Improving the Transferability of Adversarial Samples with Adversarial Transformations

Weibin Wu, Yuxin Su, Michael R. Lyu, Irwin King
Department of Computer Science and Engineering, The Chinese University of Hong Kong
{wbwu, yxsu, lyu, king}@cse.cuhk.edu.hk

Abstract

Although deep neural networks (DNNs) have achieved tremendous performance in diverse vision challenges, they are surprisingly susceptible to adversarial examples, which are born of intentionally perturbing benign samples in a human-imperceptible fashion. It thus poses security concerns on the deployment of DNNs in practice, particularly in safety- and security-sensitive domains. To investigate the robustness of DNNs, transfer-based attacks have attracted a growing interest recently due to their high practical applicability, where attackers craft adversarial samples with local models and employ the resultant samples to attack a remote black-box model. However, existing transfer-based attacks frequently suffer from low success rates due to overfitting to the adopted local model. To boost the transferability of adversarial samples, we propose to improve the robustness of synthesized adversarial samples via adversarial transformations. Specifically, we employ an adversarial transformation network to model the most harmful distortions that can destroy adversarial noises and require the synthesized adversarial samples to become resistant to such adversarial transformations. Extensive experiments on the ImageNet benchmark showcase the superiority of our method to state-of-the-art baselines in attacking both undefended and defended models.

1. Introduction

Deep neural networks (DNNs) have emerged as state-of-the-art solutions to a dizzying array of challenging vision tasks [35, 22]. Despite their astonishing performance, DNNs are surprisingly vulnerable to adversarial samples, which are crafted by purposely attaching human-imperceptible noises to legitimate images and can mislead DNNs into wrong predictions [34, 38]. It poses a severe threat to the security of DNN-based systems, especially in safety- and security-critical domains like self-driving [26, 39, 43]. Therefore, learning how to synthesize adversarial samples can serve as a crucial surrogate to evaluate the robustness of DNN-based systems before deployment [9] and spur the development of effective defenses [18, 36].

Figure 1: From left to right: An example of the clean image, the resultant image distorted by our adversarial transformation network, and the corresponding adversarial image generated by our method.

There are generally two lines of adversarial attacks studied in the literature [2]. One focuses on the white-box setting, where the attackers possess perfect knowledge about the target model [9, 17, 25]. The other considers the black-box setting, where attackers do not know the specifics of the target model, such as its architecture and parameters [28, 10]. Compared to the white-box counterpart, black-box attacks are recognized as a more realistic threat to DNN-based systems in practice [28]. Besides, among existing black-box attacks, transfer-based attacks have gained increasing interest recently due to their high practical applicability, where attackers craft adversarial samples based on local source models and directly harness the resultant adversarial examples to fool the remote black-box victims [5, 37].

However, existing transfer-based attacks frequently manifest limited transferability due to overfitting to the employed source model [9, 5, 44]. Concretely, although the generated adversarial samples can fool the source model with high success rates, they can hardly remain malicious to a different target model. Inspired by the data augmentation strategy [12, 16, 31], prior efforts have endeavored to improve the transferability of adversarial samples by training

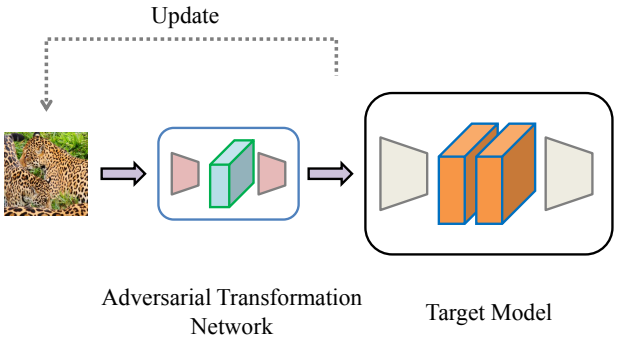

Figure 2: The diagram of our attack strategy. We proceed by first training an adversarial transformation network that can characterize the most harmful image transformations to adversarial noises. We then manufacture adversarial samples by additionally requiring them to be robust against the adversarial transformation network.

them to become robust against common image transformations, such as resizing [42], translation [6], and scaling [21]. Unfortunately, these works explicitly model the applied image distortions by employing only individual image transformations or their simple combination under a fixed distortion magnitude. Therefore, it makes the generated adversarial samples overfit to the applied image transformations and hardly resist against unknown distortions [4], leading to inferior transferability.

