Staircase Sign Method for Boosting Adversarial Attacks

Lianli Gao[1], Qilong Zhang[1], Xiaosu Zhu[1], Jingkuan Song[1], and Heng Tao Shen[1]*
[1]Center for Future Media and School of Computer Science and Engineering
[1]University of Electronic Science and Technology of China, China
qilong.zhang@std.uestc.edu.cn

Abstract

*Crafting adversarial examples for the transfer-based attack is challenging and remains a research hot spot. Currently, such attack methods are based on the hypothesis that the substitute model and the victim's model learn similar decision boundaries, and they conventionally apply Sign Method (SM) to manipulate the gradient as the resultant perturbation. Although SM is efficient, it only extracts the sign of gradient units but ignores their value difference, which inevitably leads to a serious deviation. Therefore, we propose a novel Staircase Sign Method (S^2M) to alleviate this issue, thus boosting transfer-based attacks. Technically, our method heuristically divides the gradient sign into several segments according to the values of the gradient units, and then assigns each segment with a staircase weight for better crafting adversarial perturbation. As a result, our adversarial examples perform better in both white-box and black-box manner without being more visible. Since S^2M just manipulates the resultant gradient, our method can be generally integrated into any transfer-based attacks, and the computational overhead is negligible. Extensive experiments on the ImageNet dataset demonstrate the effectiveness of our proposed methods, which significantly improve the transferability (i.e., on average, **5.1%** for normally trained models and **11.2%** for adversarially trained defenses). Our code is available at:* https://github.com/qilong-zhang/Staircase-sign-method.

1. Introduction

With the remarkable performance of deep neural networks (DNNs) in various tasks, the robustness of DNNs are becoming a hot spot of the current research. However, DNNs are pretty vulnerable to the adversarial examples [33, 11, 2, 40] which are only added with human-imperceptible perturbations but can fool state-of-the-art DNNs [32, 31, 13, 14] successfully. To make the matter worse, attacking in the physical world [30, 19, 38] is also practicable, which inevitably raises concerns in real-world applications such as self-driving cars.

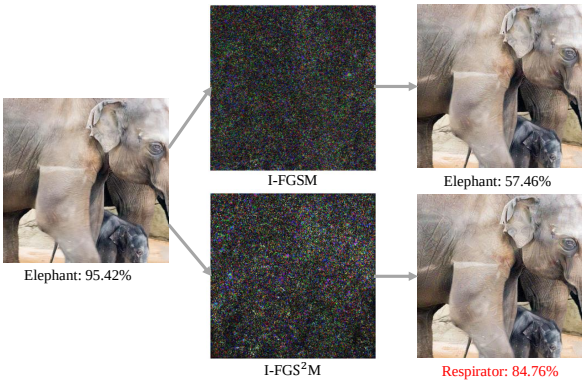


Figure 1. We generate two adversarial examples (targeted label is "respirator") for an ensemble of Inc-v4 [31], IncRes-v2 [31] and Res-152 [13], then show the classification label and confidence on the hold-out (black-box) model Inc-v3 [32]. **Left column:** the clean image. **Middle column:** the adversarial perturbations are crafted by I-FGSM and our proposed I-FGS^{2}M. **Right column:** the resultant adversarial examples.

To better evaluate the robustness of DNNs, various works have been proposed to seek the vulnerability of DNNs. Specifically, white-box attacks such as Iterative Fast Gradient Sign Method (I-FGSM) [20], Deepfool [27] and Carlini & Wagner's (C&W) method [3] can achieve impressive performance with the complete knowledge of the victim's model, *e.g.*, gradient and structure. However, deployed DNNs are usually transparent to unauthorized users for security, and thus the adversary cannot base on any knowledge of the victim's model. Therefore, resorting to cross-model transferability [33, 26, 24] of adversarial examples is a common practice. That is to say, the adversarial examples crafted via known white-box models (also called substitute model) are also dangerous for other unknown models, which makes the black-box transfer-based attack possible. In this field, Goodfellow *et al.* [11] hypothesize that the vulnerability of DNNs is their linear nature.

*Corresponding author.

Due to the fact that the magnitude of the gradient is extremely small (*e.g.* the minimal unit in gradient $\approx 10^{-9}$) and digital images usually use 8 bits per pixel, they propose Fast Gradient Sign Method (FGSM) [11] so that each pixel can be fully perturbed with only a single step. Conventionally, the following transfer-based iterative attack methods [5, 37, 22, 9] are all based on Sign Method (SM) to boost adversarial attack.

However, there is a limitation in SM, *i.e.*, ignores the difference among each unit in the gradient vector. As illustrated in Fig. 2(a), the update direction obtained by the sign function is that whether the partial derivative of loss function at each pixel is positive, negative or zero. Since the transferability phenomenon is mainly due to the fact that decision boundaries around the same data point of different models are similar, a naive application of SM results in a poor gradient estimation (as depicted in Fig. 2(c)). As a result, the adversarial examples especially for targeted ones may deviate from the *global optimal region* where most of DNNs can be fooled, thus decreasing the transferability.

Motivated by this, in this paper, we propose a **Staircase Sign Method ($\mathbf{S^2M}$)** to effectively utilize the gradient of the substitute model, thus more closely approximating the gradients of both the black-box and white-box models. Technically, our proposed method first utilizes the sign function to roughly get the gradient direction of the substitute model, then heuristically assigns different weights for each pixel by our staircase sign function. In short, we do not change the specific gradient calculated on the substitute model, but merely manipulate the resultant gradient before adding on the image. Therefore, our S^2M can be generally integrated into any transfer-based attacks, *e.g.*, the family of FGSM algorithms. Based on I-FGSM, we propose its variant I-FGS^{2}M (Algorithm 1) which can also serve as a iterative attack baseline to combine with the state-of-the-art approaches, *e.g.*, input diversity [37], Poincaré space loss [21], and patch-wise++ method [10]. To sum up, our main contributions are as follows:

- To the best of our knowledge, we are the first to point out the limitation of Sign Method which induces a poor gradient estimation, and propose a novel Staircase Sign Method to alleviate this problem.
- Our method is simple but effective and can be integrated into any transfer-based attacks. By introducing AAP to comprehensively evaluate the perturbations, we demonstrate that our attacks focus more on searching for the transferable direction.
- Extensive experiments on the ImageNet dataset [29] demonstrate that our proposed attacks consistently outperform vanilla FGSM-based non-targeted & targeted ones in both black-box and white-box manner.

2. Related Works

In this section, we first briefly review the development of transfer-based attack methods in Sec. 2.1, and then introduce several defense methods in Sec. 2.2.

2.1. Transfer-based Black-box Attacks

Unlike the white-box attack, the black-box attack cannot obtain the gradient or parameters of the victim's model. Although query-based black-box attack [4, 15, 1] can be applied in this maner, a large number of queries is computationally expensive. Thus, we turn to the transferability of adversarial examples in the remainder of this paper.

For non-targeted attacks, Goodfellow *et al.* [11] quantify the gradient by the sign function and propose single-step FGSM with the step size equal to maximum perturbation. However, perturbing the images with single-step attacks usually cannot get a high success rate on the white-box model. Therefore, Kurakin *et al.* [20] propose I-FGSM which applies FGSM multiple times with a small step size. Considering that iterative methods usually sacrifice transferability to improve the white-box performance, Dong *et al.* [5] integrate momentum term into the iterative process to avoid adversarial examples falling into local optimum. Xie *et al.* [37] apply random transformations to the input images to alleviate the overfitting problem. To effectively evade defenses, Dong *et al.* [6] propose a translation-invariant attack method to smooth the perturbation. Lin *et al.* [23] adapt Nesterov accelerated gradient and leverage scale-invariant property of DNNs to optimize the perturbations. Gao *et al.* [9] craft patch-wise noise to further increase the success rate of adversarial examples.

However, targeted attacks are more challenging which need to guide the victim's model to predict a specific targeted class with high confidence rather than just cause misclassification. In this setting, Li *et al.* [21] replace the cross-entropy loss with Poincaré distance and introduce triple loss to make adversarial examples close to the targeted label. Gao *et al.* [10] extend non-targeted PI-FGSM [9] to targeted version and adopt temperature term to push the adversarial examples into the *global optimal region* of DNNs. Instead of optimizing the output distribution with a few iterations, Zhao *et al.* [41] directly maximize the target logits with more iterations. To make the adversarial examples more transferable, several researchers [39, 18] turn to directly optimize intermediate features instead of output distribution. Although Inkawhich [16, 17] have achieved impressive performance in this way, training specific auxiliary models for each targeted class is very time-consuming.

