Making Adversarial Examples More Transferable and Indistinguishable

Junhua Zou[1], Yexin Duan[1], Boyu Li[2], Wu Zhang[1], Yu Pan[1], Zhisong Pan[1*],

[1]Command and Control Engineering College, Army Engineering University, Nanjing 210007, China
[2]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
278287847@qq.com, hotpzs@hotmail.com

Abstract

Fast gradient sign attack series are popular methods that are used to generate adversarial examples. However, most of the approaches based on fast gradient sign attack series cannot balance the indistinguishability and transferability due to the limitations of the basic sign structure. To address this problem, we propose a method, called Adam Iterative Fast Gradient Tanh Method (AI-FGTM), to generate indistinguishable adversarial examples with high transferability. Besides, smaller kernels and dynamic step size are also applied to generate adversarial examples for further increasing the attack success rates. Extensive experiments on an ImageNet-compatible dataset show that our method generates more indistinguishable adversarial examples and achieves higher attack success rates without extra running time and resource. Our best transfer-based attack NI-TI-DI-AITM can fool six classic defense models with an average success rate of 89.3% and three advanced defense models with an average success rate of 82.7%, which are higher than the state-of-the-art gradient-based attacks. Additionally, our method can also reduce nearly 20% mean perturbation. We expect that our method will serve as a new baseline for generating adversarial examples with better transferability and indistinguishability.

Introduction

Despite the great success on many tasks, deep neural networks (DNNs) have been shown that they are vulnerable to adversarial examples (Goodfellow, Shlens, and Szegedy 2015; Szegedy et al. 2014), i.e., the inputs with imperceptible perturbations can cause the incorrect results of DNNs. Moreover, a tougher problem terms transferability (Liu et al. 2017; Moosavi-Dezfooli et al. 2017) that the adversarial examples crafted by a known DNN can also fool other unknown DNNs. Consequently, adversarial examples present severe threats to the real-world applications (Athalye et al. 2018; Eykholt et al. 2018; Kurakin, Goodfellow, and Bengio 2017a) and have motivated extensive research on defense methods (Madry et al. 2018; Liao et al. 2018; Guo et al. 2018; Raghunathan, Steinhardt, and Liang 2018; Wong and Kolter 2018; Pang, Du, and Zhu 2018; Samangouei, Kabkab, and Chellappa 2018). Foolbox (Rauber et al. 2020) roughly

categorizes attack methods into three types: the gradient-based methods (Dong et al. 2018; Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2017b), the score-based methods (Narodytska and Kasiviswanathan 2017), and the decision-based methods (Brendel, Rauber, and Bethge 2018; Chen, Jordan, and Wainwright 2020). In this paper, we focus on the gradient-based methods. Although the adversarial examples crafted by using the gradient-based methods satisfy the L_p bound and continually achieve higher black-box success rates, these examples can be identified easily. In addition, the adversarial examples generated by the approaches based on the basic sign structure are limited. Taking TI-MI-FGSM (the combination of translation-invariant method (Dong et al. 2019) and momentum iterative fast gradient sign method (Dong et al. 2018)) as an example, the gradient processing steps, such as Gaussian blur, the gradient normalization, and the sign function, severely damage the gradient information. Additionally, the sign function also increases the perturbation size.

In this paper, we propose a method, called **Adam Iterative Fast Gradient Tanh Method (AI-FGTM)**, which improves the indistinguishability and transferability of adversarial examples. It is known that the fast gradient sign attack series iteratively process gradient information with transformation, normalization, and the sign function. To preserve the gradient information as much as possible, AI-FGTM modifies the major gradient processing steps. Still take TI-MI-FGSM as an example, to avoid the loss of gradient information and generate imperceptible perturbations, we replace the momentum algorithm and the sign function with Adam (Kingma and Ba 2015) and the tanh function, respectively. Then, we employ dynamic step size and smaller filters in Gaussian blur. The overview of AI-FGTM is shown in Fig. 1, and the detailed process will be given in **Methodology**. Furthermore, combining the existing attack methods with AI-FGTM can get much smaller perturbations and deliver state-of-the-art success rates. Fig. 2 shows the comparison of different examples, where the adversarial examples are crafted by three combined attacks, namely, DIM (Xie et al. 2019), TI-DIM (Dong et al. 2019) and TI-DI-AITM (the combination of TI-DIM and our method).

