Enhancing the Transferability of Adversarial Attacks through Variance Tuning

Xiaosen Wang   Kun He *
School of Computer Science and Technology, Huazhong University of Science and Technology
{xiaosen, brooklet60}@hust.edu.cn

Abstract

Deep neural networks are vulnerable to adversarial examples that mislead the models with imperceptible perturbations. Though adversarial attacks have achieved incredible success rates in the white-box setting, most existing adversaries often exhibit weak transferability in the black-box setting, especially under the scenario of attacking models with defense mechanisms. In this work, we propose a new method called variance tuning to enhance the class of iterative gradient based attack methods and improve their attack transferability. Specifically, at each iteration for the gradient calculation, instead of directly using the current gradient for the momentum accumulation, we further consider the gradient variance of the previous iteration to tune the current gradient so as to stabilize the update direction and escape from poor local optima. Empirical results on ImageNet dataset demonstrate that our method could significantly improve the transferability of gradient-based adversarial attacks. Besides, our method could be used to attack ensemble models or be integrated with various input transformations. Incorporating variance tuning with input transformations on iterative gradient-based attacks in the multi-model setting, the integrated method could achieve an average success rate of 90.1% against nine advanced defense methods, improving the current best attack performance of 85.1% significantly.

1. Introduction

Deep Neural Networks (DNNs) are known to be vulnerable to adversarial examples [31, 8], which are indistinguishable from legitimate ones by adding small perturbations, but lead to incorrect model prediction. In recent years, it has garnered an increasing interest to craft adversarial examples [22, 3, 14, 1, 16], because it not only can identify the model vulnerability [3, 1], but also can help improve the model robustness [8, 21, 32, 36]. Moreover, adversarial examples also exhibit good transferability across the models [25, 19], i.e. the adversaries crafted for one model can still fool other models, which enables black-box attacks in the real-world applications without any knowledge of the target model.

In the white-box setting that the attack method can access the architecture and parameters of the target model, existing adversarial attacks have exhibited great efficacy [14, 3, 1] but with low transferability, especially for models equipped with defense mechanisms [32, 33, 17]. To address this issue, recent works focus on improving the transferability of adversarial examples by advanced gradient calculation (e.g., Momentum, Nesterov’s accelerated gradient) [6, 18], attacking multiple models [19], or adopting various input transformations (e.g., randomly resizing and padding, translation, scale etc.) [34, 7, 18]. However, there still exists a big gap between white-box attacks and transfer-based black-box attacks with regard to the attack success rate.

In this work, we propose a novel variance tuning iterative gradient-based method to enhance the transferability of the generated adversarial examples. Different from existing

![Figure 1: Adversarial examples crafted by Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [6], Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [18] and the proposed Variance tuning MI-FGSM (VMI-FGSM) and Variance tuning NI-FGSM (VNI-FGSM) on the Inception-v3 model [30] with the maximum perturbation of $\epsilon = 16$.](image-url)
gradient-based methods that perturb the input in the gradient direction of the loss function, or momentum iterative gradient-based methods that accumulate a velocity vector in the gradient direction, at each iteration our method tunes the current gradient with the variance of gradients in the neighborhood of the previous data point. The key idea is to reduce the variance of the gradient at each iteration so as to stabilize the update direction and escape from poor local optima during the search process. Empirical results on the standard ImageNet dataset demonstrate that, compared with state-of-the-art momentum-based adversarial attacks [8, 14], the proposed method could achieve significantly higher attack success rates for black-box models, meanwhile maintain similar success rates for white-box models. For instance, the proposed method improves the success rates of the momentum based attack [6] for more than 20% using adversarial examples generated on the Inception-v3 model [30]. The adversarial examples crafted by various attacks are illustrated in Figure 1.

To further demonstrate the effectiveness of our method, we combine variance tuning with several gradient-based attacks for ensemble models [19] and integrate these attacks with various input transformations [34, 7, 18]. Extensive experiments show that our integrated method could remarkably improve the attack transferability. In addition, we compare our attack method with the state-of-the-art attack methods [17, 33, 35, 9, 20, 12, 5, 23]. Our integrated method yields an average attack success rate of 67.0%, which outperforms the baselines by a large margin of 17.5% in the single model setting, and an average success rate of 90.1%, which outperforms the baselines by a clear margin of 6.6% in the multi-model setting.

2. Related Work

Let $x$ be a benign image, and $y$ the corresponding true label. Let $f(x; \theta)$ be the classifier with parameters $\theta$ that outputs the prediction result. Let $J(x, y; \theta)$ denote the loss function of classifier $f$ (e.g. the cross-entropy loss). We define the adversarial attack as a method of finding an example $x^{adv}$ that satisfies $\|x - x^{adv}\|_p < \epsilon$ but misleads the model prediction, $f(x; \theta) \neq f(x^{adv}; \theta)$. Here $\| \cdot \|_p$ denotes the $p$-norm distance and we focus on $p = \infty$ to align with previous works.

2.1. Adversarial Attacks

Numerous adversarial attack methods have been proposed in recent years, including gradient-based methods [8, 14, 21], optimization-based methods [31, 3], score-based methods [11, 16] and decision-based methods [2, 4]. In this work, we mainly focus on the attack transferability and provide a brief overview on two branches of transfer-based attacks in this section.

2.1.1 Gradient-based Attacks

The first branch focuses on improving the transferability of gradient-based attacks by advanced gradient calculation.