To mitigate the issue of poor transferability caused by employing a fixed image transformation, a typical solution is to identify a rich collection of representative image transformations and then carefully tune a combination of them for each image. However, such a strategy can incur prohibitive computational costs. Therefore, we propose to exploit an adversarial transformation network to automate this distortion tuning process, and Figure 1 illustrates an image manipulated by our adversarial transformation network.

Figure 2 depicts the diagram of our Adversarial Transformation-enhanced Transfer Attack (ATTA). Specifically, motivated by the recent advance in applying convolutional neural networks (CNNs) to conduct diverse image manipulation tasks, like digital watermarking [45, 24] and style transfer [7], we propose to train a CNN as the adversarial transformation network by adversarial learning, which can capture the most harmful deformations to adversarial noises. After finishing the learning of the adversarial transformation network, we require the crafted adversarial samples to be able to resist the distortions introduced by the adversarial transformation network. As such, we can make the generated adversarial samples more robust and improve their transferability.

In summary, we would like to highlight the following contributions of this work:

- We propose a novel technique to improve the transferability of adversarial samples with adversarial transformations.
- We conduct extensive experiments on the ImageNet benchmark to evaluate our approach. Experimental results confirm the superiority of our method over state-of-the-art baselines in attacking both undefended and defended models.
- We show that our technology is generally complementary to other state-of-the-art schemes, suggesting it as a general strategy to boost adversarial transferability.

2. Related Work

We focus on deep image classifiers in this work. Therefore, in this section, we briefly review two lines of prior arts that are closely related to our work: synthesizing adversarial samples and defending against adversarial samples.

2.1. Synthesizing Adversarial Samples

According to the adopted threat model, there are two sorts of attacks explored in the literature to craft adversarial examples [2]. The first one assumes the white-box setting, where the target model acts as a local model, and attackers possess perfect knowledge about the target model [9]. The second one considers the black-box scenario, where the target model represents a remote model, and attackers are not informed of the particulars of the target model, such as its structures and parameters [5]. In practice, the black-box assumption can more faithfully characterize the threat to DNN-based systems [28]. Therefore, we also adopt a black-box setup in this work.

There are generally two bodies of adversarial attacks tailored for the black-box setting [13]: query-based and transfer-based attacks. Query-based attacks need to query the target model with instances of interest and exploit the feedback information to seek adversarial images [1, 10]. Nevertheless, query-based attacks usually demand excessive queries to spot an adversarial example, which may incur prohibitive query costs and render attacks more detectable [19]. By contrast, in transfer-based attacks, adversaries adopt a local model as the substitute victim to launch attacks, and directly harness the resultant adversarial samples to attack the remote target model [5]. Therefore, transfer-based attacks are grounded on the transferability of adversarial samples, which represents the phenomenon that the adversarial samples generated for a model can remain malicious to a different model. Due to their high practical applicability, transfer-based attacks have attracted unprecedented attention recently [21, 6].

Unfortunately, transfer-based attacks frequently manifest limited success due to overfitting to the employed

source model [5], especially when attacking a defended victim [6, 42, 41]. To boost the transferability of adversarial samples, prevailing solutions usually view the generation of adversarial samples as an optimization problem [21]. From this perspective, they endeavor to migrate the traditional scheme employed to improve the generalization of models to synthesize transferable adversarial samples.

In this vein, prior efforts can be further split into two groups. The first one involves applying more advanced optimization algorithms, like the momentum method and the Nesterov Accelerated Gradient [29, 5, 21, 27]. The second one is inspired by the data augmentation strategy [42, 6, 21]. Specifically, existing works along this line usually require the synthesized adversarial samples to be robust under certain image transformations that can still preserve the image content, such as resizing [42], translation [6], and scaling [21]. However, these approaches bear the deficiency of only considering individual image transformations or their simple combination under fixed distortion strength. It makes the crafted adversarial samples overfit to the applied image transformations and hardly survive under unknown distortions, which may lead to inferior transferability [4, 14, 23].

