Improving Transferability of Adversarial Examples with Virtual Step and Auxiliary Gradients

Ming Zhang, Xiaohui Kuang, Hu Li*, Zhendong Wu, Yuanping Nie, Gang Zhao
National Key Laboratory of Science and Technology on Information System Security, Beijing, China
zm_stiss@163.com, xhkuang@bupt.edu.cn, {lihu, wuzhendong, yuanpingnie}@nudt.edu.cn, zemell@foxmail.com

Abstract

Deep neural networks have been demonstrated to be vulnerable to adversarial examples, which fool networks by adding human-imperceptible perturbations to benign examples. At present, the practical transfer-based black-box attacks are attracting significant attention. However, most existing transfer-based attacks achieve only relatively limited success rates. We propose to improve the transferability of adversarial examples through the use of a virtual step and auxiliary gradients. Here, the "virtual step" refers to using an unusual step size and clipping adversarial perturbations only in the last iteration, while the "auxiliary gradients" refer to using not only gradients corresponding to the ground-truth label (for untargeted attacks), but also gradients corresponding to some other labels to generate adversarial perturbations. Our proposed virtual step and auxiliary gradients can be easily integrated into existing gradient-based attacks. Extensive experiments on ImageNet show that the adversarial examples crafted by our method can effectively transfer to different networks. For single-model attacks, our method outperforms the state-of-the-art baselines, improving the success rates by a large margin of $12\% \sim 28\%$. Our code is publicly available at https://github.com/mingcheung/Virtual-Step-and-Auxiliary-Gradients.

1 Introduction

The security of machine learning has attracted much attention for many years. For example, [Dalvi *et al.*, 2004] first proposed the concept of adversarial classification, while [Biggio *et al.*, 2013] presented evasion attacks against machine learning models. Subsequently, [Szegedy *et al.*, 2014] were the first to formally propose the concept of adversarial examples faced by deep neural networks in computer vision, which triggered a widespread research interest in adversarial attacks.

Various forms of adversarial attack have been proposed in an attempt to craft adversarial examples in the field of computer vision [Akhtar and Mian, 2018; Akhtar *et al.*, 2021]. Theses attacks can be broadly divided into two categories: white-box attacks and black-box attacks. White-box attacks assume complete knowledge of the target model, including its architecture, parameters, *etc*. White-box attacks are predominately gradient-based; examples include FGSM [Goodfellow *et al.*, 2015], I-FGSM [Kurakin *et al.*, 2017] and MI-FGSM [Dong *et al.*, 2018]. Since attackers can fully exploit model information such as gradients under white-box settings, white-box attacks can achieve high success rates and low human perceptibility. Currently, great strides have been made in research into white-box attacks. Black-box attacks, which are more practical, have attracted increasing research attention in recent years.

Except for inputs and outputs, black-box attacks assume no knowledge of the target model, making them more challenging and more practical. Currently, black-box attacks are being developed along two main directions: query-based attacks and transfer-based attacks. Query-based attacks query the target model and use its outputs to craft adversarial examples. Since too many queries being sent to the target model can arouse suspicion, the core research topic related to query-based attacks is to improve query efficiency. Transfer-based attacks craft adversarial examples on local substitute models and fool the target model by using the cross-model transferability of adversarial examples. The core research topic of transfer-based attacks centers around improving the transferability of adversarial examples.

In this work, we propose to improve the transferability of adversarial examples through the use of a virtual step and auxiliary gradients, which can be easily integrated into existing gradient-based attack methods to yield more powerful attacks. On the one hand, we use an usual step size ("virtual step") and clip adversarial perturbations only in the last iteration; on the other hand, unlike the traditional methods which use only gradients corresponding to the ground-truth label (for untargeted attacks), we use not only gradients corresponding to the ground-truth label ("main gradients"), but also gradients corresponding to some other labels ("auxiliary gradients") to generate adversarial examples. To our best knowledge, we are the first to verify the synergistic effect of using main gradients and auxiliary gradients simultaneously.

We evaluate the performance of the proposed method on both normally trained models and defense models under single-model attacks and ensemble-based attacks. The re-

*Corresponding author.

sults on ImageNet show that our method significantly outperforms the state-of-the-art baselines. For example, under single-model attacks, our method improves the attack success rates by a large margin of $12\% \sim 28\%$. We hope that our proposed attack strategy will shed light on new types of adversarial attacks and serve as a benchmark for evaluating the robustness of neural networks. This paper makes the following contributions:

- We propose the concepts of "virtual step" and "auxiliary gradients", which can be integrated into existing gradient-based methods to craft more transferable adversarial examples.
- We demonstrate that in addition to the main gradients, the auxiliary gradients can also make an contribution to generate adversarial examples.
- We systematically evaluate the performance of our proposed method, and show that the proposed method significantly outperforms the state-of-the-art baselines.