**Fast Gradient Sign Method (FGSM).** FGSM [8] is the first gradient-based attack that crafts an adversarial example $x^{adv}$ by maximizing the loss function $J(x^{adv}, y; \theta)$ with a one-step update:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y; \theta)),$$

where $\nabla_x J(x, y; \theta)$ is the gradient of loss function w.r.t. $x$ and $\text{sign}(\cdot)$ denotes the sign function.

**Iterative Fast Gradient Sign Method (I-FGSM).** I-FGSM [14] extends FGSM to an iterative version with a small step size $\alpha$:

$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(\nabla_x J(x^{adv}_t, y; \theta)),$$

$$x^{adv}_0 = x.$$

**Momentum Iterative Fast Gradient Sign Method (MI-FGSM).** MI-FGSM [6] integrates the momentum into I-FGSM and achieves much higher transferability:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x^{adv}_t} J(x^{adv}_t, y; \theta)}{\|\nabla_{x^{adv}_t} J(x^{adv}_t, y; \theta)\|_1},$$

$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(g_{t+1}),$$

where $g_0 = 0$ and $\mu$ is the decay factor.

**Nesterov Iterative Fast Gradient Sign Method (NI-FGSM).** NI-FGSM [18] adopts Nesterov’s accelerated gradient [24], and substitutes $x^{adv}_t$ in Eq. (2) with $x^{adv}_t + \alpha \cdot \mu \cdot g_t$ to further improve the transferability of MI-FGSM.

2.1.2 Input Transformations

The second branch focuses on adopting various input transformations to enhance the attack transferability.

**Diverse Input Method (DIM).** DIM [34] applies random resizing and padding to the inputs with a fixed probability, and feeds the transformed images into the classifier for the gradient calculation to improve the transferability.

**Translation-Invariant Method (TIM).** TIM [7] adopts a set of images to calculate the gradient, which performs well especially for black-box models with defense mechanisms. To reduce the calculation on gradients, Dong et al. [7] shift the image within small magnitude and approximately calculate the gradients by convolving the gradient of untranslated images with a kernel matrix.

**Scale-Invariant Method (SIM).** SIM [18] introduces the scale-invariant property and calculates the gradient over a set of images scaled by factor $1/2^i$ on the input image to enhance the transferability of the generated adversarial examples, where $i$ is hyper-parameter.
Note that different input transformations, DIM, TIM and SIM, can be naturally integrated with gradient-based attack methods. Lin et al. [18] have shown that the combination of these methods, denoted as Composite Transformation Method (CTM), is the current strongest transfer-based black-box attack method. In this work, the proposed variance tuning based method aims to improve the transferability of the gradient-based attacks (e.g. MI-FGSM, NI-FGSM), and can be combined with various input transformations to further improve the attack transferability.

2.2. Adversarial Defenses

To mitigate the threat of adversarial examples, various adversarial defense methods have been proposed. One promising defense method is called adversarial training [8, 15, 21] that injects the adversarial examples into the training data to improve the model robustness. Tramèr et al. [32] propose ensemble adversarial training by augmenting the training data with perturbations transferred from several models, which can further improve the robustness against transfer-based black-box attacks. However, such adversarial training methods, as one of the most powerful and extensively investigated defense methods, often lead to high computation cost and are difficult to be scaled to large-scale datasets and complex neural networks [15].

Guo et al. [9] utilize a set of image transformations (e.g. JPEG compression, Total Variance Minimization, etc.) on the inputs to eliminate adversarial perturbations before feeding the images to the models. Xie et al. [33] adopt random resizing and padding (R&P) on the inputs to mitigate the adversarial effect. Liao et al. [17] propose to train a high-level representation denoiser (HGD) to purify the input images. Xu et al. [35] propose two feature squeezing methods: bit reduction (Bit-Red) and spatial smoothing to detect adversarial examples. Feature distillation (FD) [20] is a JPEG-based defensive compression framework against adversarial examples. ComDefend [12] is an end-to-end image compression model to defend adversarial examples. Cohen et al. [5] adopt randomized smoothing (RS) to train a certifiably robust ImageNet classifier. Naseer et al. [23] design a neural representation purifier (NRP) model that learns to purify the adversarially perturbed images based on the automatically derived supervision.

3. Methodology

For the methodology section, we first introduce our motivation, then provide a detailed description of the proposed method. We also formulize the relationship between existing transfer-based attacks and the proposed method.

3.1. Motivation

Given a target classifier $f$ with parameters $\theta$ and a benign image $x \in X$ where $x$ is in $d$ dimensions and $X$ denotes all the legitimate images, the adversarial attack aims to find an adversarial example $x_{adv} \in X$ that satisfies:

$$f(x; \theta) \neq f(x_{adv}; \theta) \quad \text{s.t.} \quad \|x - x_{adv}\| < \epsilon.$$  (4)

For white-box attacks, we can regard the attack as an optimization problem that searches an example in the neighborhood of $x$ so as to maximize the loss function $J$ of the target classifier $f$:

$$x_{adv} = \arg\max_{x' \in x < \epsilon} J(x', y; \theta).$$  (5)

Lin et al. [18] analogize the adversarial example generation process to the neural model training process where the input $x$ can be viewed as parameters to be trained and the target model can be treated as the training set. From this perspective, the transferability of adversarial examples is similar to the generalization of the normally trained models. Therefore, current works mainly focus on better optimization algorithms (e.g. MI-FGSM, NI-FGSM) [14, 6, 18] or data augmentation (e.g. ensemble attack on multiple models or input transformations) [19, 34, 7, 18] to improve the attack transferability.