2.2. Defense Methods

With the development of adversarial examples, researchers pay more and more attention to the robustness of

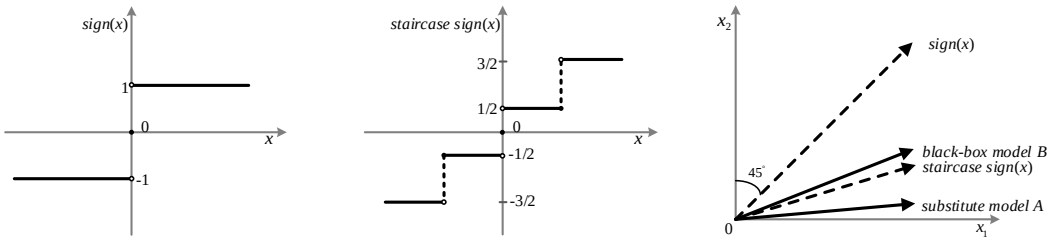


Figure 2. Sign function vs. Staircase sign function (take $K = 2$ for example). **Left & Middle:** the graph of functions. **Right:** illustration of the gradient directions of the substitute model ("A"), black-box model ("B"), and the resultant update directions of sign and staircase sign function w.r.t "A". To express more visually, here we ignore the magnitude of each perturbation. For sign function, each unit of the resultant direction is the same. By contrast, our method reflects the difference between units, and is more close to the gradient directions of both "A" and "B".

DNNs, and various defense methods are proposed to circumvent potential risks. Guo *et al.* [12] apply multiple input transformation such as JPEG compression [7], total variance minimization [28] and image quilting [8] to recover from the adversarial perturbations. Theagarajan *et al.* [34] introduce probabilistic adversarial robustness to neutralize adversarial attacks by concentrating sample probability to adversarial-free zones.

Although the above methods are efficient, *i.e.*, do not require a time-consuming training process, the adversarial training defense mechanism is more robust in practice. In this field, Madry *et al.* [25] adopt a natural saddle point formulation to cover the blind spots of DNNs. Tramèr *et al.* [35] introduce ensemble adversarial training which augments training data with perturbations transferred from other models. Xie *et al.* [36] impose constraints at the feature level by denoising technique.

3. Methodology

Before introducing our algorithm in detail, we first describe the background knowledge of generating adversarial examples. Given a DNN network $f(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{X} \longrightarrow y \in \mathcal{Y}$, it takes an input $\boldsymbol{x}$ (*i.e.*, a clean image) to predict its label y. For targeted attacks[1], it requires us to find a human-imperceptible perturbation $\boldsymbol{\delta}$ to satisfy $f(\boldsymbol{x} + \boldsymbol{\delta}) = y^*$, where $\boldsymbol{x^*} = \boldsymbol{x} + \boldsymbol{\delta}$ is the generated adversarial example and y^* is our preset targeted label.

In order to make the resultant adversarial examples invisible to the clean ones, the adversary usually sets a small perturbation upper bound ϵ, and lets $||\boldsymbol{\delta}||_\infty \leq \epsilon$. By minimizing the loss function $J(x^*, y^*)$, *e.g.*, cross entropy loss, the constrained optimization problem can be denoted as:

$$\arg\min_{x^*} J(\boldsymbol{x^*}, y^*), \qquad s.t.\ ||\boldsymbol{x^*} - \boldsymbol{x}||_\infty \leq \epsilon. \tag{1}$$

Different from the training mode of DNNs [32, 31, 13, 14], at each iteration the generation of an adversarial perturbation needs to manipulate the gradient by the sign function. For targeted attacks a resultant adversarial example at iteration $t+1$ can be formally written as:

$$\boldsymbol{x^*_{t+1}} = Clip_{\boldsymbol{x},\epsilon}\{\boldsymbol{x^*_t} - \alpha \cdot sign(\nabla_{\boldsymbol{x}} J(\boldsymbol{x^*_t}, y^*))\}, \tag{2}$$

where $Clip_{\boldsymbol{x},\epsilon}(\cdot)$ keeps the adversarial example $\boldsymbol{x^*}$ within the ϵ-ball of $\boldsymbol{x}$, and α is the step size.

3.1. Our Method

In this section, we first introduce a new metric in Sec. 3.1.1 to more comprehensively evaluate the perturbations crafted by different methods. Next, our motivation is illustrated in Sec. 3.1.2. Finally, we elaborate our method in Sec. 3.1.3 and ensemble strategy in Sec. 3.1.4.

3.1.1 Average Absolute Perturbation

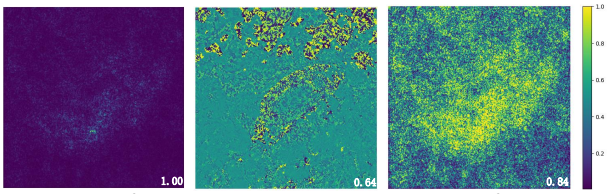
Previous works [11, 20, 5, 37, 9] conventionally focus on the ℓ_∞ norm to ensure that the adversarial examples are human-imperceptible. However, due to the fact that ℓ_∞ norm only constrains the biggest value of the perturbation vector, the magnitude of perturbation may vary considerably. For example, the stabilized update direction obtained by the momentum term [5] increases the cosine similarity of two successive perturbations, and the momentum term inevitably adds more noise to each pixel. As a result, the adversarial examples crafted by the single-step or several iterative attacks are more perceptible even under the same perturbation constraint. For instance, the resultant adversarial examples in [11, 5, 9] are more visible than those in [20].

Therefore, we introduce an *Average Absolute Perturbation (AAP)* to compute the magnitude of the adversarial perturbations for more comprehensively evaluating attack approaches. Formally, the AAP is defined as:

$$AAP = \frac{||\boldsymbol{\delta}||_1}{H \times W \times C}, \tag{3}$$

where H, W and C denote the height, width and channel of $\boldsymbol{\delta}$, respectively. Note that the lower the AAP is, the better the adversarial perturbation is.

[1] Non-targeted attacks are discussed in Appedix. C.



(a) w/o SM and S^{2}M (b) w/ SM (c) w/ S^{2}M

Figure 3. We visualize the perturbations of an image at the first iteration. All perturbations are normalized to [0, 1]. **(a)**: the gradient of an ensemble of Inc-v4 [31], IncRes-v2 [31] and Res-152 [13]; **(b)&(c)**: the results of SM and S^{2}M w.r.t (a). The numbers in the lower right corner indicate the average cosine similarity between (a) and the other two perturbations on 1,000 images. Notably, our S^{2}M not only keeps the perturbation magnitude but also has higher cosine similarity to (a).

3.1.2 Rethinking the Sign Method

To the best of our knowledge, existing transfer-based attack methods are all based on SM. In addition to the linear hypothesis [11], the motivation of this method is to modify more information for each pixel than directly adding the gradient, especially for single-step attacks. Besides, manipulating the gradients by SM for iterative attacks can quickly reach the boundary of ℓ_∞-ball with only a few iterations [5, 6].

	Hold-out ↑	Ensemble ↑	AoE [10] ↑	AAP ↓
Random-SM	0.0	0.0	0.0	15.39
I-FGSM [20]	1.2	99.9	94.7	3.4
MI-FGSM [5]	6.3	99.9	94.5	9.9
DI^2-FGSM [37]	15.4	91.8	77.8	**3.3**
TI-FGSM[2] [6]	1.6	99.9	94.0	3.4
Po-FGSM [21]	1.1	**100.0**	88.7	3.3
PI-FGSM [9]	**22.4**	99.9	**98.1**	13.2

Table 1. The average success rates (%) of targeted attacks and AAP under the same ℓ_∞ norm constraint (*i.e.* 16) on 1,000 images. The adversarial examples are crafted via an ensemble of Inc-v4 [31], IncRes-v2 [31] and Res-152 [13], and the hold-out model is Inc-v3 [32]. For Random-SM, we simply sample noise from the Gaussian distribution, then manipulated by SM.