Therefore, a straightforward remedy would involve first identifying a large corpus of image transformations that can retain the image content. Then it carefully tunes the combination of image transformations that is appropriate to each image. Unfortunately, such a process can incur prohibitive computational costs. Inspired by the recent progress in performing image manipulations with convolutional neural networks [45, 24, 7], we propose to exploit a CNN-based adversarial transformation network to mitigate the issue of explicitly modeling the employed image transformations and automate the tuning process. Specifically, our strategy proceeds by training an adversarial transformation network to model the most harmful image transformations to adversarial noises by adversarial learning. Then we require the generated adversarial samples to additionally defeat the adversarial transformation network, which can improve adversarial transferability.

2.2. Defending against Adversarial Samples

Enormous efforts have been devoted to defending against adversarial samples, which generally fall into two axes. The first one is termed adversarial training, which remains the state-of-the-art defense to date [18, 36]. Adversarial training works by injecting the generated adversarial samples into the training data to retrain the model [9]. Ensemble adversarial training is a refined successor of vanilla adversarial training [36], which employs the adversarial samples synthesized from hold-out models to augment the training data. As such, the adversarially trained models can showcase robustness against transfer-based attacks.

The second line of defenses proceeds by purifying the adversarial samples. Specifically, they pre-process the input images as a potential defense to rectify adversarial perturbations without reducing the classification accuracy on benign images. The state-of-the-art defenses of this kind include applying random resizing and padding [40], a high-level representation guided denoiser [20], randomized smoothing [4], an image compression module [14], and a JPEG-based defensive compression framework [23]. In this paper, we exploit these state-of-the-art defenses to evaluate the effectiveness of our attack against defended models.

3. Method

In this section, we detail our attack technique. We first introduce the task of crafting adversarial samples in Section 3.1. Then in Section 3.2, we elaborate on the proposed adversarial transformation network. Finally, we present our algorithm to generate adversarial samples in Section 3.3.

3.1. Problem Description

Let $\mathbf{x}$ denote a clean image with ground-truth label y. We can regard a deep image classifier as a function $f(\mathbf{x})$, which returns a probability vector, indicating the probabilities of the input belonging to each class. Given a target model f and a clean image $\mathbf{x}$, the task of attackers is to find an adversarial counterpart $\mathbf{x}^{adv}$, which satisfies the following two conditions:

$$\arg\max \ f(\mathbf{x}^{adv}) \neq y, \tag{1}$$

and

$$||\mathbf{x}^{adv} - \mathbf{x}||_p \leq \epsilon. \tag{2}$$

Here the first requirement reflects the attacker's goal of misleading the target model into wrong predictions. The second condition constrains the admissible perturbation budget for the attacker. In practice, the perturbation budget ϵ is usually a fairly small number, which ensures that the alteration to the clean image is human-imperceptible. In this work, we exploit the l_∞ norm to define the visibility of adversarial perturbations, since it is the most widely advocated measurement in the community [9, 21]. Nevertheless, our approach is generally applicable to other norm choices with simple modifications.

We employ $J(f(\mathbf{x}), y)$ to signify the training loss function of the classifier f. As such, attackers can reformulate the task of generating an adversarial sample $\mathbf{x}^{adv}$ as the following optimization problem:

$$\begin{aligned} \max_{\mathbf{x}^{adv}} \quad & J(f(\mathbf{x}^{adv}), y), \\ \text{s.t.} \quad & ||\mathbf{x}^{adv} - \mathbf{x}||_\infty \leq \epsilon. \end{aligned} \tag{3}$$

Here the attackers apply the training loss function $J(f(\mathbf{x}), y)$ as a surrogate for the original attack object function (Eq. (1)).

(a) blur (b) color & blur (c) sharpen

Figure 3: Illustrations of the output images from our adversarial transformation network T. The top row shows the clean input images, while the bottom row enumerates the corresponding images transformed by T. We discover that the learned adversarial transformation network can perform a diverse set of image manipulations, such as blurring and a combination of multiple simple transformations. Best viewed zoomed in on-screen.

In this paper, we endeavor to develop a transfer-based attack, which works by attacking a local white-box model and harnessing the crafted adversarial samples to fool the black-box victim. By escalating the transferability of adversarial samples, we can attack the target black-box model with high success rates.

3.2. Adversarial Transformation Network

We attempt to improve the transferability of adversarial samples by the data augmentation methodology [31]. It works by asking the adversarial samples to be robust against various image transformations, which may eliminate adversarial noises while still preserve the semantic meaning of the image [42]. Since only adopting a fixed transformation may lead to poor generalization to unknown ones, we endeavor to address the issue of explicitly modeling the applied image transformations by figuring out the most harmful image transformations to each adversarial image. We expect that if the generated adversarial samples can resist the toughest image deformations, they can also survive under other weaker distortions [25].

