack mode: non-targeted attack, Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/255$, $epoch = 10$, $\lambda = 0.75$, $w = 2$).

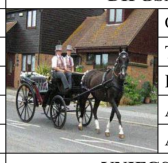
Benign image	DIFGSM	SINIFGSM	NIFGSM	PGD
				
Ori-label: 603	Ori-label: 603	Ori-label: 603	Ori-label: 603	Ori-label: 603
Tar-label: ---	Tar-label: 256	Tar-label: 256	Tar-label: 256	Tar-label: 256
Epoch: ---	Epoch: 11	Epoch: 11	Epoch: 11	Epoch: 11
Adv-label: ---	Adv-label: 603	Adv-label: 603	Adv-label: 256	Adv-label: 603
ASR: ----	ASR: 0.2750	ASR: 0.3667	ASR: 0.4917	ASR: 0.2333
VMIFGSM	**VNIFGSM**	**MIFGSM**	**OUR**	**OUR**
				
Ori-label: 603	Ori-label: 603	Ori-label: 603	Ori-label: 603	Ori-label: 603
Tar-label: 256	Tar-label: 256	Tar-label: 256	Tar-label: 256	Tar-label: 256
Epoch: 11	Epoch: 11	Epoch: 11	Epoch: **9**	Epoch: 11
Adv-label: 537	Adv-label: 537	Adv-label: 256	Adv-label: 256	Adv-label: **256**
ASR: 0.4333	ASR: 0.4417	ASR: 0.5250	ASR: 0.5167	ASR: **0.6417**

Fig. 2: Adversarial examples generated by our method and baseline methods under targeted attack. 'Ori-label' represents the original label of the image, 'Tar-label' represents the targeted label we specified, 'Epoch' represents the iteration round of the attack, and 'Adv-label' represents the predicted label of ResNet50 (Target model: ResNet50, Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 4$).

and L_∞ norm. The attack capability metrics include the Attack Success Rate (ASR) against one target model and the Average Attack Success Rate (AASR) against 13 target classifiers.

Baseline: We evaluate ten baseline methods in our study, including PGD [52], TPGD [60], DIFGSM [56], TIFGSM [53], MFGSM [53], NIFGSM [54], SINIFGSM [54], VMIFGSM [55], VNIFGSM [55], SVRE [59]. The ensemble model of SVRE are Inception_v3, Inception_v4, InceptionResnet_v2, and ResNet101.

Defense: We test seven defense methods, including Feature Squeezing (Fea. Squ.) [62], Label smoothing(Lab. Smo.) [63], Engstrom [64], Salman [65], Singh [66], Liu [67], and Shan [68]. The Fea. Squ., Engstrom, and Lab. Smo. methods utilize ResNet50 as the target model. Salman's method uses WideResNet50 as the target model. Singh utilizes ViT, and Liu employs ConvNeXt as the target model. Shan's target model is the same as work [68].

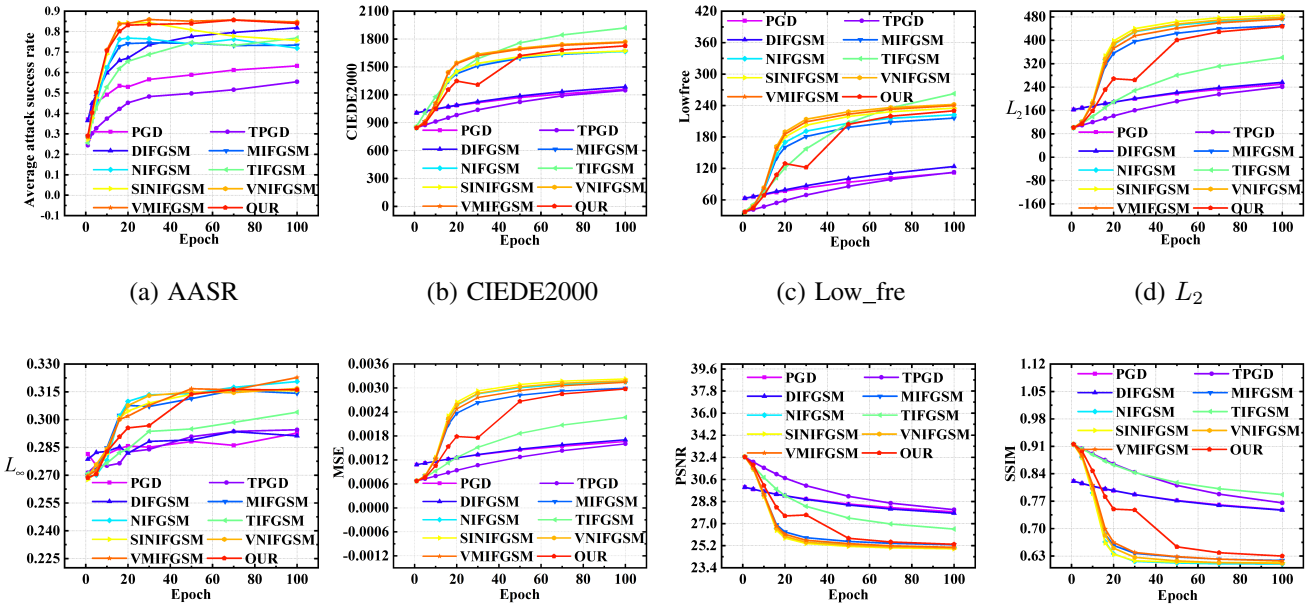
Parameter: We set the perturbation constraint for all attack methods, $\alpha = 1/500$ or $1/255$, $\epsilon = 16/255$, $\mu = 1$, $N = 5$, $w = 2$ or 4.

B. Perceptual Evaluation

Fig. 1 and Fig. 2 depict the adversarial examples generated by our method and baseline methods under non-targeted and targeted attacks. Please refer to Fig. 1 in the Appendix for the results of the baseline methods converging to the target label. To validate the perceptibility of our method, we employ six distance metrics to analyze the perceptibility of adversarial examples under non-targeted and targeted attacks, as shown in Table I and Table II. Table I presents the perceptibility of the adversarial examples generated by ten attack methods with similar attack capabilities under non-targeted attacks. Specifically, when our attack capability is comparable to VMIFGSM and VNIFGSM. Compared with VMIFGSM, our perceptibility drops by 24.08 for L_2, decreases by 13.91 for Low fre, increases by 0.04 for SSIM, and increases by 0.69 for PSNR. Table II demonstrates the perceptibility of the adversarial examples generated by nine attack methods with similar attack capabilities under targeted attack mode. This table shows that our method achieves the best perceptibility among the eight baseline methods with comparable attack capabilities, while SINIFGSM performs the worst. Notably, compared to VNIFGSM, our method improves the PSNR metric by 2.12 and reduces the SSIM by 0.088 and Low_fre

TABLE II: Targeted Attack: Perceptive measure (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Target model: ResNet50, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 4$).