Intuitively, SM is useful for improving transferability because a bigger AAP of resultant perturbation makes the feature of adversarial example close to that of the targeted class image more easily. However, relying on AAP alone is insufficient. As demonstrated in Tab. 1, Random-SM, which is not guided by the gradient of the substitute model, cannot successfully attack any models but with the highest AAP. Besides, DI^2-FGSM remarkably outperforms MI-FGSM by 9.1% (Hold-out) but only with 33.3% AAP of MI-FGSM. In other words, resorting to a more accurate gradient estimation for the black-box model in a few iterations is crucial.

Since decision boundaries of different models are similar and targeted attacks need to guide the adversarial examples into a specific territory of the targeted class, directly applying SM inevitably discards significant information of the gradient of the substitute model. As demonstrated in Fig. 3, the cosine similarity between the gradients (a) and perturbations manipulated by SM (b) is only 0.64. Consequently, the resultant perturbation badly deviates from the targeted territory, thus decreases the transferability.

3.1.3 Staircase Sign Method

Motivated by the limitation of SM, we propose a novel **Staircase Sign Method (S^{2}M)** to alleviate this problem. Fig. 2 depicts the difference between sign function and our proposed staircase sign function. Since our method merely manipulates the gradient at each iteration, it can be generally integrated into any transfer-based attacks, *e.g.*, the family of FGSM algorithms. For simplicity purpose, we only take our variant I-FGS^{2}M (summarized in Algorithm 1) as an example to demonstrate the integration process.

Technically, our method can be mainly divided into four steps. Firstly, as with the other methods, *e.g.*, [6, 37, 9], we need to compute the gradient $\boldsymbol{G}$ of the substitute model with respect to the input (in line 5):

$$\boldsymbol{G} = \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_{\boldsymbol{t}}^*, y^*)) \tag{4}$$

Secondly, we calculate the $p-$th percentile g^p of $|\boldsymbol{G}|$ (in line 7) according to the number of staircase K, where p ranges from $100/K$ to 100 with the percentile interval $\tau = 100/K$. Thirdly, we assign the staircase weights $\boldsymbol{W}$ according to g^p by Eq. (5) (in line 8):

$$\boldsymbol{W_{i,j}} = \begin{cases} \frac{\tau}{100}, & g^0 \le |\boldsymbol{G_{i,j}}| \le g^\tau, \\ \frac{3\tau}{100}, & g^\tau < |\boldsymbol{G_{i,j}}| \le g^{2\tau}, \\ \vdots & \\ \frac{(2k+1)\tau}{100}, & g^{p-\tau} < |\boldsymbol{G_{i,j}}| \le g^p, \\ \vdots & \\ \frac{(2K-1)\tau}{100}, & g^{100-\tau} < |\boldsymbol{G_{i,j}}| \le g^{100}. \end{cases} \tag{5}$$

where k ranges from 0 to $K-1$, and also equals to $p/\tau - 1$. As a result, our $\boldsymbol{W}$ is bounded in $[1/K, 2-1/K] \subseteq [0,2]$, which guarantees the resultant perturbation's AAP at each iteration is same with the one of SM (proof is shown in Appendix. A). Finally, combined with the sign direction of $\boldsymbol{G}$, we update our adversarial examples $\boldsymbol{x}_{\boldsymbol{t}}^*$ (in lines 11):

$$\boldsymbol{x}_{\boldsymbol{t+1}}^* = Clip_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_{\boldsymbol{t}}^* - \alpha \cdot sign(\boldsymbol{G}) \odot \boldsymbol{W}\}, \tag{6}$$

where $\odot$ is Hadamard product.

With the help of our S^{2}M, the poor gradient estimation problem caused by SM can be effectively alleviated. As demonstrated in Fig. 3, the cosine similarity between the gradients (a) and the perturbations manipulated by our proposed S^{2}M (c) is up to **0.84**. The adversarial examples are

[2]For consistency, here we term TI-BIM [6] as TI-FGSM.

Algorithm 1: I-FGS^{2}M

Input : The cross-entropy loss function J of our substitute model; iterations T; ℓ_∞ constraint ϵ; a clean image $\boldsymbol{x}$ (Normalized to [-1,1]) and its corresponding true label y; the targeted label y^*; the number of staircase $K(\geq 2)$;

Output: The adversarial example $\boldsymbol{x}^*$;

$\boldsymbol{x}_0^* = \boldsymbol{x}$;

$\alpha = \epsilon/T$; $\tau = 100/K$;

Initialize staircase weights $\boldsymbol{W}$ to 0, and p to $100/K$;

for $t \leftarrow 0$ **to** T **do**

  $\boldsymbol{G} = \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y^*)$;

  for $k \leftarrow 0$ **to** K **do**

    calculate the p-th percentile g^p of $|\boldsymbol{G}|$;

$$\boldsymbol{W}_{\boldsymbol{i},\boldsymbol{j}} = \begin{cases} \frac{\tau}{100}, & g^0 < |\boldsymbol{G}_{\boldsymbol{i},\boldsymbol{j}}| \leq g^{\tau}, \\ \frac{3\tau}{100}, & g^{\tau} < |\boldsymbol{G}_{\boldsymbol{i},\boldsymbol{j}}| \leq g^{2\tau}, \\ \vdots & \\ \frac{(2k+1)\tau}{100}, & g^{p-\tau} < |\boldsymbol{G}_{\boldsymbol{i},\boldsymbol{j}}| \leq g^{p}, \\ \vdots & \\ \frac{(2K-1)\tau}{100}, & g^{100-\tau} < |\boldsymbol{G}_{\boldsymbol{i},\boldsymbol{j}}| \leq g^{100}. \end{cases}$$

    $p = p + \tau$

  end

  $\boldsymbol{x}_{t+1}^* = Clip_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_t^* - \alpha \cdot sign(\boldsymbol{G}) \odot \boldsymbol{W}\}$;

  $\boldsymbol{x}_{t+1}^* = clip(\boldsymbol{x}_{t+1}^*, -1, 1)$;

end

Return $\boldsymbol{x}^* = \boldsymbol{x}_T^*$;

shown in Fig. 1. Compared with I-FGSM which cannot effectively decrease the confidence of true class, our proposed variant successfully misleads the model to classify our resultant human-imperceptible adversarial example as the target class "respirator".

3.1.4 Attacking an Ensemble of Models

To craft adversarial examples with high transferability, attacking an ensemble of models [24, 5] is a pretty effective strategy, especially effective for black-box attacks. It is mainly because crafting adversarial examples on multiple models has the potential to capture the global optimum of "blind spots" easily. In this paper, we follow the ensemble strategy of [5], which fuses the logits (the output before the softmax) of an ensemble of N models:

$$l(x) = \sum_{n=1}^{N} u_n l_n(x), \tag{7}$$

where $l_n(\cdot)$ is the logits of n-th model, and u_n is its ensemble weight with $u_n > 0$ and $\sum_{n=1}^{N} u_n = 1$.

4. Experiments

In order to demonstrate the effectiveness of our staircase sign mechanism, we choose to conduct extensive experiments on the family of FGSM methods to explore the performance of the FGS^{2}M-based attack. Firstly, we introduce the setup of experiments in Sec. 4.1. Next, we analyze the effect of staircase number in Sec. 4.2. Thirdly, the success rate for normally trained models and defense models is reported in Sec. 4.3 and Sec. 4.4, respectively. Finally, the AAP scores of different attacks are presented in Sec. 4.5.

Due to the space limitation, non-targeted attacks are discussed in Appendix. C. Notably, our non-targeted FGS^{2}M variants remarkably outperform vanilla FGSM ones by **19.1%** at most.

4.1. Setup

Networks: To comprehensively compare the performance between different attack methods, we consider nine state-of-the-art models, including six normally trained models: Inception-v3 (Inc-v3) [32], Inception V4 (Inc-v4) [31], Inception-ResNet V2 (IncRes-v2) [31], ResNet-50 (Res-50), ResNet-101 (Res-101), and ResNet-152 (Res-152) [13], and three ensemble adversarial training models: Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ [35].

