AB-FGSM: AdaBelief Optimizer and FGSM-Based Approach to Generate Adversarial Examples

Yixiang Wang,[1] Jiqiang Liu,[1*] Xiaolin Chang,[1] Jianhua Wang,[1] Ricardo J. Rodríguez[2]

[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China

[2]Dept. of Computer Science and Systems Engineering, University of Zaragoza

[1]{18112047, jqliu, xlchang, 20112051}@bjtu.edu.cn, [2]rjrodriguez@unizar.es

***Abstract*— Deep neural networks (DNNs) can be misclassified by adversarial examples, which are legitimate inputs integrated with imperceptible perturbations at the testing stage. Extensive research has made progress for white-box adversarial attacks to craft adversarial examples with a high success rate. However, these crafted examples have a low success rate in misleading black-box models with defensive mechanisms. To tackle this problem, we design an *A*da*B*elief based iterative *F*ast *G*radient *S*ign *M*ethod (AB-FGSM) to generalize adversarial examples. By integrating the AdaBelief optimizer into the iterative-FGSM (I-FGSM), the generalization of adversarial examples is boosted, considering that the AdaBelief method can find the transferable adversarial point in the ϵ ball around the legitimate input on different optimization surfaces. We carry out white-box and black-box attacks on various adversarially trained models and ensemble models to verify the effectiveness and transferability of the adversarial examples crafted by AB-FGSM. Our experimental results indicate that the proposed AB-FGSM can efficiently and effectively craft adversarial examples in the white-box setting compared with state-of-the-art attacks. In addition, the transfer rate of adversarial examples is 4% to 21% higher than that of state-of-the-art attacks in the black-box manner.**

***Index Terms*—adversarial examples, deep learning, generalization, optimization, security, transferability**

I. Introduction

The recent decade has witnessed the growth of artificial intelligence from the advancement of deep learning (DL) technologies. The significant success of DL has made it a state-of-the-art performance across multiple domains [1][2][3]. Nevertheless, the emergence of adversarial examples [4] poses an obstacle to DL techniques and their practical applications, and then the reliability and security of DL techniques challenge their users. In particular, adversarial examples are legitimate inputs along with well-designed and unnoticeable perturbations that will trick deep neural networks (DNNs) into misclassifications [5][6][7].

According to the adversary's knowledge, there are two types of adversarial example attacks: white-box attacks and black-box attacks [8][9][10]. In the white-box manner, the adversary has detailed information about the target model, including the model architecture and parameters. In black-box attacks, the adversary does not have information on the target model but can gain support from the transferability of adversarial examples to make black-box attacks possible. Transferability refers to adversarial examples crafted in a DNN model f that can still fool other DNN models f' and thus make adversarial examples more generalizable and more aggressive [4][11]. It is essential to learn to craft highly generalizable adversarial examples to increase the robustness of DNNs from two aspects. On the one hand, it takes advantage of critical DNN security issues. On the other hand, it can help recognize the vulnerability of models and improve their robustness by adversarial training before they are released.

Previously, researchers proposed first-order optimization-based one-step [12] and iterative [6][7] adversarial attacks to craft adversarial examples in the white-box setting, and this has been very successful [14][15]. However, such exposures are unsatisfactory, as recent work has demonstrated that these iterative optimization-based adversarial attacks have limited transferability [16][17][18]. That is, adversarial examples produced by these attacks have a low transfer rate on adversarially trained models, ensemble trained models [16], or models with other defensive mechanisms [17][18] in the black box. Several attempts have been made to facilitate the transferability of adversarial examples via optimization technologies [12][19][20]. For instance, Dong *et al.* [19] integrated the Momentum optimizer into the iterative Fast Gradient Sign Method (I-FGSM) [13] adversarial attacks. Nevertheless, improvements in these methods are still limited, and these adversarial example attacks do not find the most transferable adversarial point around the input. There is still a long way to go to boost transfer rates [15].

The specific goal of this paper is to address the low transferability problem mentioned above. Inspired by the fact that the adaptive optimizer AdaBelief is currently the best optimizer for convergence and generalization [21], we propose to integrate it into the I-FGSM [13] method. We call our proposal as ***A**da**B**elief based iterative **F**ast **G**radient **S**ign **M**ethod* (AB-FGSM). We focus on the effectiveness and efficiency of adversarial examples, studying whether iterative AB-FGSM method can accelerate the generation of adversarial examples in the white-box manner and improve the transferability of adversarial examples in the black-box manner. Our extensive experimental results demonstrate that our proposed method efficiently builds adversarial examples with a high white-box success rate and that these crafted adversarial examples achieve the high black-box transfer rate effectively compared to the state-of-the-art methods.

In summary, the contributions are as follows:

- We propose a novel adversarial example attack, AB-FGSM, which can efficiently and effectively find the direction to be perturbed on any optimization surface and avoid getting stuck in suboptimal areas when generating adversarial examples. We give three adversarial examples crafted by AB-FGSM in FIGURE 1.
- In the white-box manner, AB-FGSM requires the lowest perturbations and the lowest iterations to reach the 100% success rate compared to four state-of-the-art attacks (I-FGSM, MI-FGSM, NI-FGSM, and AI-FGSM). In multiple sophisticated DNNs (WideResNet, Inc-v3, Inc-v4, IncRes-v2 and Res-101 models), AB-FGSM can achieve the highest success rate stably by attacking a single model and ensemble models. More details are provided in Sections IV.C and IV.D.
- In the black-box manner, the adversarial examples crafted by AB-FGSM have the strongest transferability, which is at least 4% higher than the state-of-the-art methods compared. They can break defensive DNNs, such as ensemble trained models and adversarially trained models, with high success rates. We hope that our method can be the basis for future research.

We organize the rest of this paper as follows. Section II gives the notations and the background and comparative methods of adversarial examples. A more detailed account of AB-FGSM is provided in Section III. Section IV discusses the experimental results. Section V concludes the paper and states future work.

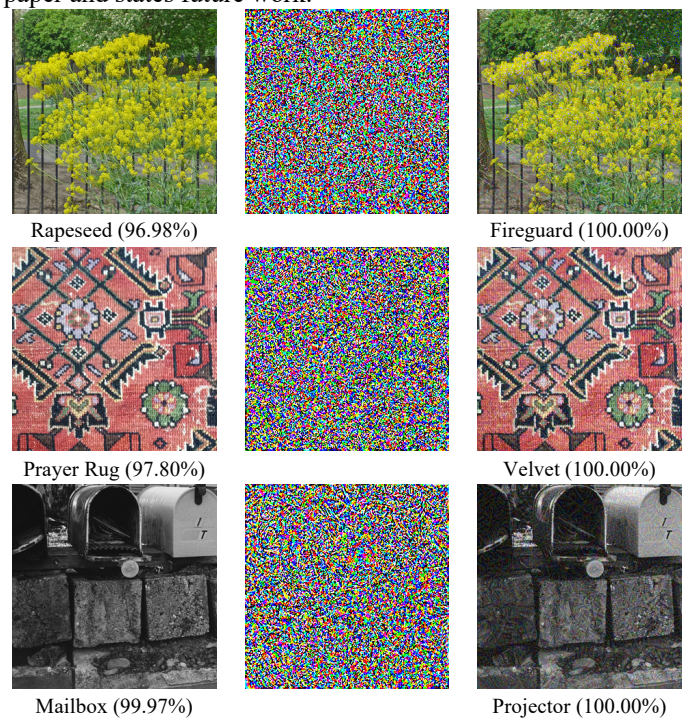


FIGURE 1 THREE ADVERSARIAL EXAMPLES CRAFTED BY AB-FGSM ARE GIVEN. **LEFT COLUMN:** THE LEGITIMATE IMAGES WITH A PREDICTION LABEL AND CONFIDENT PROBABILITY BY THE INC-V3 MODEL SHOWN BELOW. **MIDDLE COLUMN:** PERTURBATIONS CRAFTED BY AB-FGSM. **RIGHT COLUMN**: ADVERSARIAL EXAMPLES WITH THE CONFIDENCES.

II. BACKGROUND

This section provides a detailed description of basic notations and Fast Gradient Sign Method (FGSM). We then present the sophisticated version of FGSM: I-FGSM and its variants, including MI-FGSM, NI-FGSM, and AI-FGSM.

