Direction-aggregated Attack for Transferable Adversarial Examples

TIANJIN HUANG, VLADO MENKOVSKI, and YULONG PEI, Eindhoven University of Technology, Eindhoven, The Netherlands
YUHAO WANG, National University of Singapore
MYKOLA PECHENIZKIY, Eindhoven University of Technology, Eindhoven, The Netherlands

Deep neural networks are vulnerable to adversarial examples that are crafted by imposing imperceptible changes to the inputs. However, these adversarial examples are most successful in white-box settings where the model and its parameters are available. Finding adversarial examples that are transferable to other models or developed in a black-box setting is significantly more difficult. In this article, we propose the Direction-aggregated adversarial attacks that deliver transferable adversarial examples. Our method utilizes the aggregated direction during the attack process for avoiding the generated adversarial examples overfitting to the white-box model. Extensive experiments on ImageNet show that our proposed method improves the transferability of adversarial examples significantly and outperforms state-of-the-art attacks, especially against adversarial trained models. The best averaged attack success rate of our proposed method reaches 94.6% against three adversarial trained models and 94.8% against five defense methods. It also reveals that current defense approaches do not prevent transferable adversarial attacks.

CCS Concepts: • General and reference → Reliability; • Computing methodologies → Neural networks;

Additional Key Words and Phrases: Adversarial examples, transferability, deep neural network

ACM Reference format:
Tianjin Huang, Vlado Menkovski, Yulong Pei, YuHao Wang, and Mykola Pechenizkiy. 2022. Direction-aggregated Attack for Transferable Adversarial Examples. J. Emerg. Technol. Comput. Syst. 18, 3, Article 60 (April 2022), 22 pages.
https://doi.org/10.1145/3501769

1 INTRODUCTION

Deep Neural Networks (DNNs) have achieved great success in many tasks, e.g., image classification [8, 11], object detection [6], segmentation [21], and so on. However, these high-performing models have been shown to be vulnerable to adversarial examples [14, 27]. In other words, carefully crafted changes to the inputs can change the model’s prediction drastically. This fragility has raised concerns on security-sensitive tasks, such as autonomous cars, face recognition, and...
malware detection. Well-designed adversarial examples are not only useful to evaluate the robustness of models against adversarial attacks but also beneficial to improve the robustness of them [14].

Plenty of ways have been proposed to craft adversarial examples, which can be divided into white-box and black-box attacks. White-box attacks utilize complete knowledge including model architecture, model parameters, training strategy, and training method, e.g., fast gradient sign method (FGSM) [14], iterative fast gradient sign method (I-FGSM) [13], project gradient descent (PGD) [22], Deepfool [23], momentum iterative fast gradient sign method (MI-FGSM) [3], and Carlini and Wagner’s attack [1]. On the contrary, black-box attacks fool the model’s prediction without any knowledge about the model. It has been shown that adversarial examples generated by white-box attacks have the ability to fool other black-box models, which is known as the transferability property [27]. The transferability of adversarial examples enables practical black-box attacks and imposes a huge threat on real-world applications. However, the transferability of adversarial examples usually is very low, because these adversarial examples easily overfit to the white-box model, i.e., the model for generating these adversarial examples. Therefore, avoiding the overfitting problem is the key to generate transferable adversarial examples.

Deep neural networks applied to high-dimensional classification tasks are typically very complex models, in other words, the decision boundary is highly non-linear and tends to have high curvature, e.g., the decision boundary of model 1 in Figure 1. We believe that it is the high curvature of a decision boundary that makes adversarial examples decrease their ability to attack other models especially adversarial robust models\(^1\) that have smoothed decision boundary [2, 15]. As shown in Figure 1, the adversarial attack direction generated by model 1 at sample \(x\) (the black arrow line in Figure 1) tends to overfit to model 1, because this attack direction is the best direction\(^2\) for attacking model 1, but not a good direction for attacking model 2. To mitigate the issue of adversarial examples easily overfitting to the white-box model, we propose to aggregate the attack directions from the neighborhood of the input \(x\), e.g., by adding Gaussian noise or Uniform noise to the input. The green solid arrow line in Figure 1 shows the aggregated direction. It is easy to see that the green solid arrow line is a good attack direction for both model 1 and model 2. Therefore, adversarial examples generated by the aggregated direction can achieve good transferability. Based on this, we propose the direction-aggregated attack (DA-Attack) for improving the transferability of adversarial examples. Results of the extensive experiments presented in later sections show that our method achieves state-of-the-art results.

In detail, our contributions are summarized as follows:

- We propose to aggregate attack directions to stabilize the oscillation of attack directions and guide the attack direction to the generalized decision boundary and avoid overfitting to the white-box model’s decision boundary. Based on the aggregated direction, we propose our DA-Attack.
- We demonstrate experimentally that DA-Attack outperforms state-of-the-art attacks by extensive experiments on ImageNet. The best averaged attack success rate of our method achieves 94.6% against three ensemble adversarial trained models and 94.8% against five defense methods, which also reveals that current defense models are not safe to transferable adversarial attacks. We expect that the proposed DA-Attack will serve as a benchmark for evaluating the effectiveness of adversarial defense methods in the future.

---

\(^1\)In this article, it denotes a model trained with an adversarial training technique.

\(^2\)It denotes the direction that is perpendicular to the decision boundary.
Direction-aggregated Attack for Transferable Adversarial Examples

Fig. 1. A simple schematic diagram for explaining why aggregated direction can mitigate the overfitting problem of adversarial examples. Black circle and triangle markers denote samples of class 1 and class 2, respectively. Blue and red lines represent the decision boundary of model 1 and model 2, respectively. Circle with dotted line denotes a set of examples from the neighborhood of \(x\). Black arrow line denotes the attack direction \((\text{sgn}(\nabla_x L(f_\theta(x), y)))\) of model 1 at sample \(x\). Green arrow dotted lines are the attack direction at the perturbed sample with Gaussian noise. Green arrow solid lines denote the aggregated direction by the vector addition of the green arrow dotted lines.

- We experimentally show that sampling times \(N\), standard deviation \(\sigma\), iterations \(T\), and perturbation size \(\epsilon\) induced in our method play an important role in achieving the transferability of adversarial examples. Usually a bigger value in \(N\), \(\sigma\), \(T\), and \(\epsilon\) can lead to a higher transferability of the adversarial examples. However, a too large value in \(T\) and \(\sigma\) would lead to a negative effect.

The rest of this article is organized as follows. In Section 2, we present related work. In Section 3, we describe our proposed DA-Attack in detail. In Section 4, we discuss the results of the extensive experiments with DA-Attack. In Section 5, we discuss the connection of the DA-Attack to a smoothed classifier. We draw conclusions in Section 6.

2 RELATED WORK

Adversarial examples. Szegedy et al. [27] first found the existence of adversarial examples: given an input \((x, y)\) and a classifier \(f_\theta\), it is possible to find a similar input \(x^*\) such that \(f_\theta(x^*) \neq y\). A formal mathematical definition is as follows:

\[
\min_{x^*} \|x^* - x\|_p, \text{ s.t. } f_\theta(x^*) \neq y, f_\theta(x) = y, \tag{1}
\]

where \(\|\cdot\|_p\) denotes the \(L_p\) distance.