Metric \ Attack	PGD [60]	TIFGSM [53]	DIFGSM [56]	NIFGSM [54]	MIFGSM [53]	SINIFGSM [54]	VMIFGSM [55]	VNIFGSM [55]	OUR
PSNR	29.53	28.46	29.28	25.40	25.28	25.25	28.44	28.26	**30.38**
MSE	0.0012	0.0015	0.0012	0.0029	0.0030	0.0030	0.0015	0.0016	**0.0010**
L_2	178.01	226.04	187.76	438.01	450.13	453.71	224.41	233.84	**149.55**
L_∞	0.2797	0.2865	0.2818	0.3070	0.3064	0.3081	0.2875	0.2860	**0.2794**
Low_fre	70.72	156.72	79.58	189.33	196.84	213.69	111.75	116.93	**64.48**
SSIM	0.8067	0.8514	0.8017	0.6092	0.6029	0.6195	0.7826	0.7737	**0.8616**
AASR	39%	57%	58%	64%	67%	65%	65%	65%	65%



(a) AASR (b) CIEDE2000 (c) Low_fre (d) L_2

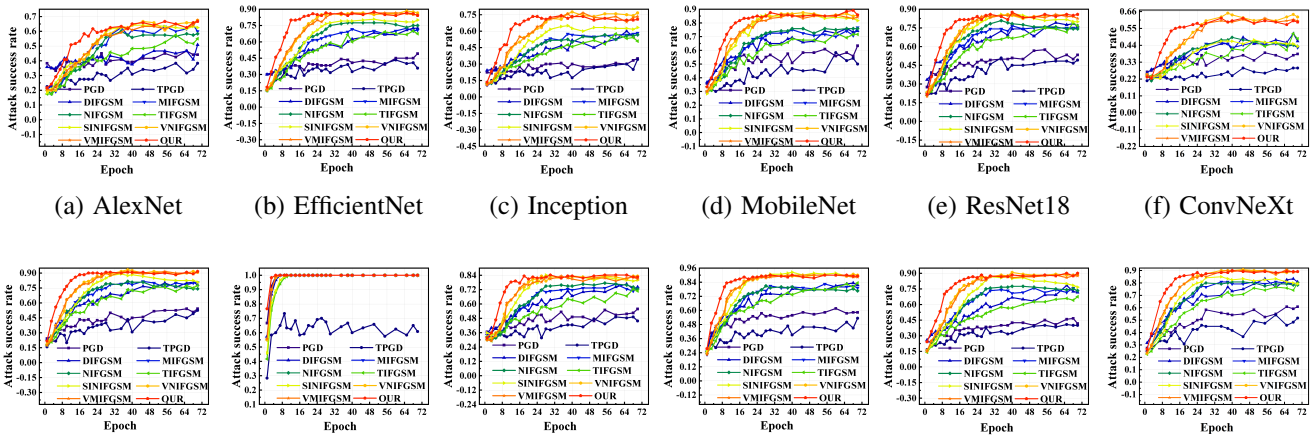
(e) L_∞ (f) MSE (g) PSNR (h) SSIM

Fig. 3: Non-targetrd Attack: Perceptive measure (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/255$, $\lambda = 0.75$, $w = 2$).

by 52.45. Compared to SINIFGSM, our method improves the PSNR metric by 5.13 and reduces the L_2 by 304.16 and Low_fre by 149.21.

To further analyze the perceptibility of adversarial examples under non-targeted attacks, we control the attack capability of our method to be similar to the most vigorous VMIFGSM by adjusting the value of λ and evaluate the perceptibility of adversarial examples at different epochs, as shown in Fig. 3. From Fig. 3(a), it can be seen that the VMIFGSM and our method have the most vigorous attack capability, while the PGD and TPGD have weaker attack capabilities. By analyzing Fig. 3(b) to 3(h), it can be observed that the perceptibility of the ten methods gradually decreases as the number of epochs increases and the perturbation increases. Among the nine baseline methods, the TPGD generates adversarial examples with the best visual perceptibility and lowest perturbation, followed by the PGD and DIFGSM. Regarding the four metrics of MSE, PSNR, SSIM, and L_2, the visual perceptibility of SINIFGSM is the worst, and the perturbation is the largest, while our perceptibility is between the above-mentioned methods. Due to the computational complexity involved in achieving comparable attack capability to the most vigorous baseline attack under targeted attacks, we do not constrain our method to have a similar attack capability in showcasing the changes in the perceptibility of adversarial examples. For the changes in the perceptibility of adversarial examples under targeted attacks, please refer to the Fig. 2 in the appendix.

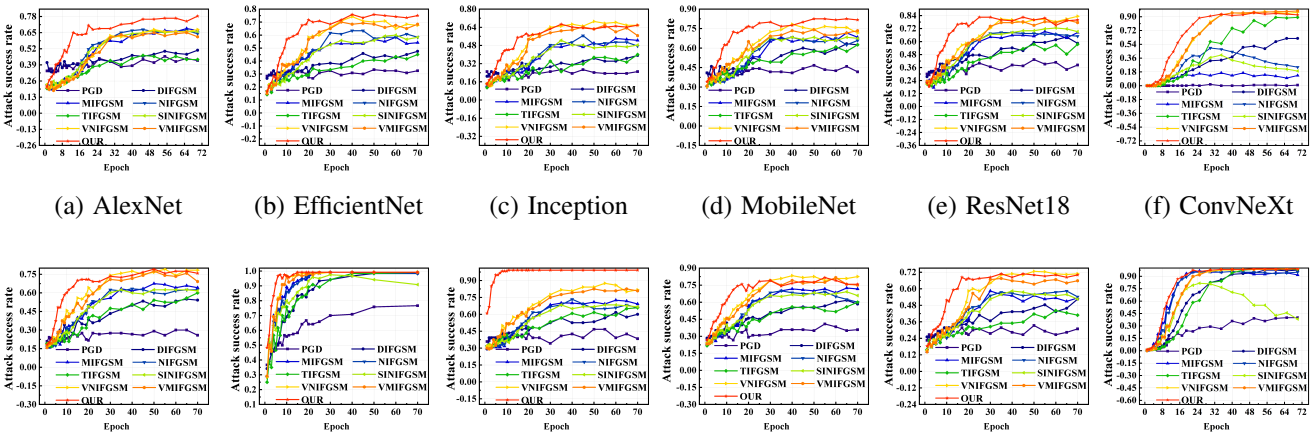
In conclusion, our method exhibits good stealthiness under the non-targeted and targeted attacks compared to the baseline methods when they have similar attack effectiveness. This is attributed to our utilization CAM is the weighting factor for perturbations, which avoids adding perturbations in unimportant regions.



(a) AlexNet (b) EfficientNet (c) Inception (d) MobileNet (e) ResNet18 (f) ConvNeXt

(g) ResNet34 (h) ResNet50 (i) VGG16 (j) WideResNet50 (k) WideResNet101 (l) RegNet

Fig. 4: Non-targeted attack: attack success rate (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 2$).