In summary, we make the following contributions:

1. Inspired by the limitations of the fast gradient sign series, we propose AI-FGTM in which the major gradient pro-
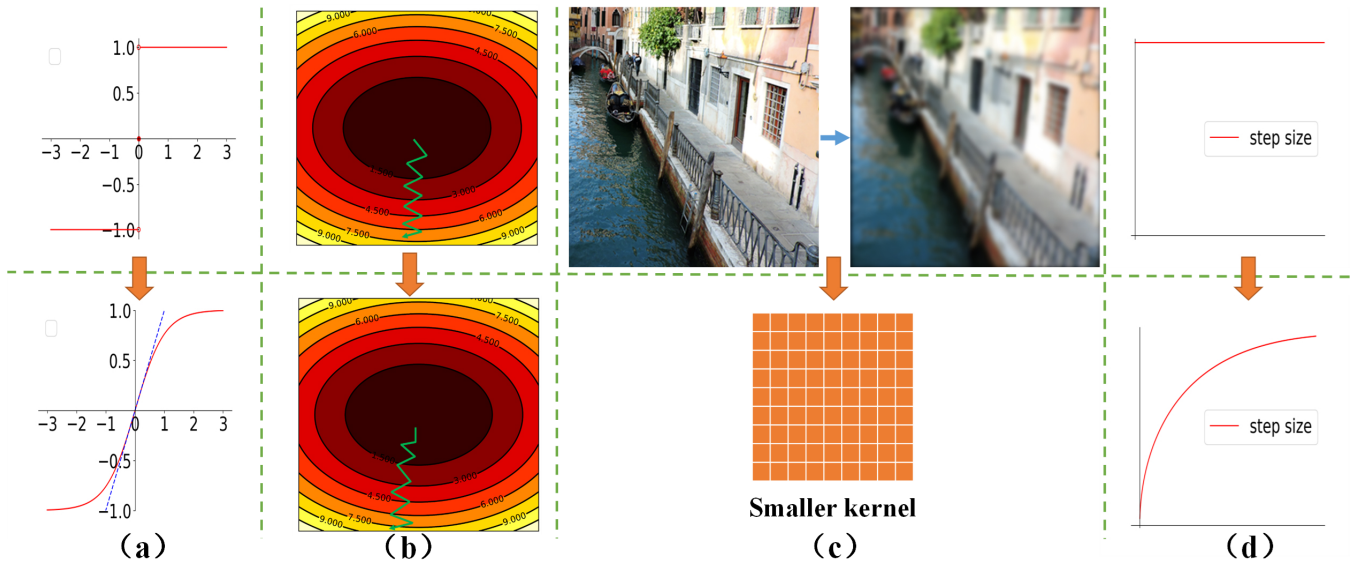
Figure 1: Overview of the our method. (a) We replace the sign function with the tanh function to generate smaller perturbations. (b) We use Adam instead of momentum method and gradient normalization to get larger losses in only ten iterations. (c) We use smaller kernels in Gaussian blur to avoid the loss of gradient information. (d) We gradually increase the step size.
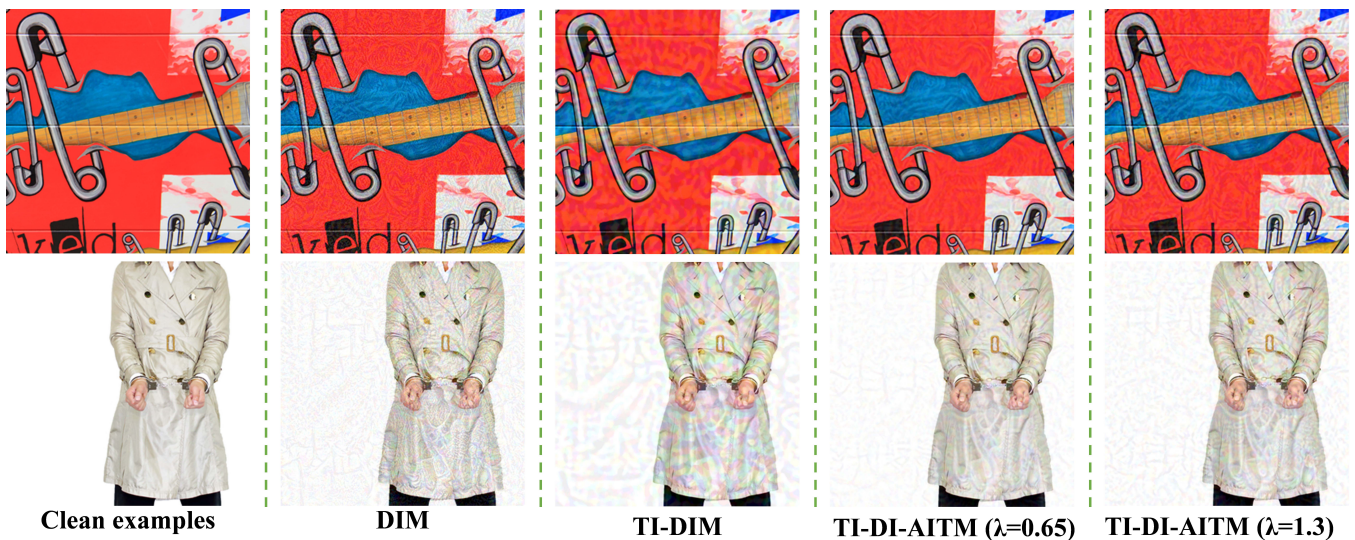

| Clean examples | DIM | TI-DIM | TI-DI-AITM (λ=0.65) | TI-DI-AITM (λ=1.3) |

Figure 2: The comparison of clean examples and adversarial examples crafted by three combined attacks. (1) TI-DI-AITM (λ=0.65) achieves similar success rates as TI-DIM and generates much more indistinguishable adversarial examples. (2) TI-DI-AITM (λ=1.3) achieves much higher success rates than TI-DIM and generates more indistinguishable adversarial examples.

cessing steps are improved to boost the indistinguishability and transferability of adversarial examples.