Dataset: To avoid cherry-picking, we conduct our experiments on ImageNet-compatible dataset[3]. This dataset is comprised of 1,000 images, and widely used in recent FGSM-based attacks [5, 37, 6, 9, 21, 10].

Parameters: Following the previous works [21, 10], in our experiments, the maximum perturbation ϵ is set to 16, the iteration T of all methods is 20, and thus the step size $\alpha = \epsilon/T = 0.8$. When attacking an ensemble of N models simultaneously, the weight for the logits of each model is equal, *i.e.*, $1/N$. For MI-FGSM, the decay factor $\mu = 1.0$. For DI-FGSM, the transformation probability $p = 0.7$. For TI-FGSM, when the victim's model is in normally trained models, the Gaussian kernel length is 5×5, and 15×15 for defense models. For Po-FGSM, we set $\lambda = 0.1$. For PI-FGSM and PI-FGSM++ [10], the amplification factor $\beta = 10$, the project factor $\gamma = 0.8$, and the project kernel length is 3×3. The temperature τ of PI-FGSM++ is set to 1.5. For our S^{2}M, the number of staircase K is set to 64. Please note that the parameters of each method are fixed no matter what methods are combined.

Evaluation Metrics: In addition to our AAP, there are three other metrics are used to evaluate the performance of targeted attacks: "Hold-out" is the success rate of the black-box models (*i.e.* transferability), "Ensemble" denotes the white-box success rates for an ensemble of models, and

[3] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset

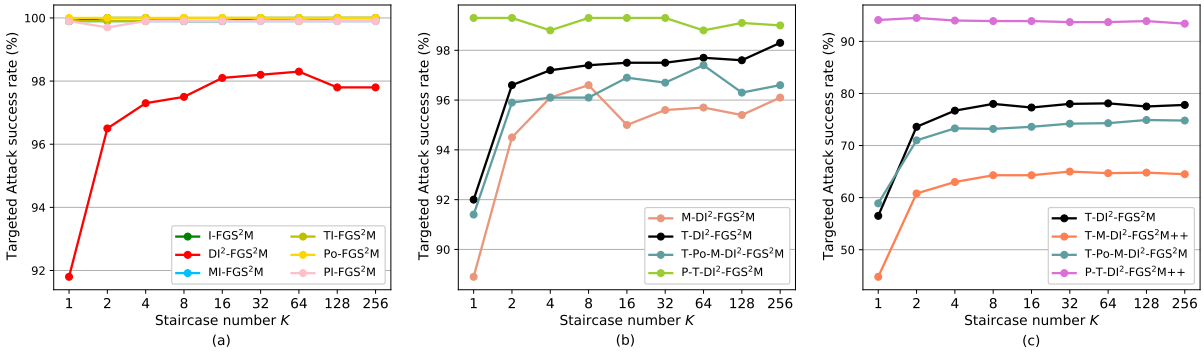


Figure 4. The success rate (%) of targeted white-box attack (Ensemble) for different methods w.r.t staircase number K ($K = 1$ denotes SM and $K \geq 2$ denotes S^{2}M). For (a) and (b), the adversarial examples are crafted via an ensemble of Inc-v4, IncRes-v2 and Res-152, and the hold-out model is Inc-v3. For (c), the white-box models are an ensemble of Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$, and the hold-out model is Inc-v3$_{ens3}$.

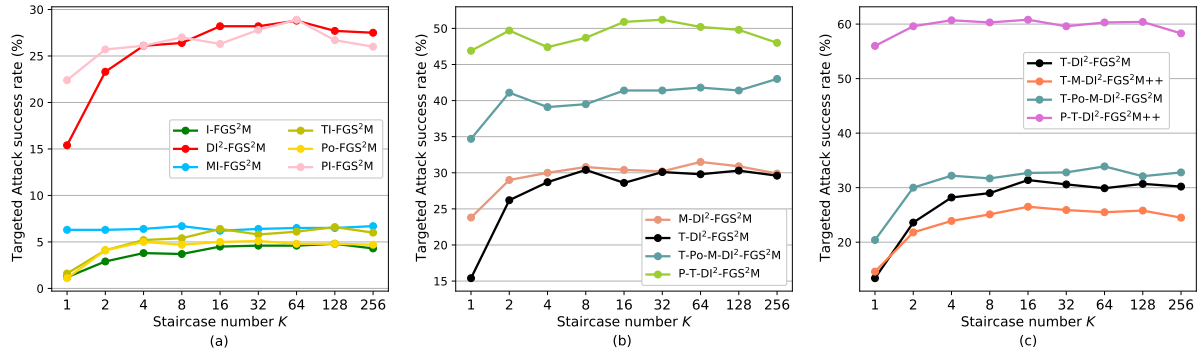


Figure 5. The success rate (%) of targeted black-box attack (Hold-out) for different methods w.r.t staircase number K ($K = 1$ denotes SM and $K \geq 2$ denotes S^{2}M). For (a) and (b), the adversarial examples are crafted via an ensemble of Inc-v4, IncRes-v2 and Res-152, and the hold-out model is Inc-v3. For (c), the white-box models are an ensemble of Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$, and the hold-out model is Inc-v3$_{ens3}$.

"AoE" [10] averages the white-box success rate of each model.

4.2. The Effect of Staircase Number K

In this section, we analyze the effect of the staircase number K for the state-of-the-art FGSM-based attacks. Here we tune $K = 2, 4, 8, ..., 256$. Please note that our methods only take $K \geq 2$ as the input. $K = 1$ denotes their corresponding FGSM-based baseline.

The experimental results of white-box attacks (Ensemble) are illustrated in Fig. 4 (the discussion on AoE is left in Appendix. B). A first glance shows that our FGS^{2}M variants have achieved an impressive improvement even when $K = 2$. When the substitute model is an ensemble of six normally trained and two ensemble adversarial training models (*i.e.* Fig. 4(c)), T-DI^2-FGS^{2}M significantly outperforms T-DI^2-FGSM by **17.1%**. As the number of staircase increases, the success rate continues to rise and then remains stable after K exceeds 64. For the methods whose success rates are already close to 100%, *e.g* Po-FGSM in Fig. 4(a), our FGS^{2}M variant does not degrade their great white-box performance.

We also depict the improvement curve for the black-box attacks (Hold-out) in Fig. 5. Compared with the vanilla FGSM implementation, our FGS^{2}M variants improve the transferability by a large margin as a whole. Specially, when transferring adversarial examples to normally trained models (Fig. 5(a)), DI-FGS^{2}M with $K = 64$ sharply increases the success rate by **13.4%** from 15.4% to 28.8%. Besides, as shown in Fig. 5(c), our methods also boost the attack performance on defenses, *i.e.*, consistently outperform the corresponding baseline attacks by **4.3%** $\sim$ **16.5%**. Considering that the curves turn to remain stable when K is big and most of methods reach the peak when $K = 64$ in Fig. 4 and Fig. 5, we set the staircase number $K = 64$ in the following experiments. Note that the computational overhead of the percentage calculation is almost negligible compared to the cost of forward pass and backpropagation.

4.3. Attacking Normally Trained Models

In this section, we compare ten FGSM-based attacks including I-FGSM, MI-FGSM, DI^2-FGSM, TI-FGSM, Po-