In this work, we treat the iterative gradient-based adversarial attack as a stochastic gradient decent (SGD) optimization process (Note that we just have one target model in the optimization process, so SGD is equivalent to gradient descent). As presented in previous works [26, 28, 13], SGD introduces a large variance due to the randomness, leading to a slow convergence. To address this issue, various variance reduction methods have been proposed to accelerate the convergence of SGD, such as SAG (stochastic average gradient) [26], SDCA (stochastic dual coordinate ascent) [28], and SVRG (stochastic variance reduced gradient) [13], which incorporate the information from the training set to reduce the variance.

Based on the above analysis, we attempt to improve the transferability of adversarial examples with the gradient variance tuning strategy. The major difference between our method and SGD with variance reduction methods (SGDVRMs) [26, 28, 13] is three-fold. First, we aim to generate highly transferable adversarial examples, which is equivalent to improving the generalization of the training models, while SGDVRMs aim to accelerate the convergence. Second, we consider the gradient variance of the examples sampled in the neighborhood of input $x$, which is equivalent to the one in the parameter space for training the neural models but SGDVRMs adopt variance in the training set. Third, our variance tuning strategy is more generalized and can be used to improve the performance of both MI-FGSM and NI-FGSM.

3.2. Variance Tuning Gradient-based Attacks

Typical gradient-based iterative attacks (e.g. I-FGSM) greedily search an adversarial example in the direction of
Algorithm 1 VMI-FGSM

**Input:** A classifier $f$ with parameters $\theta$, loss function $J$.
**Input:** A raw example $x$ with ground-truth label $y$.
**Input:** The magnitude of perturbation $\epsilon$; number of iteration $T$ and decay factor $\mu$.
**Input:** The factor $\beta$ for the upper bound of neighborhood and number of example $N$ for variance tuning.

**Output:** An adversarial example $x^{adv}$.

1: $\alpha = \epsilon / T$
2: $g_0 = 0; v_0 = 0; x_0^{adv} = x$
3: for $t = 0 \rightarrow T - 1$ do
4: Calculate the gradient $\hat{g}_{t+1} = \nabla_{x^{adv}} J(x^{adv}_t, y; \theta)$
5: Update $g_{t+1}$ by variance tuning based momentum
   
   $$g_{t+1} = \mu \cdot g_{t} + \frac{\hat{g}_{t+1} + v_{t}}{\|g_{t+1} + v_{t}\|_1} \quad (6)$$
6: Update $v_{t+1} = V(x^{adv}_t)$ by Eq. (8)
7: Update $x^{adv}_t$ by applying the sign of gradient
   
   $$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(g_{t+1}) \quad (7)$$
8: end for
9: $x^{adv} = x^{adv}_T$
10: return $x^{adv}$


the sign of the gradient at each iteration, as shown in Eq. (1), which may easily fall into poor local optima and “overfit” the model [6]. MI-FGSM [6] integrates momentum into I-FGSM for the purpose of stabilizing the update directions and escaping from poor local optima to improve the attack transferability. NI-FGSM [18] further adopts Netserov’s accelerated gradient [24] into I-FGSM to improve the transferability by leveraging its looking ahead property.

We observe that the above methods only consider the data points along the optimization path, denoted as $x_0^{adv} = x, x_1^{adv}, \ldots, x_{t-1}^{adv}, x_T^{adv} = x^{adv}$. In order to avoid overfitting and further improve the transferability of the adversarial attacks, we adapt the gradient information in the neighborhood of the previous data point to tune the gradient of current data point at each iteration.

Specifically, for any input $x \in X$, we define the gradient variance as follows.

**Definition 1 Gradient Variance.** Given a classifier $f$ with parameters $\theta$ and loss function $J(x, y; \theta)$, an arbitrary image $x \in X$ and an upper bound $\epsilon'$ for the neighborhood, the gradient variance can be defined as:

$$V^\beta_{\epsilon'}(x) = \mathbb{E}_{\|x' - x\|_p < \epsilon'}[\nabla_x J(x', y; \theta)] - \nabla_x J(x, y; \theta).$$

We simply use $V(x)$ to denote $V^\beta_{\epsilon'}(x)$ without ambiguity in the following and we set $\epsilon' = \beta \cdot \epsilon$ where $\beta$ is a hyper-parameter and $\epsilon$ is the upper bound of the perturbation magnitude. In practice, however, due to the continuity of the input space, we cannot calculate $\mathbb{E}_{\|x' - x\|_p < \epsilon'}[\nabla_x J(x', y; \theta)]$ directly. Therefore, we approximate its value by sampling $N$ examples in the neighborhood of $x$ to calculate $V(x)$:

$$V(x) = \frac{1}{N} \sum_{i=1}^{N} \nabla_x J(x^{i}, y; \theta) - \nabla_x J(x, y; \theta). \quad (8)$$

Here $x^i = x + r_i, r_i \sim U[\beta \cdot \epsilon d, (\beta \cdot \epsilon)^d], \text{and } U[a^d, b^d]$ stands for the uniform distribution in $d$ dimensions.