A. Notations

Given a DNN model, $f(x;\theta): x \in \mathrm{X} \rightarrow y \in \mathrm{Y}$, with parameters θ, generates a prediction label y corresponding to an input x, and the input has a ground truth label $C^*(x)$. In the context of adversarial settings, the DNN model is in the test stage and is regarded as a function, and the parameters all it has are fixed and not allowed to be modified. With these premises, an adversary wants to find an example x' (called an *adversarial example*) that is almost the same as x but has imperceptible differences, e.g., the differences are in the U ball in terms of the L_∞ norm distance.

As mentioned in Section I, the adversary generates the adversarial example through the adversarial generation algorithm on the DNN model with full knowledge, which is called the white box attack. Black-box attacks mean that the adversary does not know the target model but can generate adversarial examples in its substitute model and then transfer these examples to attack the unknown target model. According to the adversarial goals, there are two types of adversarial examples: non-targeted and targeted. Specifically, the prediction label for non-targeted adversarial examples is $f(x') \neq C^*(x)$. For targeted adversarial examples, the

prediction label follows the rule $f(x') = y' \neq C^*(x)$. This paper focuses on the non-targeted white-box and black-box adversarial examples.

B. Fast Gradient Sign Method and Iterative FGSM

Szegedy *et al.* [4] were the first to exploit the creation of adversarial examples using first-order gradient information, and the proposed white-box attack is called the Fast Gradient Sign Method (FGSM). Its purpose is to find an adversarial example x' such that the cross-entropy objective function value $J(x', C^*(x))$ is maximized. Notably, the difference between legitimate x and adversarial x' must be within the L_∞ norm radius Ú ball around x, i.e., $\|x - x'\|_\infty \leq$ Ú. More formally:

$$x' = x + \text{Ú} \cdot sign\left(\nabla_x J\left(x, C^*(x)\right)\right)$$

where $\nabla_x J(x, C^*(x))$ denotes the gradients of the objective function $J(\cdot)$ with respect to x and ∇ is the derivative symbol. The function $sign(\cdot)$ represents the sign of the function input. The limitation of FGSM is that FGSM only uses one-step gradients to craft adversarial examples, leading to a low generation rate.

To solve the problem of the low generation rate of FGSM, **I-FGSM** was proposed in [13]. Compared to FGSM, I-FGSM is a multistep adversarial attack. The iterative function is expressed as:

$$x'_0 = x,\ x'_{t+1} = clip_x^{\text{Ú}}\left(x'_t + \alpha \cdot sign\left(\nabla_x J\left(x, C^*(x)\right)\right)\right)$$

This means that the adversary allocates the total perturbations in *T* iterations. α is the step rate to control the perturbations added to the input in one iteration. The $clip_x^{\text{Ú}}(\cdot)$ function restricts the input into the Ú ball around x. The experimental results in [13] showed that the multistep attack can craft adversarial examples more potentially than the one-step attack.

C. Optimization-Based I-FGSM Variants

In the conventional DNN training phase, neural networks are trained with first-order gradient descent optimization algorithms. There are two families of gradient descent optimization algorithms: (i) the accelerated stochastic gradient descent (SGD) family, such as momentum [22] and Nesterov [23], and (ii) the adaptive learning rate family [21], such as Adam [24] and Adadelta [25]. The two families have their own advantages and disadvantages. Specifically, DNNs trained with the SGD family have a strong generalizability, but the convergence rate is low. The adaptive family is the opposite of the SGD family: they train DNNs faster at the cost of generalizability.

Since then, researchers have been investigating how to craft adversarial examples with more aggressiveness and transferability in I-FGSM, especially from optimization algorithms [19][20][26]. Dong *et al.* [19] integrated the momentum accelerated gradient [22] into I-FGSM to stabilize iterative directions, and the proposed attack (called **MI-FGSM**) has a stronger transferability for adversarial examples. The procedure is formalized as follows:

$$x' = x,\ g_0 = 0,\ g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x'} J\left(x', C^*(x)\right)}{\left\|\nabla_{x'} J\left(x', C^*(x)\right)\right\|_1}$$

$$x'_{t+1} = clip_x^{\text{Ú}}\left(x'_t + \alpha \cdot sign\left(g_{t+1}\right)\right)$$

where $\nabla_{x'} J(x', C^*(x))$ denotes the gradient of the cross-entropy loss to the adversarial example x', and g_t is the accumulated gradient of the first *t* iterations with a decay factor μ. $\|\cdot\|_1$ denotes the L_1 norm distance. Then, g_t is adopted for I-FGSM.

Shortly thereafter, Lin *et al.* [20] adapted the Nesterov accelerated gradient [23] to I-FGSM, as Nesterov is better than momentum for conventional optimization, which we call **NI-FGSM**. The update formulas that differ from the momentum are shown below:

$$x_t^{nes} = x'_t + \alpha \cdot \mu \cdot g_t$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J\left(x_t^{nes}, C^*(x)\right)}{\left\|\nabla_x J\left(x_t^{nes}, C^*(x)\right)\right\|_1}$$

$$x'_{t+1} = clip_x^{\text{Ú}}\left(x'_t + \alpha \cdot sign\left(g_{t+1}\right)\right)$$

Previous work only focused on the view of SGD families. Yin *et al.* [26] combined the adaptive Adam optimization method [24] with I-FGSM, and we call it **AI-FGSM**. The formulas are shown below:

$$g_t = \frac{\nabla_x J\left(x_t^{nes}, C^*(x)\right)}{\left\|\nabla_x J\left(x_t^{nes}, C^*(x)\right)\right\|_1}$$

$$m_{t+1} = \beta_1 \cdot m_t + (1-\beta_1)\cdot g_t \tag{II.1}$$

$$v_{t+1} = \beta_2 \cdot v_t + (1-\beta_2)\cdot g_t^{\,2} \tag{II.2}$$

$$\tau_{t+1} = \frac{m_{t+1}}{\varepsilon + \sqrt{v_{t+1}}}$$

$$\alpha_{t+1} = \alpha \cdot \frac{\sqrt{1-\beta_2^{i+1}}}{1-\beta_1^{i+1}} \Bigg/ \sum_{i=0}^{T-1} \frac{\sqrt{1-\beta_2^{i+1}}}{1-\beta_1^{i+1}} \tag{II.3}$$

$$x'_{t+1} = x'_t + \alpha_{t+1} \cdot \frac{\tau_{t+1}}{\left\|\tau_{t+1}\right\|_2} \tag{II.4}$$

where m_t and v_t denote the exponential moving average (EMA) of g_t and $g_t^{\,2}$ in Eq.(II.1) and Eq. (II.2), respectively. β_1 and β_2 are exponential decay rates to smooth g_t and $g_t^{\,2}$. The perturbation is represented as τ_t and ε is a small positive value to avoid the divisor being 0. The iterative form in AI-FGSM is a variant of the standard adaptive Adam method. The difference focuses on Eq. (II.3). The traditional Adam optimization method does not have the term $\sum_{i=0}^{T-1} \frac{\sqrt{1-\beta_2^{i+1}}}{1-\beta_1^{i+1}}$. The authors added it as it can normalize the step size α in the iteration. However, the authors in [26] did not use the $sign(\cdot)$ function in Eq. (II.4) when crafting adversarial examples, which is unreasonable, and we correct this in the experiments.

From I-FGSM to AI-FGSM, we conclude that all adversarial example methods to improve the transferability are the combination of I-FGSM and a single optimization method. In the model training phase, we cannot optimize a DNN model with two optimizers simultaneously. Therefore, it is intuitive to optimize the adversarial examples using one optimization method rather than multiple methods.

In addition, all the previous works have limitations. Momentum and Nesterov are implemented to make DNN models more generalizable, but the impact of generalization becomes very limited in adversarial settings. Likewise, the adversarial examples crafted by AI-FGSM do not generalize well, as Adam-trained generalize poorly. This is validated with our experiments in Sections IV.C and IV.D. To address these limitations, we propose a novel adversarial example attack called AB-IFGSM, which combines the AdaBelief optimizer with the iterative Fast Gradient Sign Method.

III. Methodology

This section first describes how to combine the AdaBelief optimizer with the iterative Fast Gradient Sign Method and the function of AB-FGSM will be explained in Section III.A. Then, we illustrate how our proposed method attacks single and ensemble models in Section III.B.