2. We show that AI-FGTM integrated with other transfer-based attacks can obtain much smaller perturbations and larger losses than the current sign attack series.

3. The empirical experiments prove that without extra running time and resource, our best attack fools six classic defense models with an average success rate of 89.3% and three advanced defense models with an average success rate of 82.7%, which are higher than the state-of-

the-art gradient-based attacks.

Review Of Existing Attack Methods

Problem Definition

Let $\left\{(f_{Wi})_{i\in[N]}, (f_{Bj})_{j\in[M]}\right\}$ be a set of pre-trained classifiers, where $(f_{Wi})_{i\in[N]}$ denotes the white-box classifiers and $(f_{Bj})_{j\in[M]}$ represents the unknown classifiers. Given a clean example x, it can be correctly classified to the ground-truth label y^{true} by all pre-trained classifiers. It is

possible to craft an adversarial example x^{adv} that satisfies $\left\| x^{adv} - x \right\|_p \leq \varepsilon$ by using the white-box classifiers, where p could be 0, 1, 2, ∞, and ε is the perturbation size. In this paper, we focus on non-targeted attack with $p = \infty$. Note that, the adversarial example x^{adv} can mislead the white-box classifiers and the unknown classifiers simultaneously.

The Gradient-Based Methods

Here, we introduce the family of the gradient-based methods.

Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) establishes the basic framework of the gradient-based methods. It efficiently crafts an adversarial example x^{adv} by using one-step update while maximizing the loss function $J(x^{adv}, y^{true})$ of a given classifier as

$$x^{adv} = x + \varepsilon \cdot \text{sign}(\nabla_x J(x, y^{true})), \qquad (1)$$

where $\nabla_x J(\cdot, \cdot)$ computes the gradient of the loss function w.r.t. x, $\text{sign}(\cdot)$ is the sign function, and ε is the given scalar value that restricts the L_∞ norm of the perturbation.

Basic Iterative Method (BIM) (Kurakin, Goodfellow, and Bengio 2017a) is the iterative version of FGSM that performs better in white-box attack but less effective in transfer-based attack. It iteratively updates the adversarial example x_t^{adv} with a small step size α as

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha \cdot \text{sign}(\nabla_{x_t^{adv}} J(x_t^{adv}, y^{true})) \right\}, \qquad (2)$$

where $\alpha = \varepsilon / T$ with T denoting the number of iterations. $\text{Clip}_\varepsilon^x \{\cdot\}$ performs per-pixel clipping as

$$\text{Clip}_\varepsilon^x \{x'\} = \min \{255, x + \varepsilon, \max \{0, x - \varepsilon, x'\}\}. \qquad (3)$$

Momentum Iterative Fast Gradient Sign Method (MI-FGSM) (Dong et al. 2018) enhances the transferability of adversarial examples by incorporating momentum term into gradient process, given as

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(x_t^{adv}, y^{true})}{\left\| \nabla_{x_t^{adv}} J(x_t^{adv}, y^{true}) \right\|_1}, \qquad (4)$$

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}) \right\}, \qquad (5)$$

where g_{t+1} denotes the accumulated gradient at $(t+1)_{th}$ iteration , and μ is the decay factor of g_{t+1}.

Nesterov Iterative Method (NIM) (Lin et al. 2020) integrates an anticipatory update into MI-FGSM and further increases the transferability of adversarial examples. The update procedures are expressed as

$$x_t^{nes} = x_t^{adv} + \alpha \cdot \mu \cdot g_t, \qquad (6)$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{nes}} J(x_t^{nes}, y^{true})}{\left\| \nabla_{x_t^{nes}} J(x_t^{nes}, y^{true}) \right\|_1}, \qquad (7)$$

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}) \right\}. \qquad (8)$$

Scale-Invariant Method (SIM) (Lin et al. 2020) applies the scale copies of the input image to further improve the

transferability. However, SIM requires much more running time and resource.

Diverse Input Method (DIM) (Xie et al. 2019) applies random resizing and padding to the adversarial examples with the probability p at each iteration. DIM can be easily integrated into other gradient-based methods to further boost the transferability of adversarial examples. The transformation function $T(x_t^{adv}, p)$ is

$$T(x_t^{adv}, p) = \begin{cases} T(x_t^{adv}), p \\ x_t^{adv}, (1-p) \end{cases}. \qquad (9)$$