Specifically, let H signify an image transformation function with parameter θ_H, which can be a composition of multiple simple image transformations, such as blurring and coloring. $H(\mathbf{x})$ thus denotes the transformed image given an input sample $\mathbf{x}$. As per Eq. (3), we can formulate the task of searching for the most harmful image transformations to an adversarial image $\mathbf{x}^{adv}$ as the following min-max problem:

$$\begin{aligned} \min_{\theta_H} \max_{\mathbf{x}^{adv}} \quad & J(f(H(\mathbf{x}^{adv})), y), \\ \text{s.t.} \quad & ||\mathbf{x}^{adv} - \mathbf{x}||_\infty \leq \epsilon, \\ & \arg\max \ f(H(\mathbf{x})) = y. \end{aligned} \tag{4}$$

Recall that y is the ground-truth label of the legitimate image $\mathbf{x}$. Here the inner maximization problem corresponds to finding an adversarial image $\mathbf{x}^{adv}$. In contrast, the outer minimization problem accounts for optimizing the transformation parameters to rectify the adversarial image, so that they become no longer malicious. The second constraint ensures that the learned image transformations can maintain the content of the clean image.

A straightforward way to solve the optimization problem of Eq. (4) involves first spotting all candidate image transformations, and then tuning their combinations and distortion strengths for each adversarial image. However, such a process can incur prohibitive computational costs. Motivated by the recent success of deep learning-based image manipulation techniques [45, 24], we propose to train a CNN-based adversarial transformation network to automate the process of tuning the most harmful image transformations to each adversarial image.

Specifically, we relax the optimization problem of Eq. (4) by restricting the hypothesis space of the transformation function H to be some class of convolutional neural networks $T(\mathbf{x}; \theta_T)$ parameterized with θ_T. Therefore, the optimization problem of Eq. (4) now reduces to the task as follows.

$$\begin{aligned} \min_{\theta_T} \max_{\mathbf{x}^{adv}} \quad & J(f(T(\mathbf{x}^{adv})), y), \\ \text{s.t.} \quad & ||\mathbf{x}^{adv} - \mathbf{x}||_\infty \leq \epsilon, \\ & \arg\max \ f(T(\mathbf{x})) = y. \end{aligned} \tag{5}$$

Employing CNN to model the applied transformations affords two-fold merits. The first one is that CNNs possess the capacity to generate a cornucopia of image distortions, as demonstrated in Figure 3. It ensures that although we have reduced the hypothesis space of the transformation function H to be some class of convolutional neural networks, the constrained hypothesis space of the transformation function H is still large enough. Therefore, the solution to the relaxed optimization problem of Eq. (5) is fairly close to the optimal of the original task of Eq. (4). The second virtue is that we can learn the CNN function in an end-to-end fashion, which automates the tuning of the exploited transformations for each adversarial image. Therefore, it is faster and more convenient by circumventing the prohibitive overhead of manually tuning.

To train the CNN-based adversarial transformation network, we resort to the adversarial learning scheme [9, 8] to

Algorithm 1 Adversarial Transformation Network Training

Require: The fooling object function L_{fool}, the training loss function L_T of the adversarial transformation network, and a clean image $\mathbf{x}$

Require: The perturbation budget ϵ, the iteration numbers K_{outer} and K_{inner}

Initialize $\mathbf{x}^{adv} = \mathbf{x}$
Randomly initialize θ_T
for $k_{outer} = 1$ to K_{outer} **do**
  for $k_{inner} = 1$ to K_{inner} **do**
    Update $\mathbf{x}^{adv} = \mathbf{x}^{adv} - \mathrm{Adam}(L_{fool})$
    Clip $\mathbf{x}^{adv} = \mathrm{Clip}^{\epsilon}_{\mathbf{x}}\{\mathbf{x}^{adv}\}$
  end for
  Update $\theta_T = \theta_T - \mathrm{Adam}(L_T)$
end for
return the parameter θ_T of the learned adversarial transformation network

Algorithm 2 Adversarial Sample Generation

Require: A classifier f, the attack object function L_{attack}, the adversarial transformation network T, a clean image $\mathbf{x}$, and its ground-truth label y