A. AdaBelief Iterative Fast Gradient Sign Method

Algo. 1. shows the steps of our proposed **AB-FGSM attack**, which is based on the AdaBelief optimizer. This optimizer is proposed to solve the generalization problem of adaptive optimization, as opposed to stochastic gradient descent (SGD) optimization methods [21]. The AdaBelief optimizer solved this problem by adding a "belief" coefficient $1/\sqrt{s_t}$ shown in Eq. (III.5), which is easily modified from the Adam optimizer. The detailed formulas of the Adam and Adabelief optimizers are shown below:

$$g_t = \nabla_\theta f(\theta),\ \ m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t,\ \ \hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{III.1}$$

Adam:

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^{\,2},\ \ \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{III.2}$$

$$\Delta\theta_t = \alpha \cdot \frac{1}{\sqrt{\hat{v}_t} + \varepsilon} \cdot \hat{m}_t \tag{III.3}$$

AdaBelief:

$$s_t = \beta_2 s_{t-1} + (1-\beta_2)\left(g_t - m_t\right)^2,\ \ \hat{s}_t = \frac{s_t + \varepsilon}{1-\beta_2^t} \tag{III.4}$$

$$\Delta\theta_t = \alpha \cdot \frac{1}{\sqrt{\hat{s}_t} + \varepsilon} \cdot \hat{m}_t \tag{III.5}$$

Algo. 1. AB-FGSM

Input: The model learned function $f(\cdot)$ with the cross-entropy objective function $J(\cdot)$; a legitimate input x, and its ground-truth label $C^*(x)$; the total iterations T with each step t; the size of the perturbation U; a fine-tuned step rate α; AdaBelief factors includes exponential decay rates β_1, β_2; a denominator stability parameter ε.

Output: An adversarial example x' with $\|x - x'\|_\infty < $ Ú;

Initialize $m_0 \leftarrow 0,\ s_0 \leftarrow 0,\ t \leftarrow 0,\ x'_0 \leftarrow x$

while $t < T$ **do:**

  $t = t + 1$

  $g_t = \nabla_x J\left(\theta, x'_{t-1}, C^*(x)\right)$

  $\gamma = \sum_{i=1}^{t} \frac{\sqrt{1-\beta_2^{\,i+1}}}{1-\beta_1^{\,i+1}}$

  $m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$

  $s_t = \beta_2 \cdot s_{t-1} + (1-\beta_2) \cdot (g_t - m_t)^2$

  $s_t = \max(s_{t-1}, s_t)$

  $\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{s}_t = \frac{s_t + \varepsilon}{1-\beta_2^t}$

  $x'_t = x'_{t-1} + \frac{\alpha}{\gamma} \cdot sign\left(\frac{\hat{m}_t}{\sqrt{\hat{s}_t + \varepsilon}}\right)$

  $x'_t = Clip_x^{Ú}(x'_t)$

end while

return x'

Here, Eq. (III.1) are the universal basic EMAs of g_t in the Adam and AdaBelief optimizers. Eq. (III.2) and Eq. (III.3) are the Adam update functions, and Eq. (III.4) and Eq. (III.5) are the AdaBelief update functions. The improvement of the Adam optimizer in the AdaBelief optimizer is shown in Eq. (III.4), compared with Eq. (III.2) in the Adam optimizer.

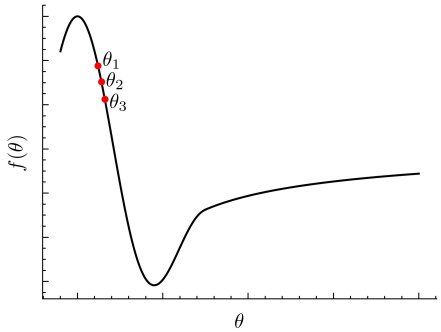


FIGURE 2 ILLUSTRATION OF THE OPTIMIZATION SURFACE.

The term $(g_t - m_t)^2$ in Eq. (III.4) promises that the AdaBelief method can constantly update parameters on any objective surface. We use FIGURE 2 as an example to illustrate the disadvantage of the Adam method. $f(\theta)$ represents an imaginary optimization surface with respect to parameter θ. When the surface is steep and *t* is in the period $\theta_1 \sim \theta_3$, the gradient g_t is large, and we want the parameters to update faster. In this case, m_t and v_t in Eq. (III.1) and (III.2) are large, causing $1/\sqrt{v_t}$ to be very small in Eq. (III.3) and then slowing down the update process. This phenomenon is the shortcoming of the Adam method since the performance

of Adam is far from our goals.

However, in the AdaBelief settings, s_t in Eq. (III.4) is small since g_t and m_t are close on the steep optimization surface, which leads to $\frac{1}{\sqrt{s_t}+\varepsilon}$ in Eq. (III.5) and thus promoting an expected rapid convergence. This is why $1/\sqrt{s_t}$ is called the "belief" term, as it can reflect the belief of the gradient prediction: if $1/\sqrt{s_t}$ is small, then the AdaBelief method believes in taking a small step to update parameters; otherwise, AdaBelief takes a large step.

Intuitively, we integrate the AdaBelief method into I-FGSM to improve the transferability of adversarial examples due to the faster convergence rate and stronger generalization performance of the AdaBelief method. We call it AB-FGSM algorithm, the procedure of which is shown in **Algo. 1**.

Specifically, AB-FGSM takes *T* iterations to generate adversarial examples, as **while** function does. In each iteration, we first calculate the gradient of the current input x', as described in line 4. Then, in lines 6 and 7, we calculate the EMAs of g_t and $\left(g_t - m_t\right)^2$. In line 9, we compute the bias correction of m_t and s_t. The reason to perform bias correction is to help correct the biased estimation of m_t and s_t in the first few iterations. There is no difference with the standard AdaBelief method in these three lines.

The difference between AdaBelief and AB-FGSM is in lines 5, 8, 10, and 11 in **Algo.1**. In line 5, inspired by the previous work of AI-FGSM [26], we also add a normalized term γ to further fine-tune the step size. However, our normalized term differs from that of AI-FGSM. We only add the normalized values from the first iterations to the current iteration *t*, but theirs added all normalized values initially, which we believe is counterintuitive. The benefit of our proposal is that only the previous normalized terms affect the step size, and the normalized terms that have not yet been iterated have no effect on the current step size. We think this is more reasonable than theirs. In line 8, the AMSGrad skill [27], the choice of the bigger EMA values, is used in our algorithm to help AB-FGSM avoid convergence to the suboptimal point. Lines 10 and 11 are the adaptation of I-FGSM to AB-FGSM.

B. Attacking Single and Ensemble Models

We follow the previous experimental pattern [11][12][18] to validate the effectiveness of AB-FGSM: attacking single and ensemble models in the white-box and black-box manners. In this section, we will give a detailed description.

Traditionally, white-box and black-box attacks are merely conducted on a single DNN model. Liu *et al.* [29] verified that an adversarial example remains adversarial for multiple ensemble models, and then it is more likely to transfer to other models as well. Dong *et al.* [19] further emphasized that adversarial examples that can threaten an ensemble model have greater transferability. Therefore, an adversarial example is more transferable if it can escape the ensemble model.

An ensemble model is an approach that combines multiple models to obtain better prediction and boosts the model robustness since the prediction of the ensemble model can be seen as voting for the prediction of each submodel. Only when the predictions of almost all submodels are the same can the ensemble model output the agreed prediction. Similarly, the adversarial example must fool all submodels in an ensemble model, which can cause misclassification. This implicitly reflects the transferability of the adversarial example and why it is necessary to test the effectiveness of AB-FGSM on the ensemble model. In the ensemble setup, if the submodel is a machine learning model, there are several ways to aggregate the predictions, such as weighted average and max voting. If the submodel is the deep learning model, the most useful aggregation of the ensemble prediction is generally fusing the weighted logit outputs. For instance, the output of a *K*-ensemble model is:

$$L(x) = \sum_{k=1}^{K} w_k \cdot l_k(x) \tag{III.6}$$

where w_i and $l_i(x)$ denote the weight and the logit outputs of the *i*-th model and $\sum w_i = 1$. In the experiments, we not only attack a single DNN model in the white-box and black-box manners but also attack the ensemble models in the black-box manner.

IV. Experiments and Analysis

In this section, extensive experiments are carried out to validate the efficiency and effectiveness of our proposed method. We first specify the dataset, models and metrics used in the experiments in Section IV.A. Then, in Section IV.B, we investigate the effect of hyperparameters on the performance of AB-FGSM attack. The results of single and ensemble model attacks by the proposed method and other baseline attacks are discussed in Sections IV.C and IV.D.

A. Setup

We choose two datasets to validate the effectiveness of our proposed method: ImageNet [30] and CIFAR-10 [31]. The information of each dataset is shown in TABLE I. There is one thing that needs to be stressed, which is that in the ImageNet dataset, we select a corresponding data point in 1000 classes that can be correctly classified by the model, considering that the number of validation sets in ImageNet is too large, and this setup was also established in previous papers [19], [20], [26].