Following Reference [27], many related researches have emerged. On the one hand, some of them propose to generate adversarial examples that can be applied in the physical world [5, 13]. On the other hand, some of them focus on reducing the minimal size of adversarial perturbations and improving the attack success rates [1, 3, 14, 23]. Among these researches, the attack success rates under the black-box setting is still low, especially against adversarial trained models, i.e., the model is trained by adversarial training technique, which can effectively defend against adversarial examples [22]. Recently, several papers improve the attack success rates based on
transferable adversarial attacks. Inkawhich et al. [9] generate more transferable adversarial examples by enlarging the distance between adversarial examples and clean samples in feature space. Their intuition is from the fact that deep feature representations of models are transferable. Similar in utilizing feature representations, Zhou et al. [32] improve the transferability by reducing the variations of adversarial perturbations via constructing a new regularization based on feature representations. Liu et al. [19] demonstrate that the transferability can be improved by attacking an ensemble of substitute models. This method suffers from expensive computational cost, since multiple models are needed to be trained first. Li et al. [16] further reduce the computation cost of the method by attacking “Ghost Networks” where the “Ghost Networks” are generated from a basic trained model. Xie et al. [31] believe that overfitting to the white-box model decreases the transferability of adversarial examples, therefore they induce the data augmentation technique to mitigate the overfitting issue. Specifically, they apply random transformations to the inputs and calculate gradient based on the transformed inputs. Dong et al. [4] find that different models make predictions based on different discriminative regions of the input, which decreases the transferability of adversarial examples. Based on this intuition, they propose a translation-invariant attack by averaging the gradients from an ensemble of images composed of the image and its translated versions. Similarly, Lin et al. [18] enhance the transferability of adversarial examples by averaging gradients from an ensemble of images composed of the image and its scaled versions. Besides, Lin et al. [18] also demonstrate that Nesterov accelerated gradient can further improve the transferability of adversarial examples. Wu and Zhu [29] improve the transferability of adversarial examples by smoothing the loss surface. Our method is degraded to this method when the attack direction of each step is the gradient of loss w.r.t. the inputs. Naseer et al. [24] propose “domain-agnostic” adversarial perturbations, which can be used to fool models learned from different domains.

Defense against adversarial examples. Correspondingly, many methods have been proposed to defend against these adversarial examples. Usually, the ability of a model for defending adversarial examples is referred to adversarial robustness. It measures a model’s resilience against adversarial examples. Goodfellow, Shlen, and Szegedy [14] and Madry et al. [22] effectively improve a model’s adversarial robustness by adversarial training technique. That is, it trains model based on on-the-fly generated adversarial examples \( x^* \) bounded by uniformly \( \varepsilon \)-ball of the input \( x \) (i.e., \( \| x^* - x \| \leq \varepsilon \)). Tramer et al. [28] further improve adversarial robustness by ensemble adversarial training where the model is trained on adversarial examples generated from multiple pretrained models. Cohen, Rosenfeld and Kolter [2] build guaranteed adversarial robust model by transforming a base classifier \( f \) into a smoothed classifier’s \( g \). Specifically, the prediction of \( g(x) \) is defined to be the class that \( f \) is most likely to classify the random variable \( N(x, \sigma^2 I) \) as. However, several papers try to defend against adversarial examples by purifying or reducing adversarial perturbations. Xie et al. [30] and Guo et al. [7] impose transformations, e.g., image cropping, rescaling, quilting, padding and so on, on input images at inference time to reduce the adversarial perturbations, and therefore increase accuracy of the model’s performance on adversarial examples. Liao et al. [17] propose a U-net-based denoiser to purify the adversarial perturbations.

### 3 METHODOLOGY

In this section, we first introduce notation and then provide details of our method.

#### 3.1 Notation

We specify the notations that are used in this article by the following list:

- \( x \) and \( y \) denote a clean image and the corresponding true label, respectively.
- \( x^* \) denotes the adversarial example.
- \( f_\theta(x) \) denotes a deep neural network.
Direction-aggregated Attack for Transferable Adversarial Examples

- $L(f_\theta(x), y)$ represents the Cross-entropy loss.
- $\text{sgn}(\cdot)$ denotes the sign function.
- $\nabla_x L(\cdot)$ denotes the gradient of $L(\cdot)$ with respect to $x$.
- $\text{Clip}_x^\epsilon(\cdot)$ function limits the generated adversarial example $x^*$ to the $\epsilon$ max-norm ball of $x$.
- $\epsilon$ is the allowed maximum perturbation size of the adversarial perturbation.
- $\alpha$ is the step size for PGD/FGSM-based adversarial attacks.
- $\mathcal{N}(0, \sigma^2 I)$ denotes Gaussian distribution with mean 0 and standard deviation $\sigma$.
- $U(a, b)$ is Uniform distribution.
- $\epsilon$ denotes a small random noise and can be generated from Gaussian distribution or Uniform distribution. In this article, we adopt Gaussian noise by default.
- $|\cdot|$ denotes the number of elements of a set.
- $D^*$ denotes a set of adversarial examples.

3.2 Gradient-based Adversarial Attack Methods

Several adversarial attacks will be integrated into our proposed method. We give a brief introduction of them in this section.

**Fast Gradient Sign Method (FGSM)** [14] generates adversarial examples by adding a fixed magnitude along the sign of gradients of the loss function, which is formalized as follows:

$$x^* = x + \epsilon \cdot \text{sgn}(\nabla_x L(f_\theta(x), y))$$  \hspace{1cm} (2)

**Iterative Fast Gradient Sign Method (I-FGSM)** [12] is a multi-step variant of FGSM and restricts the perturbed size to the $\epsilon$ max-norm ball. With the initialization $x^*_0 = x$, the perturbed data in $t$-th step $x^*_t$ can be expressed as follows:

$$x^*_t = \text{Clip}_x^\epsilon\{x^*_t - 1 + \alpha \cdot \text{sgn}(\nabla_x L(f_\theta(x^*_{t-1}), y))\}.$$  \hspace{1cm} (3)

**Momentum iterative fast gradient sign method (MI-FGSM)** [3] integrates momentum into I-FGSM method for stabilizing optimization, which can be expressed as follows:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x L(f_\theta(x^*_t), y)}{\|\nabla_x L(f_\theta(x^*_t), y)\|_1},$$  \hspace{1cm} (4)

$$x^*_{t+1} = \text{Clip}_x^\epsilon\{x^*_t + \alpha \cdot \text{sgn}(g_t)\}.$$  \hspace{1cm} (5)

where $g_t$ is the accumulated gradient at iteration $t$ and $\mu$ is the decay factor of the momentum term.

**Diverse Inputs Method(DIM)** [31] calculates gradient based on random transformed inputs. The transformation contains random resizing and padding with a given probability. Formally, it can be expressed as follows:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x L(f_\theta(T(x^*_t; p)), y)}{\|\nabla_x L(f_\theta(T(x^*_t; p)), y)\|_1},$$  \hspace{1cm} (6)

$$x^*_{t+1} = \text{Clip}_x^\epsilon\{x^*_t + \alpha \cdot \text{sgn}(g_{t+1})\},$$  \hspace{1cm} (7)

where $T(\cdot; p)$ is the stochastic transformation function and $p$ is the transformation probability.