2 Related Work

Adversarial attacks. [Szegedy *et al.*, 2014] were the first to demonstrate that deep neural networks have counter-intuitive properties, and accordingly proposed a box-constrained L-BFGS method for finding adversarial examples. [Goodfellow *et al.*, 2015] proposed the fast gradient sign method (FGSM) to efficiently craft adversarial examples. [Kurakin *et al.*, 2017] extended FGSM to an iterative version, whilte [Madry *et al.*, 2018] proposed the projected gradient descent method, in which the ℓ_∞-norm of the perturbations is bounded by the clipping operation—the projection. The above are known as white-box attacks. In terms of black-box attacks, Boundary attack [Brendel *et al.*, 2018], qFool attack [Liu *et al.*, 2019] and HopSkipJump attack [Chen *et al.*, 2020] explore the decision boundary to launch query-based attacks. [Xie *et al.*, 2019b] proposed to improve the transferability of adversarial examples by creating diverse input patterns. [Dong *et al.*, 2019] developed a translation-invariant attack capable of generating more transferable adversarial examples against the defense models. [Li *et al.*, 2020] proposed the ghost networks, which can be used to efficiently construct substitute models in transfer-based attacks.

Adversarial defenses. While demonstrating the existence of adversarial examples in deep neural networks, [Szegedy *et al.*, 2014] also identified a potential direction for defending against adversarial examples—adversarial training. Adversarial training [Madry *et al.*, 2018; Tramer *et al.*, 2018] defends against adversarial perturbations by injecting adversarial examples into the training data. At present, adversarial training is hampered by low efficiency and can obtain only limited robustness. [Guo *et al.*, 2018] and [Xie *et al.*, 2018] proposed to use input transformations to mitigate adversarial effects. [Liao *et al.*, 2018] introduced a high-level representation guided denoiser (HGD) defense method. [Xie *et al.*, 2019a] proposed to improve adversarial robustness by performing feature denoising, which is recommended to be used in combination with adversarial training.

3 Methodology

Let $\boldsymbol{x}$ denote a benign example, while y^{true} denotes the corresponding ground-truth label. Given a classifier f, we apply an imperceptible perturbation to $\boldsymbol{x}$ to craft an adversarial example $\boldsymbol{x}^{adv}$ that is capable of fooling the classifier, *i.e.*, $f(\boldsymbol{x}^{adv}) \neq y^{true}$ (for untargeted attacks). In most cases, the L_p-norm is used to bound adversarial perturbations in order to achieve imperceptibility, *i.e.*, $\|\boldsymbol{x}^{adv} - \boldsymbol{x}\|_p \leq \epsilon$. If the loss function of the classifier is denoted as J, the process for crafting adversarial examples is usually derived by maximizing the loss function $J(\boldsymbol{x}^{adv}, y^{true})$, which can be formally expressed as follows:

$$\arg\max_{\boldsymbol{x}^{adv}} J(\boldsymbol{x}^{adv}, y^{true}); \text{ s.t.} \|\boldsymbol{x}^{adv} - \boldsymbol{x}\|_p \leq \epsilon \qquad (1)$$

For white-box attacks, the above optimization problem can be solved by calculating the gradients of the loss function with respect to the input. For transfer-based black-box attacks, we can employ a substitute white-box model to craft adversarial examples, which are then expected to be misclassified by the target black-box model due to the transferability of adversarial examples.

3.1 Family of Gradient-based Adversarial Attacks

[Goodfellow *et al.*, 2015] proposed a one-step gradient-based method, known as the fast gradient sign method (FGSM), for efficiently crafting adversarial examples. Subsequently, several variants of FGSM were proposed. In this section, we briefly introduce three typical variants.

Iterative Fast Gradient Sign Method (I-FGSM). [Kurakin *et al.*, 2017] extended FGSM to an iterative version, which can be expressed as follows:

$$\boldsymbol{x}_{t+1}^{adv} = \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{true}))\} \quad (2)$$

where $\boldsymbol{x}_0^{adv} = \boldsymbol{x}$, while $\boldsymbol{x}_i^{adv}$ denotes the perturbed example at the i-th iteration; $\text{Clip}_{\boldsymbol{x},\epsilon}\{\cdot\}$ indicates that the perturbed example is clipped within the ϵ-ball of the original example $\boldsymbol{x}$; α is the step size at each iteration (normally, $\alpha = \epsilon/T$, where T is the total number of iterations.).