(a) AlexNet (b) EfficientNet (c) Inception (d) MobileNet (e) ResNet18 (f) ConvNeXt

(g) ResNet34 (h) ResNet50 (i) VGG16 (j) WideResNet50 (k) WideResNet101 (l) RegNet

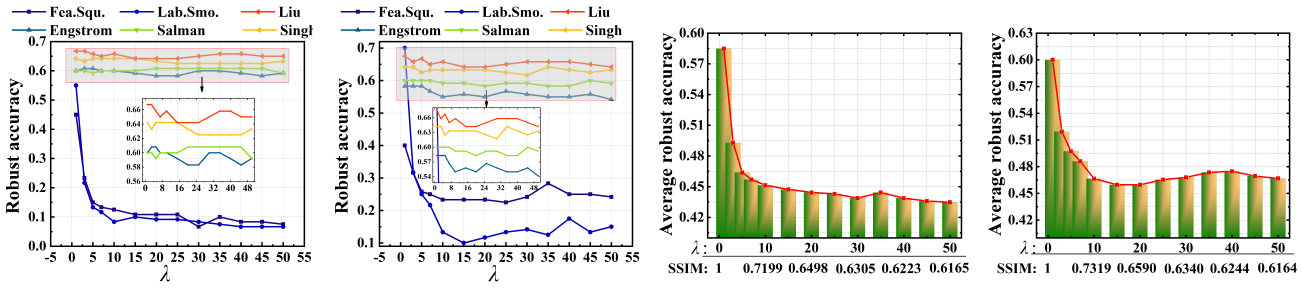
Fig. 5: Targeted attack: the attack success rate that can mislead the gradient substitution model into classifying as a specified label and can also mislead the targeted model (CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 4$).

C. Attack Performance Analysis

Fig. 4 demonstrates the impact of adversarial examples generated at different epochs on the performance of the target model under non-targeted attacks. From Fig. 4(a) to 4(l), it can be seen that with the increase of attack epochs, the attack ability of adversarial examples generated by ten attack methods will gradually increase until convergence. Compared with the nine baseline methods, our method has the most vigorous attack ability, followed by VMIFGSM. Specifically, our method has a faster convergence rate. Its attack ability can converge to the level of the most powerful attack method around the 20th epoch, reducing about ten epochs compared with the baseline methods. At the same time, our method has a significantly better attack ability than the nine baseline methods in the first 20 epochs. This effect is practical for all 12 target models.

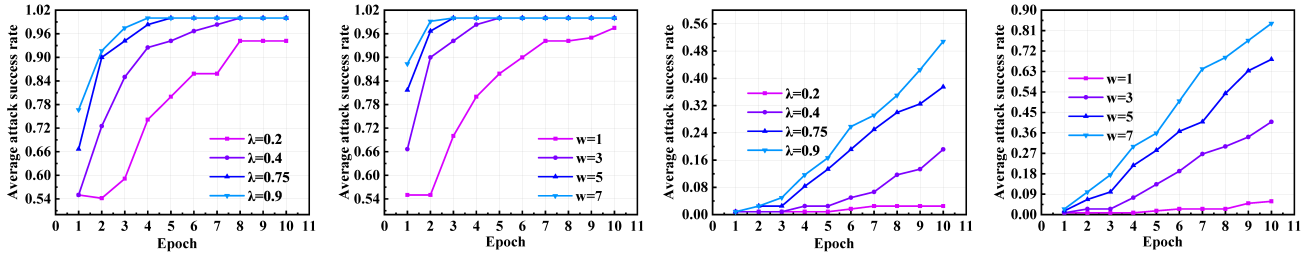
To demonstrate this ability clearly, we show the attack ability of adversarial examples generated at the 10th epoch in non-targeted attack, as shown in Table III. From the table, it can be seen that our method has a significantly better attack ability than nine baseline methods. Compared with the most substantial attack method VMIFGSM, our method improves the average attack ability against 13 target models by 11.69%, and compared with the most perceptible attack method TPGD, it improves the average attack ability by 37.15%. Compared to VMIFGSM, our method achieves 17% and 15% higher attack success rates against the EfficientNet and RegNet.

Fig. 5 illustrates the impact of adversarial examples generated at different epochs on the performance of the target model under targeted attacks. As the number of attack epochs increases, the attack capabilities of the nine attack methods gradually increase until convergence. Our method consistently



(a) Robust: Non-targeted (b) Robust: Targeted (c) Ave. robust: Non-targeted (d) Ave. robust: Targeted

Fig. 6: Non-targeted attack and targeted attack: robustness analysis (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Target model: ResNet50, Non-targeted attack: w=2, Targeted attack: w=4).



(a) $w = 3$: Non-targeted (b) λ=0.75: Non-targeted (c) $w = 3$: Targeted (d) λ=0.75: Targeted

Fig. 7: Non-targeted attack and targeted attack: the influence of λ and w on the attack capability of adversarial examples (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/255$).

exhibits significant advantages over the eight baseline methods in attack capability, convergence speed, and low-iteration attack capability in targeted attack. The attack capabilities of the baseline methods do not increase significantly in the first 20 epochs. Our method significantly eight baseline methods regarding attack capability against the AlexNet, EfficientNet, VGG16, ConvNeXt, ResNet18, WideResNet101, and MobileNet.

To demonstrate this ability clearly, we show the attack ability of adversarial examples generated at the 11th epoch in the targeted attack, as shown in Table IV. Table IV presents the attack capabilities of our method and the baseline methods in targeted attack mode. As the number of attack epochs increases, the attack capabilities of the nine methods gradually increase until convergence. Similar to non-targeted attacks, our method exhibits the fastest convergence and the most vigorous attack capability, significantly outperforming the eight baseline methods at low attack epochs.

In summary, our method outperforms the baseline methods with the fastest convergence speed and most vital attack capability, significantly surpassing the baseline methods, especially at low attack epochs.

D. Robustness Analysis

To evaluate the evasion capability of our method against different defense methods, we present the impact of various attack methods on the robustness under the non-targeted and targeted attacks, as shown in Table V. The table shows that compared to the baseline methods, our method has the most significant impact on the robustness under non-targeted and targeted attacks. Notably, robustness is most severely affected for defense methods such as Fea. Squ. [62] and Lab. Smo. [63], followed by Shan [67]'s adversarial training method, while the impact on Engstrom [64]'s defense is relatively weaker. Although our method exhibits a decrease in attack capability compared to the undefended scenario, it still outperforms the baseline methods. While specific baseline methods show increased attack capability against defense methods compared to the undefended model, they have already reached their limits. Thus, it can be inferred that our method exhibits a certain level of evasion capability against defense methods.

Fig. 6 illustrates the relationship between perceptibility changes and the model's robustness with defense methods. We control the transformation parameter λ to achieve variations in the perceptibility of adversarial examples, where larger values result in more significant perturbations and poorer image perceptibility. Fig. 6(a) and 6(b) depict the changes in the robustness of the model with defense methods for non-targeted and targeted attacks. These subplots show that as image perceptibility decreases, the robustness of models with defense methods also gradually decreases. Fig. 6(c) and 6(d) present the average robustness of various defense methods, where image perceptibility is evaluated using λ and the SSIM. These subplots demonstrate that as image perceptibility decreases, the robustness of models with defense methods also decreases.

TABLE III: Non-targeted Attack: the success rate of non-targeted attacks in the 10th epoch (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $epoch = 10$, $\lambda = 0.75$, $w = 2$).