**Translation-invariant Method(TIM)** [4] generates an adversarial example by an ensemble of translated inputs and it was demonstrated to be equivalent to convolve the gradient at the
untranslated image. Specifically, it can be expressed as follows:

\[
g_{t+1} = \mu \cdot g_t + \frac{W \ast \nabla L(f_\theta(x^*_t), y)}{\|W \ast \nabla L(f_\theta(x^*_t), y)\|_1},
\]

\[
x^*_{t+1} = \text{Clip}_x(x^*_t + \alpha \cdot \text{sgn}(g_t)),
\]

where * is the convolutional operation and \(W\) is the kernel matrix of size \((2k + 1) \times (2k + 1)\). Following Reference [4], a Gaussian kernel is chosen for our experiments. It is defined as: \(W_{i,j} = \frac{1}{2\pi\sigma^2} \exp(-\frac{\|i - j\|^2}{2\sigma^2})\) where the standard deviation \(\sigma = k/\sqrt{3}\) and \(W_{i,j} = \sum_{l,j} W_{i,j}\).

### 3.3 Direction-aggregated Attack (DA-Attack)

In Figure 1, we illustrated that adversarial examples could overfit to the white-box model due to the very complex decision boundary decreasing their transferability. We mitigate this problem with overfitting adversarial examples by aggregating the attack directions of a set of examples from the neighborhood of the input. We integrate the aggregated direction to basic adversarial attacks, i.e., Fast Gradient Sign Method (FGSM) [27], Iterative Fast Gradient Sign Method (I-FGSM) [12], and Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [3], for improving their transferability. Besides, to further enhance the transferability, we combine our method with other transferable adversarial attacks, i.e., Diverse Input Method (DIM) [31], Translation-Invariant Method (TIM) [4], TI-DIM [4]. Concretely, the update procedures for each attack are formalized as follows.

**DA-FGSM.** To mitigate the effect of overfitting to the specific model and improve the transferability of adversarial examples for FGSM attack, we propose the Direction-aggregated FGSM (DA-FGSM). The attack direction is replaced with the aggregated direction, which are achieved by aggregating the attack directions of a set of examples from the neighborhood of the input \(x\). In practice, we generate the set of examples by adding small perturbations to the input, i.e., adding Gaussian noise or Uniform noise to the input. In this article, we adopt Gaussian noise as the default choice. We further provide the evidence that Uniform noise can reach the same performance as Gaussian noise. Formally, it can be represented as follows:

\[
x^* = x + \epsilon \cdot \text{sgn}\left(\sum_{i=0}^{N} (\text{sgn}(\nabla L(f_\theta(x + \epsilon_i), y)))\right),
\]

where \(N\) denotes the sampling times from certain noise distribution. The \(\text{sgn}(\nabla L(f_\theta(x + \epsilon_i), y))\) denotes one specific attack direction. We aggregate the \(N\) attack directions by the sum operation.

**DA-I-FGSM.** To improve the transferability for I-FGSM. We propose the Direction-aggregated I-FGSM (DA-I-FGSM). The attack direction at each iteration is replaced with the aggregated direction. The update procedure can be formalized as follows:

\[
x^*_t = \text{Clip}_x\left(x^*_{t-1} + \alpha \cdot \text{sgn}\left(\sum_{i=0}^{N} (\text{sgn}(\nabla L(f_\theta(x^*_t + \epsilon_i), y)))\right)\right).
\]

**DA-MI-FGSM.** We integrate the momentum term into DA-I-FGSM for improving the attack ability, which is called Momentum Direction-aggregated I-FGSM (DA-MI-FGSM). The update procedure of DA-MI-FGSM can be expressed as follows:

\[
g_a = \sum_{i=0}^{N} (\text{sgn}(\nabla L(f_\theta(x^*_t + \epsilon_i), y)))
\]
\[
g_t = \mu \cdot g_{t-1} + \frac{g_a}{\|g_a\|_1},
\]
\[
x^*_{t} = \text{Clip}_x\left(x^*_{t-1} + \alpha \cdot \text{sgn}(g_t)\right),
\]
where $g_t$ is the accumulated gradient at iteration $t$ and $\mu$ is the decay factor of the momentum term, and $g_a$ is the aggregated direction.

**DA-DIM.** We combine our proposed DA-MI-FGSM with DIM for further improving the transferability of adversarial examples and denote it as Direction-aggregated DIM (DA-DIM). The update procedure is similar to DA-MI-FGSM, with the replacement of Equation (12) by the following equation:

$$g_a = \sum_{i=0}^{N} (\text{sgn}(\nabla_x L(f_\theta(T(x^*_t + \varepsilon_i; p)), y))).$$  \hfill (15)

**DA-TIM.** Similar to DA-DIM, we combine DA-MI-FGSM with TIM and denote it as Direction-aggregated TIM (DA-TIM). Likewise, the update procedure is similar to DA-MI-FGSM, with the replacement of Equation (13) by the following equation:

$$g_t = \mu \cdot g_{t-1} + \frac{W \cdot g_a}{\|W \cdot g_a\|_1}.$$  \hfill (16)

**DA-TI-DIM.** Following Reference [18], we combine DA-MI-FGSM with TIM and DIM together and denote it as Direction-aggregated TI-DIM (DA-TI-DIM). The update procedure can be presented as follows:

$$g_a = \sum_{i=0}^{N} (\text{sgn}(\nabla_x L(f_\theta(T(x^*_t + \varepsilon_i; p)), y))),$$  \hfill (17)

$$g_t = \mu \cdot g_{t-1} + \frac{W \cdot g_a}{\|W \cdot g_a\|_1},$$  \hfill (18)

$$x^*_t = \text{Clip}_x (x^*_t + \alpha \cdot \text{sgn}(g_t)).$$  \hfill (19)

The pseudocode of DA-MI-FGSM is summarized in Algorithm 1 and the code is provided.\footnote{https://github.com/Juintin/DA-Attack.git.}

**ALGORITHM 1: DA-MI-FGSM**

**Require:** A input image $x$ with true label $y$; a classifier $f$ with loss function $L$; perturbation size $\varepsilon$; maximum iterations $T$; Gaussian distribution $N(0, \sigma^2 I)$; The decay factor $\mu$; the aggregated direction $g_a$.

**Ensure:** An adversarial example $x^*$

1: $\alpha = \varepsilon / T$
2: $x^*_0 = x$; $g_0 = 0$
3: for $t = 1$ to $T$
4: \hspace{1em} $g_a$ = 0
5: \hspace{1em} for $i = 0$ to $N$
6: \hspace{2em} Get $\varepsilon_i \sim N(0, \sigma^2 I)$
7: \hspace{2em} Aggregate attack directions as $g_a = g_a + \text{sgn}(\nabla_x L(f_\theta(T(x^*_t - 1 + \varepsilon_i), y)))$
8: \hspace{1em} end for
9: \hspace{1em} Update $g_t = \mu \cdot g_{t-1} + \frac{g_a}{\|g_a\|_1}$
10: \hspace{1em} Update $x^*_t = \text{Clip}_x (x^*_t + \alpha \cdot \text{sgn}(g_t))$
11: end for
12: $x^* = x^*_T$
13: return $x^*$

ACM Journal on Emerging Technologies in Computing Systems, Vol. 18, No. 3, Article 60. Pub. date: April 2022.
4 EXPERIMENTS

We evaluate the effectiveness of DA-Attack empirically. We first introduce the dataset and experimental settings. Then, we show the performance of our method against normal and defense models. Finally, we analyze the influence of the parameters $N$, $\sigma$, $\epsilon$, $T$, and $\alpha$ on achieving the transferability of adversarial examples.