	Attacks	-Inc-v3	-Inc-v4	-Res-152	-IncRes	Avg.
Ensemble (White-box)	I	99.9 / **100.0**	99.9 / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**
	MI	**99.9** / **99.9**	99.9 / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**
	DI^2	91.8 / **98.3**	93.3 / **98.2**	94.5 / **98.7**	94.8 / **98.5**	93.6 / **98.4**
	TI	**99.9** / **99.9**	99.8 / **99.9**	97.0 / **99.9**	**100.0** / **100.0**	99.2 / **99.9**
	Po	**100.0** / **100.0**	99.9 / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**	**100.0** / **100.0**
	PI	**100.0** / 99.4	**100.0** / 99.9	**100.0** / 99.9	**99.8** / **99.8**	**100.0** / 99.8
	M-DI^2	88.9 / **95.7**	90.4 / **96.7**	91.0 / **96.2**	92.8 / **98.3**	90.8 / **96.7**
	T-DI^2	92.0 / **97.7**	92.2 / **97.7**	93.2 / **98.2**	94.0 / **98.5**	92.9 / **98.0**
	T-Po-M-DI^2	91.4 / **97.4**	93.0 / **97.4**	91.9 / **96.5**	94.9 / **97.5**	92.8 / **97.2**
	P-T-DI^2	**99.3** / 98.8	**99.4** / 98.5	**99.4** / 99.0	**99.6** / 99.4	**99.4** / 98.9
AoE (White-box)	I	94.7 / **97.2**	88.6 / **93.3**	92.5 / **97.0**	89.7 / **93.1**	91.4 / **95.2**
	MI	94.5 / **96.6**	90.1 / **93.5**	93.4 / **97.0**	90.5 / **93.2**	92.1 / **95.1**
	DI^2	77.8 / **89.1**	76.3 / **86.8**	84.4 / **93.7**	77.8 / **86.0**	79.1 / **88.9**
	TI	94.0 / **97.0**	87.2 / **92.3**	92.5 / **96.6**	88.6 / **92.3**	90.6 / **94.6**
	Po	88.7 / **92.8**	82.4 / **88.5**	78.6 / **85.9**	87.1 / **91.6**	84.2 / **89.7**
	PI	**98.1** / 97.9	**97.3** / 97.2	**98.5** / 97.8	96.6 / **96.9**	**97.6** / 97.5
	M-DI^2	75.4 / **84.8**	74.6 / **83.7**	80.9 / **89.8**	76.6 / **85.0**	76.9 / **85.8**
	T-DI^2	78.6 / **89.1**	75.9 / **87.0**	83.8 / **93.6**	76.8 / **86.6**	78.8 / **89.1**
	T-Po-M-DI^2	79.4 / **86.3**	76.4 / **83.8**	77.0 / **84.6**	78.0 / **84.6**	77.7 / **84.8**
	P-T-DI^2	**95.1** / 94.0	**93.3** / 92.7	**97.3** / 96.7	92.7 / **93.5**	**94.6** / 94.2
Hold-out (Black-box)	I	1.2 / **4.6**	1.2 / **3.4**	0.0 / **1.1**	0.9 / **1.9**	0.8 / **2.8**
	MI	6.3 / **6.5**	3.6 / **3.7**	**1.6** / 1.4	3.0 / **3.6**	3.6 / **3.8**
	DI^2	15.4 / **28.8**	13.8 / **27.6**	3.2 / **8.6**	9.4 / **20.6**	10.5 / **21.4**
	TI	1.6 / **6.1**	1.3 / **4.2**	0.3 / **1.3**	0.8 / **3.2**	1.0 / **3.7**
	Po	1.1 / **4.8**	0.9 / **2.9**	0.0 / **0.4**	0.3 / **2.3**	0.6 / **2.6**
	PI	22.4 / **28.9**	17.2 / **23.2**	4.2 / **6.2**	13.9 / **20.9**	14.4 / **19.8**
	M-DI^2	23.8 / **31.5**	24.1 / **30.8**	12.3 / **14.2**	21.3 / **28.3**	20.4 / **26.2**
	T-DI^2	15.4 / **29.8**	15.9 / **30.6**	3.9 / **9.7**	11.3 / **25.1**	11.6 / **23.8**
	T-Po-M-DI^2	34.7 / **41.8**	32.3 / **40.4**	17.3 / **18.0**	28.3 / **34.4**	28.2 / **33.7**
	P-T-DI^2	46.9 / **50.2**	47.1 / **50.8**	14.2 / **19.4**	41.3 / **44.7**	37.4 / **41.3**

Table 2. The success rate (%) of targeted FGSM-based/FGS^{2}M-based attacks. We study four models—Inc-v3, Inc-v4, Res-152 and IncRes-v2, and adversarial examples are crafted via an ensemble of three of them. In each column, "-" denote the hold-out model. We comprehensively report the results from three metrics, *i.e.*, Ensemble, AoE and Hold-out.

	Attacks	-Inc-v3$_{ens3}$	-Inc-v3$_{ens4}$	-IncRes-v2$_{ens}$	Avg.
Ensemble (White-box)	T-DI^2	56.5 / **78.1**	56.5 / **77.8**	55.2 / **76.7**	56.1 / **77.5**
	T-M-DI^2	44.8 / **64.7**	45.4 / **64.9**	48.9 / **68.4**	46.4 / **66.0**
	T-Po-M-DI^2	58.9 / **74.3**	58.3 / **75.3**	60.6 / **76.9**	59.3 / **75.5**
	P-T-DI^2++	**94.1** / 93.7	**94.3** / 93.3	94.2 / **95.0**	**94.2** / 94.0
AoE (White-box)	T-DI^2	43.4 / **65.1**	44.6 / **65.5**	46.2 / **68.1**	44.7 / **66.2**
	T-M-DI^2	34.7 / **52.1**	36.2 / **52.4**	37.8 / **54.5**	36.2 / **53.0**
	T-Po-M-DI^2	48.5 / **63.8**	48.9 / **63.8**	50.1 / **65.5**	49.2 / **64.4**
	P-T-DI^2++	**87.5** / 87.0	**87.3** / 86.7	**88.4** / 87.5	**87.7** / 87.1
Hold-out (Black-box)	T-DI^2	13.4 / **29.9**	12.6 / **30.0**	10.4 / **29.0**	12.1 / **29.6**
	T-M-DI^2	14.6 / **25.5**	14.5 / **25.2**	14.2 / **24.3**	14.4 / **25.0**
	T-Po-M-DI^2	20.4 / **33.9**	20.0 / **32.5**	19.2 / **30.8**	19.9 / **32.4**
	P-T-DI^2++	56.0 / **60.3**	56.5 / **58.1**	45.5 / **51.7**	52.7 / **56.7**

Table 3. The success rate (%) of targeted FGSM-based/FGS^{2}M-based attacks. We study nine models—Inc-v3, Inc-v4, Res-152, Res-101, Res-50, IncRes-v2, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$, and adversarial examples are crafted via an ensemble of eight of them. In each column, "-" denote the hold-out model. We comprehensively report the results from three metrics, *i.e.*, Ensemble, AoE and Hold-out.

FGSM, M-DI^2-FGSM, T-DI^2-FGSM, T-Po-M-DI^2-FGSM, P-T-DI^2-FGSM with our FGS^{2}M variants. In this experiment, four models including Inc-v3, Inc-v4, Res-152, and IncRes-v2 are considered. At each iteration, we select one model as the hold-out model to test the transferability, and an ensemble of the rest three with the weight of each model $1/3$ serves as the substitute model.

As indicated in Tab. 2, our proposed FGS^{2}M variants effectively boost both the white-box and black-box attacks. On average, they increase the success rate in Ensemble, AoE and Hold-out cases by **2.1%**, **5.2%** and **5.1%**, respectively. This demonstrates that our adversarial examples are more close to the *global optimal region*.

From the results of Tab. 2, we also observe that several methods, especially for these integrated with diversity input patterns (DI^2), suffer badly from SM which cannot well utilize the gradient with respect to the random input transformation. Specifically, DI^2-FGSM only successfully attack 93.6% images against the substitute model (Ensemble). The average success rate of each white-box model (AoE) is even

reduced to 79.1%, and merely 10.5% images transfer to the black-box model (Hold-out) on average. With the help of our S^{2}M at each iteration, we dramatically alleviate the poor gradient estimation problem, that is, increasing the success rate in Ensemble and AoE cases by **4.8%** and **9.8%**, respectively. Furthermore, in the Hold-out case our DI^2-FGS^{2}M remarkably improves the transferability by **10.9%**.

Another observation from the results is that the staircase sign perturbation seems to be less effective on vanilla MI but more effective for other momentum-based methods in the black-box manner, *e.g.*, M-DI^2. It may be because that the update direction derivated by our FGS^{2}M variants suffers more from the noise curing [21] caused by vanilla MI, thus causing a lack of diversity and adaptability of noise.

4.4. Attacking Adversarially Trained Defenses

Adversarially trained defenses are shown to effectively withstand the adversarial examples. Therefore, in this experiment, we consider all nine models, which are introduced in Sec. 4.1. To conduct the experiment, we select one of the ensemble adversarial training models (*e.g.* Inc-v3$_{ens4}$) as the hold-out model to evaluate the transferability, and then craft adversarial examples with an ensemble of the rest eight models with the weight of each model $1/8$. Here we compare four stronger FGSM-based attacks including T-DI^2-FGSM, T-M-DI^2-FGSM, T-Po-M-DI^2-FGSM, P-T-DI^2-FGSM++ with our FGS^{2}M variants.