After obtaining the gradient variance, we can tune the gradient of $x^{adv}_t$ at $t$-th iteration with the gradient variance $V(x^{adv}_t)$ at the $(t - 1)$-th iteration to stabilize the update direction. The algorithm of variance tuning MI-FGSM, denoted as VMI-FGSM, is summarized in Algorithm 1. Note that our method is generally applicable to any gradient-based attack method and we can easily extend VMI-FGSM to variance tuning NI-FGSM (VNI-FGSM), and integrate these methods with DIM, TIM and SIM as in [18].

3.3. Relationships among Various Attacks

This work focuses on the transferability of adversarial attacks derived from FGSM. Here we summarize the relationships among these attacks, as illustrated in Figure 2. If the factor $\beta$ for the upper bound of neighborhood is set to 0, VMI-FGSM and VNI-FGSM degrade to MI-FGSM and NI-FGSM, respectively. If the decay factor $\mu = 0$, both MI-FGSM and NI-FGSM degrade to I-FGSM. If the iteration number $T = 1$, I-FGSM degrades to FGSM. Moreover, we can integrate the above attacks with various input transformations, i.e. DIM, TIM, SIM, to obtain more powerful adversarial attacks and these derived methods follow the same discipline.
### 4. Experiments

To validate the efficacy of the proposed variance tuning based method, we conduct extensive experiments on standard ImageNet dataset [27]. In this section, we first specify the experimental setup, then compare our method with competitive baselines under various experimental settings and quantify the attack efficacy on nine advanced defense models. Experimental results demonstrate that our method can significantly improve the transferability of the baselines in various settings. In the end, we provide further investigations on hyper-parameters $N$ and $\beta$ used for variance tuning.

#### 4.1. Experimental Setup

**Dataset.** We randomly pick 1,000 clean images from the ILSVRC 2012 validation set [27], which are almost correctly classified by all the testing models as in [6, 18].

**Models.** We consider four normally trained networks, including Inception-v3 (Inc-v3) [30], Inception-v4 (Inc-v4), Inception-Resnet-v2 (IncRes-v2) [29] and Resnet-v2-101 (Res-101) [10] and three adversarially trained models, namely Inc-v3\textsubscript{ens}, Inc-v3\textsubscript{ens3} and IncRes-v2\textsubscript{ens} [32]. Besides, we include nine advanced defense models that are robust against black-box adversarial attacks on the ImageNet dataset, i.e. HGD [17], R&P [33], NIPS-r3 \(^1\), Bit-Red [35], JPEG [9], FD [20], ComDefend [12], RS [5] and NRP [23].

**Baselines.** We take two momentum based iterative attacks, i.e. MI-FGSM [6] and NI-FGSM [18], as our baselines. Besides, we integrate the proposed method with various input transformations, i.e. DIM [34], TIM [7], SIM and CTM [18], denoted as VM(N)I-DI-FGSM, VM(N)I-TI-FGSM, VM(N)I-SI-FGSM and VM(N)I-CT-FGSM respectively, to further validate the efficacy of our method.

**Hyper-parameters.** We follow the attack setting in [6] with the maximum perturbation of $\epsilon = 16$, number of iteration $T = 10$ and step size $\alpha = 1.6$. For MI-FGSM and NI-FGSM, we set the decay factor $\mu = 1.0$. For DIM, the transformation probability is set to 0.5. For TIM, we adopt the Gaussian kernel with kernel size $7 \times 7$. For SIM, the number of scale copies is 5 (i.e. $i = 0, 1, 2, 3, 4$). For the proposed method, we set $N = 20$ and $\beta = 1.5$.

#### 4.2. Attack a Single Model

We first perform four adversarial attacks, namely MI-FGSM, NI-FGSM, our variance tuning based methods VMI-FGSM and VNI-FGSM, on a single neural network. We craft adversarial examples on normally trained networks and test them on all the seven neural networks we consider. The success rates, which are the misclassification rates of the corresponding models on adversarial examples, are shown in Table 1. The models we attack are on rows and the seven models we test are on columns.