4.1 Experimental Settings

Datasets. Following the strategy used in Reference [18], a set of 1,000 images (denoted as $D$) that are correctly classified by all testing models are randomly selected from ILSVRC 2012 validation set. For a fair comparison with state-of-the-art methods, we use the same 1,000 images in Reference [18].

Models. Four normal trained models and three ensemble adversarial trained models are used for evaluating adversarial examples, which are Inception-V3 (Inc-V3) [26], Inception-v4 (Inc-V4) [25], Inception-Resnet-v2 (IncRes-V2) [25], Resnet-V2 (Res-101) [8], Inc-V3$_{ens3}$, Inc-V3$_{ens4}$, and IncRes-V2$_{ens}$ [28], respectively. Besides, five advanced defense methods are considered for further evaluating the effectiveness of our method. Specifically, the selected advanced defense methods are high-level representation guided denoiser (HGD) [17], random resizing and padding (R&P) [30], NIPS-r3,$^5$ feature distillation (FD) [20], and purifying perturbations by image compression (Comdefend) [10].

Baselines. Several most recently proposed methods aiming at generating transferable adversarial examples are taken as baselines:

- DIM [31], which generates transferable examples by random resizing input images;
- TIM [4], which generates transferable examples by a set of translated images;
- SI-NI-FGSM [18], which generates transferable examples by scaled images and nesterov accelerated gradients; and
- The combinations of DIM, TIM, and SI-NI-FGSM, namely, TI-DIM [4], SI-NI-TIM [18], SI-NI-DIM [18], and SI-NI-TI-DIM [18] attacks.

Considering that we completely follow the experimental settings in Reference [18], all the baseline results except for the attack success rates against FD and ComDefend in Table 6 are from Reference [18].

Hyper-Parameters. We follow the settings in Reference [18] for all hyper-parameters, the maximum perturbation $\epsilon$ is set to 16 and the number of iterations $T$ is set to 12 as default values. Accordingly $\alpha = \epsilon / T$. The momentum parameter $\mu$ is set to 1.0. For DIM and TI-DIM methods, the transformation probability is set to 0.5. For TIM method, Gaussian kernel is adopted as our baseline experiments and kernel size is set to $7 \times 7$. For SI-NI-FGSM, SI-NI-TIM, SI-NI-DIM, and SI-NI-TI-DIM methods, the number of scales is set to 5. For our DA-Attack, sampling times $N$ and standard deviation $\sigma$ are set to 30 and 0.05, respectively.

Criteria. We use the attack success rates to reflect the ability of adversarial examples attacking a model. The attack success rates is defined as follows:

$$\frac{100 \times \sum_{i=1}^{M} \left[ \arg \max_{j} f_j(x^*_i) \neq y_i \right]}{M},$$

where $(x^*_i, y_i) \in D^*$ and $M$ is the number of adversarial examples in $D^*$.

$^4$https://github.com/JHL-HUST/SI-NI-FGSM.

$^5$https://github.com/anlthms/nips-2017/tree/master/mmd.

ACM Journal on Emerging Technologies in Computing Systems, Vol. 18, No. 3, Article 60. Pub. date: April 2022.
Table 1. The Attack Success Rates (%) Against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens3}, Inc-V3\textsubscript{ens4}, and IncRes-V2\textsubscript{ens} Models

| Attack          | Inc-V3\textsuperscript{*} | Inc-V4 | IncRes-V2 | Res-101 | Inc-V3\textsubscript{ens3} | Inc-V3\textsubscript{ens4} | IncRes-V2\textsubscript{ens} |
|-----------------|----------------------------|--------|-----------|---------|-----------------------------|-----------------------------|-----------------------------|
| FGSM            | 67.1                       | 26.7   | 25        | 24.4    | 10.5                        | 10                          | 4.5                         |
| I-FGSM          | 99.9                       | 20.7   | 18.5      | 15.3    | 3.6                         | 5.8                         | 2.9                         |
| PGD             | 99.5                       | 17.3   | 15.1      | 13.1    | 6.1                         | 5.6                         | 3.1                         |
| C&W             | 100                        | 18.4   | 16.2      | 14.3    | 3.8                         | 4.7                         | 2.7                         |
| MI-FGSM         | 100.0                      | 40.0   | 38.2      | 32.3    | 12.5                        | 12.8                        | 6.8                         |
| DIM             | 98.7                       | 67.7   | 62.9      | 54      | 20.5                        | 18.4                        | 9.7                         |
| TIM             | 100                        | 47.8   | 42.8      | 39.5    | 24                          | 21.4                        | 12.9                        |
| SI-NI-FGSM      | 100                        | 76     | 73.3      | 67.6    | 31.6                        | 30                          | 17.4                        |
| DA-FGSM(Ours)   | 87.6                       | 47     | 43.6      | 42      | 18.3                        | 17.4                        | 9.5                         |
| DA-I-FGSM(Ours) | 99.8                       | 44     | 39.2      | 34.3    | 23.7                        | 22.4                        | 12.4                        |
| DA-MI-FGSM(Ours)| 99.8                       | 80.6   | 78.5      | 72.2    | **40.6**                    | **40.4**                    | **26.5**                    |

The adversarial examples are generated based on Inc-V3 model by normal adversarial attacks, including FGSM, I-FGSM, PGD, C&W, and transferable adversarial attacks, including DIM, TIM, SI-NI-FGSM, DA-FGSM, DA-I-FGSM, and DA-MI-FGSM attacks. * denotes the white-box model being attacked.

4.2 Single-model Attacks

We first evaluate the effectiveness of DA-Attack based on the single model. DIM [31], TIM [4], and SI-NI-FGSM [18] and their combinations, i.e., SI-NI-TIM, TI-DIM, and SI-NI-TI-DIM, are taken as baselines. Besides, several popular normal adversarial attacks, i.e., FGSM, I-FGSM, MI-FGSM, PGD, and C&W, are utilized to show the effectiveness of our method.

Comparison with normal and transferable attacks. The attack success rates of DIM, TIM, SI-NI-FGSM, normal attacks, and our proposed method are shown in Table 1. The adversarial examples are crafted based on Inc-V3 model. From Table 1, it can be observed:

- Adversarial examples are much easier to attack normal trained models than adversarial trained models.
- Adversarial examples generated by transferable attacks have much higher attack success rates against black-box models than normal attacks.
- Our proposed M-ADI-FGSM attack outperforms the current state-of-the-art SI-NI-FGSM attack by 4.6%–10.4%. Besides, DA-FGSM and DA-I-FGSM attacks without momentum acceleration still achieve remarkable results compared with normal attacks, which demonstrate the effectiveness of the aggregated direction.