Diverse Inputs Iterative Fast Gradient Sign Method (DI^2-FGSM). [Xie *et al.*, 2019b] proposed a variant of FGSM designed to improve the transferability of adversarial examples by creating diverse input patterns, which can be expressed as follows:

$$\boldsymbol{x}_{t+1}^{adv} = \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(T(\boldsymbol{x}_t^{adv}; p), y^{true}))\} \quad (3)$$

where $T(\boldsymbol{x}_t^{adv}; p)$ denotes the application of a random transformation to the input $\boldsymbol{x}_t^{adv}$ with a probability p. The transformation includes random resizing and padding.

Translation-Invariant Iterative Fast Gradient Sign Method (TI^2-FGSM). [Dong *et al.*, 2019] proposed a translation-invariant method dedicated to crafting more transferable adversarial examples to combat the defense models. The method is finally implemented by convolving the gradient at the untranslated example with a pre-defined kernel, which can be expressed as follows:

$$\boldsymbol{x}_{t+1}^{adv} = \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\boldsymbol{W} * \nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{true}))\} \quad (4)$$

where $\boldsymbol{W}$ denotes a pre-defined kernel, which can be uniform, linear or Gaussian.

3.2 Motivation

The traditional iterative attacks, *e.g.*, I-FGSM, greedily perturb the examples in the gradient direction of loss function, thus easily falling into the poor local maximum [Xie *et al.*, 2019b]. We experimentally determine that the untargeted I-FGSM attack is almost identical to the most-likely-class targeted I-FGSM attack; in other words, an untargeted adversarial example generated by I-FGSM is usually misclassified to the neighboring class of the original ground-truth class.

We divide the examples involved in iterative attacks into three categories: 1) the original benign example $\boldsymbol{x}$, namely $\boldsymbol{x}_0^{adv}$; 2) the intermediate perturbed examples $\boldsymbol{x}^{inter}$, namely $\boldsymbol{x}_i^{adv}$, where $i \in [1, T-1]$, and T is the number of iterations; 3) the final adversarial example $\boldsymbol{x}^{adv}$, namely $\boldsymbol{x}_T^{adv}$. In order to generate more transferable adversarial examples, we put forward the following two hypotheses[1].

Hypothesis 1. *If the final adversarial example $\boldsymbol{x}^{adv}$ is misclassified to a class that is far away from the ground-truth class on the substitute model, it is more likely to be misclassified by the target model.*

The intuition behind this hypothesis is that the farther the class of the adversarial example is from that of the benign example, the lower the similarity between the two examples; therefore, the adversarial example is more likely to be misclassified by the target model.

Hypothesis 2. *If the intermediate perturbed examples $\boldsymbol{x}^{inter}$ are misclassified to different classes on the substitute model, the final adversarial example $\boldsymbol{x}^{adv}$ is more likely to be misclassified by the target model.*

Let $c(\boldsymbol{x}')$ denote the classified class of $\boldsymbol{x}'$ on the substitute model. We think $\{c(\boldsymbol{x}), c(\boldsymbol{x}^{inter}), c(\boldsymbol{x}^{adv})\}$ are all potential classified classes of $\boldsymbol{x}^{adv}$ on the target model. Therefore, the probability the final adversarial example being misclassified on the target model can be increased by increasing the number of $c(\boldsymbol{x}^{inter})$, *i.e.*, improving the diversity of misclassification classes of the intermediate perturbed examples. For the target model, this seems to change what is essentially a coin-toss game into a dice-toss game.

Based on the above two hypotheses, the method we envisage for generating more transferable adversarial examples is illustrated in Figure 1. Below we will present how to implement such attack method.

3.3 Virtual Step and Auxiliary Gradients Method

We propose two techniques to verify the above two hypotheses and to implement a novel attack method that can generate adversarial examples with more transferability.

Virtual Step. This technique is designed to verify the first hypothesis. To avoid exceeding the limit of the perturbation threshold ϵ, the traditional iterative attacks (such as I-FGSM) usually set the step size α to a small value (normally ϵ/T), and clip the intermediate perturbed examples (*i.e.*, using the $\text{Clip}_{\boldsymbol{x},\epsilon}\{\cdot\}$ operation) during each iteration, which will cause the attack to fall into the poor local maximum [Xie *et al.*, 2019b]. In our method, we propose to set the step size α to a larger value and only clip the perturbed example generated in the last iteration, *i.e.*, $\text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_T^{adv}\}$; this can help the attack to avoid converging on the local optimum and increase the likelihood that the final adversarial example will be misclassified to a class far away from the ground-truth class. Since the step size used in the attack is not the real perturbation amplitude of the examples, we refer to it as "virtual step".