<table>
<thead>
<tr>
<th>Model</th>
<th>Attack</th>
<th>Inc-v3</th>
<th>Inc-v4</th>
<th>IncRes-v2</th>
<th>Res-101</th>
<th>Inc-v3\textsubscript{ens3}</th>
<th>Inc-v3\textsubscript{ens4}</th>
<th>IncRes-v2\textsubscript{ens}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc-v3</td>
<td>MI-FGSM</td>
<td>100.0*</td>
<td>43.6</td>
<td>42.4</td>
<td>35.7</td>
<td>13.1</td>
<td>12.8</td>
<td>6.2</td>
</tr>
<tr>
<td></td>
<td>VMI-FGSM</td>
<td>100.0*</td>
<td>71.7</td>
<td>68.1</td>
<td>60.2</td>
<td>32.8</td>
<td>31.2</td>
<td>17.5</td>
</tr>
<tr>
<td></td>
<td>NI-FGSM</td>
<td>100.0*</td>
<td>51.7</td>
<td>50.3</td>
<td>41.3</td>
<td>13.5</td>
<td>13.2</td>
<td>6.0</td>
</tr>
<tr>
<td></td>
<td>VNI-FGSM</td>
<td>100.0*</td>
<td>46.0</td>
<td>41.0</td>
<td>56.0</td>
<td>35.0</td>
<td>32.8</td>
<td>18.8</td>
</tr>
<tr>
<td>Inc-v4</td>
<td>MI-FGSM</td>
<td>56.3</td>
<td>99.7*</td>
<td>46.6</td>
<td>41.0</td>
<td>16.3</td>
<td>14.8</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td>VMI-FGSM</td>
<td>77.9</td>
<td>99.8*</td>
<td>71.2</td>
<td>62.2</td>
<td>38.2</td>
<td>38.7</td>
<td>23.2</td>
</tr>
<tr>
<td></td>
<td>NI-FGSM</td>
<td>63.1</td>
<td>100.0*</td>
<td>51.8</td>
<td>45.8</td>
<td>15.4</td>
<td>13.6</td>
<td>6.7</td>
</tr>
<tr>
<td></td>
<td>VNI-FGSM</td>
<td>83.4</td>
<td>99.9*</td>
<td>76.1</td>
<td>66.9</td>
<td>40.0</td>
<td>37.7</td>
<td>24.5</td>
</tr>
<tr>
<td>IncRes-v2</td>
<td>MI-FGSM</td>
<td>60.7</td>
<td>51.1</td>
<td>97.9*</td>
<td>46.8</td>
<td>21.2</td>
<td>16.0</td>
<td>11.9</td>
</tr>
<tr>
<td></td>
<td>VMI-FGSM</td>
<td>77.9</td>
<td>72.1</td>
<td>97.9*</td>
<td>67.7</td>
<td>46.4</td>
<td>40.8</td>
<td>34.4</td>
</tr>
<tr>
<td></td>
<td>NI-FGSM</td>
<td>62.8</td>
<td>54.7</td>
<td>99.1*</td>
<td>46.0</td>
<td>20.0</td>
<td>15.1</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>VNI-FGSM</td>
<td>80.8</td>
<td>76.9</td>
<td>98.5*</td>
<td>69.8</td>
<td>47.9</td>
<td>40.3</td>
<td>34.2</td>
</tr>
<tr>
<td>Res-101</td>
<td>MI-FGSM</td>
<td>58.1</td>
<td>51.6</td>
<td>50.5</td>
<td>99.3*</td>
<td>23.9</td>
<td>21.5</td>
<td>12.7</td>
</tr>
<tr>
<td></td>
<td>VMI-FGSM</td>
<td>75.1</td>
<td>68.9</td>
<td>70.5</td>
<td>99.2*</td>
<td>45.2</td>
<td>41.4</td>
<td>30.1</td>
</tr>
<tr>
<td></td>
<td>NI-FGSM</td>
<td>65.6</td>
<td>58.3</td>
<td>57.0</td>
<td>99.4*</td>
<td>24.5</td>
<td>21.4</td>
<td>11.7</td>
</tr>
<tr>
<td></td>
<td>VNI-FGSM</td>
<td>79.8</td>
<td>74.6</td>
<td>73.2</td>
<td>99.7*</td>
<td>46.1</td>
<td>42.5</td>
<td>32.1</td>
</tr>
</tbody>
</table>

Table 1: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks. The adversarial examples are crafted on Inc-v3, Inc-v4, IncRes-v2, and Res-101 respectively. * indicates the white-box model.

---

\(^1\)https://github.com/anlthms/nips-2017/tree/master/mmd
Figure 3: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks enhanced by DIM, TIM and SIM respectively. The adversarial examples are generated on Inc-v3 model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Attack</th>
<th>Inc-v3</th>
<th>Inc-v4</th>
<th>IncRes-v2</th>
<th>Res-101</th>
<th>Inc-v3_{ens3}</th>
<th>Inc-v3_{ens4}</th>
<th>IncRes-v2_{ens}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc-v3</td>
<td>MI-CT-FGSM</td>
<td>98.7*</td>
<td>85.4</td>
<td>80.6</td>
<td>76.0</td>
<td>64.1</td>
<td>62.1</td>
<td>45.2</td>
</tr>
<tr>
<td></td>
<td>VMI-CT-FGSM</td>
<td>99.3*</td>
<td>88.6</td>
<td>86.7</td>
<td>82.9</td>
<td>78.6</td>
<td>76.2</td>
<td>64.7</td>
</tr>
<tr>
<td></td>
<td>NI-CT-FGSM</td>
<td>98.9*</td>
<td>84.1</td>
<td>80.0</td>
<td>74.5</td>
<td>60.0</td>
<td>56.2</td>
<td>41.0</td>
</tr>
<tr>
<td></td>
<td>VNI-CT-FGSM</td>
<td>99.5*</td>
<td>91.2</td>
<td>89.0</td>
<td>85.3</td>
<td>78.6</td>
<td>76.7</td>
<td>65.3</td>
</tr>
<tr>
<td>Inc-v4</td>
<td>MI-CT-FGSM</td>
<td>87.2</td>
<td>98.6*</td>
<td>83.3</td>
<td>78.3</td>
<td>72.2</td>
<td>67.2</td>
<td>57.3</td>
</tr>
<tr>
<td></td>
<td>VMI-CT-FGSM</td>
<td>90.0</td>
<td>98.8*</td>
<td>86.6</td>
<td>81.9</td>
<td>78.3</td>
<td>76.6</td>
<td>68.3</td>
</tr>
<tr>
<td></td>
<td>NI-CT-FGSM</td>
<td>87.8</td>
<td>99.4*</td>
<td>82.5</td>
<td>75.9</td>
<td>65.8</td>
<td>62.6</td>
<td>51.3</td>
</tr>
<tr>
<td></td>
<td>VNI-CT-FGSM</td>
<td>92.1</td>
<td>99.2*</td>
<td>89.2</td>
<td>85.1</td>
<td>80.1</td>
<td>78.3</td>
<td>70.4</td>
</tr>
<tr>
<td>IncRes-v2</td>
<td>MI-CT-FGSM</td>
<td>87.9</td>
<td>85.7</td>
<td>97.1*</td>
<td>83.0</td>
<td>77.6</td>
<td>74.6</td>
<td>72.0</td>
</tr>
<tr>
<td></td>
<td>VMI-CT-FGSM</td>
<td>88.9</td>
<td>87.0</td>
<td>97.0*</td>
<td>85.0</td>
<td>83.4</td>
<td>80.5</td>
<td>79.4</td>
</tr>
<tr>
<td></td>
<td>NI-CT-FGSM</td>
<td>90.2</td>
<td>87.0</td>
<td>99.4*</td>
<td>83.2</td>
<td>75.0</td>
<td>68.9</td>
<td>65.1</td>
</tr>
<tr>
<td></td>
<td>VNI-CT-FGSM</td>
<td>92.9</td>
<td>90.6</td>
<td>99.0*</td>
<td>88.2</td>
<td>85.2</td>
<td>82.5</td>
<td>81.8</td>
</tr>
<tr>
<td>Res-101</td>
<td>MI-CT-FGSM</td>
<td>86.5</td>
<td>81.8</td>
<td>83.2</td>
<td>98.9*</td>
<td>77.0</td>
<td>72.3</td>
<td>61.9</td>
</tr>
<tr>
<td></td>
<td>VMI-CT-FGSM</td>
<td>86.9</td>
<td>84.2</td>
<td>86.4</td>
<td>98.6*</td>
<td>81.0</td>
<td>78.6</td>
<td>71.6</td>
</tr>
<tr>
<td></td>
<td>NI-CT-FGSM</td>
<td>86.1</td>
<td>82.2</td>
<td>83.3</td>
<td>98.5*</td>
<td>70.0</td>
<td>68.5</td>
<td>54.6</td>
</tr>
<tr>
<td></td>
<td>VNI-CT-FGSM</td>
<td>90.7</td>
<td>85.5</td>
<td>87.2</td>
<td>99.1*</td>
<td>82.6</td>
<td>79.7</td>
<td>73.3</td>
</tr>
</tbody>
</table>