Translation-Invariant Method (TIM) (Dong et al. 2019) optimizes an adversarial example by an ensemble of translated examples as

$$x_{t+1}^{adv} = \sum_{i,j} T_{ij}(x_t^{adv}), s.t. \|x_t^{adv} - x^{real}\|_\infty \leq \epsilon, \qquad (10)$$

where $T_{ij}(x_t^{adv})$ denotes the translation function that respectively shifts input x_t^{adv} by i and j pixels along the two-dimensions. And TIM calculates the gradient of the loss function at a point $\hat{x}_t^{adv}$, convolves the gradient with a pre-defined Gaussian filter and updates as

$$\begin{aligned} g' &= \nabla_{x_t^{adv}} (\sum_{i,j} w_{ij} J(T_{ij}(x_t^{adv}), y^{true}))|_{x_t^{adv} = \hat{x}_t^{adv}} \\ &\approx \sum_{i,j} w_{ij} T_{-i-j}(\nabla_{x_t^{adv}} J(x_t^{adv}, y^{true}))|_{x_t^{adv} = \hat{x}_t^{adv}} , \\ &\approx W * \nabla_{x_t^{adv}} J(x_t^{adv}, y^{true}) \end{aligned} \qquad (11)$$

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha \cdot \text{sign}(g') \right\}. \qquad (12)$$

Note that, with limited running time and computing resources, the combination of NIM, TIM and DIM (NI-TI-DIM) is strong transfer-based attack method so far.

Methodology

Motivations

Based on the contradiction that the adversarial examples achieve high success rates but can be identified easily, our observations are shown as follows:

1. The sign function in gradient-based methods has two main disadvantages. On the one hand, the sign function normalizes all the gradient values to 1, -1 or 0, and thus leads to the loss of gradient information. On the other hand, the sign function normalizes some small gradient values to 1 or -1, which increases the perturbation size. While the tanh function not only normalizes the large gradient values as the sign function, but also maintains the small gradient values as function $y = x$. Therefore, the tanh function can replace the sign function and reduce the perturbation size.

2. With iterations $T = 10$, the applications of Nesterov's accelerated gradient (NAG) (Lin et al. 2020) and the momentum algorithm (Dong et al. 2018) in adversarial attacks demonstrate that we can migrate other methods to generate adversarial examples. Moreover, the t_{th} gradient $\nabla_{x_t^{adv}} J(x_t^{adv}, y^{true})$ is normalized by the L_1

distance of itself before the momentum algorithm. Intuitively, due to the performance of traditional convergence algorithms, Adam can achieve larger losses than the momentum algorithm in such small number of iterations. Additionally, Adam can normalize the gradient with $m_t/\sqrt{v_t + \delta}$, where m_t denotes the first moment vector, v_t is the second moment vector and $\delta = 10^{-8}$.

3. Traditional convergence algorithms employ learning rate decay to improve the model performance. Existing gradient-based methods set stable step size $\alpha = \varepsilon/T$. In intuition, we can improve the transferability with the step size change. Different from the traditional convergence algorithms, the attack methods with the ε-ball restriction aim to maximize the loss function of the target models. Hence, we use the increasing step size with $\sum_{t=0}^{T-1} \alpha_t = \varepsilon$.

4. Dong et al. (Dong et al. 2019) show that Gaussian blur with a large kernel improves the transferability of adversarial examples. However, Gaussian blur with larger kernels leads to the loss of the gradient information. Using the modifications mentioned above, the gradient information is preserved and plays a more important role in generating adversarial examples. Consequently, we apply smaller kernels in Gaussian blur to avoid the loss of the gradient information.

Based on the above four observations, we propose AI-FGTM to craft the adversarial examples which are expected to be more transferable and indistinguishable.

AI-FGTM

Adam (Kingma and Ba 2015) uses the exponential moving averages of squared past gradients to mitigate the rapid decay of learning rate. Essentially, this algorithm limits the reliance of update to only the past few gradients by the following simple recursion:

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) g_{t+1}, \qquad (13)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) g_{t+1}^2, \qquad (14)$$

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\sqrt{(1 - \beta_2^t)}}{1 - \beta_1^t} \cdot \frac{m_{t+1}}{\sqrt{v_{t+1} + \delta}}, \qquad (15)$$

where m_t denotes the first moment vector, v_t represents the second moment vector, β_1 and β_2 are the exponential decay rates.

Due to the opposite optimization objectives, we apply Adam into adversarial attack with some modifications. Starting with $x_0^{adv} = x$, $m_0 = 0$ and $v_0 = 0$, the first moment estimate and the second moment estimate are presented as follows:

$$m_{t+1} = m_t + \mu_1 \cdot \nabla_{x_t^{adv}} J(x_t^{adv}, y^{true}), \qquad (16)$$

$$v_{t+1} = v_t + \mu_2 \cdot (\nabla_{x_t^{adv}} J(x_t^{adv}, y^{true}))^2, \qquad (17)$$

where μ_1 and μ_2 denote the first moment factor and second moment factor, respectively. We replace the sign function with the tanh function and update x_{t+1}^{adv} as

$$\alpha_t = \frac{\varepsilon}{\sum_{t=0}^{T-1} \frac{1-\beta_1^{t+1}}{\sqrt{(1-\beta_2^{t+1})}}} \frac{1 - \beta_1^{t+1}}{\sqrt{(1 - \beta_2^{t+1})}}, \qquad (18)$$