Require: The perturbation budget ϵ and iteration number K

Ensure: $||\mathbf{x}^{adv} - \mathbf{x}||_\infty \leq \epsilon$

$\epsilon' = \frac{\epsilon}{K}$
$\mathbf{x}^{adv}_0 = \mathbf{x}$
for $k = 0$ to $K - 1$ **do**
  $\mathbf{x}^{adv}_{k+1} = \mathrm{Clip}^{\epsilon}_{\mathbf{x}}\{\mathbf{x}^{adv}_k + \epsilon' \ \mathrm{sign}(\frac{\partial L_{attack}}{\partial \mathbf{x}})\}$
end for
return $\mathbf{x}^{adv} = \mathbf{x}^{adv}_K$

solve the optimization problem of Eq. (5). Specifically, we first define the following training loss function of the adversarial transformation network:

$$\begin{aligned} L_T =& J(f(T(\mathbf{x}^{adv})), y) + \alpha_1 J(f(T(\mathbf{x})), y) \\ &+ \alpha_2 ||\mathbf{x}^{adv} - T(\mathbf{x}^{adv})||^2. \end{aligned} \tag{6}$$

Here the first term reflects the adversarial transformation network's pursuit of counteracting the adversarial noises, namely, rendering the adversarial sample no longer destructive to the target image classifier after the pre-processing of the adversarial transformation network. In contrast, the second term requires the adversarial transformation network to retain the content of the clean image, so that it will not incur misclassification of the target model on distorted legitimate images. The last term constrains the distortion strength introduced by the adversarial transformation network. It serves as a regularizer to alleviate the overfitting issue during the training of the adversarial transformation network. In this work, we employ the l_2 norm to formulate the transformation magnitude for simplicity. Nonetheless, we can also adopt other semantic measurements, like the distance calculated on the feature space of a pre-trained deep model [31]. α_1 and α_2 are the scalar weights to balance the contributions of each term in Eq. (6).

For the inner maximization problem of Eq. (5), we propose the following fooling object function L_{fool} to search for the adversarial instance $\mathbf{x}^{adv}$:

$$L_{fool} = -J(f(T(\mathbf{x}^{adv})), y) - \beta J(f(\mathbf{x}^{adv}), y). \tag{7}$$

Here the second term exploits the training loss function of the target model as the surrogate to seek an adversarial example $\mathbf{x}^{adv}$. Moreover, the first term takes into account the deformation induced by the adversarial transformation network, and endeavors to make the adversarial example remain malicious under the adversarial transformation network. β is the scalar weight to control the strength of each term in Eq. (7).

The above definitions of the outer and inner training loss functions lead us to an end-to-end training algorithm of the adversarial transformation network, which is detailed in Algorithm 1. In short, we alternate the searching for the adversarial example and the training of the adversarial transformation network, which amount to the optimization of the inner maximization problem and the outer minimization task of Eq. (5), respectively. Here we employ an Adam optimizer [15] to compute the updating value ($\mathrm{Adam}(\cdot)$) in each iteration. Additionally, we apply the function $\mathrm{Clip}^{\epsilon}_{\mathbf{x}}$ to clip the resultant adversarial sample to be within the ϵ-neighborhood of the source image $\mathbf{x}$ in the l_∞ space. Therefore, we can satisfy the norm constraint for the adversarial sample in Eq. (5).

3.3. Adversarial Sample Generation

After finishing the training of the adversarial transformation network, we can view the learned adversarial transformation network as a pre-processing module, and attach it to the target image classification model, as depicted in Figure 2. As a result, we can regard the cascaded adversarial transformation network and image classifier as another victim model to attack. Therefore, we define the following attack object function for the attackers:

$$L_{attack} = J(f(\mathbf{x}^{adv}), y) + \gamma J(f(T(\mathbf{x}^{adv})), y), \tag{8}$$

where γ is the scalar weight to trade-off the contributions of each term in Eq. (8).