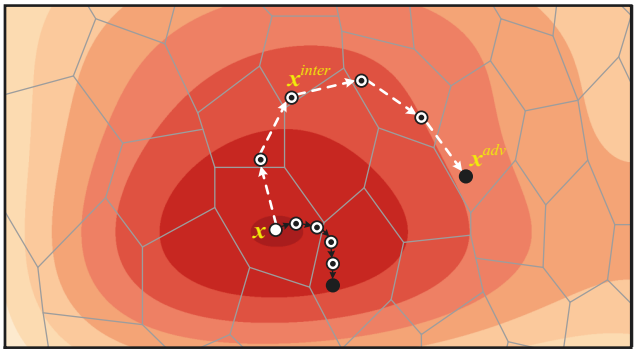

Figure 1: Illustrations of our envisaged attack (indicated by the white dashed arrow) and the traditional iterative gradient-based attack (indicated by the black dashed arrow). The irregular polygons represent the boundaries of each class.

Auxiliary Gradients. This technique is designed to verify the second hypothesis. By using the gradients of the loss function with respect to the ground-truth label y^{true}, traditional iterative attacks (such as I-FGSM) perturb the examples in approximately one fixed direction, which is not conducive to making the intermediate perturbed examples be misclassified to different classes. We propose to use not only the gradients corresponding to the ground-truth label (main gradients), but also the gradients corresponding to some other labels (auxiliary gradients) to generate adversarial perturbations. For an I-FGSM using auxiliary gradients , it will perturb the examples both in the ascending direction of the main gradients and the descending direction of the auxiliary gradients. Through the use of auxiliary gradients, the intermediate perturbed examples may potentially be misclassified to the auxiliary classes.

Based on the above analysis, the iterative fast gradient sign method with virtual step and auxiliary gradients (abbreviated to VA-I-FGSM) can be formulated as follows:

$$\begin{aligned} \boldsymbol{x}_{t+1}^{adv} &= \boldsymbol{x}_t^{adv} \pm \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y_+^{true}, y_-^{aux})); \\ \boldsymbol{x}^{adv} &= \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_T^{adv}\} \end{aligned} \tag{5}$$

where α denotes the virtual step, y_+^{true} denotes that $\boldsymbol{x}_t^{adv}$ is updated by adding the sign of the loss gradients with respect to y^{true}, and y_-^{aux} denotes that $\boldsymbol{x}_t^{adv}$ is updated by subtracting the sign of the loss gradients with respect to y^{aux}. Note that there may be more than one y^{aux} and that $y^{aux} \neq y^{true}$.

A detailed description of VA-I-FGSM is presented in Algorithm 1. In each iteration, in addition to updating the example with the main gradients corresponding to y^{true}, we ran-

[1]These hypotheses are for untargeted attacks. We leave research on targeted attacks for our future work.

Algorithm 1 VA-I-FGSM for crafting adversarial examples.

Input: A classifier f with loss function J; a benign example $\boldsymbol{x}$ and its true label y^{true}; the label set $\mathbb{C}$; the number of iterations T; the perturbation threshold ϵ; the virtual step size α; the number of auxiliary labels n_{aux}.
Output: The adversarial example $\boldsymbol{x}^{adv}$.

Let $\boldsymbol{x}_0^{adv} \leftarrow \boldsymbol{x}$; $t \leftarrow 0$.
while $t < T$ **do**
  $\boldsymbol{x}_{tmp}^{adv} \leftarrow \boldsymbol{x}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{true}))$
  $\mathbb{C}^{aux} \leftarrow \text{RandomSelect}(\mathbb{C} \backslash y^{true}, n_{aux})$
  for y^{aux} in $\mathbb{C}^{aux}$ **do**
    $\boldsymbol{x}_{tmp}^{adv} \leftarrow \boldsymbol{x}_{tmp}^{adv} - \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^{adv}, y^{aux}))$
  end for
  $\boldsymbol{x}_{t+1}^{adv} \leftarrow \boldsymbol{x}_{tmp}^{adv}$
  $t \leftarrow t + 1$
end while
return $\boldsymbol{x}^{adv} \leftarrow \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}_T^{adv}\}$

domly select n_{aux} labels from $\mathbb{C} \backslash y^{true}$ to construct auxiliary gradients for generating additional adversarial perturbations. $\mathbb{C} \backslash y^{true}$ represents the set formed by removing the element y^{true} from set $\mathbb{C}$. No clipping operation is performed for the intermediate perturbed examples, and only the final adversarial example is clipped to satisfy the perturbation threshold.

Our proposed virtual step and auxiliary gradients can be similarly integrated into DI^2-FGSM and TI^2-FGSM to create VA-DI^2-FGSM and VA-TI^2-FGSM, respectively.