Besides, it is worthy noting that adversarial examples from I-FGSM attack are less transferable than that from FGSM attack (by comparing I-FGSM with FGSM in Table 1), which shows the evidence that adversarial examples overfitting to the white-box model decreases the transferability. And the transferability is improved by adding a momentum term during generating adversarial examples (by comparing MI-FGSM with I-FGSM in Table 1), which conforms the claim in Reference [3]. Interestingly, the combination of Direction Aggregation and momentum can greatly improve the transferability again (by comparing MI-FGSM with DA-MI-FGSM in Table 1). We conjecture that it is because the proposed Direction Aggregation technique is orthogonal to the momentum technique. Intuitively, Direction Aggregation technique stabilize the attack direction by reducing the oscillation of each update direction during the iterations while momentum stabilize the attack direction by accumulating historical update directions.

Comparison with the extensions of DIM and TIM. To fully evaluate DA-TIM, DA-DIM, and DA-TI-DIM attacks, adversarial examples are crafted by these attacks based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models, respectively. We test it against the four normal trained and three
Table 2. Comparison of TIM, SI-NI-TIM, and the DA-TIM Extension

| Model       | Attack             | Inc-V3 | Inc-V4 | IncRes-V2 | Res-101 | Inc-V3ens3 | Inc-V3ens4 | IncRes-V2ens |
|-------------|--------------------|--------|--------|-----------|---------|------------|------------|-------------|
| TIM         |                    |        |        |           |         |            |            |             |
| Inc-V3      | SI-NI-TIM          | 100*   | 77.2   | 75.8      | 66.5    | 51.8       | 45.9       | 35.5        |
| DA-TIM(Ours)|                   | 99.8*  | 80.9   | 77.9      | 71.8    | 66.9       | 65.2       | 51.2        |
| TIM         |                    |        |        |           |         |            |            |             |
| Inc-V4      | SI-NI-TIM          | 83.5   | 100*   | 76.6      | 68.9    | 57.8       | 54.3       | 42.9        |
| DA-TIM(Ours)|                   | 84.2   | 98.4*  | 77.7      | 69.3    | 66.8       | 65.9       | 56.4        |
| IncRes-V2   | SI-NI-TIM          | 86.4   | 83.2   | 99.5*     | 77.2    | 66.1       | 60.2       | 57.1        |
| DA-TIM(Ours)|                   |        |        |           |         |            |            |             |
| Res-101     | SI-NI-TIM          | 78.3   | 74.1   | 73        | 99.8*   | 58.9       | 53.9       | 43.1        |
| DA-TIM(Ours)|                   | 78.6   | 74.7   | 76        | 99.2*   | 72.1       | 69.7       | 62.7        |

The attack success rates (%) are shown in the table. Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101, respectively. * denotes the attack success rates under white-box attacks.

Table 3. Comparison of DIM, SI-NI-DIM, and the DA-DIM Extension

| Model       | Attack             | Inc-V3 | Inc-V4 | IncRes-V2 | Res-101 | Inc-V3ens3 | Inc-V3ens4 | IncRes-V2ens |
|-------------|--------------------|--------|--------|-----------|---------|------------|------------|-------------|
| DIM         |                    |        |        |           |         |            |            |             |
| Inc-V3      | SI-NI-DIM          | 99.6*  | 84.7   | 81.7      | 75.4    | 36.9       | 34.6       | 20.2        |
| DA-DIM(Ours)|                   | 99.5*  | 89     | 87.3      | 81.2    | 57.1       | 56.6       | 38.8        |
| DIM         |                    |        |        |           |         |            |            |             |
| Inc-V4      | SI-NI-DIM          | 89.7   | 99.3*  | 84.5      | 78.5    | 47.6       | 45         | 28.9        |
| DA-DIM(Ours)|                   | 90.8   | 98.1*  | 87.1      | 80.9    | 62.1       | 62.9       | 49.7        |
| IncRes-V2   | SI-NI-DIM          | 89.7   | 86.4   | 99.1*     | 81.2    | 55         | 48.2       | 38.1        |
| DA-DIM(Ours)|                   |        |        |           |         |            |            |             |
| Res-101     | SI-NI-DIM          | 88.7   | 84.2   | 84.4      | 99.3*   | 53.4       | 48         | 33.2        |
| DA-DIM(Ours)|                   | 90.9   | 87.7   | 89.4      | 99.2*   | 75.3       | 72.6       | 62.9        |

The numbers in table denote the attack success rates (%). Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101, respectively, using DIM, SI-NI-DIM, and DA-DIM methods. * denotes the attack success rates under white-box attacks.

ensemble adversarial trained models. The evaluation results are shown in Tables 2, 3, and 4. It can be observed from these results:

- The combinations of our method and DIM, TIM methods can greatly improve the transferability of adversarial examples, which indicates that our method is orthogonal to these methods.
- Our method outperforms the state-of-the-art attacks across all conducted experiments, i.e., SI-NI-TIM, SI-NI-DIM, and SI-NI-TI-DIM, except for adversarial examples crafted on IncRes-V2 model. Besides, the attack success rates of our method against the adversarial trained models outperform state-of-the-art attacks by large margins.

For the exception that our method does not outperform the state-of-the-art results for adversarial examples crafted on IncRes-V2 model, it may be because the adversarial examples generated by our method underfit the IncRes-V2 model somehow, since the attack success rates for the white-box model IncRes-V2 is only around 95% and 4%–5% lower than the SI-NI-TIM/DIM method. One possible solution for this "underfit" problem is to increase the Iterations T. The results in
Table 4. Comparison of TI-DIM, SI-NI-TI-DIM, and the DA-TI-DIM Extension

| Model     | Attack  | Inc-V3 | Inc-V4 | IncRes-V2 | Res-101 | Inc-V3_{ens3} | Inc-V3_{ens4} | IncRes-V2_{ens2} |
|-----------|---------|--------|--------|-----------|---------|-------------|-------------|-----------------|
|           | TI-DIM  | 98.5*  | 66.1   | 63        | 56.1    | 38.6        | 34.9        | 22.5            |
| Inc-V3    | SI-NI-TI-DIM | 99.6*  | 85.5   | 80.9      | 75.7    | 61.5        | 56.9        | 40.7            |
|           | DA-TI-DIM (Ours) | 99.6*  | 88.3   | 85.1      | 80.3    | 77.4        | 76.8        | 62.9            |
| Inc-V4    | TI-DIM  | 72.5   | 97.8*  | 63.4      | 54.5    | 38.1        | 35.2        | 25.3            |
|           | SI-NI-TI-DIM | 88.1   | 99.3*  | 83.7      | 77      | 65          | 63.1        | 49.4            |
|           | DA-TI-DIM (Ours) | 88.8   | 97.8*  | 83.9      | 78.3    | 75.7        | 75.7        | 68.1            |
| IncRes-V2 | TI-DIM  | 73.2   | 67.5   | 92.4*     | 61.3    | 46.4        | 40.2        | 35.8            |
|           | SI-NI-TI-DIM | 89.6   | 87     | 99.1*     | 83.9    | 74          | 67.9        | 63.7            |
|           | NS-TI-DIM (Ours) | 84.2   | 83.5   | 94.5*     | 78.3    | 76.1        | 73.1        | 72.8            |
| Res-101   | TI-DIM  | 74.9   | 69.8   | 70.5      | 98.7*   | 52.6        | 49.1        | 37.8            |
|           | SI-NI-TI-DIM | 86.4   | 82.6   | 84.6      | 99*     | 72.6        | 66.8        | 56.4            |
|           | DA-TI-DIM (Ours) | 88.1   | 83.8   | 86.2      | 99.3*   | 82.6        | 82.2        | 76.2            |

The numbers in table denote the attack success rates (%). Adversarial examples are generated based on Inc-V3, Inc-V4, IncRes-V2, and Res-101, respectively, using TI-DIM, SI-NI-TI-DIM, and DA-TI-DIM methods. * denotes the attack success rates under white-box attacks.