TABLE I. DATASET INFORMATION

Dataset	Size	Class	Number
ImageNet	$299\times299\times3$	1000	1000
CIFAR-10	$32\times32\times3$	10	10000

In the experiments, we consider six models for each dataset, including normal-trained and adversarially-trained models. Concretely, in the ImageNet dataset, we study four ImageNet models: Inception v3 (Inc-v3) [32], Inception v4 (Inc-v4), Inception Resnet v2 (IncRes-v2) [33] and Resnet v2-101 (Res-101) [34] for normally trained models. For adversarially trained models, we consider the $\text{Inc-v3}_{\text{ens3}}$ and $\text{Inc-v3}_{\text{adv}}$ models. Here, the suffix *adv* means the adversarially-trained model, while the suffix *ens3* means an adversarially trained model with three ensemble models. For the CIFAR-10 dataset, four normal-trained CIFAR-10 models are considered: ResNet-18 (Res-18) [34], VGG-18 [35], DenseNet [36] and GoogleNet [32]. Meanwhile, two adversarially-trained CIFAR-10 models, WideResNet with adversarial training defensive method [37] ($\text{WideRes-28}_{\text{adv}}$) and WideResNet with adversarial training by data generated by the GAN model ($\text{WideRes-28}_{\text{GAN}}$) [38], are also considered. We choose these models because they are classic, representative and widely used in image classification [39].

For comparison, we compare our method with four iterative FGSM variants: I-FGSM, MI-FGSM, NI-FGSM, and AI-FGSM, which are described in Section II. They belong to the FGSM family and are combinations of optimization methods and FGSM algorithms. We do not take FGSM into account because previous work demonstrated its weakness in improving the transferability of adversarial examples. Meanwhile, other adversarial attacks, especially those which aim to minimize the perturbation, such as CW [5] and FAB [42], are not considered since previous work has proven that their transferability is very limited [40]. The transferability of PGD attack has been verified to be lower than that of AI-FGSM [26]. Therefore, we compare our approach with FGSM families.

In terms of hyperparameters, we follow the settings described in [19]. That is, the maximum perturbation U is 16, and the momentum factor μ is 1.0. The maximum number of iterations *T* in our experiments is set to 10. The value α in both MI-FGSM and NI-FGSM is $\acute{y}T$. In AI-FGSM, the exponential decay rates β_1 and β_2 are determined to be 0.99 and 0.999, respectively. Finally, ε is set to 10^{-14} . The setting of hyperparameters of AB-FGSM is discussed in more detail in Section IV.B.

For the evaluation metric, we use the success rate, which is calculated as:

$$Success\ Rate = \frac{\#\text{misclassified samples}}{\#\text{ correctly classified samples}} \tag{IV.1}$$

where *#misclassified samples* denote the number of misclassified examples, i.e., adversarial examples. *#correctly classified samples* denotes the number of samples that were classified correctly. It is reasonable to choose the success rate to evaluate our experiments since we pay attention to the transferability of adversarial examples and are not concerned with perturbation size. From this point, the success rate is certainly an intuitive and appropriate metric. Meanwhile, in the white-box attack, the success rate is also called the *generation rate*.

In the black-box attack, the success rate is called the *transfer rate*. However, in the transfer rate, the meanings of the denominator and numerator are different compared with Eq. (IV.1). Concretely, the denominator in the transfer rate is the number of adversarial examples, and the numerator represents the number of adversarial examples that successfully fool the new model. A high transfer rate means better transferability of adversarial examples.

In hardware settings, we implement our attack in the TensorFlow Python library [41]. We use an Intel Xeon Silver 4114 CPU with a single NVIDIA TITAN XP GPU for experiments.

B. Hyperparameter Analysis of AB-FGSM

In this section, we investigate the influence of the hyperparameters (the step rate α , the iteration T , the smooth parameters β_1 , β_2 , the perturbation size U and stability parameter ε) on the success rates of the proposed AB-FGSM. It is important to emphasize that we only investigate the effect of the above hyperparameters on the ImageNet models instead of the CIFAR-10 models because the ImageNet models are more sophisticated and complex than the CIFAR-10 models, and testing hyperparameters on the larger and deeper models is more convincing. Moreover, the hyperparameters of sophisticated models are usually effective on small-scale models. We will directly use the appropriate hyperparameters tested on the ImageNet models to the CIFAR-10 models.

1) Influence of the step rate α

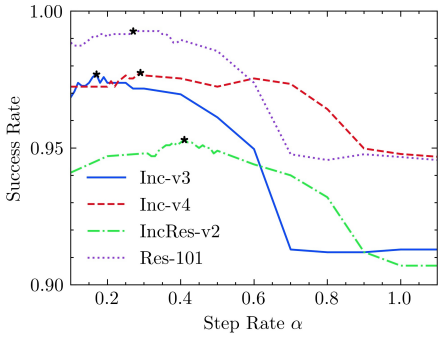


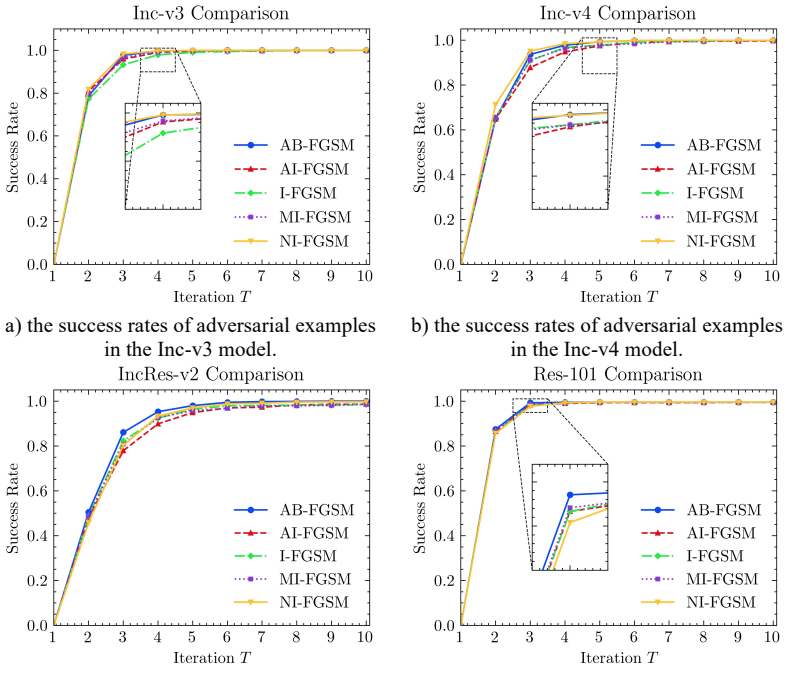
FIGURE 3 SUCCESS RATES OF AB-FGSM ATTACK ON FOUR IMAGENET MODELS WHEN THE STEP RATE α VARIES FROM 0 TO 1.1.

This subsection primarily discusses the step rate α. It is known that learning rate is a critical element when training a neural network. Here, the step rate α in the adversarial methods plays the same role as in the learning rate of the DNN training phase. Hence, it is practical to find out the step rates that can make the AB-FGSM attack method efficient.

FIGURE 3 shows, under different step rates α, the success rate curves of the adversarial examples generated by AB-FGSM. We preset iteration $T = 4$ on the IncRes-v2 model and $T = 3$ in the other three models, as our extensive experimental results indicate that AB-FGSM is empirically sensitive to the iteration settings. For instance, when we attack the IncRes-v2 model by AB-FGSM with iteration $T = 4$, a modification of the step rate will dramatically change the success rates of the other iterations. However, when we set $T = 3$, these changes are not noticeable. Other hyperparameters are the same as those suggested by the AdaBelief optimizer [21].

As seen in FIGURE 3, each curve follows the trend of fluctuating upward and then downward. Furthermore, we can see that the optimal step rate of AB-FGSM is diverse when attacking different models, and all of them are greater than 0.1 and less than 0.5. We highlight them by the symbol '*'. The optimal step rate can help to achieve the highest success rate with smaller iterations. Additionally, a step rate in a reasonable range can still cause the attack to achieve the highest success rate, but at the cost of more iterations. We suggest that the step rate be set in the range (0.1, 0.5). We use the optimal step rates to investigate the effect of other hyperparameters on AB-FGSM performance.

2) Influence of the number of iterations T



a) the success rates of adversarial examples in the Inc-v3 model.

b) the success rates of adversarial examples in the Inc-v4 model.

c) the success rates of adversarial examples in the IncRes-v2 model.

d) the success rates of adversarial examples in the Res-101 model.