4 Experiments

4.1 Experimental Setup

Dataset. We use a subset dataset[2] of ImageNet to conduct the experiments. This subset dataset consists of 1000 images and was used in the NIPS 2017 adversarial competition. All the images are resized to $299 \times 299 \times 3$ pixels.

Models. We consider four normally trained models, *i.e.*, Inception-v3 (Inc-v3) [Szegedy *et al.*, 2016], Xception (Xcep) [Chollet, 2017], Inception-Resnet-v2 (IncRes-v2) [Szegedy *et al.*, 2017] and ResNet-152-v2 (Res152-v2) [He *et al.*, 2016], and three defense models, *i.e.*, input transformation through JPEG compression (JPEG) [Guo *et al.*, 2018], input transformation through random resizing and padding (R&P, rank-2 submission in the NIPS 2017 defense competition) [Xie *et al.*, 2018], ResNeXt101 DenoiseAll (ResX101-dn, rank-1 submission in CAAD 2018) [Xie *et al.*, 2019a]. All models are publicly available[3].

Implementation details. For all attacks, the maximum perturbation of each pixel is set to $\epsilon = 16$. The total number of iterations is set to $T = \min(\epsilon + 4, 1.25\epsilon)$ [Kurakin *et al.*, 2017]. For I-FGSM, DI^2-FGSM and TI^2-FGSM, the step size is set to $\alpha = \epsilon/T$. For DI^2-FGSM, the transformation operations $T(\boldsymbol{x}; p)$ first randomly resize the input to a $rnd \times rnd \times 3$ image, with $rnd \in [299, 330)$, then pad to size $330 \times 330 \times 3$ in a random manner. The transformation probability p is set to be 0.5. For TI^2-FGSM, $\boldsymbol{W}$ is set to be a 15×15 Gaussian kernel. In experiments, the pixel values of all images are scaled to $[0, 1]$. Correspondingly, the ϵ is scaled to $16/255$.

[2]https://www.kaggle.com/c/nips-2017-non-targeted-adversarial-attack/data

[3]https://keras.io/api/applications/

4.2 Setting the Hyperparameters

We use a grid search to find the optimal values for hyperparameters α and n_{aux}. We first attack Inc-v3 by VA-I-FGSM, VA-DI^2-FGSM and VA-TI^2-FGSM under white-box settings, and then transfer the adversarial examples to Xcep, IncRes-v2 and Res152-v2. Figure 2 presents the attack success rates of the adversarial examples crafted on Inc-v3 by VA-I-FGSM. It can be seen that, for the three different black-box models, hyperparameters associated with high attack success rates have almost the same distribution. For Xcep, the attack success rate is the highest when $\alpha = 0.007$ and $n_{aux} = 3$; for IncRes-v2, the attack success rate is the highest when $\alpha = 0.007$ and $n_{aux} = 3$; for Res152-v2, the attack success rate is the highest when $\alpha = 0.008$ and $n_{aux} = 2$. Thus, for VA-I-FGSM, we set $\alpha = 0.007$ and $n_{aux} = 3$ according to the majority rule.

Similarly, we have searched the optimal hyperparameters for VA-DI^2-FGSM and VA-TI^2-FGSM. For VA-DI^2-FGSM, the hyperparameters are set to $\alpha = 0.009$ and $n_{aux} = 1$; for VA-TI^2-FGSM, the hyperparameters are set to $\alpha = 0.009$ and $n_{aux} = 4$. The following experiments are all based on the above hyperparameters.

4.3 Attacking a Single Model

We first conduct attacks on a single model. We craft adversarial examples only on normally trained models under white-box settings, and test them on all seven models. The attack success rates are shown in Table 1.

From Table 1, we can observe that, for black-box attacks, our proposed methods with virtual step and auxiliary gradients outperforms the baseline methods on all models, including normally trained and defense models. It is worth noting that, with the exception of ResX101-dn, our methods improve the attack success rates by a large margin of $12\% \sim 28\%$ on all other models. For example, if the adversarial examples are crafted on Inv-v3, VA-I-FGSM achieves an attack success rate of 41.6% on Xcep, while the baseline I-FGSM obtains an attack success rate of only 13.8%. On ResX101-dn, our methods are slightly superior to the baselines. Since the classification accuracy of ResX101-dn on benign examples is as low as 82.1%, we treat ResX101-dn as a special case that may be countered by adaptive attacks. For white-box attacks, only VA-TI^2-FGSM is slightly inferior to TI^2-FGSM in some cases, while VA-I-FGSM and VA-DI^2-FGSM surpass I-FGSM and DI^2-FGSM, respectively.

Figure 3 visualizes two randomly selected benign images and their corresponding adversarial images crafted by different methods. Overall, all adversarial images are visually indistinguishable from benign images. The adversarial perturbations generated by our methods are slightly more perceptible for humans than those generated by baseline methods. However, it should be noted that the adversarial images crafted by our methods still do not exceed the perturbation threshold of $\epsilon = 16$.