To resolve the optimization problem of Eq. (8), we can now turn to any backbone optimization algorithm to find an approximate solution. In this paper, we apply the basic iterative method [17], since it is simple and efficient. Algo-

	Attack	Res-v2	Inc-v3	Inc-v4	IncRes-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{adv}$
Res-v2	FGSM	85.4	43.7	35.2	33.2	22.6	22.2	14.3
	BIM	95.6	46.8	38.0	36.2	27.6	25.3	17.4
	DIM	97.9	66.3	57.2	55.6	30.5	29.6	20.8
	TIM	98.8	65.2	59.8	57.4	35.6	31.7	25.8
	SIM	98.8	**67.3**	57.4	57.4	38.1	30.1	26.7
	MI-FGSM	98.2	57.9	53.9	49.4	33.0	29.2	21.8
	NI-FGSM	98.6	62.2	55.5	53.3	33.1	28.9	21.1
	ATTA (Ours)	**99.8**	64.3	**61.8**	**59.2**	**42.1**	**38.9**	**29.1**
Inc-v3	FGSM	34.3	72.8	29.8	27.1	14.9	13.6	17.9
	BIM	33.2	99.9	32.3	29.8	11.8	11.5	17.6
	DIM	39.2	**100**	39.2	37.6	23.2	24.3	14.0
	TIM	39.2	**100**	44.3	45.8	23.2	24.9	16.4
	SIM	40.1	**100**	42.9	46.4	22.8	24.3	16.9
	MI-FGSM	36.2	**100**	44.4	42.7	22.5	22.4	16.5
	NI-FGSM	38.0	**100**	47.4	46.4	23.2	22.4	16.4
	ATTA (Ours)	**44.8**	**100**	**52.9**	**53.2**	**25.1**	**27.9**	**18.8**
Inc-v4	FGSM	31.7	32.9	49.7	28.2	11.9	13.1	6.2
	BIM	37.9	59.1	99.1	30.9	14.7	14.7	7.1
	DIM	40.8	64.3	**99.6**	39.4	24.6	24.8	15.2
	TIM	41.4	64.3	**99.6**	48.2	25.7	25.2	16.9
	SIM	41.4	61.9	**99.6**	49.7	27.9	25.2	17.4
	MI-FGSM	40.1	58.8	**99.6**	44.4	27.0	25.1	18.1
	NI-FGSM	42.9	62.4	**99.6**	51.8	25.4	24.1	17.6
	ATTA (Ours)	**43.8**	**66.8**	**99.6**	**59.2**	**32.1**	**29.2**	**20.8**
IncRes-v2	FGSM	29.3	31.0	23.5	42.8	13.1	12.7	7.3
	BIM	39.6	58.5	23.5	42.8	15.2	13.1	7.1
	DIM	41.3	63.4	58.3	97.7	30.7	29.2	19.8
	TIM	43.1	62.9	55.4	**98.9**	31.8	29.2	20.6
	SIM	42.1	60.9	52.7	**98.9**	29.6	29.2	20.9
	MI-FGSM	39.9	56.8	48.6	97.7	19.6	26.0	21.7
	NI-FGSM	39.7	59.1	51.2	**98.9**	25.6	25.2	20.6
	ATTA (Ours)	**44.8**	**68.9**	**65.2**	**98.9**	**33.0**	**31.9**	**24.3**

Table 1: Success rates (%) of different attacks against seven models. The first column lists the source model adopted to craft adversarial samples, while the first row shows the target model.

rithm 2 elaborates on our procedure to synthesize adversarial samples.

4. Experiments

In this section, we conduct experiments to evaluate the effectiveness of our approach. We first state the experimental setup in Section 4.1. Then in Section 4.2, we offer the results of our attacks against both cutting-edge undefended and defended models. We follow by an in-depth investigation of our approach in Section 4.3. We finally verify the complementary effect of our strategy on other compatible state-of-the-art approaches in Section 4.4.

4.1. Experimental Setup

We center on attacking image classifiers trained on the ImageNet dataset [30], which is the most widely recognized benchmark task for transfer-based attacks [6, 3]. We follow the protocol of the state-of-the-art baseline [21] to set up the experiments, which we detail as follows.

Dataset. We employ the ILSVRC 2012 training partition [30] as the development set to develop our attack, where we train the adversarial transformation network and fine-tune the hyper-parameters. For the test data adopted to evaluate our method, we randomly sample 1000 images of different categories from the ILSVRC 2012 validation set [30]. We also ensure that nearly all of the selected test images can be correctly classified by every model exploited in this paper.