As demonstrated in Tab. 3, regardless of the attacks are white-box or black-box, our methods generally surpass the vanilla FGSM-based methods. Specifically, in the white-box manner, FGS^{2}M-based attacks, on average, outperform FGSM-based ones by **14.3%** (Ensemble) and **13.2%** (AoE). This again demonstrates that our method can effectively alleviate the poor gradient estimation problem caused by SM. Besides, even under the more challenging black-box attack manner, our proposed attacks, on average, significantly improves the transferability by an increase of **11.2%** (Hold-out). Notably, compared with T-DI^2-FGSM, which only successfully transfers 10.4% adversarial examples to IncRes-v2$_{ens}$, our T-DI^2-FGS^{2}M achieves an higher transferability, *i.e.*, **29.0%**, which is about **3**$\times$.

4.5. The Comparison on AAP

Due to the fact that ℓ_∞ norm only constrains the maximum value in a perturbation vector, it ignores the magnitude of AAP. As demonstrated in Tab. 4, the AAP of several methods, *e.g.*, MI-FGSM and PI-FGSM, are much bigger than their baseline I-FGSM. Intuitively, the performance of attacks can benefit from the increase of AAP [11, 9, 10]. But if we boost adversarial attacks with a smaller AAP cost, the vulnerability of DNNs can be discovered more comprehensively, and the resultant adversarial examples are less visible. Therefore, we experimentally analyze the resultant AAP of FGSM-based attacks and their corresponding FGS^{2}M variants here.

	FGSM	FGS^{2}M (Ours)	Diff.
I	**3.34**	3.60	0.27
MI	**9.86**	9.96	0.10
DI^2	**3.29**	3.63	0.34
TI	**3.36**	3.62	0.26
Po	**3.30**	3.57	0.27
PI	13.21	**12.69**	-0.53
M-DI^2	10.14	**9.99**	-0.15
T-DI^2	**3.34**	3.67	0.33
T-Po-M-DI^2	10.16	**9.99**	-0.17
P-T-DI^2	13.33	**12.72**	-0.60
T-DI^2	**4.05**	4.51	0.46
T-M-DI^2	10.14	**9.91**	-0.23
T-Po-M-DI^2	9.77	**9.71**	-0.06
P-T-DI^2++	13.69	**12.87**	-0.82

Table 4. The AAP of FGSM-based and FGS^{2}M-based attacks ($||\boldsymbol{\delta}||_\infty = 16$). The methods listed in first ten rows craft adversarial examples for an ensemble of Inc-v4, IncRes-v2 and Res-152, and the rest of methods craft adversarial examples for an ensemble of Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$.

As indicated in Tab. 4, the AAP difference between FGSM-based and FGS^{2}M-based attacks ranges from -0.82 to 0.46, which demonstrates our proposed methods focus more on searching for the transferable direction rather than indirectly benefit from a large distortion. As demonstrated in Tab. 2, Tab. 3 and Tab. 4, for methods with relatively small AAP, *e.g.*, T-DI^2-FGSM, our T-DI^2-FGS^{2}M effectively increases the success rate on the hold-out model by **14.4%**, with a cost of just extra 0.33 AAP. For methods with relatively bigger AAP such as P-T-DI^2-FGSM++, our FGS^{2}M variant eliminates redundant noise equivalent to 0.82 per pixel but further improves transferability by **4.3%**.

5. Conclusion

In this paper, we rethink the limitation of Sign Method (SM) in state-of-the-art transfer-based attacks and experimentally demonstrate that it causes poor gradient estimation. To address this issue, we propose a simple but effective Staircase Sign Method (S^{2}M) to boost transfer-based attacks. With the help of staircase weights, our methods can more effectively fool both white-box models and black-box models. Different from several methods which significantly increase the Average Absolute Perturbation (AAP) of resultant adversarial perturbation, ours are generally with a negligible increase or even a decrease of AAP. Extensive experiments on the ImageNet dataset demonstrate the effectiveness of our FGS^{2}M-based attacks, which significantly improves the transferability by **5.1%** for normally trained models and **11.2%** for adversarially trained defenses on average. Therefore, our method can serve as an effective baseline to boost adversarial attacks and evaluate the robustness of various deep neural networks.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020. 2

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. *CoRR*, abs/1708.06131, 2017. 1

[3] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*, 2017. 1

[4] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Bhavani M. Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha, editors, *AISec@CCS*, 2017. 2

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 2, 3, 4, 5, 11, 12

[6] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 2, 4, 5, 11, 12

[7] Dziugaite, Gintare Karolina, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of jpg compression on adversarial images. *CoRR*, abs/1608.00853, 2016. 3

[8] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001. 3

[9] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Hengtao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, 2020. 2, 3, 4, 5, 8, 11, 12

[10] Lianli Gao, Qilong Zhang, Jingkuan Song, and Heng Tao Shen. Patch-wise++ perturbation for adversarial targeted attacks. *CoRR*, abs/2012.15503, 2020. 2, 4, 5, 6, 8, 11

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 1, 2, 3, 4, 8

[12] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3, 4, 5, 11

[14] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1, 3

[15] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In Jennifer G. Dy and Andreas Krause, editors, *ICML*, 2018. 2

[16] Nathan Inkawhich, Kevin J. Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *ICLR*. OpenReview.net, 2020. 2

[17] Nathan Inkawhich, Kevin J. Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020. 2

[18] Nathan Inkawhich, Wei Wen, Hai (Helen) Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *CVPR*, 2019. 2

[19] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2017. 1

[20] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017. 1, 2, 3, 4, 12

[21] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR*, 2020. 2, 4, 5, 8, 11

[22] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020. 2, 12

[23] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020. 2

[24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017. 1, 5

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 3

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 1

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 1

[28] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 3

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 2

[30] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *SIGSAC*, 2016. 1

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 1, 3, 4, 5, 11

[32] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1, 3, 4, 5, 11

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus.

Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2014. 1

[34] Rajkumar Theagarajan, Ming Chen, Bir Bhanu, and Jing Zhang. Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness. In *CVPR*, 2019. 3

[35] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: attacks and defenses. In *ICLR*, 2018. 3, 5, 11

[36] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 3

[37] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 2, 3, 4, 5, 11, 12

[38] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020. 1

[39] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *NeurIPS*, 2014. 2

[40] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *CVPR*, 2020. 1

[41] Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. On success and simplicity: A second look at transferable targeted attacks. *CoRR*, abs/2012.11207, 2020. 2

A. The AAP of S^{2}M and SM

Let δ_t denotes the resultant perturbation at iteration t, then the AAP of SM is equal to 1 since the absolute value of each pixel at δ is 1[4]. For S^{2}M, δ_t is equal to $sign(\boldsymbol{G}) \odot \boldsymbol{W}$, where $\boldsymbol{W}$ is defined in Eq. (5) is our staircase weights. Due to the fact that our $\boldsymbol{W}$ is assigned in terms of percentiles $\boldsymbol{g^p}$, that is to say, each weight interval is of equal length. Therefore, if we set the number of staircase to K, then the percentile interval $\tau = 100/K$ and the number of pixels at each part is $\frac{H\times W\times C}{K}$. So the AAP can be calculated as:

$$\begin{aligned} AAP &= \frac{\frac{H\times W\times C}{K} \times (\frac{\tau}{100} + \frac{3\tau}{100} + \ldots + \frac{(2K-1)\tau}{100})}{H \times W \times C} \\ &= \frac{1}{K} \times \frac{1}{100}(\tau + 3\tau + \ldots + (2K-1)\tau) \\ &= \frac{1}{200} \times (\tau + (2K-1)\tau) \\ &= \frac{K}{100} \times \tau \\ &= \frac{K}{100} \times \frac{100}{K} \\ &= 1. \end{aligned} \tag{8}$$

As proved in Eq. (8), the AAP of S^{2}M at each iteration is equal to the result of SM.