Model \ Attack	SVRE	PGD	TPGD	DIFGSM	MIFGSM	NIFGSM	TIFGSM	SINIFGSM	VNIFGSM	VMIFGSM	OUR
ResNet18	0.15	0.44	0.26	0.50	0.47	0.52	0.38	0.50	0.55	0.57	**0.73**
ResNet34	0.10	0.37	0.20	0.51	0.48	0.52	0.43	0.48	0.63	0.63	**0.77**
ResNet50	0.16	1.00	0.63	1.00	1.00	1.00	0.99	1.00	1.00	1.00	**1.00**
AlexNet	0.08	0.39	0.28	0.42	0.33	0.29	0.28	0.36	0.31	0.33	**0.41**
MobileNet	0.18	0.50	0.36	0.53	0.51	0.47	0.42	0.57	0.58	0.63	**0.72**
WideResNet50	0.14	0.48	0.33	0.54	0.50	0.51	0.52	0.57	0.68	0.70	**0.83**
WideResNet101	0.08	0.30	0.25	0.38	0.42	0.41	0.33	0.41	0.53	0.56	**0.73**
VGG16	0.43	0.16	0.33	0.45	0.42	0.43	0.41	0.49	0.59	0.61	**0.74**
Inception	0.23	0.10	0.19	0.28	0.26	0.28	0.25	0.26	0.37	0.40	**0.48**
EfficientNet	0.35	0.11	0.30	0.38	0.39	0.40	0.33	0.40	0.49	0.51	**0.68**
ConvNeXt	0.28	0.16	0.25	0.32	0.29	0.28	0.27	0.28	0.33	0.36	**0.48**
ViT	0.15	0.05	0.11	0.14	0.14	0.17	0.17	0.15	0.23	0.23	**0.33**
RegNet	0.42	0.18	0.33	0.51	0.53	0.51	0.41	0.48	0.62	0.60	**0.75**

TABLE IV: Targeted Attack: the attack success rate in the 11th epoch that can mislead the gradient substitution model into classifying as a specified label and can also mislead the targeted model (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $epoch = 11$, $\lambda = 0.75$, $w = 4$).

Model \ Attack	PGD [52]	DIFGSM [56]	MIFGSM [53]	NIFGSM [54]	TIFGSM [53]	SINIFGSM [54]	VNIFGSM [55]	VMIFGSM [55]	OUR
ResNet18	0.3667	0.4333	0.3750	0.3500	0.2750	0.3833	0.4417	0.4583	**0.7000**
ResNet34	0.2750	0.2667	0.3417	0.3583	0.2750	0.3250	0.4417	0.4583	**0.6333**
ResNet50	0.5750	0.7333	0.9167	0.9000	0.7583	0.8167	0.9667	0.9417	**0.9750**
AlexNet	0.3583	0.4000	0.3083	0.3000	0.2833	0.3167	0.2917	0.2750	**0.6500**
MobileNet	0.4167	0.4417	0.3750	0.4083	0.3833	0.3917	0.5167	0.4917	**0.7250**
WideResNet50	0.2667	0.3917	0.3833	0.3917	0.3750	0.4000	0.5333	0.4750	**0.6667**
WideResNet101	0.2583	0.3167	0.2917	0.3417	0.2583	0.3083	0.3083	0.3417	**0.5333**
VGG16	0.4083	0.4417	0.4333	0.4417	0.3917	0.4000	0.5750	0.5167	**0.9917**
Inception	0.2500	0.2667	0.2167	0.2417	0.2333	0.2250	0.3417	0.3250	**0.4500**
EfficientNet	0.3417	0.3000	0.3667	0.3417	0.2917	0.3083	0.3917	0.3750	**0.5917**
ConvNeXt	0.0000	0.0700	0.0900	0.10	0.0600	0.1000	0.1000	0.1300	**0.3700**
ViT	0.4800	0.2600	0.5600	0.5300	0.0900	0.4500	0.4100	0.4600	**0.6800**
RegNet	0.1400	0.1800	0.5100	0.4600	0.1100	0.2400	0.2800	0.3000	**0.6500**

Overall, as the perceptual quality of images deteriorates, the robustness of models with defense methods decreases for both non-targeted and targeted attacks.

In conclusion, our method exhibits a certain level of evasion against defense methods in non-targeted and targeted attacks, making it valuable for evaluating the robustness of models.

E. Ablation Study

Fig. 7 analyzes the influence of two parameters, λ and w, on our method's attack capability of generating adversarial examples. It can be seen from the figure that the attack capability of generating adversarial examples increases with the increase of both parameters, and the change in the value of w has a more significant effect on the attack capability of generating adversarial examples. Moreover, compared with the non-targeted attack mode, the parameter change significantly impacts the attack capability of generating adversarial examples in the targeted attack.

To validate the rationality of using CAM as perturbation weights, we demonstrate the attack capabilities of generating adversarial examples using different perturbation weighting methods, as shown in Table VI. Fix indicates adding a fixed perturbation value to all pixels. Uniform represents adding perturbations that follow a uniform distribution. Gauss represents

TABLE V: Robustness analysis (accuracy of models with defense methods) (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Non-targeted attack: *w*=2, Targeted attack: *w*=4).

Method / Attack	Non-targeted Attack							Targeted Attack						
	Fea. Squ.	**Lab. Smo.**	**Engstrom**	**Salman**	**Singh**	**Liu**	**Shan**	**Fea. Squ.**	**Lab. Smo.**	**Engstrom**	**Salman**	**Singh**	**Liu**	**Shan**
BENIGN	——	——	61%	65%	68%	73%	——	——	——	61%	65%	68%	73%	——
DIFGSM	37%	47%	**59%**	**59%**	64%	67%	58%	45%	54%	59%	**59%**	64%	67%	52%
MIFGSM	32%	28%	61%	60%	64%	67%	58%	40%	41%	58%	59%	64%	65%	52%
NIFGSM	36%	27%	61%	60%	64%	68%	47%	41%	38%	58%	59%	64%	66%	41%
PGD	47%	52%	60%	**59%**	64%	68%	66%	51%	63%	58%	59%	65%	66%	42%
SINIFGSM	28%	41%	59%	**59%**	64%	66%	64%	45%	49%	58%	60%	63%	66%	52%
TIFGSM	43%	53%	62%	**59%**	64%	68%	64%	44%	58%	58%	59%	63%	66%	50%
TPGD	46%	61%	60%	60%	64%	68%	55%			——				
VNIFGSM	21%	23%	61%	60%	64%	67%	60%	31%	27%	58%	61%	63%	67%	46%
VMIFGSM	23%	22%	61%	60%	64%	66%	45%	33%	24%	59%	61%	63%	67%	48%
OUR	**16%**	**12%**	61%	**59%**	**64%**	**66%**	**43%**	**25%**	**21%**	**57%**	61%	**63%**	**65%**	**39%**

TABLE VI: CAM effectiveness analysis (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, Non-targeted attack: *w*=2, Targeted attack: *w*=4).