Figure 7(c) also indicate that the attack success rates for normal models can be improved a lot by increasing the Iterations T. Besides, we notice that the improvement of combining DA technique and DIM/TIM implemented on different white-box models are different. We think it may be caused by the different degree of non-linearity on the decision boundaries of different white-box models. Intuitively, the higher degree of non-linearity the decision boundary is, the larger improvements of transferability the DA technique can make.

**Visibility.** We visualize five randomly selected pairs of adversarial examples generated by TIM, DIM, SI-NI-FGSM, and DA-MI-FGSM attacks, respectively, and their corresponding clean images in Figure 2. We can see that the adversarial examples generated by our method are similar to those generated by other methods in visibility, and all these adversarial examples are hard to be distinguished from their corresponding clean images by humans.

### 4.3 Ensemble-based Attacks

We also evaluate the performance of our method under ensemble-based attacks. Liu et al. [19] have shown that attacking multiple models simultaneously can generate more transferable adversarial examples. It is because if an adversarial example can attack multiple models successfully, it can more likely attack yet another model successfully.

We follow the ensemble-based attack strategy proposed in Reference [3], which fuses the logit activations of multiple models to generate adversarial examples. In this experiment, we generate adversarial examples by attacking Inc-V3, Inc-V4, IncRes-V2, and Res-101 models simultaneously with equal ensemble weights. In Table 5, we show the attack success rates for DA-DIM, DA-TIM, and DA-TI-DIM attacks and baselines. It shows that our method outperforms these baselines across all experiments. The highest attack success rate is achieved by our DA-TI-DIM attack and the average attack success rates against the three robust models reach 94.6%.

### 4.4 Attacking Other Defense Models

We also study the performance of our method on defense models. We test it against HGD [17], R&P [30], NIPS-r3, FD [20], and ComDefend [10] defense methods. HGD, R&P, and NIPS-r3 were the top 3 defense methods in NIPS 2017 defense competition. FD and ComDefend are recently published defense methods for purifying adversarial perturbations. TI-DIM [4] and SI-NI-TI-DIM attacks [18] are presented as baselines. Adversarial examples are generated based on the ensemble
Fig. 2. Visualization of randomly selected clean images and their corresponding adversarial examples. All examples are generated by TIM, DIM, SI-NI-FGSM, and DA-MI-FGSM attacks, respectively.

Table 5. The Attack Success Rates (%) Against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens3}, Inc-V3\textsubscript{ens4}, and IncRes-V2\textsubscript{ens} Models

| Attack          | Inc-V3\* | Inc-V4\* | IncRes-V2\* | Res-101\* | Inc-V3\textsubscript{ens3} | Inc-V3\textsubscript{ens4} | IncRes-V2\textsubscript{ens} | Average |
|-----------------|----------|----------|-------------|------------|-----------------------------|-----------------------------|-----------------------------|---------|
| DIM             | 99.7     | 99.2     | 98.9        | 98.9       | 66.4                        | 60.9                        | 41.6                        | 56.3    |
| SI-NI-DIM       | 100      | 100      | 100         | 99.9       | 88.2                        | 85.1                        | 69.7                        | 81      |
| **DA-DIM (Ours)** | 99.9     | 99.8     | 99.7        | 99.8       | 91                          | **90.1**                    | 60.9                        | 56.3    |
| TIM             | 99.9     | 99.3     | 99.3        | 99.8       | 71.6                        | 67                          | 53.2                        | 63.9    |
| SI-NI-TIM       | 100      | 100      | 100         | 100        | 93.2                        | 90.1                        | 84.5                        | 89.2    |
| **DA-TIM (Ours)** | 99.8     | 99.8     | 99.2        | 99.6       | **93.4**                    | **92.1**                    | 90.1                        | 89.2    |
| TI-DIM          | 99.6     | 98.8     | 98.8        | 98.9       | 85.2                        | 80.2                        | 73.3                        | 79.5    |
| SI-NI-TI-DIM    | 99.9     | 99.9     | 99.9        | 99.9       | 96                          | 94.3                        | 90.3                        | 93.5    |
| **DA-TI-DIM (Ours)** | 99.8     | 99.8     | 99.6        | 99.6       | **96.2**                    | **94.7**                    | **93**                      | **94.6**|

Adversarial examples are generated based on the ensemble of Inc-V3, Inc-V4, IncRes-V2, and Res-101 models using DIM, SI-NI-DIM, TIM, SI-NI-TIM, TI-DIM, SI-NI-TI-DIM, DA-DIM, DA-TIM, and DA-TI-DIM attacks, respectively. Average column denotes the averaged attack success rates against the three robust models. * denotes the white-box model being attacked.

of Inc-V3, Inc-V4, IncRes-V2, and Res-101 models. The attack success rates against FD and ComDefend defense are based on IncRes-V2\textsubscript{ens} model.

As shown in Table 6, our model achieves state-of-the-art results and reaches 94.8% for averaged attack success rates, which indicates current defense methods are not safe to transferable adversarial attacks.
Table 6. The Attack Success Rates Against the Five Advanced Defense Models

| Attack          | HGD | R&P | NIPS-r3 | FD  | ComDefend | Average |
|-----------------|-----|-----|---------|-----|-----------|---------|
| TI-DIM          | 84.8| 75.3| 80.7    | 84.2| 79.6      | 80.9    |
| SI-NI-TI-DIM    | 96.1| 91.3| 94.4    | 93.7| 91.9      | 93.5    |
| DA-TI-DIM (Ours)| 96.1| 93.6| 94.8    | 94.4| 94.3      | 94.8    |

Fig. 3. The cosine similarity of adversarial perturbations generated from Inc-V3, Inc-V4, IncRes-V2, and Res-101 models.

4.5 Similarity of Adversarial Perturbations

To further understand the proposed Direction-aggregated attack, we plot the cosine similarity of adversarial perturbations generated from multiple white-box models, i.e., Inc-V3, Inc-V4, IncRes-V2, and Res-101 models. The results are shown in Figure 3.

In Figure 3, the cosine similarity of adversarial perturbations generated by the propose Direction-aggregated attack is generally higher than other baseline attacks. It is in line with our expectation, since the aggregated direction could reduce the oscillation of each update direction in generating adversarial perturbations. Besides, we notice that the cosine similarity of adversarial perturbations on DA-FGSM is not significantly higher than FGSM. We conjecture that it is due to the adversarial perturbations generated by FGSM “underfit” the white-box model, which limits the similarity of adversarial perturbations.

4.6 Parameter Analysis

In this section, we conduct a series of experiments to study the impact of different hyper-parameters on the transferability of adversarial examples.

Sampling Times $N$. We explore the influence of sampling times $N$ upon the transferability of adversarial examples. Figure 4 shows the attack success rates (%) against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$, and IncRes-V2$_{ens}$ models under black-box settings. The generation of adversarial examples is based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models, respectively, with standard deviation $\sigma$ setting as 0.05.