B. The Effect of Staircase Number K on AoE

The results of white-box manner (Ensemble) and black-box manner (Hold-out) are shown in Sec. 4. Here we depict the experimental results of white-box attacks (AoE [10]) in Fig. 6. We also adopt the same experimental setups with Sec. 4 including dataset, parameters, and networks here.

In this section, we analyze the effect of the staircase number K for state-of-the-art FGSM-based attacks. Here we tune $K = 2, 4, 8, ..., 256$. Similar to the observation in Sec. 4.2, our FGS^{2}M variants can also improve the success rates by a large margin even when $K = 2$ and the success rate continues to increase and then remain stable after K exceeds 64.

Since AoE averages the success rate of each white-box model, a bigger score of AoE means more adversarial examples can be successfully attacked against the white-box models. Put differently, this result demonstrates that our S^{2}M effectively pushes the adversarial example into *global optimal region*.

C. Experiments for Non-targeted Attacks

Due to space limitation, we just discuss the more challenging targeted attacks in our paper. Since the poor gradient estimation problem is caused by SM, crafting adversarial perturbations by our proposed S^{2}M can also boost non-targeted attacks. In this section, we report our experimental results to demonstrate the effectiveness of our methods. By replacing the SM of I-FGSM with S^{2}M, our non-targeted I-FGS^{2}M is derivated (summarized in Algorithm 2).

Algorithm 2: Non-targeted I-FGS^{2}M

Input : The cross-entropy loss function J of our substitute model; iterations T; ℓ_∞ constraint ϵ; a clean image $\boldsymbol{x}$ (Normalized to [-1,1]) and its corresponding true label y; the number of staircase $K(\geq 2)$;

Output: The adversarial example $\boldsymbol{x^*}$;

$\boldsymbol{x_0^*} = \boldsymbol{x}$;
$\alpha = \epsilon/T$; $\tau = 100/K$;
Initialize staircase weights $\boldsymbol{W}$ to 0, and p to $100/K$;
for $t \leftarrow 0$ **to** T **do**
  $\boldsymbol{G} = \nabla_{\boldsymbol{x}} J(\boldsymbol{x_t^*}, y)$;
  for $k \leftarrow 0$ **to** K **do**
    calculate the p-th percentile g^p of $|\boldsymbol{G}|$;
    $\boldsymbol{W_{i,j}} =$

$$\begin{cases} \frac{\tau}{100}, & g^0 < |\boldsymbol{G_{i,j}}| \leq g^\tau, \\ \frac{3\tau}{100}, & g^\tau < |\boldsymbol{G_{i,j}}| \leq g^{2\tau}, \\ \vdots & \\ \frac{(2k+1)\tau}{100}, & g^{p-\tau} < |\boldsymbol{G_{i,j}}| \leq g^p, \\ \vdots & \\ \frac{(2K-1)\tau}{100}, & g^{100-\tau} < |\boldsymbol{G_{i,j}}| \leq g^{100}. \end{cases}$$

    $p = p + \tau$
  end
  $\boldsymbol{x_{t+1}^*} = Clip_{\boldsymbol{x},\epsilon}\{\boldsymbol{x_t^*} + \alpha \cdot sign(\boldsymbol{G}) \odot \boldsymbol{W}\}$;
  $\boldsymbol{x_{t+1}^*} = clip(\boldsymbol{x_{t+1}^*}, -1, 1)$;
end
Return $\boldsymbol{x^*} = \boldsymbol{x_T^*}$;

C.1. Setup

Networks: Here we consider nine state-of-the-art models, including six normally trained models: Inception-v3 (Inc-v3) [32], Inception V4 (Inc-v4) [31], Inception-ResNet V2 (IncRes-v2) [31], ResNet-50 (Res-50), ResNet-101 (Res-101), and ResNet-152 (Res-152) [13], and three ensemble adversarial training models: Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ [35].

Dataset: To avoid cherry-picking, we conduct our experiments on ImageNet-compatible dataset[5]. This dataset is comprised of 1,000 images, and widely used in FGSM-based attacks [5, 37, 6, 9, 21, 10].

[4] here we do not consider the step size

[5] https://github.com/tensorflow/cleverhans/tree/master/examples/nips17_adversarial_competition/dataset

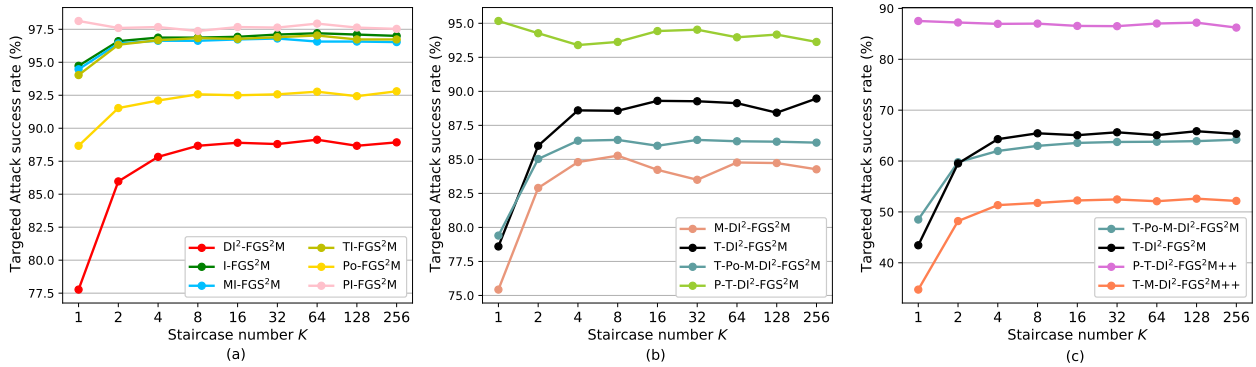


Figure 6. The success rate (%) of targeted white-box attack (AoE) for different methods w.r.t staircase number K ($K = 1$ denotes SM and $K \geq 2$ denotes S^{2}M). For (a) and (b), the adversarial examples are crafted via an ensemble of Inc-v4, IncRes-v2 and Res-152, and the hold-out model is Inc-v3. For (c), the white-box models are an ensemble of Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$, and the hold-out model is Inc-v3$_{ens3}$.

Parameters: Following the previous works [9, 22], in our experiments, the maximum perturbation ϵ is set to 16, the iteration T of all methods is 10, and thus the step size $\alpha = \epsilon/T = 1.6$. For MI-FGSM [5], the decay factor $\mu = 1.0$. For DI-FGSM [37], the transformation probability $p = 0.7$. For TI-FGSM [6], the Gaussian kernel length is 15×15. For PI-FGSM [9], the amplification factor $\beta = 5$[6], the project factor $\gamma = 1.0$, and the project kernel length is 3×3. For SI-FGSM [22], the number of scale copies $m = 5$. For our S^{2}M, the number of staircase K is set to 64. Please note that the parameters of each method are fixed no matter what methods are combined.

C.2. Staircase Sign Method for Non-targeted Attacks

In this section, we compare six FGSM-based attacks including I-FGSM [20], MI-FGSM, DI^2-FGSM, TI-FGSM, SI-FGSM and PI-FGSM with our FGS^{2}M variants. In this experiment, we study nine models introduced in Sec. C.1. Since non-targeted attacks are less challenging than targeted attacks, here we craft adversarial examples via one model instead of an ensemble of models. More specially, we consider four well-known normally trained models including Inc-v3, Inc-v4, IncRes-v2 and Res-152 as our substitute model.

As demonstrated in Tab. 5, Tab. 6, Tab. 7 and Tab. 8, our FGS^{2}M-based attacks consistently outperform FGSM-based ones in both the white-box and black-box manner. For the black-box manner, we significantly improve the transferability by **8.2%** on average. Furthermore, when adversarial examples are crafted via IncRes-v2 by SI-FGS^{2}M, we can remarkably transfer an extra **19.1%** adversarial examples to Inc-v3$_{ens3}$. For the white-box manner, we observe that our FGS^{2}M variants further increase the success rate of white-box attack toward 100%. As demonstrated in Tab. 5, I-FGSM only successfully attacks Inc-v3 with a 99.2% success rate. But with the help of our staircase weights, our I-FGS^{2}M effectively fool Inc-v3 on all the images, *i.e.*, **100%**.