Non-targeted Attack													
Method / Model	**ResNet18**	**ResNet34**	**ResNet50**	**Alexnet**	**MobileNet**	**WideResNet50**	**WideResNet101**	**VGG16**	**Inception**	**EfficientNet**	**ConvNeXt**	**ViT**	**RegNet**
Uniform	0.43	0.50	**1.00**	0.00	0.00	0.00	0.00	0.45	0.00	0.00	0.29	0.16	0.50
Gauss	0.23	0.17	0.25	0.00	0.00	0.00	0.00	0.31	0.00	0.00	0.24	0.08	0.28
Clam [69]	0.51	0.52	0.98	0.00	0.01	0.00	0.00	0.52	0.00	0.00	0.31	0.19	0.47
Fix [54]	0.55	0.63	1.00	0.31	0.58	0.68	0.53	0.59	0.37	0.49	0.33	0.23	0.62
OUR	**0.73**	**0.77**	**1.00**	**0.41**	**0.72**	**0.83**	**0.73**	**0.74**	**0.48**	**0.68**	**0.48**	**0.33**	**0.75**
Targeted Attack													
Method / Model	**Resnet18**	**Resnet34**	**Resnet50**	**Alexnet**	**MobileNet**	**WideResNet50**	**WideResNet101**	**VGG16**	**Inception**	**EfficientNet**	**ConvNeXt**	**ViT**	**RegNet**
Uniform	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Gauss	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
CALM [69]	0.67	0.28	0.29	0.58	0.28	0.20	0.08	0.27	0.11	0.18	0.08	0.29	0.19
Fix [54]	0.44	0.44	0.97	0.29	0.52	0.53	0.31	0.58	0.34	0.39	0.10	0.41	0.28
OUR	**0.70**	**0.63**	**0.98**	**0.65**	**0.73**	**0.67**	**0.53**	**0.99**	**0.45**	**0.59**	**0.37**	**0.68**	**0.65**

TABLE VII: Attack Success Ratio of different modules (Target model: WideResNet101, $\epsilon = 16/255$, $\alpha = 1/500$, $epoch = 7$, $\lambda = 0.75$, Non-targeted attack: *w*=2, Targeted attack: *w*=4).

Mode / Module	Module 1	Module 2	Module 3	ALL
Non-targeted attack	0.5333	0.525	0.6583	**0.7000**
Targeted attack	0.4583	0.3667	0.5250	**0.5333**

adding perturbations that follow the Gaussian distribution. CALM [69] represents the CALM method based on attribute attribution. The table shows that our method exhibits the strongest attack capability and transferability for generating adversarial examples compared to the four perturbation-adding methods. The Fix method follows closely, while Gauss and Uniform exhibit the weakest attack capability and transferability. The adversarial example generated by the CALM [69] method possesses certain attack capabilities but weaker transferability. This validates the rationality of using CAM as perturbation weights.

To validate the effectiveness of our method, we analyze the attack capability of adversarial examples under different modules, as shown in Table VII. Module 1 only includes the first term of the loss function, Module 2 includes both the first and second terms of the loss function, Module 3 includes the first term of the loss function and uses class activation mapping scores as perturbation weights, and ALL includes all modules. The table shows that when generating

adversarial examples under both targeted and non-targeted attack modes, the attack capability is the weakest when only using Module 1, the strongest when using all modules, and the second strongest when only using Module 3. Comparing the performance of Module 1 and Module 3, it can be seen that the addition of Module 3 significantly improves the attack capability of adversarial examples. The combination of Module 2 and Module 3 further enhances the attack capability of adversarial examples.

Regarding the analysis of computational costs, please refer to Appendix Table III.

VI. Conclusion

Although many studies have focused on the issue of model transferability in attacks, there still exist problems of poor stealthiness and low attack effectiveness in generating adversarial examples. To address these issues, we propose a class activation mapping ensemble attack. This method considers each pixel feature's role in the image, using the CAMs to improve attack performance. We attack the ResNet50 model and test the transferability of adversarial examples on models such as WideResNet101, Inception, and ResNet34. Experimental results show that our method can generate adversarial examples with better transferability and perform better under low-round attacks.

In the future, we will continue to focus on research on the transferability of model ensemble attacks, aiming to improve the transferability of attack methods significantly. We will also try more attack methods and techniques, including combining reinforcement learning and meta-learning, to improve attack performance.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (NSFC) [grant numbers 62172377, 61872205], the Shandong Provincial Natural Science Foundation [grant number ZR2019MF018], and the Startup Research Foundation for Distinguished Scholars No. 202112016. This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

References

[1] Z. Deng, J. Shi, and J. Zhu, "Neuralef: Deconstructing kernels by deep neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 4976–4992.

[2] Z. Huang, Y. Wang, C. Li, and H. He, "Going deeper into permutation-sensitive graph neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 9377–9409.

[3] F. Brau, G. Rossolini, A. Biondi, and G. C. Buttazzo, "On the minimal adversarial perturbation for deep neural networks with provable estimation error," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 4, pp. 5038–5052, 2023.

[4] S. Yang, E. Yang, B. Han, Y. Liu, M. Xu, G. Niu, and T. Liu, "Estimating instance-dependent bayes-label transition matrix using a deep neural network," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 25 302–25 312.

[5] Y. Shen, A. Sowmya, Y. Luo, X. Liang, D. Shen, and J. Ke, "A federated learning system for histopathology image analysis with an orchestral stain-normalization GAN," *IEEE Transactions on Medical Imaging (TMI)*, vol. 42, no. 7, pp. 1969–1981, 2023.

[6] R. Yasrab, Z. Fu, H. Zhao, L. H. Lee, H. Sharma, L. Drukker, A. T. Papageorgiou, and J. A. Noble, "A machine learning method for automated description and workflow analysis of first trimester ultrasound scans," *IEEE Transactions on Medical Imaging (TMI)*, vol. 42, no. 5, pp. 1301–1313, 2023.

[7] Y. Ding, Q. Li, Z. Li, and Y. Li, "Multi-modal medical image segmentation with deep learning: a review," *IEEE Transactions on Medical Imaging (TMI)*, vol. 41, no. 1, pp. 20–36, 2022.

[8] S. Chang, Y. Gao, M. J. Pomeroy, T. Bai, H. Zhang, S. Lu, P. J. Pickhardt, A. Gupta, M. Reiter, E. S. Gould, and Z. Liang, "Exploring dual-energy CT spectral information for machine learning-driven lesion diagnosis in pre-log domain," *IEEE Transactions on Medical Imaging (TMI)*, vol. 42, no. 6, pp. 1835–1845, 2023.

[9] S. Zhang, Y. Li, D. Yang, Q. Wang, and Y. Liu, "Deep learning-based malware detection with improved robustness," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022, pp. 1021–1036.

[10] X. Liu, Z. Zhang, X. Wang, B. Liu, F. Li, and X. Xie, "Detecting stealthy adversarial examples with deep learning models," in *Proceedings of the 2023 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2023, pp. 1125–1140.

[11] X. Wu, S. Liu, X. Hu, J. Xu, and W. Wang, "Robust deep learning for intrusion detection: A review," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 17, no. 9, pp. 2292–2307, 2022.

[12] H. Chen, Q. Zhang, Y. Liu, Y. Liu, X. Yin, and W. Wang, "Adversarial training of deep learning models for malware detection: A case study," *ACM Transactions on Privacy and Security (TOPS)*, vol. 26, no. 1, pp. 1–26, 2023.

[13] M. Gupta and P. Agrawal, "Compression of deep learning models for text: A survey," in *Proceedings of the ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2022, pp. 61:1–61:55.