Target model. We attack both undefended and defended models. For undefended models, we consider multiple top-performance models with diversified architectures, incorporating ResNet v2 (Res-v2) [11, 12], Inception v3 (Inc-v3) [33], Inception v4 (Inc-v4) [32], and Inception-ResNet v2 (IncRes-v2) [32]. For defended models, we focus on several cutting-edge adversarially trained models, since adversarial training is arguably the most effective and promising

Attack	HGD	R&P	NIPS-r3	FD	ComDefend	RS	Average
FGSM	8.9	16.8	23.1	19.2	13.4	6.8	14.7
BIM	12.1	19.3	23.8	21.8	17.2	8.9	17.2
DIM	79.5	74.7	81.9	76.4	72.3	42.3	71.2
TIM	73.3	69.8	79.4	78.2	69.2	36.2	67.7
SIM	76.2	77.7	84.2	79.8	75.4	39.3	72.1
MI-FGSM	33.4	27.2	42.1	47.3	42.8	29.9	37.1
NI-FGSM	35.2	30.3	40.8	49.2	44.9	32.3	38.8
ATTA (Ours)	**85.9**	**83.2**	**89.5**	**84.4**	**79.9**	**47.4**	**78.4**

Table 2: Success rates (%) of different attacks against advanced defense methods.

Structure	Res-v2	Inc-v3	Inc-v4	IncRes-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{adv}$
Conv (4, 3)	39.7	100	42.8	44.9	19.3	19.5	16.2
Conv (16, 3)	**44.8**	**100**	**52.9**	**53.2**	**25.1**	**27.9**	**18.8**
Conv (32, 3)	34.8	100	31.6	32.2	15.9	13.6	16.6
Conv (32, 32, 3)	33.8	100	34.1	31.3	12.3	11.9	17.9

Table 3: Success rates (%) of our attack when varying the complexity of the adversarial transformation network. The first row shows the target model.

defense to date [25]. Specifically, we explore adversarially trained Inception-ResNet v2 (IncRes-v2$_{adv}$), adversarially trained Inception v3 with deceptive samples from an ensemble of three models (Inc-v3$_{ens3}$) and four models (Inc-v3$_{ens4}$), respectively [36, 18].

Furthermore, we study another line of state-of-the-art defenses that aims to rectify adversarial samples. These defenses cover high-level representation guided denoiser (HGD) [20], random resizing and padding (R&P) [40], NIPS-r3[1], feature distillation (FD) [23], compression defense (ComDefend) [14], and randomized smoothing (RS) [4].

Baseline. We compare our approach with two sorts of baselines. The first one represents top-performance white-box attacks that manifest greater transferability than the other white-box techniques [6], including Fast Gradient Sign Method (FGSM) [9] and Basic Iterative Method (BIM) [17]. The second category incorporates state-of-the-art transfer-based attacks, embracing Diverse Input Method (DIM) [42], Translation-Invariant Method (TIM) [6], Scale-Invariant Method (SIM) [21], Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [5], and Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [21]. Similar to us, they also seek to boost the transferability of adversarial samples from the perspective of optimization and generalization, either by employing more advanced optimizers or by data augmentation.

Parameter. For the adversarial transformation network, we adopt a two-layer CNN: $T(\mathbf{x}) = \text{Conv}_{3\times3} \circ \text{Leaky ReLU} \circ \text{Conv}_{16\times3}(\mathbf{x})$, where Conv indicates a convolutional layer with the denotation of $\text{Conv}_{\text{kernel size}\times\text{number}}$.

[1] https://github.com/anlthms/nips-2017/tree/master/mmd

For benchmark attacks, we employ the recommended parameters in their original implementation for fair comparisons. Following [21, 5], we set the perturbation budget $\epsilon = 16$ for all attacks. The iteration numbers K, K_{outer}, and K_{inner} are set to 10. We determine the best hyper-parameters of our algorithm with grid search on the development set. The weight parameters are 1.0, 10, 1.0, and 1.0 for α_1, α_2, β, and γ, respectively.

4.2. Attacking Results

Here we assess the performance of our attacks against both undefended and defended models. Specifically, for a given source model, we mount attacks on it and directly apply the result adversarial samples to fool the other different models, which amounts to the black-box setting. We also test the attacking results on the source model itself, which corresponds to the white-box setting.