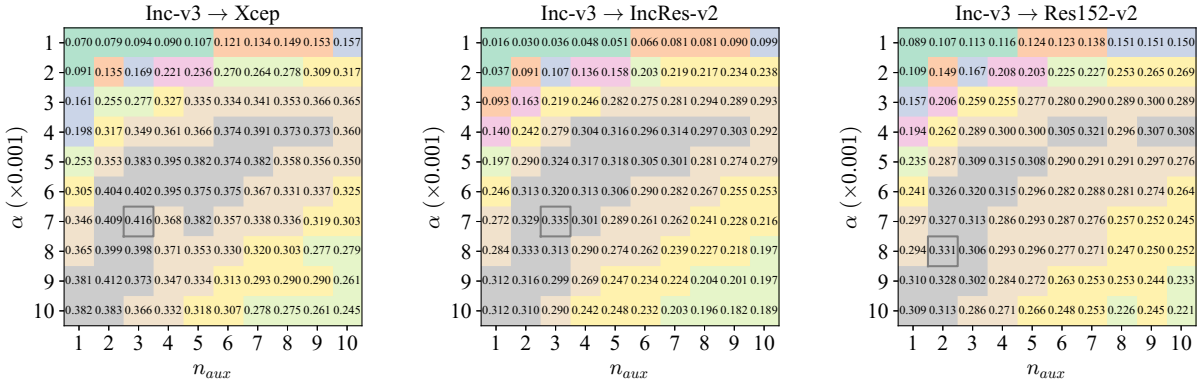

Figure 2: Attack success rates on Xcep, IncRes-v2 and Res152-v2. The adversarial examples are crafted on Inc-v3 using VA-I-FGSM.

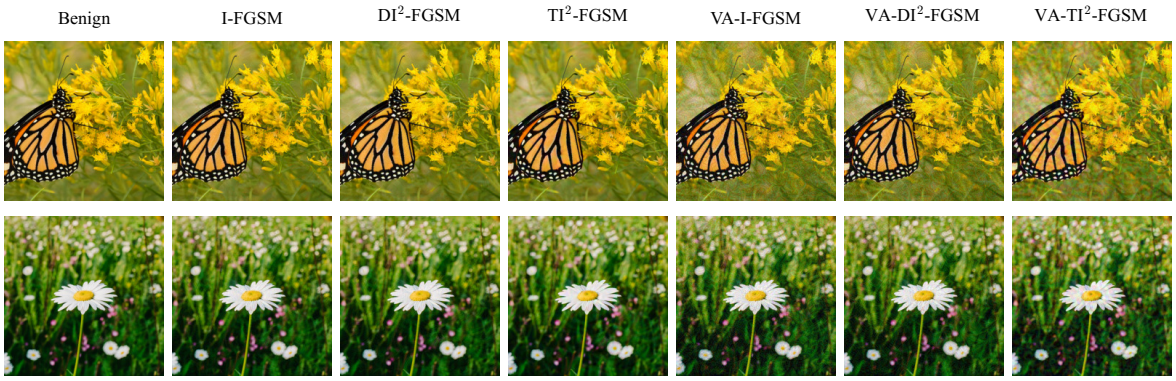

Figure 3: Visualization of randomly selected benign images and their corresponding adversarial images. The adversarial images are crafted on Inc-v3 using different methods with a maximum perturbation of $\epsilon = 16$.

4.4 Attacking an Ensemble of Models

In this section, we test the performance of our methods on an ensemble of models. We adopt the ensemble strategy proposed in [Dong *et al.*, 2018], which fuses the logits of different models. The experimental results on normally trained models are presented in Table 2. We can observe that the methods with virtual step and auxiliary gradients outperform all baseline methods.

The experimental results on defense models are presented in Table 3. We can observe that VA-I-FGSM and VA-DI^2-FGSM outperform I-FGSM and DI^2-FGSM, respectively. On JPEG and R&P defense models, VA-TI^2-FGSM is inferior to TI^2-FGSM.

4.5 Ablation Study

In this section, we conduct an ablation study to analyze the influence of virtual step and auxiliary gradients in attacks. We take VA-TI^2-FGSM as an example. For VA-TI^2-FGSM, the optimal hyperparameters are $\alpha = 0.009$ and $n_{aux} = 4$.

Virtual step size α. We set the number of auxiliary labels to $n_{aux} = 0$, *i.e.*, no auxiliary gradients are used in attacks. The virtual step size α varies from 0.001 to 0.01. In this case, VA-TI^2-FGSM degrades to TI^2-FGSM with only a virtual step.