From Figure 4, we can see that the attack success rates are growing with the increase of sampling times. In detail, the curve is growing fast when sampling times $N$ is less than 30 and the trend of growth tends to be flattening when sampling times $N$ is greater than 30. Besides, the growing
The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3<sub>ens3</sub>, Inc-V3<sub>ens4</sub>, and IncRes-V2<sub>ens</sub> models when varying sampling times N ranging from 10 to 50. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, by DA-MI-FGSM attack.

The trends of Figures 4(a)–4(d) are similar, which indicates that the influence of sampling times N on the transferability is little sensitive to the white-box model.

**σ in Gaussian Distribution.** Standard deviation σ controls the shape of Gaussian distribution and plays an important role in Gaussian noise generation. We study the influence of σ upon the transferability of adversarial examples. Figure 5 shows the attack success rates against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3<sub>ens3</sub>, Inc-V3<sub>ens4</sub>, and IncRes-V2<sub>ens</sub> models under black-box attacks. Adversarial examples in this experiment are crafted based on Inc-V3, Inc-V4, IncRes-V2, and Res-101 models, respectively, with sampling times N = 30.

From Figure 5, we can see that the attack success rates have a surge increasing at first, then the growing trends tend to be flattening. The surge increasing of the attack success rates indicates that the parameter σ plays an important role in our method. Besides, the similar trends among Figures 5(a)–5(d) indicate that the influence of σ on achieving transferability is insensitive to the white-box model.

It is deserved to note that a very large σ is not encouraged for our method for the two reasons: (1) a larger σ indicates larger perturbation size will be generated (Figure 1), thus more sampling times are needed to cover the sampling region; (2) noise sampling from a very large σ might already be too large to flip the prediction and consequently disturb the attack direction.

**Perturbation Size ε.** We study the impact of perturbation size ε on the attack success rates. We set sampling times N and standard deviation σ to 30 and 0.05, respectively. We fix step size.
The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens3}, Inc-V3\textsubscript{ens4}, and IncRes-V2\textsubscript{ens} models when varying $\sigma$ from 0 to 0.09. The adversarial examples are generated based on Inc-V3 (Figure 5(a)), Inc-V4 (Figure 5(b)), IncRes-V2 (Figure 5(c)), and Res-101 (Figure 5(d)) models, respectively, using DA-MI-FGSM attack.

From Figure 6, we observe that the attack success rates increase steadily as perturbation size $\epsilon$ increases on both adversarial trained models and normal trained models.

**Iterations $T$.** We study the impact of iterations $T$ on the transferability of adversarial examples. Similarly, we set sampling times $N$ and standard deviation $\sigma$ to 30 and 0.05, respectively. We fix perturbation size $\epsilon$ to 16 and step size $\alpha$ to $\frac{16}{10}$. We generate adversarial examples based on normal trained models. Then these adversarial examples are tested on the other models under black-box settings. The total iterations $T$ varies from 5 to 22 and the results are shown in Figure 7.

From Figure 7, we can see that the attack success rates are growing significantly when $T$ is less than 10. However, the attack success rates start to be flattening/slightly growing on the normal trained models and slightly decrease on the adversarial trained models after $T$ is greater than 10. It is worthy to note that the perturbation size reaches the maximum perturbation size, because the $\alpha$ is set to $\frac{16}{10}$, which could be the reason why the trends start to be flattening after $T = 10$. Besides, we conjecture that the adversarial examples overfit to the white-box model to some extent when $T$ is greater than 10, which decreases its transferability. A similar phenomenon can be found on $I-\text{FGSM}$ attack, in which the adoption of multiple iterations decreases its transferability. A possible reason
Fig. 6. The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3ens3, Inc-V4ens4, and IncRes-V2ens models when varying $\epsilon$ from 10 to 16. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, using DA-MI-FGSM attack.

for the steady/slightly increase on the normal trained models when $T > 10$ is that the decision boundary of the white-box model is more similar to that of the normal trained models than that of the adversarial trained models.

**Step size $\alpha$.** We study the impact of step size $\alpha$ on the transferability of adversarial examples. Similarly, we set sampling times $N$ and standard deviation $\sigma$ to 30 and 0.05, respectively. We fix perturbation size $\epsilon$ to 16 and iterations $T$ to 16. We generate adversarial examples based on normal trained models. Then, we test these adversarial examples on the other models under black-box settings. The step size $\alpha$ varies from 16/8 to 16/16 and the results are shown in Figure 8.

From Figure 8, it can be seen that the attack success rates are consistently increasing with the decrease of $\alpha$ on the adversarial trained models while keep a flat/slightly decreasing trend on normal trained models. The reason for the different trends between normal trained models and adversarial trained models might be because the correctly classified samples by normal trained models are very difficult to conduct the transferable attack. To show the evidence for our conjecture, we provide a ratio metric to indicate the percentage of the samples correctly classified by normal models are also correctly classified by the adversarial trained model. We denote $S_{IncV3} = \{x \in D^* | f_{\theta}^{IncV3}(x) = y\}$ where the mark IncV3 denotes the name of the model. The ratio is formulated as follows:

$$\text{Ratio} = \frac{|S_{IncV3} \cup S_{IncV4} \cup S_{IncResV2} \cup S_{Res101} \cap S_{\text{robust}}|}{|S_{IncV3} \cup S_{IncV4} \cup S_{IncResV2} \cup S_{Res101}|},$$

(21)

where the mark robust denotes the surrogate name of adversarial trained models.
From Figure 9, we can see that around 90% or more than 90% of the samples that correctly classified by normal trained models are also correctly classified by adversarial trained models. It implies that these samples are difficult to be transferred to attack the black-box models. Therefore, the transferability of these samples improved by reducing $\alpha$ may not be enough to attack the black-box models successfully.

## 5 CONNECTION OF THE DA-ATTACK TO A SMOOTHED CLASSIFIER

Our method mitigates the overfitting problem by aggregating the attack directions of a set of examples around the input $x$, which is different with DIM, TIM, and SI-NI-FGSM attacks. Essentially, these methods are based on geometric transformations of the inputs, e.g., scale and translation. The successful boosting of the performance of combinations of PA-Attack with DIM or TIM (Tables 2, 3, and 4) also provides the evidence that our method is orthogonal to these attacks.

For a better understanding of our method, we provide an analyse of connection of DA-Attack to a smoothed classifier. A reasonable assumption is that adversarial examples generated by a non-smoothed classifier are more easily overfitted than that generated by a smoothed classifier. We take the Gaussian noise smoothed classifier as an example. Formally, given a Gaussian function $g(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp{-\frac{t^2}{2\sigma^2}}$, the Gaussian noise smoothed classifier can be presented as

---

Fig. 7. The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$ and IncRes-V2$_{ens}$ models when varying $T$ from 5 to 22. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, using DA-MI-FGSM attack.
Fig. 8. The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens}, Inc-V4\textsubscript{ens}, and IncRes-V2\textsubscript{ens} models when varying $\alpha$ from $\frac{16}{8}$ to $\frac{16}{16}$. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, using DA-MI-FGSM attack.