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha_t \cdot \tanh(\lambda \frac{m_{t+1}}{\sqrt{v_{t+1} + \delta}}) \right\}, \quad (19)$$

where β_1 and β_2 are exponential decay rates, and λ denotes the scale factor. Specifically, α_t is the increasing step size with $\sum_{t=0}^{T-1} \alpha_t = \varepsilon$. Then the tanh function reduces the perturbations of adversarial examples without any success rate reduction. Furthermore, $m_{t+1}/(\sqrt{v_{t+1}} + \delta)$ replaces the L_1 normalization and the first moment estimate of Eq. 4 due to the fact that Adam has faster divergence speed than momentum attack algorithm (as shown in Fig. 1(b)).

The Combination Of AI-FGTM And NIM

Inspired by NIM that integrates an anticipatory update into MI-FGSM. Similarly, we can also integrate an anticipatory update into AI-FGTM. We first calculate the step size in each iteration as Eq. 18, and the Nesterov term can be expressed as

$$x_t^{nes} = x_t^{adv} + \alpha_t \cdot \frac{m_t}{\sqrt{v_t + \delta}}. \qquad (20)$$

The remaining update procedures are similar to Eq. 16, Eq. 17 and Eq. 19, which can be expressed as

$$m_{t+1} = m_t + \mu_1 \cdot \nabla_{x_t^{nes}} J(x_t^{nes}, y^{true}), \qquad (21)$$

$$v_{t+1} = v_t + \mu_2 \cdot (\nabla_{x_t^{nes}} J(x_t^{nes}, y^{true}))^2, \qquad (22)$$

$$x_{t+1}^{adv} = \text{Clip}_\varepsilon^x \left\{ x_t^{adv} + \alpha_t \cdot \tanh(\lambda \frac{m_{t+1}}{\sqrt{v_{t+1} + \delta}}) \right\}. \quad (23)$$

We summarize NI-TI-DI-AITM as the combination of AI-FGTM, NIM, TIM and DIM, and the procedure is given in Algorithm 1.

Experiments

In this section, we provide extensive experimental results on an ImageNet-compatible dataset to validate our method. First, we introduce the experimental setting. Then, we compare the running efficiency of different transfer-based attacks. Next, we present the ablation study of the effects of different parts of our method. Finally, we compare the results of the baseline attacks. Table 1 presents the definitions of the abbreviations used in this paper.

Abbreviation	Definition
TI-DIM	The combination of MI-FGSM, TIM and DIM
NI-TI-DIM	The combination of MI-FGSM, NIM, TIM and DIM
SI-NI-TI-DIM	The combination of MI-FGSM, NIM, SIM, TIM and DIM
TI-DI-AITM	The combination of AI-FGTM, TIM and DIM
NI-TI-DI-AITM	The combination of AI-FGTM, NIM, TIM and DIM
SI-NI-TI-DI-AITM	The combination of AI-FGTM, NIM, SIM, TIM and DIM

Table 1: Abbreviations used in the paper

Attack	Inc-v3	Inc-v4	IncRes-v2	Res-v2-101	Model ensemble
TI-DIM	172.8	261.2	277.8	234.0	767.5
NI-TI-DIM	174.5	238.9	291.8	243.0	830.2
SI-NI-TI-DIM	608.2	1086.3	1156.2	1096.2	3490.2
TI-DI-AITM	170.6	258.5	280.4	239.3	762.7
NI-TI-DI-AITM	173.5	253.7	288.1	242.1	770.1
SI-NI-TI-DI-AITM	603.6	1103.9	1119.4	1123.1	3341.6

Table 2: The running time (s) of generating 1000 adversarial examples for Inc-v3, Inc-v4, IncRes-v2, Res-v2-101 and the ensemble of theses four models.

Algorithm 1: NI-TI-DI-AITM

Input: A clean example x and its ground-truth label y^{true};
Parameters: The perturbation size ε; the iteration number T; the decay factors μ_1 and μ_2; the exponential decay rates β_1 and β_2; the scale factor λ; the probability p.
Output: An adversarial example x^{adv}.

1: $m_0 = 0$; $v_0 = 0$; $x_0^{adv} = x$;
2: **for** $t = 0$ to $T - 1$ **do**
3:     Input x_t^{adv};
4:     Update step size α by Eq. 18;
5:     Obtain the Nesterov term x_t^{nes} by Eq. 20;
6:     Obtain the diverse input $T(x_t^{nes}, p)$ by Eq. 9;
7:     Compute the gradient $\nabla_{x_t^{adv}} J(T(x_t^{nes}, p), y^{true})$;
8:     Obtain processed gradient g' by Eq. 11;
9:     Update m_{t+1} by $m_{t+1} = m_t + \mu_1 \cdot g'$;
10:    Update v_{t+1} by $v_{t+1} = v_t + \mu_2 \cdot (g')^2$;
11:    Update x_{t+1}^{adv} by Eq. 23;
12: **end for**
13: **Return** $x^{adv} = x_t^{adv}$.