FIGURE 4 SUCCESS RATES OF ADVERSARIAL EXAMPLES GENERATED BY FIVE ATTACKS WITH ITERATIONS T RANGING FROM 1 TO 10.

This subsection aims to investigate the influence of iterations on AB-FGSM performance compared to four baseline attacks. FIGURE 4 shows the success rate curves of adversarial examples generated by five adversarial attacks in different models when the hyperparameter iteration ranges from 1 to 10. Here, the step sizes are optimal depending on the models.

As seen in FIGURE 4.a, when attacking the Inc-v3 model, each attack method achieves a 100% success rate when $T=10$. However, these attacks require multiple iterations to achieve a 100% success rate. Specifically, when the number of iterations is less than 6, there is a gap between each attack. We focus on the condition $T=4$, as this condition represents the efficiency of five attack methods. The most efficient attacks are AB-FGSM and NI-FGSM, and both success rates are almost 100%. The following are MI-FGSM and AI-FGSM. I-FGSM has the worst performance, which is expected since I-FGSM is conventional and basal.

When attacking the Inc-v4 model, the results shown in FIGURE 4.b are similar to those of the Inc-v3 model with little difference. More specifically, NI-FGSM is also the first to achieve the best success rate. Our proposed AB-FGSM follows NI-FGSM to achieve the second-best success rate. Unlike FIGURE 4.a, the success rates of MI-FGSM and I-FGSM overlap. Surprisingly, AI-FGSM has the worst performance, and this is unforeseen but reasonable: when the optimization surface in the Inc-v4 model is relatively convex, the Adam-based convergence method is slowly optimized on the surface, as discussed in Section III.A. This is the reason why AI-FGSM needs more iterations to achieve the best success rate.

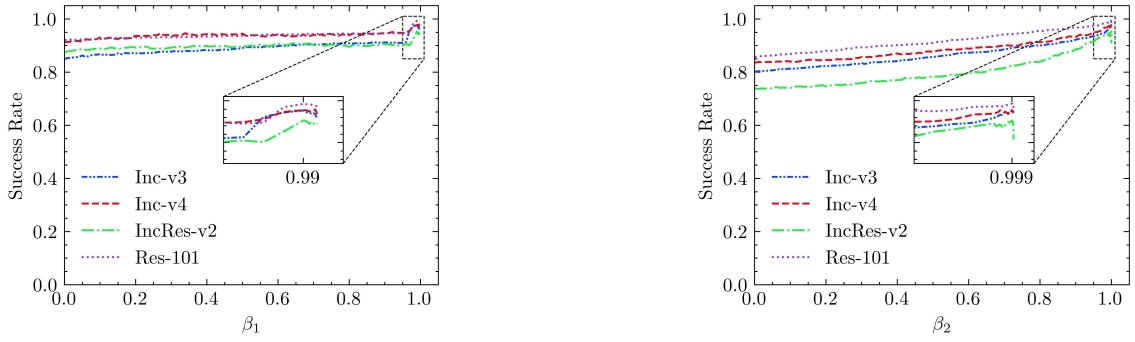
For the success rates in the IncRes-v2 model (shown in FIGURE 4.c), the discussed phenomenon shown in the Inc-v3 and Inc-v4 models becomes diverse. Specifically, our proposed attack was the first to achieve the best success rate and outperform the others. NI-FGSM attack has the lowest success rate in the first three iterations but gradually approaches the AB-FGSM in the following iterations. AI-FGSM is as poor as in the Inc-v4 model. As before, MI-FGSM and I-FGSM still overlap. We assume that MI-FGSM has limited improvements to I-FGSM in terms of the efficiency of generating adversarial examples in the white-box manner.

In FIGURE 4.d, we can see the trends in the success rates when generating adversarial examples in the Res-101 model using the five attacks. Five curves are more difficult to distinguish than before, and it is easy to distinguish five curves when $T=3$. We find that AB-FGSM is also the first to achieve a 100% success rate in four iterations. The second-best performance is MI-FGSM, which performs better than the previous three models. The followers are I-FGSM and AI-FGSM since their curves overlap. NI-FGSM performs poorly in the first three iterations but reaches the best success rate in the subsequent few iterations

Generally, NI-FGSM works well in the Inc-v3 and Inc-v4 models but not as expected in the IncRes-v2 and Res-101 models. The performance of MI-FGSM and I-FGSM overlaps in the Inc-v4 and IncRes-v2 models, and we infer that the momentum skill has a minor improvement for I-FGSM in the aspect of the generation efficiency of adversarial examples. Due to the limitation of the Adam optimizer, AI-FGSM requires more iterations to achieve the highest success rate in any model. In general, all baselines cannot perform stably in the four models. Our proposed AB-FGSM achieves the highest attack success rate with minor iterations in four models, indicating that AB-FGSM can generate adversarial examples with high efficiency in any model.

3) Influence of the hyperparameters β_1 and β_2

In this subsection, we will investigate how hyperparameters β_1 and β_2 influence the performance of AB-FGSM. As explained in Section III.A, the role of β_1 and β_2 is to control the EMA decay rates.



a. β_1 b. β_2

FIGURE 5 SUCCESS RATES OF ADVERSARIAL EXAMPLES GENERATED BY AB-FGSM ATTACK IN FOUR MODELS WITH β VARYING FROM 0 TO 1.0.

We choose β_1 ranging from 0 to 1.0 (shown in FIGURE 5.a) to see the optimal value for each model. The optimal value of the four models is 0.99, which indicates that the different models have no impact on the optimal β_1 value. An interesting phenomenon is that when β_1 is in (0.99, 1.0), the success rates drop sharply to 0. It is reasonable that when β_1 approaches 1, $\hat{m}_t$ will be too large and will add many perturbations when generating adversarial examples. However, β_1 is suggested as a value of 0.9 in the Adam and AdaBelief optimizers when training DNNs. Here, we give the optimal β_1 when crafting adversarial examples, and it is understandable that β_1 has different optimal values because the tasks differ.

FIGURE 5.b provides an overview of the success rates when β_2 changes in [0, 1.0). This figure shows that the optimal β_2 value for all models is 0.999, which is the same as suggested in the Adam and AdaBelief optimizers, and that this optimal β_2

value is also independent of the model architecture. We also adjust β_2 to [0.9991, 1), and the success rates continually decrease to 0.

Experimentally, $\beta_1 = 0.99$ and $\beta_2 = 0.999$ are optimal in the adversarial example optimization problems, regardless of the model architecture.

4) Influence of the size of perturbations U

This subsection mainly verifies the relationship between the perturbation size and the success rate of the adversarial attacks in the white-box and black-box manners. Concretely, we use five adversarial attacks to generate adversarial examples in the Inc-v3 model (solid lines in FIGURE 6). Then, the generated adversarial examples attack the black-box IncRes-v2 model (dotted lines in FIGURE 6).

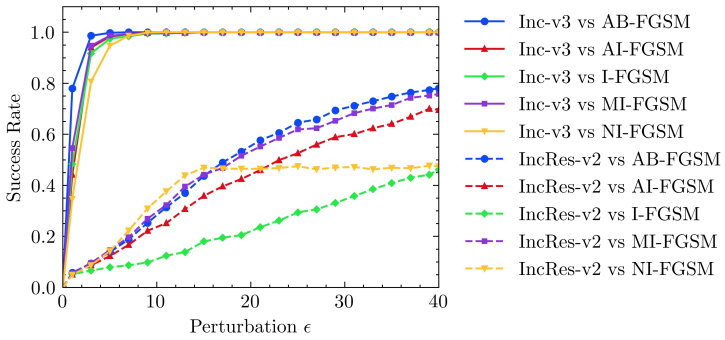


FIGURE 6 SUCCESS RATES OF ADVERSARIAL EXAMPLES GENERATED BY FIVE ATTACK METHODS IN THE INC-V3 MODEL AGAINST THE WHITE-BOX INC-V3 MODEL AND THE BLACK-BOX INCRES-V2 MODEL WITH PERTURBATIONS RANGING FROM 0 TO 40.

As demonstrated in FIGURE 6, our proposed AB-FGSM simply acquires 5 perturbation sizes to achieve a 100% success rate in the white-box manner. This phenomenon means that our proposed method can find the best direction to add distortion and take full advantage of the perturbations to generate adversarial examples. Meanwhile, other methods obtain a 94% to 98% success rate in 5 perturbation sizes. The surprising thing about FIGURE 6 is that NI-FGSM has the worst performance among the five methods, completely different from its performance in FIGURE 4. This result strongly verifies the efficiency and effectiveness of our AB-FGSM and indirectly implies that NI-FGSM cannot find a suitable direction to add effective perturbations. Furthermore, no significant difference is evident between AI-FGSM, MI-FGSM, and I-FGSM, and they are all slightly better than NI-FGSM.