The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens3}, Inc-V3\textsubscript{ens4} and IncRes-V2\textsubscript{ens} models when varying $\alpha$ from $\frac{16}{8}$ to $\frac{16}{16}$. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, using DA-MI-FGSM attack.

The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3\textsubscript{ens3}, Inc-V3\textsubscript{ens4} and IncRes-V2\textsubscript{ens} models when varying $\alpha$ from $\frac{16}{8}$ to $\frac{16}{16}$. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models, respectively, using DA-MI-FGSM attack.

In practice, the Equation (22) can be empirical estimated by Monte Carlo sampling. That is, $\Phi(f)(x) = \sum_{i=1}^{N} \sum_{x} g(x + \epsilon_i) f(x + \epsilon_i)$. Accordingly, the gradient of $\Phi(f)(x)$ can be presented as follows:

$$\nabla_x \Phi(f)(x) = \frac{1}{N} \sum_{i=1}^{N} \nabla_x f(x + \epsilon_i), \epsilon_i \in \mathcal{N}(0, \sigma^2 I).$$  \hspace{1cm} (23)$$

Comparing Equation (23) with Equation (10), it can be observed that when we use the gradient instead of the projected gradient as the update direction, i.e., drop the sign function in Equation (10), Equation (10) will be equivalent to Equation (23). Therefore, our method will be degraded to generate adversarial examples by a smoothed classifier when we use the gradient as the attack direction directly, which also implies that DA-Attack can mitigate the overfitting issue of adversarial examples.

Actually, the smoothed classifier also could be smoothed by other noise, e.g., Uniform noise, where $g(t)$ is replaced with the uniform distribution function. Similarly, Gaussian noise is not the only choice for our DA-Attack. Uniform noise is applicable too. To provide empirical evidence...
Fig. 9. The percentage of the samples that are correctly classified by both normal models and the adversarial trained model. Adversarial examples generated by different models are shown in panels (a–d), respectively.

for this, we conduct further experiments by replacing Gaussian noise with Uniform noise (Equation (12)) sampled from $U(-0.08, 0.08)$. Other hyper-parameters are set the same as in the preceding experiments (Section 4). The results are shown in Figure 10, from which we observe that DA-Attack with Uniform noise reaches the same performance as when Gaussian noise was added. This experiment illustrates that the choice of the type of perturbations is not the key factor for DA-Attack.

6 CONCLUSION

In this article, we propose to improve the transferability of adversarial examples by aggregating attack directions of a set of examples around the neighborhood of the input. Our proposed DA-Attack makes use of such aggregated direction. Our extensive experiments on ImageNet with single model attacks and ensemble-based attacks show that our method outperforms state-of-the-art attacks. This result is consistent across all experiments except for the experiments made on IncRes-V2 model. The best averaged attack success rate of our method reaches 94.6% against three adversarial trained models and 94.8% against five defense methods under black-box attacks. Our results also reveal current defense models are not safe to transferable adversarial attacks, and therefore, new defense mechanisms are needed.

We outline several potential approaches for defending against transferable adversarial examples. The essence of existing transferable adversarial examples is that the decision boundaries of the trained models are similar. Therefore, one simple defense approach is to train ensemble models with diversified decision boundaries in order that the decision boundary of each base model is less similar with that of the white-box model. Another way is to use transferable adversarial examples as training instances, i.e., simply adding them to the training data. This idea is similar to adversarial
Fig. 10. The attack success rates (%) of black-box attack against Inc-V3, Inc-V4, IncRes-V2, Res-101, Inc-V3_{ens3}, Inc-V3_{ens4}, and IncRes-V2_{ens} models. The adversarial examples are generated based on Inc-V3 (a), Inc-V4 (b), IncRes-V2 (c), and Res-101 (d) models using DA-MI-FGSM attack with Gaussian noise and Uniform noise, respectively.

training. The challenge here, however, is how to generate the on-the-fly transferable adversarial examples efficiently.

REFERENCES

[1] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP’17). IEEE, 39–57.

[2] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In Proceedings of the 36th International Conference on Machine Learning, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1310–1320. Retrieved from http://proceedings.mlr.press/v97/cohen19c.html.

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. 2018. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 9185–9193.

[4] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4312–4321.

[5] Gamaleldin F. Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification. In Advances in Neural Information Processing Systems, Vol. 2018. Neural Information Processing Systems Foundation, 842–852.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2014.81 arXiv:1311.2524.

[7] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. 2018. Countering adversarial images using input transformations. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=Syl7CIWCb.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 770–778.

[9] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. 2019. Feature space perturbations yield more transferable adversarial examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 7066–7074.

[10] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. 2019. Comdefend: An efficient image compression model to defend adversarial examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 6084–6092.

[11] Alex Krizhevsky and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. Neural Information Processing Systems. https://doi.org/10.1016/j.procyt.2014.09.007 ArXiv:1102.0183.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. Retrieved from https://arXiv:1607.02533.

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/pdf?id=BJm4T4Kgx.

[14] J. Shlens, I. Goodfellow, and C. Szegedy. 2015. explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=nLAxjsnlDzg.

[15] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. 2019. Certified adversarial robustness with additive gaussian noise. In Proceedings of the Conference on Neural Information Processing Systems (Neurips’19). Retrieved from https://proceedings.neurips.cc/paper/2019/file/335cd1b90bfa4ec70b39d08a4ae0cf2d-Paper.pdf.

[16] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. 2020. Learning transferable adversarial examples via ghost networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34. 11458–11465.

[17] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1778–1787.

[18] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. 2020. Nesterov accelerated gradient and scale invariance for adversarial attacks. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=SJHlwkBvDH.

[19] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into transferable adversarial examples and black-box attacks. In Proceedings of the International Conference on Learning Representations. Retrieved from https://arXiv:1611.02770.

[20] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. 2019. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR’19). IEEE, 860–868.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’15). https://doi.org/10.1109/CVPR.2015.7298965 ArXiv:1411.4038.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations. Retrieved from https://arxiv.org/pdf/1706.06083.pdf.

[23] SeyedMohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2016. IEEE Computer Society, 2574–2582. https://doi.org/10.1109/CVPR.2016.282.

[24] Naseer Muzammal, Khan Salman, Khan Muhammad Haris, Shahbaz Khan Fahad, and Fatih Porikli. 2019. Cross-Domain transferability of adversarial perturbations. In Proceedings of the 33rd Conference on Neural information Processing Systems (NeurIPS’19).

[25] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the 31st AAAI Conference on Artificial Intelligence.

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2818–2826.

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations. Retrieved from http://arxiv.org/abs/1312.6199.

[28] Florian Tramér, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2018. Ensemble adversarial training: Attacks and defenses. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=rkZvS-ZR.

[29] Lei Wu and Zhanxing Zhu. 2020. Towards understanding and improving the transferability of adversarial examples in deep neural networks. In Proceedings of the Asian Conference on Machine Learning. PMLR, 837–850.
[30] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. In Proceedings of the International Conference on Learning Representations. Retrieved from https://openreview.net/forum?id=Sk9yuql0Z.

[31] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. 2019. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2730–2739.

[32] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. 2018. Transferable adversarial perturbations. In Proceedings of the European Conference on Computer Vision (ECCV'18). 452–467.

Received October 2020; revised November 2021; accepted November 2021