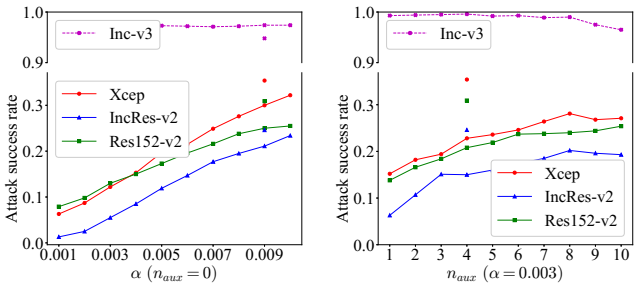

Figure 4: The attack success rates of VA-TI^2-FGSM when varying α (left) or n_{aux} (right). The dashed lines denote white-box models, and the solid lines denote black-box models. The outlier points indicate the benchmark attack success rates obtained by setting $\alpha = 0.009$ and $n_{aux} = 4$.

The results are shown on the left of Figure 4. We observe that the white-box attack success rates remain stable as α increases. The black-box attack success rates increase with increasing α, but gradually flatten out. Given $\alpha \in [0.001, 0.01]$, the black-box attack success rates do not reach the benchmark values. Moreover, if $\alpha \gg \epsilon/T$, the perturbations will tend to become increasingly human-perceptible in practice.

Model	Attack	Inc-v3	Xcep	IncRes-v2	Res152-v2	JPEG	R&P	ResX101-dn
Inc-v3	I-FGSM	0.973	0.138	0.091	0.150	0.127	0.201	0.180
	VA-I-FGSM	**0.999**	**0.416**	**0.335**	**0.313**	**0.306**	**0.414**	**0.186**
	DI^2-FGSM	0.976	0.291	0.219	0.255	0.246	0.402	0.182
	VA-DI^2-FGSM	**0.999**	**0.508**	**0.457**	**0.433**	**0.439**	**0.602**	**0.190**
	TI^2-FGSM	**0.973**	0.129	0.051	0.132	0.155	0.177	0.187
	VA-TI^2-FGSM	0.948	**0.354**	**0.246**	**0.309**	**0.361**	**0.404**	**0.198**
Xcep	I-FGSM	0.234	0.978	0.122	0.182	0.169	0.233	0.183
	VA-I-FGSM	**0.457**	**1.000**	**0.339**	**0.364**	**0.355**	**0.451**	**0.188**
	DI^2-FGSM	0.494	0.981	0.345	0.375	0.339	0.514	0.186
	VA-DI^2-FGSM	**0.662**	**0.999**	**0.510**	**0.524**	**0.535**	**0.669**	**0.188**
	TI^2-FGSM	0.237	**0.976**	0.106	0.185	0.235	0.255	0.187
	VA-TI^2-FGSM	**0.433**	0.956	**0.261**	**0.319**	**0.388**	**0.418**	**0.196**
IncRes-v2	I-FGSM	0.315	0.217	**0.995**	0.202	0.182	0.260	0.182
	VA-I-FGSM	**0.498**	**0.411**	0.992	**0.325**	**0.343**	**0.431**	**0.188**
	DI^2-FGSM	0.557	0.441	0.988	0.387	0.371	0.523	0.183
	VA-DI^2-FGSM	**0.736**	**0.648**	**0.999**	**0.528**	**0.580**	**0.691**	**0.189**
	TI^2-FGSM	0.281	0.217	**0.969**	0.197	0.251	0.262	0.187
	VA-TI^2-FGSM	**0.442**	**0.374**	0.788	**0.326**	**0.400**	**0.423**	**0.195**
Res152-v2	I-FGSM	0.254	0.195	0.089	0.970	0.143	0.223	0.185
	VA-I-FGSM	**0.462**	**0.462**	**0.330**	**1.000**	**0.334**	**0.478**	**0.194**
	DI^2-FGSM	0.508	0.475	0.372	0.971	0.343	0.502	0.186
	VA-DI^2-FGSM	**0.698**	**0.684**	**0.560**	**0.999**	**0.544**	**0.705**	**0.197**
	TI^2-FGSM	0.209	0.180	0.077	0.967	0.193	0.222	0.189
	VA-TI^2-FGSM	**0.402**	**0.372**	**0.266**	**0.968**	**0.357**	**0.390**	**0.198**

Table 1: Attack success rates of single-model attacks. The diagonal blocks indicate white-box attacks, while the off-diagonal blocks indicate black-box attacks.