[6]Due to the fact that our $\boldsymbol{W} \subseteq [0,2]$, adopting any $\beta > 5$ will result in an inaccurate gradient estimation even at the first iteration.

	Attacks	Inc-v3*	Inc-v4	IncRes-v2	Res-152	Res-101	Res-50	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
Inc-v3	I	99.2 / **100.0**	30.0 / **40.6**	21.5 / **34.9**	18.9 / **26.6**	20.8 / **29.6**	23.3 / **32.6**	12.1 / **16.0**	12.1 / **17.3**	4.9 / **8.4**	18.0 / **25.8**
	MI	99.2 / **100.0**	55.7 / **57.5**	51.2 / **55.1**	**44.0** / **44.0**	44.6 / **46.9**	49.9 / **51.8**	21.9 / **22.7**	20.7 / **23.1**	11.0 / **11.6**	37.4 / **39.1**
	DI	99.9 / **100.0**	52.5 / **67.0**	42.5 / **57.8**	32.4 / **42.9**	36.0 / **48.4**	41.4 / **52.4**	13.9 / **22.0**	14.6 / **22.7**	6.9 / **11.6**	30.0 / **40.6**
	TI	99.1 / **100.0**	27.5 / **35.9**	14.0 / **24.5**	16.3 / **23.0**	17.9 / **25.6**	22.1 / **28.1**	17.8 / **28.0**	16.5 / **26.6**	10.4 / **16.9**	17.8 / **26.1**
	SI	**100.0 / 100.0**	53.8 / **69.3**	47.2 / **64.9**	39.4 / **53.5**	45.3 / **58.9**	48.7 / **61.3**	21.7 / **33.4**	22.6 / **37.2**	10.8 / **20.1**	36.2 / **49.8**
	PI	**100.0 / 100.0**	54.5 / **62.9**	47.4 / **55.9**	39.7 / **47.6**	43.0 / **48.7**	48.2 / **52.1**	26.3 / **31.3**	25.4 / **29.0**	15.5 / **19.2**	37.5 / **43.3**

Table 5. The success rate (%) of non-targeted FGSM-based/FGS^{2}M-based attacks w.r.t adversarial examples crafted via Inc-v3. We study nine models—Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ here.

	Attacks	Inc-v3	Inc-v4*	IncRes-v2	Res-152	Res-101	Res-50	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
Inc-v4	I	43.2 / **56.9**	99.2 / **100.0**	26.3 / **39.2**	25.2 / **34.6**	25.9 / **36.1**	30.9 / **39.8**	12.0 / **16.9**	12.6 / **19.0**	6.4 / **10.5**	22.8 / **31.6**
	MI	70.8 / **73.1**	99.2 / **100.0**	58.1 / **60.4**	52.2 / **53.5**	53.8 / **54.8**	56.6 / **59.1**	23.8 / **26.0**	23.8 / **25.0**	12.7 / **13.9**	44.0 / **45.7**
	DI	64.5 / **75.3**	99.1 / **100.0**	48.3 / **63.5**	38.6 / **48.5**	40.0 / **50.4**	44.3 / **54.5**	16.0 / **21.5**	16.3 / **22.9**	8.6 / **13.7**	34.6 / **43.8**
	TI	36.8 / **46.6**	99.2 / **100.0**	16.8 / **27.9**	20.8 / **27.9**	18.9 / **26.9**	22.3 / **31.9**	16.2 / **26.8**	19.8 / **27.7**	11.6 / **17.9**	20.4 / **29.2**
	SI	72.0 / **81.7**	**100.0 / 100.0**	57.0 / **71.7**	51.1 / **62.3**	52.1 / **64.2**	56.7 / **68.6**	26.6 / **43.7**	28.0 / **45.4**	16.9 / **29.8**	45.1 / **58.4**
	PI	68.7 / **74.9**	**100.0 / 100.0**	51.4 / **60.2**	45.4 / **53.3**	44.5 / **52.8**	52.2 / **57.7**	28.2 / **35.4**	27.8 / **33.6**	19.7 / **23.6**	42.2 / **48.9**

Table 6. The success rate (%) of non-targeted FGSM-based/FGS^{2}M-based attacks w.r.t adversarial examples crafted via Inc-v4. We study nine models—Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ here.

	Attacks	Inc-v3	Inc-v4	IncRes-v2*	Res-152	Res-101	Res-50	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
IncRes-v2	I	46.7 / **59.5**	38.2 / **49.0**	99.2 / **100.0**	25.4 / **36.5**	28.2 / **39.9**	30.7 / **42.1**	13.2 / **21.4**	13.0 / **19.5**	8.3 / **15.2**	25.5 / **35.4**
	MI	**76.1** / 75.8	67.9 / **68.8**	99.2 / **100.0**	**57.6** / 56.3	57.9 / **58.9**	61.3 / **63.3**	32.0 / **34.7**	28.4 / **28.8**	20.5 / **22.0**	50.2 / **51.1**
	DI	71.4 / **79.5**	65.3 / **76.6**	98.5 / **99.7**	47.8 / **58.3**	49.6 / **59.8**	54.38 / **64.7**	19.5 / **31.0**	19.1 / **28.1**	12.2 / **22.6**	42.5 / **52.6**
	TI	43.5 / **52.2**	41.2 / **47.8**	98.8 / **99.9**	26.3 / **31.4**	28.6 / **34.5**	30.2 / **38.0**	26.7 / **35.6**	24.7 / **36.4**	20.8 / **32.4**	30.3 / **38.5**
	SI	74.0 / **83.9**	64.2 / **75.7**	**99.9 / 99.9**	51.7 / **66.1**	52.9 / **66.5**	60.5 / **73.3**	29.6 / **48.7**	28.8 / **44.3**	22.1 / **40.7**	48.0 / **62.4**
	PI	72.6 / **79.0**	64.1 / **72.8**	**100.0 / 100.0**	52.2 / **59.0**	53.8 / **61.2**	56.9 / **63.9**	34.3 / **43.4**	30.9 / **38.5**	25.6 / **33.2**	48.8 / **56.4**

Table 7. The success rate (%) of non-targeted FGSM-based/FGS^{2}M-based attacks w.r.t adversarial examples crafted via IncRes-v2. We study nine models—Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ here.

	Attacks	Inc-v3	Inc-v4	IncRes-v2	Res-152*	Res-101	Res-50	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$	Avg.
IncRes-v2	I	31.3 / **43.8**	25.9 / **35.6**	17.7 / **31.6**	98.7 / **99.5**	67.3 / **80.8**	66.1 / **78.0**	12.2 / **17.7**	13.3 / **19.4**	7.6 / **12.6**	30.2 / **39.9**
	MI	55.9 / **59.5**	50.0 / **52.2**	45.9 / **50.3**	98.7 / **99.5**	85.2 / **88.0**	83.3 / **87.6**	26.9 / **29.7**	25.7 / **26.8**	15.3 / **16.4**	48.5 / **51.3**
	DI	60.6 / **74.0**	56.5 / **68.2**	51.0 / **65.7**	98.4 / **99.6**	86.8 / **93.6**	84.2 / **92.0**	21.2 / **33.6**	20.1 / **31.9**	13.0 / **21.6**	49.2 / **60.1**
	TI	25.5 / **32.6**	21.9 / **28.2**	11.0 / **19.1**	98.2 / **99.2**	53.3 / **62.8**	48.2 / **56.3**	18.4 / **26.2**	18.7 / **25.9**	12.6 / **20.0**	26.2 / **33.9**
	SI	43.6 / **56.5**	40.3 / **50.5**	32.4 / **47.1**	**99.8 / 99.8**	84.7 / **91.6**	83.7 / **89.9**	19.0 / **33.0**	18.9 / **31.4**	12.5 / **22.4**	41.9 / **52.8**
	PI	57.5 / **63.9**	50.3 / **57.8**	47.4 / **55.0**	99.6 / **99.7**	82.7 / **90.6**	81.8 / **87.9**	31.7 / **38.1**	29.5 / **37.4**	21.2 / **27.1**	50.3 / **57.2**

Table 8. The success rate (%) of non-targeted FGSM-based/FGS^{2}M-based attacks w.r.t adversarial examples crafted via Res-152. We study nine models—Inc-v3, Inc-v4, IncRes-v2, Res-152, Res-101, Res-50, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$ here.