Table 2: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks enhanced by CTM. * indicates the white-box model.

43.6% and 13.1%, respectively. We also illustrate several adversarial images generated on Inc-v3 model by various attack methods in Appendix A.

4.3. Attack with Input Transformations

Several input transformations, e.g. DIM, [7], TIM [34] and SIM [18], have been incorporated into the gradient-based adversarial attacks to improve the transferability. Here we integrate our methods into these input transformations and demonstrate the proposed variance tuning strategy could further enhance the transferability. We report the success rates of black-box attacks in Figure 3, where the adversarial examples are generated on Inc-v3 model. The results for adversarial examples generated on other three models are reported in Appendix B.

The results show that the success rates against black-box models are improved by a large margin with the variance tuning strategy regardless of the attack algorithms or the white-box models to be attacked. In general, the methods equipped with our variance tuning strategy consistently outperform the baseline attacks by 10%~30%.

Lin et al. [18] have shown that CTM, the combination of DIM, TIM and SIM, could help the gradient-based attacks achieve state-of-the-art transferability. We also combine CTM with our method to further improve the transferability. As depicted in Table 2, the success rates could be further improved remarkably on various models, especially against adversarially trained models.

4.4. Attack an Ensemble of Models

Liu et al. [19] have shown that attacking multiple models simultaneously could improve the transferability of the generated adversarial examples. In this subsection, we adopt the ensemble attack method in [6], which fuses the logit out-
puts of different models, to demonstrate that our variance tuning method could further improve the transferability of adversarial attacks in the multi-model setting. Specifically, we attack the ensemble of four normally trained models, \textit{i.e.} Inc-v3, Inc-v4, IncRes-v2 and Res-101 by averaging the logit outputs of the models using various attacks with or without input transformations.

As shown in Table 3, our methods (VMI-FGSM, VNI-FGSM) could significantly enhance the transferability of the baselines more than 25\% for MI-FGSM and 30\% for NI-FGSM on the adversarially trained models. Even though the attacks with CTM could achieve good enough transferability, our methods (VMI-CT-FGSM, VNI-CT-FGSM) could further improve the transferability significantly. In particular, VNI-CT-FGSM achieves the success rates of 92.3\% \sim 95.5\% against three adversarially trained models, indicating the vulnerability of current defense mechanisms. Besides, in the white-box setting, our methods could still maintain similar success rates as the baselines.

### 4.5. Attack Advanced Defense Models

To further validate the effectiveness of the proposed method in practice, except for the normally trained models and adversarially trained models, we also evaluate our methods on nine extra models with advanced defenses, including the top-3 defense methods in the NIPS competition (HGD (rank-1) [17], R&P (rank-2) [33] and NIPS-r3 (rank-3)), and six recently proposed defense methods, namely Bit-Red [35], JPEG [9], FD [20], ComDefend [12], RS [5] and NRP [23].