[14] W. Zhang, J. Liu, X. Li, and H. Chen, "A survey of deep learning techniques in natural language processing," in *Proceedings of the IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2023, pp. 1–15.

[15] H. Tran, D. Lu, and G. Zhang, "Exploiting the local parabolic landscapes of adversarial losses to accelerate black-box adversarial attack," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 317–334.

[16] Y. Yang, P. Liu, X. Zhang, and Q. Huang, "Universal adversarial training via wasserstein distance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 13 702–13 711.

[17] B. Chen, Y. Feng, T. Dai, J. Bai, Y. Jiang, S. Xia, and X. Wang, "Adversarial examples generation for deep product quantization networks on image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 2, pp. 1388–1404, 2023.

[18] G. Dhillon, J. Rajasegaran, P. K. Mookiah, C. P. Lim, and S. Raman, "Robustnessand adversarial training for object detection via self-supervised learning," in *Proceedings of the 30th ACM International Conference on Multimedia (ICM)*, 2022, pp. 1627–1635.

[19] A. Banerjee, U. Bhattacharya, and A. Bera, "Learning unseen emotions from gestures via semantically-conditioned zero-shot perception with adversarial autoencoders," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 3–10.

[20] Z. Wei, J. Chen, Z. Wu, and Y. Jiang, "Boosting the transferability of video adversarial examples via temporal translation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 2659–2667.

[21] K. Li, Y. Liu, X. Ao, and Q. He, "Revisiting graph adversarial attack and defense from a data distribution perspective," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[22] L. Pan, C. Hang, A. Sil, and S. Potdar, "Improved text classification via contrastive adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 11 130–11 138.

[23] B. Wang, L. Zhang, D. Zhou, Y. Cao, and J. Ding, "Neural topic

modeling based on cycle adversarial training and contrastive learning," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, pp. 9720–9731.

[24] Y. Li, Y. Sun, Z. Xu, J. Cao, Y. Li, R. Li, H. Chen, S. Cheung, Y. Liu, and Y. Xiao, "Regexscalpel: Regular expression denial of service (redos) defense by localize-and-fix," in *Proceedings of the USENIX Security Symposium (USENIX Security)*, 2022, pp. 4183–4200.

[25] L. Chen, Y. Zhang, Y. Song, L. Liu, and J. Wang, "Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 18 689–18 698.

[26] H. C. Moon, S. R. Joty, and X. Chi, "Gradmask: Gradient-guided token masking for textual adversarial example detection," in *Proceedings of the ACM Knowledge Discovery and Data Mining (SIGKDD)*, 2022, pp. 3603–3613.

[27] E. Tavan and M. Najafi, "Marsan at semeval-2023 task 10: Can adversarial training with help of a graph convolutional network detect explainable sexism?" in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, pp. 1011–1020.

[28] V. Bhat, P. Jyothi, and P. Bhattacharyya, "Adversarial training for low-resource disfluency correction," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, pp. 8112–8122.

[29] E. Altinisik, H. Sajjad, H. T. Sencar, S. Messaoud, and S. Chawla, "Impact of adversarial training on robustness and generalizability of language models," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, pp. 7828–7840.

[30] Y. Li, Z. Li, Y. Gao, and C. Liu, "White-box multi-objective adversarial attack on dialogue generation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023, pp. 1778–1792.

[31] S. Shan, W. Ding, E. Wenger, H. Zheng, and B. Y. Zhao, "Post-breach recovery: protection against white-box adversarial examples for leaked DNN models," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2022, pp. 2611–2625.

[32] C. Zhang, P. Benz, A. Karjauv, J. Cho, K. Zhang, and I. S. Kweon, "Investigating top-k white-box and transferable black-box attack," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 15 064–15 073.

[33] D. Lee, S. Moon, J. Lee, and H. O. Song, "Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, pp. 12 478–12 497.

[34] Y. Dai, H. Luo, and L. Chen, "Follow-the-perturbed-leader for adversarial markov decision processes with bandit feedback," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[35] Y. Xue and U. Roshan, "Accuracy of white box and black box adversarial attacks on a sign activation 01 loss neural network ensemble," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[36] J. Liu, Y. Kang, D. Tang, K. Song, C. Sun, X. Wang, W. Lu, and X. Liu, "Order-disorder: Imitation adversarial attacks for black-box neural ranking models," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2022, pp. 2025–2039.

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

[38] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[39] G. Severi, J. Meyer, S. Coull, and A. Oprea, "{Explanation-Guided} backdoor poisoning attacks against malware classifiers," in *Proceedings of the USENIX security symposium (USENIX security)*, 2021, pp. 1487–1504.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the International Conference on MachineLearning (ICML)*, 2021, pp. 10 096–10 106.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[44] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proceedings of the British Machine Vision Conference 2016 (BMVC)*, 2016, pp. 1–12.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[47] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 976–11 986.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[49] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 428–10 436.

[50] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[51] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[52] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[53] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9185–9193.

[54] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[55] X. Wang and K. He, "Enhancing the transferability of adversarial attacks through variance tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1924–1933.

[56] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2730–2739.

[57] X. Wang, X. He, J. Wang, and K. He, "Admix: Enhancing the transferability of adversarial attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021, pp. 16 158–16 167.

[58] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[59] Y. Xiong, J. Lin, M. Zhang, J. E. Hopcroft, and K. He, "Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability," in *Proceedings of the IEEE/CVF Conference on*

TABLE I: Non-targeted Attack: attack success rate at different epoch (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Target model: ResNet50, ϵ = 16/255, α = 1/500, λ = 0.75, w = 2).

Epoch \ Attack	PGD	TPGD	DIFGSM	MIFGSM	NIFGSM	TIFGSM	SINIFGSM	VNIFGSM	VMIFGSM	OUR
1	0.25833	0.24167	0.25833	0.22500	0.23333	0.25000	0.25000	0.23333	0.22500	**0.27500**
3	0.40833	0.31667	0.38333	0.32500	0.31667	0.27500	0.33333	0.34167	0.32500	**0.49167**
5	0.43333	0.28333	0.45833	0.41667	0.38333	0.34167	0.39167	0.44167	0.46667	**0.57500**
7	0.45833	0.35000	0.50000	0.45833	0.45000	0.42500	0.49167	0.53333	0.51667	0.**70000**
9	0.47500	0.32500	0.56667	0.50000	0.48333	0.49167	0.51667	0.56667	0.59167	**0.80000**
10	0.48333	0.32500	0.54167	0.50000	0.50833	0.51667	0.56667	0.67500	0.70000	**0.83333**
12	0.45833	0.34167	0.56667	0.56667	0.57500	0.57500	0.64167	0.73333	0.76667	**0.84167**
14	0.52500	0.41667	0.58333	0.60833	0.64167	0.57500	0.68333	0.78333	0.78333	**0.85833**
16	0.52500	0.40000	0.63333	0.63333	0.68333	0.60000	0.75000	0.79167	0.80000	**0.86667**
18	0.50000	0.36667	0.63333	0.70000	0.71667	0.59167	0.81667	0.85000	**0.83333**	**0.88333**
20	0.52500	0.40000	0.70000	0.73333	0.74167	0.62500	0.85833	0.85833	**0.86667**	**0.86667**
22	0.55000	0.42500	0.63333	0.74167	0.75000	0.62500	0.85833	0.87500	0.85833	**0.88333**
24	0.53333	0.39167	0.65833	0.75833	0.75000	0.66667	0.88333	0.86667	0.88333	**0.88333**
26	0.57500	0.38333	0.73333	0.77500	0.77500	0.68333	0.89167	0.88333	0.88333	**0.89167**

Computer Vision and Pattern Recognition (CVPR), 2022, pp. 14 983–14 992.