Experimental Setting

Dataset. We utilize 1000 images [1] which are used in the NIPS 2017 adversarial competition to conduct the following experiments.

Models. In this paper, we employ thirteen models to perform the following experiments. Four non-defense models (Inception v3 (Inc-v3) (Szegedy et al. 2016), Inception v4 (Inc-v4), Inception ResNet v2 (IncRes-v2) (Szegedy et al. 2017), and ResNet v2-101 (Res-v2-101) (He et al. 2016)) are used as white-box models to craft adversarial examples. Six defense models (Inc-v3ens3, Inc-v3ens4, In-

cResv2ens (Tramèr et al. 2018), high-level representation guided denoiser (HGD) (Liao et al. 2018), input transformation through random resizing and padding (R&P) (Xie et al. 2018), and rank-3 submission [2] in the NIPS 2017 adversarial competition) are employed as classic models to evaluate the crafted adversarial examples. In addition, we also evaluate the attacks with three advanced defenses (Feature Distillation (Liu et al. 2019), Comdefend (Jia et al. 2019), and Randomized Smoothing (Cohen, Rosenfeld, and Kolter 2019)).

Baselines. We focus on the comparison of TI-DIM, NI-TI-DIM, TI-DI-AITM and NI-TI-DI-AITM, where TI-DIM and NI-TI-DIM are both the state-of-the-art methods.

Hyper-parameters. According to TI-DIM (Dong et al. 2019) and NI-FGSM (Lin et al. 2020), we set the maximum perturbation $\varepsilon = 16$, and the number of iteration $T = 10$. Specifically, we set the kernel size to 15×15 in normal TI-DIM and NI-TI-DIM while 9×9 in TI-DI-AITM. The exploration of appropriate settings of our method is illustrated in **Appendix**.

The mean perturbation size. For an adversarial example x^{adv} with the size of $M \times N \times 3$, the mean perturbation size P_m can be calculated as

$$P_m = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{3} \left| x_{ijk}^{adv} - x_{ijk} \right|}{M \times N \times 3}, \quad (24)$$

where x_{ijk} denotes the value of channel k of the image x at coordinates (i, j).

The Comparison Of Running Efficiency

We compare the running time of each attack mentioned in Table 1 using a piece of Nvidia GPU GTX 1080 Ti. Table 2 shows the running time under single-model setting and model ensemble setting. It can be seen that attacks combined with our method AI-FGTM do not cost extra running.

Attack	tanh	Adam	smaller kernels	dynamic step size	mean success rate (%)	mean perturbation
					82.0	10.46
	✓				82.4	9.14
TI-DIM		✓			83.6	9.20
	✓	✓			83.1	7.86
	✓	✓	✓		86.5	7.82
	✓	✓	✓	✓	88.0	8.11

Table 3: Ablation study of the effects of the tanh function, the Adam optimizer, the kernel size and dynamic step size. The adversarial examples are generated for Inc-v3, Inc-v4, IncRes-v2, Res-v2-101 respectively using TI-DIM and TI-DIM with different parts of our method. We compare the mean perturbations and the mean attack success rates of the generated adversarial examples against six classic defense models.

	Attack	Inc-v3ens3	Inc-v3ens4	IncRes-v2ens	HGD	R&P	NIPS-r3
	TI-DIM	46.5	47.3	38.1	38.0	36.9	41.1
Inc-v3	TI-DI-AITM	**53.8**	**53.3**	**39.0**	**40.2**	**39.1**	**45.7**
	NI-TI-DIM	48.3	48.6	36.9	37.3	36.8	42.5
	NI-TI-DI-AITM	**51.9**	**52.4**	**38.2**	**39.3**	**38.1**	**44.6**
	TI-DIM	48.2	47.9	39.1	40.6	39.3	41.5
Inc-v4	TI-DI-AITM	**53.2**	**51.8**	**42.4**	**43.7**	**42.5**	**44.6**
	NI-TI-DIM	52.4	51.8	41.3	41.9	41.1	42.7
	NI-TI-DI-AITM	**54.8**	**53.7**	**41.7**	**43.9**	**43.2**	**44.1**
	TI-DIM	60.8	59.6	59.3	58.4	60.7	61.3
IncRes-v2	TI-DI-AITM	**64.9**	**61.8**	**62.1**	**62.7**	**64.8**	**65.1**
	NI-TI-DIM	61.5	60.4	59.9	60.1	62.2	63.1
	NI-TI-DI-AITM	**66.5**	**63.8**	**62.0**	**63.2**	**65.6**	**65.8**
	TI-DIM	56.1	55.4	49.8	51.3	50.4	52.3
Res-v2-101	TI-DI-AITM	**62.8**	**62.8**	**54.4**	**55.3**	**54.2**	**57.1**
	NI-TI-DIM	59.5	57.7	50.4	51.9	50.8	54.6
	NI-TI-DI-AITM	**64.0**	**61.0**	**54.6**	**54.8**	**53.4**	**57.6**

Table 4: The success rates (%) of adversarial attacks against six defense models under single-model setting. The adversarial examples are generated for Inc-v3, Inc-v4, IncRes-v2, Res-v2-101 respectively using TI-DIM, NI-TI-DIM, TI-DI-AITM and NI-TI-DI-AITM.