Since MI-CT-FGSM and NI-CT-FGSM exhibit the best transferability among the existing attack methods [18], we compare our methods with the two attacks with adversarial examples crafted on Inc-v3 model and the ensemble models as in Section 4.4, respectively. The results are shown in Table 4. In the single model setting, the proposed methods achieve an average success rate of 66.5\% for VMI-CT-FGSM and 67.0\% for VNI-CT-FGSM, which outperforms the baseline attacks for more than 13.5\% and 17.5\% respectively. In the multi-model setting, our methods achieve an average success rate of 88.9\% for VMI-CT-FGSM and 90.1\% for VNI-CT-FGSM, which outperforms the baseline attacks for more than 3.8\% and 6.6\% respectively. Note that in the multi-model setting, our methods achieve the success rates of more than 77\% against the defense model with Randomize Smoothing (RS) [5] that provides certified defense. And our methods achieve the success rates of more than 83\% against the defense model with Neural Representation Purifier (NRP) which is a recently proposed powerful defense method and exhibits great robustness against DIM and DI-TIM [23], raising a new security issue for the development of more robust deep learning models.

### Table 3: The success rates (%) on seven models in the multi-model setting by various gradient-based iterative attacks. * indicates the white-box models.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Inc-v3</th>
<th>Inc-v4</th>
<th>IncRes-v2</th>
<th>Res-101</th>
<th>Inc-v3_{ens}</th>
<th>Inc-v4_{ens}</th>
<th>IncRes-v2_{ens}</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI-FGSM</td>
<td>99.9*</td>
<td>98.2*</td>
<td>95.3*</td>
<td>99.9*</td>
<td>39.4</td>
<td>35.3</td>
<td>24.2</td>
</tr>
<tr>
<td>VMI-FGSM</td>
<td>99.7*</td>
<td>98.5*</td>
<td>96.0*</td>
<td>99.9*</td>
<td>67.6</td>
<td>62.9</td>
<td>50.7</td>
</tr>
<tr>
<td>NI-FGSM</td>
<td>99.8*</td>
<td>98.9*</td>
<td>98.9*</td>
<td>99.8*</td>
<td>41.0</td>
<td>33.5</td>
<td>23.1</td>
</tr>
<tr>
<td>VNI-FGSM</td>
<td>99.9*</td>
<td>99.6*</td>
<td>98.6*</td>
<td>99.9*</td>
<td>71.3</td>
<td>66.0</td>
<td>52.9</td>
</tr>
<tr>
<td>MI-CT-FGSM</td>
<td>99.6*</td>
<td>99.1*</td>
<td>97.4*</td>
<td>99.7*</td>
<td>91.3</td>
<td>89.6</td>
<td>86.8</td>
</tr>
<tr>
<td>VMI-CT-FGSM</td>
<td>99.7*</td>
<td>99.2*</td>
<td>98.4*</td>
<td>99.9*</td>
<td>93.6</td>
<td>92.4</td>
<td>91.0</td>
</tr>
<tr>
<td>NI-CT-FGSM</td>
<td>100.0*</td>
<td>99.9*</td>
<td>99.9*</td>
<td>100.0*</td>
<td>95.5</td>
<td>94.5</td>
<td>92.3</td>
</tr>
<tr>
<td>VNI-CT-FGSM</td>
<td>100.0*</td>
<td>99.9*</td>
<td>99.6*</td>
<td>100.0*</td>
<td>95.5</td>
<td>94.5</td>
<td>92.3</td>
</tr>
</tbody>
</table>

### Table 4: The success rates (%) on nine models with advanced defense mechanism by various gradient-based iterative attacks enhanced by CTM. The adversarial examples are generated on Inc-v3 model and the ensemble of models respectively.

<table>
<thead>
<tr>
<th>Model</th>
<th>Attack</th>
<th>HGD</th>
<th>R&amp;P</th>
<th>NIPS-r3</th>
<th>Bit-Red</th>
<th>JPEG</th>
<th>FD</th>
<th>ComDefend</th>
<th>RS</th>
<th>NRP</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inc-v3</td>
<td>MI-CT-FGSM</td>
<td>56.6</td>
<td>44.9</td>
<td>52.5</td>
<td>36.2</td>
<td>77.3</td>
<td>60.0</td>
<td>80.1</td>
<td>40.3</td>
<td>29.3</td>
<td>53.0</td>
</tr>
<tr>
<td>VMI-CT-FGSM</td>
<td>73.1</td>
<td>65.1</td>
<td>70.3</td>
<td>49.5</td>
<td>85.4</td>
<td>72.4</td>
<td>86.0</td>
<td>51.9</td>
<td>45.2</td>
<td>66.5</td>
<td></td>
</tr>
<tr>
<td>NI-CT-FGSM</td>
<td>50.4</td>
<td>39.4</td>
<td>47.4</td>
<td>34.3</td>
<td>76.0</td>
<td>58.6</td>
<td>77.7</td>
<td>36.9</td>
<td>24.8</td>
<td>49.5</td>
<td></td>
</tr>
<tr>
<td>VNI-CT-FGSM</td>
<td>73.4</td>
<td>64.5</td>
<td>70.6</td>
<td>51.2</td>
<td>86.8</td>
<td>73.5</td>
<td>87.3</td>
<td>52.1</td>
<td>43.9</td>
<td>67.0</td>
<td></td>
</tr>
<tr>
<td>Ens</td>
<td>MI-CT-FGSM</td>
<td>91.0</td>
<td>87.7</td>
<td>89.0</td>
<td>75.9</td>
<td>94.2</td>
<td>88.8</td>
<td>95.1</td>
<td>68.1</td>
<td>76.1</td>
<td>85.1</td>
</tr>
<tr>
<td>VMI-CT-FGSM</td>
<td>92.9</td>
<td>91.0</td>
<td>92.3</td>
<td>80.9</td>
<td>95.4</td>
<td>91.0</td>
<td>96.2</td>
<td>77.0</td>
<td>83.2</td>
<td>88.9</td>
<td></td>
</tr>
<tr>
<td>NI-CT-FGSM</td>
<td>91.3</td>
<td>85.6</td>
<td>89.0</td>
<td>72.3</td>
<td>95.9</td>
<td>89.5</td>
<td>95.4</td>
<td>63.2</td>
<td>69.5</td>
<td>83.5</td>
<td></td>
</tr>
<tr>
<td>VNI-CT-FGSM</td>
<td>94.7</td>
<td>92.4</td>
<td>93.4</td>
<td>82.3</td>
<td>97.1</td>
<td>92.6</td>
<td>97.4</td>
<td>77.4</td>
<td>84.0</td>
<td>90.1</td>
<td></td>
</tr>
</tbody>
</table>
4.6. Ablation Study on Hyper-parameters