[60] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proceedings of the International conference on machine learning (ICML)*, 2019, pp. 7472–7482.

[61] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li, "Axiom-based grad-cam: Towards accurate visualization and explanation of cnns," *arXiv preprint arXiv:2008.02312*, 2020.

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

[63] C. Zhang, P. Jiang, Q. Hou, and Y. Wei, "Delving deep into label smoothing," *IEEE Transactions on Medical Imaging (TMI)*, vol. 30, pp. 5984–5996, 2021.

[64] L. Engstrom, A. Ilyas, and H. Salmans, "Robustness (python library)," 2019. [Online]. Available: https://github.com/MadryLab/robustness

[65] H. Salman, A. Ilyas, and Engstrom, "Do adversarially robust imagenet models transfer better?" *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 3533–3545, 2020.

[66] N. D. Singh, F. Croce, and M. Hein, "Revisiting adversarial training for imagenet: architectures, training and generalization across threat models," *arXiv preprint arXiv:2303.01870*, 2023.

[67] C. Liu, Y. Dong, W. Xiang, X. Yang, H. Su, J. Zhu, Y. Chen, Y. He, H. Xue, and S. Zheng, "A comprehensive study on robustness of image classification models: Benchmarking and rethinking," *arXiv preprint arXiv:2302.14301*, 2023.

[68] S. Shan, E. Wenger, and B. Wang, "Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2020, pp. 67–83.

[69] J. M. Kim, J. Choe, Z. Akata, and S. J. Oh, "Keep calm and improve visual feature attribution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 8350–8360.

APPENDIX

PERCEPTUAL EVALUATION

Fig. 1 extends DIFGSM, SINIFGSM, VMIFGSM, and VNIFGSM, four baseline methods, to generate adversarial examples in the target attack mode converging to the target label. Due to the weaker attack capability of the PGD method, it did not converge to the target label even with an increased number of epochs.

Analyzing the perceptibility of nine attack methods under targeted attack mode at different epochs with the same attack capability involves a significant computational burden. To facilitate computation, when analyzing the perceptibility of adversarial examples at different epochs, we did not constrain the attack effectiveness of the baseline method under the targeted attack mode. Fig. 2 illustrates the variations in image perceptibility, measured by six distance metrics under targeted attack at different epochs. Fig. 2(a) presents the attack capability of adversarial examples at different epochs. From this figure, it is evident that the attack capability of our method gradually increases with the increase in epochs until it converges. Moreover, our approach exhibits significantly higher attack capability at low epochs than the baseline method. Fig. 2(b) to 2(h) demonstrate the perceptibility of adversarial examples evaluated using different perceptual metrics. From these subfigures, it can be observed that the perceptibility of adversarial examples generated by all attack methods gradually converges as the number of epochs increases. Specifically, before the attack effectiveness of adversarial examples converges, our method exhibits lower perceptibility than baseline methods. This is because, at low epochs, even though our method focuses on adding perturbations in critical regions with weighted perturbations, the weighted perturbations are more significant than the unweighted perturbations. As a result, our method shows poor perceptibility compared to the baseline methods in fewer epochs. Nevertheless, the attack capability of the baseline method is significantly lower than ours. As the number of epochs increases, the baseline method adds minor perturbations across the entire image. At lower epochs, there is not much change in perceptibility. However, the perceptibility starts to deteriorate when the number of epochs reaches a

TABLE II: Targeted attack: attack success rate at different epoch (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, Target model: ResNet50, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 4$).

Epoch \ Attack	PGD [52]	DIFGSM [56]	MIFGSM [53]	NIFGSM [54]	TIFGSM [53]	SINIFGSM [54]	VNIFGSM [55]	VMIFGSM [55]	OUR
1	0.0000	0.0083	0.0083	0.0083	0.0000	0.0083	0.0083	0.0083	**0.0167**
2	0.0167	0.0000	0.0250	0.0250	0.0167	0.0250	0.0250	0.0250	**0.0250**
3	0.0250	0.0167	0.0250	0.0250	0.0000	0.0167	0.0250	0.0250	**0.0583**
4	0.0417	0.0333	0.1250	0.0917	0.0333	0.0500	0.0500	0.0667	**0.1667**
5	0.0750	0.0750	0.1333	0.0917	0.0250	0.0417	0.0500	0.0583	**0.1917**
6	0.1000	0.0750	0.2667	0.2083	0.0500	0.1667	0.1333	0.1417	**0.3000**
7	0.1333	0.1083	0.2833	0.2083	0.0500	0.1333	0.1417	0.1417	**0.3083**
8	0.1250	0.1917	0.4417	0.3583	0.1250	0.2917	0.3083	0.3000	**0.4167**
9	0.1417	0.225	0.4333	0.3500	0.1167	0.2667	0.3083	0.3000	**0.4750**
10	0.1917	0.2333	0.5333	0.4917	0.2083	0.3833	0.3583	0.3667	**0.6083**
11	0.2333	0.2750	0.5250	0.4917	0.1833	0.3667	0.3667	0.3500	**0.6417**
13	0.2500	0.3250	0.7250	0.6333	0.2500	0.4167	0.4417	0.4333	**0.7333**
17	0.3250	0.4833	0.8750	0.8667	0.3833	0.6667	0.7083	0.7083	**0.8833**
19	0.3667	0.5167	0.9083	0.9167	0.4250	0.7083	0.7917	0.8167	**0.9250**
22	0.3333	0.6167	0.9583	**0.9750**	0.6167	0.8250	0.8833	0.8917	0.9667
25	0.3583	0.6750	0.9750	**0.9833**	0.6583	0.8583	0.9500	0.9417	**0.9833**

TABLE III: Average time cost (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, ϵ = 16/255, α = 1/500, *epoch* = 11, λ = 0.75, Non-targeted attack: *w*=2, Targeted attack: *w*=4).

Mode \ Model	DIFGSM [56]	MIFGSM [53]	NIFGSM [54]	TIFGSM [53]	SINIFGSM [54]	VNIFGSM [55]	VMIFGSM [55]	OUR
Non-targeted Attack	0.942011s	0.540739s	0.505559s	1.055500s	1.464435s	1.648041s	1.642374s	1.393054s
Targeted Attack	1.197577s	0.765150s	0.808836s	1.273745s	3.828091s	2.452281s	2.695090s	1.629163s

TABLE IV: The rationale behind using separate substitute models for gradient and CAM computation under non-targeted attack.