In the black-box attacks, the dotted lines in FIGURE 6 show that there has been a steady increase in all adversarial attacks except NI-FGSM when the perturbation increases. Significantly, the success rate curve of MI-FGSM is slightly higher than that of AB-FGSM, but when the size of perturbation is greater than 17, AB-FGSM gradually outperforms MI-FGSM. The performance of AI-FGSM is inferior to that of MI-FGSM by an average of 6%. The worst performance is achieved by I-FGSM, which implies that although the momemtum skill has limited improvement in I-FGSM in the white-box manner, it dramatically improves the transferability of the adversarial examples generated by I-FGSM in the black-box manner.

Interestingly, what can be seen in the figure is the trend of NI-FGSM. First, it increases when the size of the perturbation is less than 15. After that, it levels off regardless of whether the size of the perturbation increases. This phenomenon experimentally confirms the above inference: NI-FGSM has limited ability to find the direction to add perturbation.

In summary, our proposed AB-FGSM can generate adversarial examples with minimal perturbations and simultaneously improve the transferability of adversarial examples.

5) Influence of the denominator stability parameter ε

The term ε in **Algo.1**. plays a role in making the divisor not 0. Therefore, it should be as small as possible mathematically. We run ε in $[10^{-20}, 0.1]$ and find that if ε is sufficiently small, it has less influence on the performance of AB-FGSM in any model, so we deliberately omit it to show it graphically.

6) Discussion

Here, we summarize each optimal hyperparameter of AB-FGSM as follows: the convenient hyperparameter range α is in (0.1, 0.5), and in this range, the iteration T assigned as 10 is sufficient. The optimal hyperparameters β_1 and β_2 are 0.99 and 0.999, respectively, regardless of the model architecture. In terms of the size of the perturbation, it is sufficient to establish Ú> 4 , but for the convenience of later experimental comparison, we set Ú= 16 . These hyperparameters are also directly used in the CIFAR-10 models to generate adversarial examples except U since U is set to 8 in the CIFAR-10 dataset.

C. Analysis of Attacking the Single Model

1) ImageNet dataset

This section aims to investigate the transferability of adversarial examples generated by five methods. TABLE II compares the results of attacking six models with five adversarial attacks. The symbol '*' denotes the results of the white-box attacks, and normal numbers are the results of the black-box attacks. The best score is in bold, and the second-best score is in italics to highlight the results.

What stands out in the table is that in the Inc-v3 model, the generation rate of all the methods is 100%, but the transfer rates are different from each other. Specifically, I-FGSM has the lowest transfer rate among the five methods, which meets expectations since the vanilla optimization method only finds a point among the U ball vicinity around x to fool the model. The U ball space contains the points that can fool the model, but this vanilla iterative method cannot find the best transferability points. The other sophisticated optimization methods not only consider the generation rate but also guarantee transferability. Therefore, the results of the other methods work better than I-FGSM. Among them, to our surprise, NI-FGSM performs worse than MI-FGSM, where the transfer rates of NI-FGSM on IncRes-v2~ $\text{Inc-v3}_{\text{adv}}$ models are 1%~4% lower than those of MI-FGSM. Notably, our AB-FGSM method works best in terms of both the generation rate and transfer rate. The transfer rate is almost 10%~38% higher than that of I-FGSM and 3%~6% higher than that of the second-best MI-FGSM.

Regarding the Inc-v4 model, the results are different from those of Inc-v3. In this case, the adversarial examples generated by I-FGSM have limited transferability, and the generation rates of all the methods are almost the same. However, the transfer rates of the five methods gradually increase from I-FGSM to AB-FGSM, except for the IncRes-v2 and $\text{Inc-v3}_{\text{ens3}}$ models. In these two models, our method achieves the second-best scores. NI-FGSM performs better than MI-FGSM, contrary to the Inc-v3 model, and implies that NI-FGSM performs well in complicated models.

TABLE II. SUCCESS RATES (%) OF FIVE ADVERSARIAL ATTACK METHODS AGAINST SIX **SINGLE** MODELS IN THE WHITE-BOX AND BLACK-BOX MANNERS ON THE IMAGENET DATASET

	Method	Inc-v3	Inc-v4	IncRes-v2	Res-101	$\text{Inc-v3}_{\text{ens3}}$	$\text{Inc-v3}_{\text{adv}}$
Inc-v3	I-FGSM	**100.0***	24.7	17.5	21.4	26.1	19.5
	MI-FGSM	**100.0***	51.4	46.0	44.1	**35.3**	33.8
	NI-FGSM	**100.0***	53.7	43.7	42.3	34.4	30.5
	AI-FGSM	**100.0***	44.7	38.5	37.4	33.2	28.8
	AB-FGSM	**100.0***	**55.2**	**51.2**	**47.3**	*34.6*	**39.0**
Inc-v4	I-FGSM	38.5	99.8*	18.8	24.2	28.5	21.4
	MI-FGSM	65.7	99.8*	51.4	49.9	37.5	36.4
	NI-FGSM	70.1	**99.9***	54.4	51.8	**38.7**	36.8
	AI-FGSM	71.8	99.7*	**56.9**	54.6	37.1	43.9
	AB-FGSM	**72.2**	**99.9***	*56.0*	**53.2**	*37.8*	**44.3**
IncRes-v2	I-FGSM	40.0	35.8	98.5*	29.7	30.8	24.1
	MI-FGSM	69.3	61.5	98.6*	52.5	39.1	40.1
	NI-FGSM	71.9	65.5	99.7*	55.6	**41.6**	40.6
	AI-FGSM	64.4	58.1	98.6*	49.7	38.6	38.2
	AB-FGSM	**76.6**	**68.5**	**100.0***	**58.6**	*41.0*	**45.0**
Res-101	I-FGSM	36.7	29.5	23.5	*99.5**	29.0	22.7
	MI-FGSM	61.4	55.3	46.2	*99.5**	**40.9**	35.7
	NI-FGSM	62.2	55.8	47.0	**99.6***	38.9	34.9
	AI-FGSM	58.2	50.7	42.8	*99.5**	38.7	32.7
	AB-FGSM	**69.3**	**62.8**	**55.7**	*99.5**	36.2	**41.6**

Another intriguing point is that the performance of AI-FGSM is slightly worse than ours, suggesting that AI-FGSM is not as stable as our AB-FGSM method. Surprisingly, our AB-FGSM achieves acceptable performance. Our method achieves the best transferability in the Inc-v3, Res-101, and $\text{Inc-v3}_{\text{adv}}$ models. Compared to the second-best AI-FGSM, the improvement is only 1%, which is limited, but ours is still better than MI-FGSM and NI-FGSM.

The performance of the five methods in the IncRes-v2 and Res-101 models is relatively consistent. In particular, NI-FGSM outperforms MI-FGSM, further validating our observation: NI-FGSM performs better than MI-FGSM as the model becomes complex. AI-FGSM performs poorly in these two models and is even worse than MI-FGSM, further illustrating the unreliability of AI-FGSM. Our proposed method achieves a 100% generation rate and the strongest transferability, whose scores are between 3% and 5% higher than those of the second-best NI-FGSM method. For the Res-101 model, the generation rates of the five methods are almost the same. Our approach achieves the best transferability scores among the four models in the black-box manner, which are approximately 7% higher than the second-best NI-FGSM.

In summary, our proposed method, AB-FGSM, can maintain a high generation rate, and the adversarial examples generated by it have strong transferability when attacking a single model. We also found an intriguing phenomenon from the experimental results: the adversarial examples generated by AI-FGSM are not transferable all the time.

2) CIFAR-10 dataset

TABLE III is the comparison of baselines on the CIFAR-10 dataset. We use the hyperparameters discussed in Section IV.**Error! Reference source not found.** to attack these CIFAR-10 models. In the white-box manner, a straightforward phenomenon is that our proposed AB-FGSM can always reach the best generation rate, where we obtain the 100% generation rate on VGG-13, DenseNet and GoogleNet and the best generation rate on the Res-18 model. I-FGSM obtains the second-best generation rate, whose generation rate is merely lower than ours. The generation rates of MI-FGSM and AI-FGSM are almost the same and lower than ours. Surprisingly, NI-FGSM performs poorly, and it obtains the lowest generation rate. Considering the excellent generation rate of AI-FGSM and the poor generation rate of NI-FGSM, we infer that the adaptive optimizer can achieve remarkable performance on the small-size dataset, and NI-FGSM is not stable and reliable on all the datasets.