We conduct a series of ablation experiments to study the impact of two parameters, $N$ and $\beta$, used in our variance tuning strategy. All the adversarial examples are generated on Inc-v3 model without input transformations, which achieve success rates of 100% for different values of the two parameters in the white-box setting.

On the upper bound of neighborhood. In Figure 4, we study the influence of the neighborhood size, determined by parameter $\beta$, on the success rates in the black-box setting where $N$ is fixed to 20. When $\beta = 0$, VMI-FGSM and VNI-FGSM degrade to MI-FGSM and NI-FGSM, respectively, and achieve the lowest transferability. When $\beta = 1/5$, although the neighborhood is very small, our variance tuning strategy could improve the transferability remarkably. As we increase $\beta$, the transferability increases and achieves the peak for normally trained models when $\beta = 3/2$ but it is still increasing against the adversarially trained models. In order to achieve the trade-off for the transferability on normally trained models and adversarial trained models, we choose $\beta = 3/2$ in our experiments.

On the number of sampled examples in the neighborhood. We then study the influence of the number of sampled examples $N$ on the success rates in the black-box setting ($\beta$ is fixed to 3/2). As depicted in Figure 5, when $N = 0$, VMI-FGSM and VNI-FGSM also degrade to MI-FGSM and NI-FGSM, respectively, and achieve the lowest transferability. When $N = 20$, the transferability of adversarial examples is improved significantly and the transferability increases slowly when we continue increase $N$. Note that a bigger value of $N$ means a higher computation cost. To balance the transferability and computation cost, we set $N = 20$ in our experiments.

In summary, $\beta$ plays a key role in improving the transferability while $N$ exhibits little impact when $N > 20$. In our experiments, we set $\beta = 3/2$ and $N = 20$.

5. Conclusion

In this work, we propose a variance tuning method to enhance the transferability of the iterative gradient-based adversarial attacks. Specifically, for any input of a neural classifier, we define its gradient variance as the difference between the mean gradient of the neighborhood and its own gradient. Then we adopt the gradient variance of the data point at the previous iteration along the optimization path to tune the current gradient. Extensive experiments demonstrate that the variance tuning method could significantly improve the transferability of the existing competitive attacks, MI-FGSM and NI-FGSM, while maintains similar success rates in the white-box setting.

The variance tuning method could be generally applicable to any gradient-based attacks. We apply our method to attack ensemble models and integrate it with advanced input transformation methods (e.g., DIM, TIM, SIM) to further enhance the transferability. Empirical results on nine advanced defense models show that our integrated method could reach an average success rate of at least 90.1%, outperforming the state-of-the-art attacks for 6.6% on average, indicating the insufficiency of current defense techniques.

Considering the remarkable improvement for the attack transferability, our method might not be limited to adversarial attacks and could be possibly served as an optimization method to improve the generalization of normally trained models. We will explore this in our future work.
References


[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large...


Appendix

In the appendix, we first visualize more adversarial examples generated by various attacks. Then we provide more results for our methods integrated with DIM, TIM or SIM on the other three normally trained models, i.e. Inc-v4, IncRes-v2 and Res-101.

A. Visualizations on Adversarial Examples

We visualize eight randomly selected benign images and their corresponding adversarial examples crafted by various attacks in Figure 6. The adversarial examples are crafted on Inc-v3 model, using MI-FGSM, NI-FGSM, VMI-FGSM and VNI-FGSM, respectively. It can be observed that these crafted adversarial examples are human-imperceptible.

B. More Attacks with Input Transformations

Here we further provide the attack results of our methods with input transformations on the other three models, i.e. Inc-v4, IncRes-v2 and Res-101. The results are depicted in Figure 7 for DIM, Figure 8 for TIM and Figure 9 for SIM. Our methods can improve the transferability of these input transformations remarkably, especially against the adversarially trained models. The results are consistent with the results of adversarial examples crafted on Inc-v3 model.
Figure 6: Adversarial examples crafted by MI-FGSM [6], NI-FGSM [18], the proposed VMI-FGSM and VNI-FGSM on Inc-v3 model [30] with the maximum perturbation of $\epsilon = 16$. 
Figure 7: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks enhanced by DIM.

Figure 8: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks enhanced by TIM.

Figure 9: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks enhanced by SIM.