Attack	-Inc-v3	-Xcep	-IncRes-v2	-Res152-v2
I-FGSM	0.573	0.447	0.362	0.341
VA-I-FGSM	**0.659**	**0.603**	**0.525**	**0.475**
DI^2-FGSM	0.846	0.766	0.704	0.670
VA-DI^2-FGSM	**0.914**	**0.880**	**0.827**	**0.749**
TI^2-FGSM	0.549	0.438	0.321	0.329
VA-TI^2-FGSM	**0.556**	**0.476**	**0.375**	**0.413**

Table 2: Attack success rates of ensemble-based attacks on normally trained models. "-Inc-v3" indicates that the adversarial examples are crafted on an ensemble of Xcep, IncRes-v2 and Res152-v2, and black-box attacks are performed on Inc-v3. The meaning of the other symbols can be deduced by analogy.

Attack	JPEG	R&P	ResX101-dn
I-FGSM	0.472	0.577	0.188
VA-I-FGSM	**0.501**	**0.618**	**0.198**
DI^2-FGSM	0.785	0.868	0.193
VA-DI^2-FGSM	**0.849**	**0.911**	**0.202**
TI^2-FGSM	**0.578**	**0.574**	0.198
VA-TI^2-FGSM	0.485	0.526	**0.208**

Table 3: Attack success rates of ensemble-based attacks on defense models. The adversarial examples are craft on an ensemble of Inc-v3, Xcep, IncRes-v2 and Res152-v2.

It is therefore unrealistic to rely only on the virtual step to improve the transferability of adversarial examples.

Number of auxiliary labels n_{aux}. We set the virtual step size to $\alpha = \epsilon/T \approx 0.003$, *i.e.*, the step size usually used by traditional iterative attacks. The number of auxiliary labels n_{aux} varies from 1 to 10. In this case, VA-TI^2-FGSM degrades to TI^2-FGSM with only auxiliary gradients. The results are shown on the right of Figure 4. It can be seen that if $n_{aux} \leq 4$, the black-box success rates increase significantly; by contrast, if $n_{aux} > 4$, the black-box attack success rates increase moderately and the white-box attack success rates tend to drop. Given $n_{aux} \in [1, 10]$, the black-box attack success rates do not reach the benchmark values. Moreover, the time cost for generating adversarial examples increases with increasing n_{aux}. It is therefore impractical to rely only on the auxiliary gradients to improve the transferability.

5 Conclusion

In this paper, we propose to improve the transferability of adversarial examples through the use of a virtual step and auxiliary gradients, which can be easily integrated into existing gradient-based attacks. Extensive experiments on ImageNet show that our method significantly outperforms the baselines. The ablation study further verifies that both the virtual step and auxiliary gradients are necessary to achieving the improved results. In future, we will further explore the potential use of auxiliary gradients in generating adversarial examples.

References

[Akhtar and Mian, 2018] Naveed Akhtar and Ajmal Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access*, 2018.

[Akhtar *et al.*, 2021] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey. *IEEE Access*, 2021.

[Biggio *et al.*, 2013] 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. In *Proc. of ECML*, 2013.

[Brendel *et al.*, 2018] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *Proc. of ICLR*, 2018.

[Chen *et al.*, 2020] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In *Proc. of SP*, 2020.

[Chollet, 2017] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proc. of CVPR*, 2017.

[Dalvi *et al.*, 2004] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proc. of KDD*, 2004.

[Dong *et al.*, 2018] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. In *Proc. of CVPR*, 2018.

[Dong *et al.*, 2019] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. In *Proc. of CVPR*, 2019.

[Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proc. of ICLR*, 2015.

[Guo *et al.*, 2018] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering Adversarial Images using Input Transformations. In *Proc. of ICLR*, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity Mappings in Deep Residual Networks. In *Proc. of ECCV*, 2016.

[Kurakin *et al.*, 2017] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Proc. of ICLR*, 2017.

[Li *et al.*, 2020] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning Transferable Adversarial Examples via Ghost Networks. In *Proc. of AAAI*, 2020.

[Liao *et al.*, 2018] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser. In *Proc. of CVPR*, 2018.

[Liu *et al.*, 2019] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *Proc. of ICCV*, 2019.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. of ICLR*, 2018.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. of ICLR*, 2014.

[Szegedy *et al.*, 2016] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proc. of CVPR*, 2016.

[Szegedy *et al.*, 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proc. of AAAI*, 2017.

[Tramer *et al.*, 2018] Florian Tramer, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. In *Proc. of ICLR*, 2018.

[Xie *et al.*, 2018] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating Adversarial Effects Through Randomization. In *Proc. of ICLR*, 2018.

[Xie *et al.*, 2019a] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature Denoising for Improving Adversarial Robustness. In *Proc. of CVPR*, 2019.

[Xie *et al.*, 2019b] 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 *Proc. of CVPR*, 2019.