In terms of black-box transfer attacks, our adversarial examples are still the most transferable. Concretely, whether our AB-FGSM generates adversarial examples on the Res-18, DenseNet, and GoogleNet models, these adversarial examples can always obtain the best transfer rate on the Res-18, VGG-13, DenseNet, GoogleNet and $\text{WideRes-28}_{\text{GAN}}$ models and the second-best transfer rate on the $\text{WideRes-28}_{\text{adv}}$ model. When AB-FGSM generates adversarial examples on the VGG-13 model, the transfer rate of adversarial examples is generally the second-best. The overall transfer rate of AB-FGSM is 1% larger than that of the second-best baseline. The improvement is limited since the tested model becomes small; therefore, the optimization surface becomes smooth, and the most transferable point is easy to find. The performance of MI-FGSM and AI-FGSM is almost the same but lower than ours. NI-FGSM is slightly better than I-FGSM, which indicates that NI-FGSM is sensitive to the dataset size and not reliable and stable. It is noticeable that the transfer rates of $\text{WideRes-28}_{\text{adv}}$ and $\text{WideRes-28}_{\text{GAN}}$ are relatively low. This is reasonable since their defensive methods play a role and are effective in the small-size dataset compared with the large-scale ImageNet dataset. Although the two models have effective defensive methods, our AB-FGSM can still obtain the best transfer rate in the $\text{WideRes-28}_{\text{GAN}}$ model and the second-best transfer rate in the $\text{WideRes-28}_{\text{adv}}$ model.

In conclusion, our AB-FGSM can generate adversarial examples on any model and dataset with high generation rates and transfer rates. Other baselines, especially AI-FGSM and NI-FGSM, are not reliable and stable for generating high transferability adversarial examples.

TABLE III. SUCCESS RATES (%) OF FIVE ADVERSARIAL ATTACK METHODS AGAINST SIX **SINGLE** MODELS IN THE WHITE-BOX AND BLACK-BOX MANNERS ON THE CIFAR-10 DATASET

	Method	Res-18	VGG-13	DenseNet	GoogleNet	$\text{WideRes-28}_{\text{adv}}$	$\text{WideRes-28}_{\text{GAN}}$
Res-18	I-FGSM	97.96*	50.82	54.00	56.93	9.09	10.61
	MI-FGSM	95.81*	62.80	60.83	65.06	9.63	11.00
	NI-FGSM	94.99*	56.60	56.97	61.34	9.63	10.93
	AI-FGSM	95.99*	62.73	60.87	65.19	**9.73**	11.04
	AB-FGSM	**98.51***	**63.85**	**61.87**	**66.35**	*9.68*	**11.14**
VGG-13	I-FGSM	70.36	99.99*	61.48	60.63	9.05	10.65
	MI-FGSM	**80.19**	99.86*	**70.86**	**71.11**	9.82	11.05
	NI-FGSM	68.22	99.09*	60.39	60.10	9.55	10.82
	AI-FGSM	77.42	99.65*	68.96	68.88	9.72	**11.17**
	AB-FGSM	*78.19*	**100.00***	*69.66*	*69.69*	**9.91**	*11.08*
DenseNet	I-FGSM	56.94	45.32	**100.00***	63.99	8.84	10.55
	MI-FGSM	72.70	63.99	**100.00***	78.02	**9.52**	10.68
	NI-FGSM	60.98	52.50	99.90*	67.13	9.16	10.48
	AI-FGSM	73.03	**64.15**	**100.00***	77.58	9.25	10.65
	AB-FGSM	**74.30**	*64.03*	**100.00***	**79.14**	*9.33*	**10.73**
GoogleNet	I-FGSM	75.30	57.63	73.17	**100.00***	9.47	10.88
	MI-FGSM	82.37	69.52	80.14	**100.00***	*9.86*	11.13
	NI-FGSM	71.51	60.22	70.55	99.96*	9.72	11.07
	AI-FGSM	82.37	69.18	80.21	**100.00***	**9.91**	11.10
	AB-FGSM	**83.26**	**69.31**	**80.87**	**100.00***	*9.86*	**11.15**

D. Analysis of Attacking Ensemble Models

1) ImageNet dataset

In this section, we analyze the performance of the five methods attacking ensemble models. TABLE IV gives the success rates of the five adversarial attack methods against ensemble models in white-box and black-box manners. In the column '-Inc-v3', 'Ensemble' denotes that a multiple model containing the other five models except the Inc-v3 model are merged by Eq. (III.6). 'Hold-out' indicates that the adversarial examples generated by one method on the previous ensemble model are transferred to the Inc-v3 model.

TABLE IV. SUCCESS RATES (%) OF FIVE ADVERSARIAL ATTACK METHODS AGAINST **ENSEMBLE** MODELS IN THE WHITE-BOX AND BLACK-BOX MANNERS ON THE IMAGENET DATASET

	Method	-Inc-v3	-Inc-v4	-IncRes-v2	-Res-101	-Inc-v3$_{ens3}$	-Inc-v3$_{adv}$	Avg.
Ensemble	I-FGSM	96.7	97.0	**98.7**	95.8	96.5	98.2	97.2
	MI-FGSM	96.3	96.5	98.6	96.2	96.3	97.3	96.9
	NI-FGSM	97.9	**98.2**	98.0	**98.1**	**99.5**	98.7	98.4
	AI-FGSM	96.2	96.7	98.4	95.6	95.2	97.9	96.7
	AB-FGSM	**98.6**	**98.2**	98.5	*98.0*	*99.4*	**99.1**	**98.6**
Hold-out	I-FGSM	55.0	45.8	39.4	36.1	25.6	24.8	37.8
	MI-FGSM	76.3	73.6	68.8	64.0	34.5	37.3	59.1
	NI-FGSM	52.7	50.6	56.1	44.6	32.9	30.4	44.6
	AI-FGSM	67.2	63.6	58.5	53.2	29.9	37.9	51.7
	AB-FGSM	**85.2**	**82.2**	**85.2**	**69.7**	**36.5**	**38.7**	**66.3**

As shown in TABLE IV, there is not much difference in the performance of the five methods in the ensemble white-box manner. Of the average generation rates, the best generation rate is only 1.9% higher than the worst. Specifically, our method achieves the best generation rate in the '-Inc-v3', '-Inc-v4' and '-Inc-v3$_{adv}$' ensemble models and the second-best in the '-Res-101' and '-Inc-v3$_{ens3}$' ensemble models, where the two second-best generation rates are only 0.1% lower than those of the best. Our method works relatively well in the '-IncRes-v2' ensemble model, where the generation rate is the third best and only 0.2% lower than the best I-FGSM generation rate. The performance of NI-FGSM is close to that of our proposed AB-FGSM. Their performance on average is only 0.2% lower than ours. And our method in the '-Inc-v3' ensemble performs almost 1% better than NI-FGSM. Other methods, such as MI-FGSM and AI-FGSM, are inferior to ours, and their average performance is 3% less than that of AB-FGSM. I-FGSM performs at the medium level, whose generation rate is 1.4% lower than ours.

However, the transfer rates of the five methods in ensemble models are highly divergent. Specifically, the transfer rate of our proposed AB-FGSM is 7.2%~28.5% higher than that of other methods on average across all models, which strongly verifies the transferability of the adversarial examples generated by AB-FGSM. NI-FGSM exceeds our expectations and only performs better than I-FGSM and 21.7% lower than ours on average. This can be explained because the optimization surface of the ensemble model is more complex than that of a single model. The Nesterov method cannot find the most transferable adversarial point on the complex surface. In contrast, the momentum method is theoretically as simple as the Nesterov method, but it can better handle the complex optimization surface and thus achieve the second-best performance. This is why the momentum method is used more often than the Nesterov method in training models. The performance of AI-FGSM is generally good, with a transfer rate that is 7.4% lower than MI-FGSM and 14.6% lower than ours. The worst performance is obtained with I-FGSM, which is in line with expectations.

In conclusion, our proposed AB-FGSM can consistently generate adversarial examples, either in single or ensemble models. In addition, the adversarial examples generated by our approach can be transferred to other models with a high attack success rate.