Method \ Target model	ResNet18	ResNet34	ResNet50	Alexnet	MobileNet	WideResNet50	WideResNet101	VGG16	Inception	Efficientnet	ConvNeXt	ViT	RegNet
Same substitute model	0.4750	0.5000	**1.0000**	0.0000	0.0000	0.0000	0.0000	0.4750	0.0000	0.0000	0.2917	0.1250	0.5167
Multiple substitute models	**0.7300**	**0.7700**	**1.0000**	**0.4100**	**0.7200**	**0.8300**	**0.7300**	**0.7400**	**0.4800**	**0.6800**	**0.4800**	**0.3300**	**0.7500**

certain threshold. Additionally, by analyzing the results after the 40th epoch, it can be concluded that our method exhibits attack effectiveness comparable to the most vigorous attack, VMIFGSM. At the same time, the perceptibility falls between the optimal perceptual method and the most vigorous attack method. This conclusion aligns with the findings in non-targeted attacks.

Attack Performance Analysis

Table I displays the attack capabilities of our method and baseline methods in the non-targeted attack. As the number of attack epochs increases, the attack capabilities of the nine attack methods gradually strengthen until convergence. Compared to the baseline methods, our method exhibits the fastest convergence and vigorous attack capability, significantly outperforming the baseline methods at low attack epochs. Starting from the 3rd epoch, our attack capability surpasses the most robust method, VMIFGSM, by at least 10%. In particular, at the 9th iteration, our method's attack capability is improved by 20.83%.

Table II shows the attack capabilities of our method and baseline methods in targeted attacks. As the number of attack epochs increases, the attack capabilities of the nine attack methods gradually increase until convergence. Similar to non-targeted attacks, our method converges the fastest and has the strongest attack capability, significantly outperforming the eight baseline methods at low attack epochs. Starting from the 4th iteration, our attack capability surpasses VMIFGSM, the strongest method, by at least 10%. Particularly, at the 10th iteration, our method's attack capability increases by 24.16% compared to VMIFGSM, and at the 11th iteration, it increases by 29.17%.

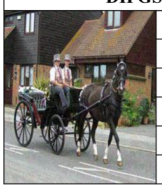
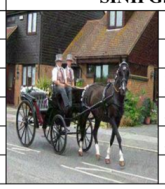
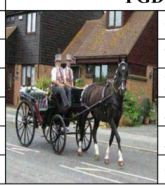
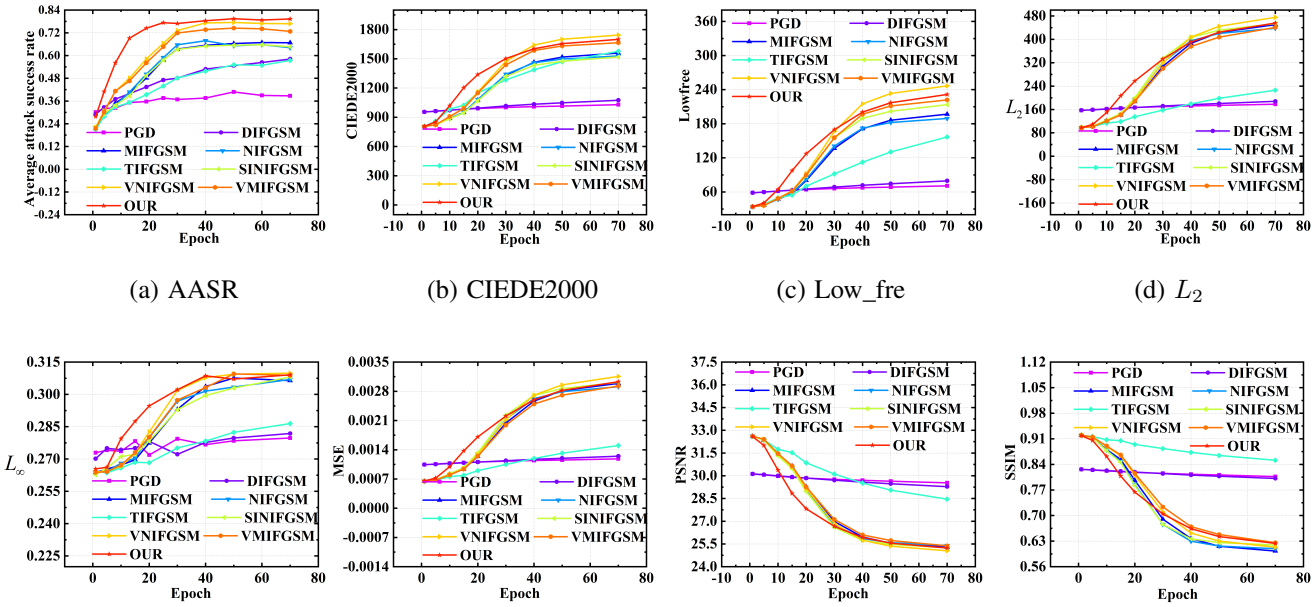
DIFGSM		SINIFGSM		VMIFGSM		VNIFGSM		PGD	
	Ori-label: 603 Tar-label: 256 Epoch: 30 Adv-label: 256 ASR: 0.9417 O-ASR:0.9917		Ori-label: 603 Tar-label: 256 Epoch: 20 Adv-label:256 ASR: 0.9333 O-ASR:0.9917		Ori-label: 603 Tar-label: 256 Epoch: 15 Adv-label: 537 ASR: 0.9750 O-ASR:0.9917		Ori-label: 603 Tar-label: 256 Epoch: 15 Adv-label: 537 ASR:0.9750 O-ASR:0.9917		Ori-label: 603 Tar-label: 256 Epoch: 70 Adv-label: 603 ASR: 0.7667 O-ASR:0.9917

Fig. 1: Adversarial examples generated by our method and baseline methods under targeted attack (Target model: ResNet50, Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/500$, $\lambda = 0.75$, $w = 4$).



(a) AASR (b) CIEDE2000 (c) Low_fre (d) L_2

(e) L_∞ (f) MSE (g) PSNR (h) SSIM

Fig. 2: Targetrd Attack: Perceptive measure (Gradient substitute model: ResNet50, CAMs substitute models: WideResNet101, Inception, and ResNet34, $\epsilon = 16/255$, $\alpha = 1/255$, $\lambda = 0.75$, $w = 4$).

Ablation Study

To analyze the computational cost of generating adversarial examples using our method, we present the average time cost for generating a single adversarial example under non-targeted and targeted attack modes when different attack methods achieve comparable attack performance. The results are shown in Table III. The table shows that generating adversarial examples for targeted attacks incurs higher time costs than non-targeted attacks. This is because targeted attacks are more challenging and require multiple epochs. In both non-targeted and targeted attack modes, the average time taken by our method to generate each image falls between the fastest method, MIFGSM, and the slowest method, SINIFGSM. Therefore, although some additional time is involved in computing the perturbation weights, it does not significantly exacerbate the overall time cost. Our method exhibits higher time costs than less performant attack methods such as DIFGSM, MIFGSM, and NIFGSM. However, it is significantly more efficient than most power attack methods like VNIFGSM and VMIFGSM.

Table IV demonstrates the rationale for using different substitute models for computing gradients and heatmaps. From this table, it is evident that adversarial examples generated using different substitute models exhibit stronger transferability than adversarial examples generated using the same substitute model. This validates the effectiveness of using different substitute models for generating adversarial examples.