Additionally, SIM requires at least two pieces of GPUs under model ensemble setting and costs much more running time than other attacks under both single-model setting and model ensemble setting. Therefore, we exclude SIM in the following experiments.

Ablation Study

Table 3 shows the ablation study of the effects of different parts of our method. We compare the mean perturbation and the mean success rates against six classic defense models. Our observations are shown as follow:

1. Both of the tanh function and Adam can reduce the perturbation size. Additionally, Adam can also improve the transferability of adversarial examples.

2. The combination of the tanh function and Adam can greatly reduce the perturbation size, but only slightly improve the transferability of adversarial examples.

3. Using smaller kernels and dynamic step size can improve the transferability of adversarial examples even using dynamic step size slightly increase the perturbation size.

The Validation Results In The Single-Model Attack Scenario

In this section, we compare the success rates of AI-FGTM based attacks and the baseline attacks against six classic defenses. We generate adversarial attacks for Inc-v3, Incv4, IncRes-v2, and Res-v2-101 by severally using TI-DIM, TI-DI-AITM, NI-TI-DIM and NI-TI-DI-AITM.

As shown in Table 4, we find that our attack method consistently outperforms the baseline attacks by a large margin. Furthermore, according to Table 4 and Fig. 3(b), we observe that our method can generate adversarial examples with much better transferability and indistinguishability.

The Validation Results In The Model Ensemble Attack Scenario

In this section, we present the success rates of adversarial examples generated for an ensemble of four non-defense models. Table 5 gives the results of transfer-based attacks against six classic defense models. It shows that our methods achieve higher success rates than baseline attacks. In particular, without extra running time and resource, TI-DI-AITM and NI-TI-DI-AITM fool six defense models with an aver-
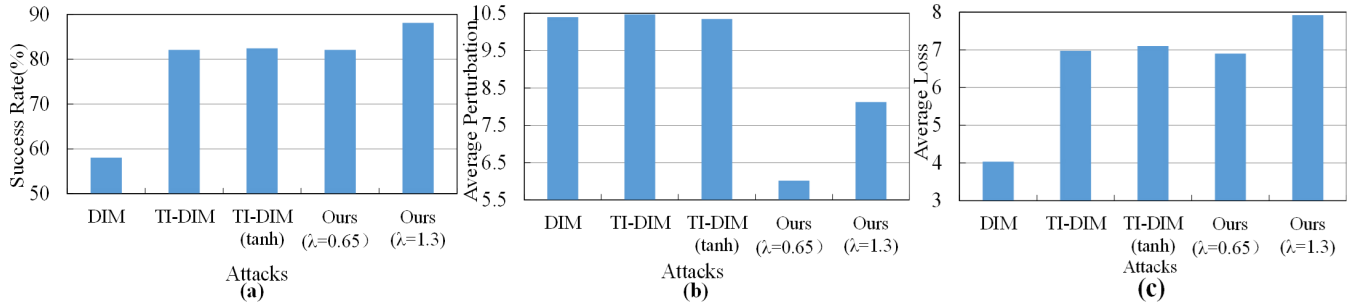
Figure 3: Results of adversarial examples generated for the ensemble of Inc-v3, Inc-v4, IncRes-v2, and Res-v2-101 using different attacks.

Attack	Inc-v3ens3	Inc-v3ens4	IncRes-v2ens	HGD	R&P	NIPS-r3	Average
TI-DIM	83.9	83.2	78.4	81.9	81.2	83.6	82.0
TI-DI-AITM	**90.2**	**88.5**	**85.4**	**88.3**	**87.1**	**88.7**	**88.0**
NI-TI-DIM	85.5	85.9	80.1	83.6	82.9	84.3	83.7
NI-TI-DI-AITM	**91.8**	**90.3**	**85.8**	**89.4**	**88.6**	**90.1**	**89.3**

Table 5: The success rates (%) of adversarial attacks against six defense models under model ensemble setting. The adversarial examples are generated for the ensemble of Inc-v3, Inc-v4, IncRes-v2, and Res-v2-101 using TI-DIM, NI-TI-DIM, TI-DI-AITM and NI-TI-DI-AITM.

Attack	Feature Distillation	Comdefend	Randomized Smoothing	Average
TI-DIM	83.1	78.2	49.9	70.4
TI-DI-AITM	90.6	87.9	63.7	80.7
NI-TI-DIM	82.1	84.7	58.6	75.1
NI-TI-DI-AITM	**91.4**	**90.3**	**66.4**	**82.7**

Table 6: The success rates (%) of adversarial attacks against Feature Distillation, Comdefend, and Randomized Smoothing under model ensemble setting. The adversarial examples are generated for the ensemble of Inc-v3, Inc-v4, IncRes-v2, and Res-v2-101 using TI-DIM, NI-TI-DIM, TI-DI-AITM and NI-TI-DI-AITM.

age success rate of 88.0% and 89.3%, respectively, which are higher than the state-of-the-art gradient-based attacks.