2) CIFAR-10 dataset

TABLE V. SUCCESS RATES (%) OF FIVE ADVERSARIAL ATTACK METHODS AGAINST **ENSEMBLE** MODELS IN THE WHITE-BOX AND BLACK-BOX MANNERS ON THE CIFAR-10 DATASET

	Method	-Res-18	-VGG-13	-DenseNet	-GoogleNet	- WideRes-28$_{adv}$	- WideRes-28$_{GAN}$	Avg.
Ensemble	I-FGSM	**99.79**	**99.69**	99.69	**99.68**	**99.90**	**99.79**	**99.75**
	MI-FGSM	**99.79**	99.48	99.27	99.37	*99.79*	**99.79**	99.54
	NI-FGSM	96.45	96.44	95.31	96.11	98.43	98.64	96.55
	AI-FGSM	**99.79**	99.37	99.27	99.37	*99.79*	**99.79**	99.52
	AB-FGSM	**99.79**	*99.58*	**99.79**	*99.47*	*99.79*	**99.79**	*99.68*
Hold-out	I-FGSM	95.62	84.73	93.74	90.55	8.81	11.33	74.69
	MI-FGSM	97.70	**92.05**	**94.99**	94.43	9.65	11.54	77.76
	NI-FGSM	87.79	77.30	80.92	79.41	9.76	12.28	67.04
	AI-FGSM	97.39	91.42	94.79	94.43	9.76	11.65	77.56
	AB-FGSM	**97.81**	*91.84*	*94.89*	**95.06**	**10.07**	**11.65**	**77.93**

This section verifies the transferability of adversarial examples against the ensemble models on the CIFAR-10 dataset, as shown in TABLE V. When attacking the ensemble models, I-FGSM performs best, and its generation rates are the largest. However, our generation rate is only 0.1% lower than that of I-FGSM. MI-FGSM and AI-FGSM perform the same, and their generation rate is 0.2% lower than ours. NI-FGSM performs poorly throughout, and its generation rates are 3% lower than those of other attacks.

For the transferability, our AB-FGSM achieves the best transfer rates on the '-Res-18', '-GoogleNet', '- WideRes-28_{adv}', and ' WideRes-28_{GAN}' models and is 4% higher than other baselines on average. MI-FGSM is close to ours, whose transfer rate is 0.2% lower than ours. AI-FGSM generally performs well, with transfer rates 0.4% lower than ours and 0.2% lower than MI-FGSM. Although I-FGSM's generation rate is high, its transfer rate is limited compared with ours and MI-FGSM since no optimization skills are used to help find the transferable adversarial point. NI-FGSM performs poorly, and its transfer rates are almost 11% lower than ours. Specifically, its performance on the CIFAR-10 dataset is as poor as that on the ImageNet dataset. We infer that the ability of the Nesterov method to find the most transferable adversarial point is limited.

In brief, our AB-FGSM can consistently and stably generate adversarial examples on different datasets by the single or ensemble models. And the transferability of the generated adversarial examples is the best among the baselines in most cases.

V. Conclusions and Future Work

In this paper, we propose an iterative method to generate adversarial samples dubbed AB-FGSM. This method efficiently finds the transferable adversarial point in a legitimate input on different optimization surfaces. The extensive experimental results presented in this paper demonstrate that our method can efficiently and effectively generate adversarial examples with a higher generation rate in the white-box manner and higher transfer rates in the black-box manner. The transfer rate is 4% and 22% higher than those of the state-of-the-art attack methods.

In future work, we aim to investigate the adaptation of AB-FGSM to defensive methods such as adversarial training to increase the robustness of DNNs.

References

[1] H. Onizuka, Z. Hayirci, D. Thomas, A. Sugimoto, H. Uchiyama, and R. Taniguchi, "TetraTSDF: 3D Human Reconstruction From a Single Image With a Tetrahedral Outer Shell," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6011–6020.

[2] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation," Mar. 2019. Available: http://arxiv.org/abs/1903.11816.

[3] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, p. 6.

[4] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," presented at the International Conference on Learning Representations, 2015.

[5] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 39–57. doi: 10.1109/SP.2017.49.

[6] F. Croce and M. Hein, "Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks," in *International Conference on Machine Learning*, Nov. 2020, pp. 2206–2216.

[7] F. Croce and M. Hein, "Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack," in *International Conference on Machine Learning*, Nov. 2020, pp. 2196–2205.

[8] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, Saarbrucken, Mar. 2016, pp. 372–387. doi: 10.1109/EuroSP.2016.36.

[9] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019, doi: 10.1109/TNNLS.2018.2886017.

[10] N. Liu, M. Du, R. Guo, H. Liu, and X. Hu, "Adversarial Attacks and Defenses: An Interpretation Perspective," *ACM SIGKDD Explor. Newsl.*, vol. 23, no. 1, pp. 86–99, May 2021, doi: 10/gmmdx6.

[11] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples," *ArXiv160507277 Cs*, May 2016. Available: http://arxiv.org/abs/1605.07277.

[12] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 4307–4316. doi: 10.1109/CVPR.2019.00444.

[13] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," presented at the International Conference on Learning Representations Workshop, 2017.

[14] Y. Wang, J. Liu, X. Chang, J. Mišić, and V. B. Mišić, "IWA: Integrated Gradient-based White-box Attacks for Fooling Deep Neural Networks," Int. J. Intell. Syst., Oct. 2021, doi: 10.1002/int.22720.

[15] W. Wu, Y. Su, X. Chen, S. Zhao, I. King, M. R. Lye, and Y. Tai, "Boosting the Transferability of Adversarial Samples via Attention," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1161–1170.

[16] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," Apr. 2020. Available: http://arxiv.org/abs/1705.07204.

[17] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering Adversarial Images using Input Transformations," presented at the International Conference on Learning Representations, Feb. 2018.

[18] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating Adversarial Effects Through Randomization," presented at the International Conference on Learning Representations, Feb. 2018.

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting Adversarial Attacks with Momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 9185–9193. doi: 10.1109/CVPR.2018.00957.

[20] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks," presented at the International Conference on Learning Representations, Feb. 2020.

[21] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, "AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 18795–18806.

[22] N. Qian, "On the Momentum Term in Gradient Descent Learning Algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999, doi: 10.1016/S0893-6080(98)00116-6.

[23] Y. E. NESTEROV, "A Method for Solving the Convex Programming Problem with Convergence Rate O(1/k^2)," *Dokl Akad Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[24] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” presented at the 3rd International Conference for Learning Representations, San Diego, California, USA, 2015.
[25] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” *ArXiv12125701 Cs*, Dec. 2012. Available: http://arxiv.org/abs/1212.5701.
[26] H. Yin, H. Zhang, J. Wang, and R. Dou, “Boosting Adversarial Attacks on Neural Networks with Better Optimizer,” *Secur. Commun. Netw.*, vol. 2021, p. e9983309, Jun. 2021, doi: 10.1155/2021/9983309.
[27] S. J. Reddi, S. Kale, and S. Kumar, “On the Convergence of Adam and Beyond,” presented at the International Conference on Learning Representations, Feb. 2018.
[28] H. Yin, H. Zhang, J. Wang, and R. Dou, “Improving the Transferability of Adversarial Examples with the Adam Optimizer,” *ArXiv201200567 Cs*, Dec. 2020. Available: http://arxiv.org/abs/2012.00567.
[29] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into Transferable Adversarial Examples and Black-box Attacks,” *ICLR 2017*.
[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 248–255. doi: 10/cvc7xp.
[31] K. Alex, “Learning Multiple Layers of Features from Tiny Images,” University of Toronto, Toronto, 2009.
[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.
[33] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, Feb. 2017, pp. 4278–4284.
[34] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 630–645. doi: 10.1007/978-3-319-46493-0_38.
[35] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015. Available: http://arxiv.org/abs/1409.1556.
[36] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 4700–4708.
[37] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, “Geometry-aware Instance-reweighted Adversarial Training,” presented at the International Conference on Learning Representations, Sep. 2020.
[38] S. Gowal, S.-A. Rebuffi, O. Wiles, F. Stimberg, D. A. Calian, and T. Mann, “Improving Robustness using Generated Data,” presented at the Advances in Neural Information Processing Systems, May 2021.
[39] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of Tricks for Image Classification with Convolutional Neural Networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2019, pp. 558–567.
[40] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2725–2734. doi: 10.1109/CVPR.2019.00284.
[41] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, and *et al.*, “TensorFlow: A System for Large-scale Machine Learning,” in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, USA, Nov. 2016, pp. 265–283.
[42] F. Croce and M. Hein, “Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack,” in International Conference on Machine Learning, Nov. 2020, pp. 2196–2205.