We also validate our method by comparing the different results between DIM, TI-DIM and TI-DI-AI-FGTM in Fig. 3. Adversarial examples are generated for the ensemble of Inc-v3, Inc-v4, IncRes-v2 and Res-v2-101 using different attack methods. Fig. 3 (a) shows that the tanh function does not hurt the performance and Adam can boost the attack success rates. Fig. 3 (b) shows that our method significantly reduce the mean perturbation size. In particular, our method reduces 40% perturbation while delivering the stable performance. Fig. 3 (c) shows that our approach with $\lambda = 1.3$ obtains the largest loss of all the methods.

We evaluate the attacks with three more advanced defenses, namely Feature Distillation (Liu et al. 2019), Comdefend (Jia et al. 2019) and Randomized Smoothing (Cohen, Rosenfeld, and Kolter 2019). Table 6 shows the success rates of TI-DIM, TI-DI-AITM, NI-TI-DIM and NI-TI-DI-AITM against these defenses in the ensemble attack scenario.

In Table 6, we find that the attacks with AI-FGTM consistently outperform the attacks with MI-FGSM. In general, our methods can fool these defenses with high success rates.

Based on the above experimental results, it is reasonable to state that our method can generate adversarial examples with much better indistinguishability and transferability. Meanwhile, our method also raises the security challenge for the development of more effective defense models.

Conclusion

In this paper, we propose AI-FGTM to craft adversarial examples that are much indistinguishable and transferable. AI-FGTM modifies the major gradient processing steps of the basic sign structure to address the limitations faced by the existing basic sign involved methods. Compared with the state-of-the-art attacks, extensive experiments on an ImageNet-compatible dataset show that our method generates more indistinguishable adversarial examples and achieves higher attack success rates without extra running time and resource. Our best attack NI-TI-DI-AITM can fool six classic defense models with an average success rate of 89.3% and fool three advanced defense models with an average success rate of 82.7%, which are higher than the state-of-the-art gradient-based attacks. Additionally, our method reduces nearly 20% mean perturbation.

Acknowledgements

References

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2018. Synthesizing Robust Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning*, 284–293.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *6th International Conference on Learning Representations*.

Chen, J.; Jordan, M. I.; and Wainwright, M. J. 2020. Hop-SkipJumpAttack: A Query-Efficient Decision-Based Attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, 1277–1294. IEEE.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, 1310–1320.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks With Momentum. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 9185–9193.

Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4312–4321.

Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; and Song, D. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 1625–1634.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations*.

Guo, C.; Rana, M.; Cissé, M.; and van der Maaten, L. 2018. Countering Adversarial Images using Input Transformations. In *6th International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity Mappings in Deep Residual Networks. In *Computer Vision - ECCV 2016 - 14th European Conference*, 630–645.

Jia, X.; Wei, X.; Cao, X.; and Foroosh, H. 2019. ComDefend: An Efficient Image Compression Model to Defend Adversarial Examples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6084–6092.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017a. Adversarial examples in the physical world. In *5th International Conference on Learning Representations*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017b. Adversarial Machine Learning at Scale. In *5th International Conference on Learning Representations*.

Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; and Zhu, J. 2018. Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 1778–1787.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2020. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2017. Delving into Transferable Adversarial Examples and Black-box Attacks. In *5th International Conference on Learning Representations*.

Liu, Z.; Liu, Q.; Liu, T.; Xu, N.; Lin, X.; Wang, Y.; and Wen, W. 2019. Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples. In *IEEE Conference on Computer Vision and Pattern Recognition*, 860–868.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations*.

Moosavi-Dezfooli, S.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal Adversarial Perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 86–94.

Narodytska, N.; and Kasiviswanathan, S. P. 2017. Simple Black-Box Adversarial Attacks on Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, 1310–1318. IEEE Computer Society.

Pang, T.; Du, C.; and Zhu, J. 2018. Max-Mahalanobis Linear Discriminant Analysis Networks. In *Proceedings of the 35th International Conference on Machine Learning*, 4013–4022.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified Defenses against Adversarial Examples. In *6th International Conference on Learning Representations*.

Rauber, J.; Zimmermann, R. S.; Bethge, M.; and Brendel, W. 2020. Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX. *J. Open Source Softw.*, 5(53): 2607.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models. In *6th International Conference on Learning Representations*.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the Impact of

Residual Connections on Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4278–4284.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I. J.; Boneh, D.; and McDaniel, P. D. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *6th International Conference on Learning Representations*.

Wong, E.; and Kolter, J. Z. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the 35th International Conference on Machine Learning*, 5283–5292.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. L. 2018. Mitigating Adversarial Effects Through Randomization. In *6th International Conference on Learning Representations*.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving Transferability of Adversarial Examples With Input Diversity. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2730–2739.