Table 1 reports the attacking performance of different methods against both undefended and adversarially trained models. Our attack achieves nearly **100%** success rates under the white-box scenarios. More importantly, we can see that under the black-box settings, our technique can drastically improve the transferability of BIM. For instance, when applying Inc-v3 as the source model, our attacking performance exceeds BIM by over **14.4%** on average. Besides, our attack consistently outperforms all state-of-the-art baselines by a significant margin under the black-box settings, which further corroborates the superiority of our strategy on synthesizing transferable adversarial samples.

We also evaluate the success rates of different attacks against other advanced defenses. Table 2 shows the results when adopting Inc-v3 as the source model to attack other models defended with different mechanisms. Our at-

	Attack	Res-v2	Inc-v3	Inc-v4	IncRes-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{adv}$
Res-v2	SI-NI-TI-DIM	**99.8**	78.3	70.2	71.8	34.9	35.9	30.2
	AT-SI-NI-TI-DIM (Ours)	**99.8**	**80.1**	**74.9**	**74.9**	**36.8**	**37.3**	**33.2**
Inc-v3	SI-NI-TI-DIM	48.3	**100**	54.3	56.2	**27.8**	28.1	24.5
	AT-SI-NI-TI-DIM (Ours)	**49.1**	**100**	**55.9**	**57.1**	**27.8**	**28.6**	**24.9**
Inc-v4	SI-NI-TI-DIM	49.5	72.1	**99.6**	60.3	33.2	31.8	26.9
	AT-SI-NI-TI-DIM (Ours)	**50.4**	**75.2**	**99.6**	**62.8**	**33.9**	**32.3**	**27.6**
IncRes-v2	SI-NI-TI-DIM	50.1	72.9	69.6	**98.9**	34.5	32.7	27.4
	AT-SI-NI-TI-DIM (Ours)	**55.3**	**77.8**	**74.2**	**98.9**	**36.5**	**34.9**	**29.1**

Table 4: Attack success rates (%) when combining our strategy with compatible algorithms. The first column lists the source model adopted to craft adversarial samples, while the first row shows the target model.

tacks achieve an average success rate of **78.4%**, defeating all state-of-the-art attacks by a significant margin of over **6.3%**. It further evidences the effectiveness of our attacks against both top-performance undefended and defended models, and raises a new security concern for developing more robust defenses.

4.3. Further Analysis

As shown in Algorithm 2, our attack is built upon BIM by augmenting an adversarial transformation network. Therefore, comparing the performance of BIM and our method in Table 1 and Table 2 constitutes an ablation study. The remarkable advance of our attack over BIM verifies the contribution of the proposed adversarial transformation network.

We then analyze the effect of the complexity of the adversarial transformation network. Specifically, we adjust the structures of the adversarial transformation network and perform attacks as in Section 4.2. We present the results when exploiting Inc-v3 as the source model in Table 3. We indicate the architecture of the adversarial transformation network in the format of Conv $(a, b, ...)$, where we specify the kernel size of each convolutional layer in parentheses. The number of kernels is three across all convolutional layers. From Table 3, we can observe that over simple or sophisticated structures can deteriorate our attack performance, since the former hardly owns enough representation capacity, while the latter can make the adversarial transformation network overfit to the backbone attack algorithm.

4.4. Complementary Effect of Our Technique

In principle, our strategy is compatible with other state-of-the-art transfer-based attacks. Therefore, we can conveniently combine our technique with these attacks. To validate the complementary effect of our technology, we experiment with the state-of-the-art integrated transfer-based attack (SI-NI-TI-DIM) [21], which is a composition of SIM, NI-FGSM, TIM, and DIM. Specifically, to integrate our strategy with SI-NI-TI-DIM, we just need to first regard the cascaded adversarial transformation network and image classifier as another victim model. Then we attack both the cascaded network and the original classifier with SI-NI-TI-DIM. We denote the combination of our ATTA and SI-NI-TI-DIM as AT-SI-NI-TI-DIM.

We conduct similar experiments as in Section 4.2, and Table 4 states the results. We make the following observations. First, our attack (AT-SI-NI-TI-DIM) can attain almost **100%** success rates under the white-box context. Second, our method can consistently promote the success rates of the state-of-the-art baseline by a considerable margin, under all black-box cases. Therefore, it affirms the complementary effect of our technique.

5. Conclusion

In this work, we introduce a novel technique, Adversarial Transformation-enhanced Transfer Attack (ATTA), to boost the transferability of adversarial samples. Inspired by the data augmentation methodology, it features training a CNN-based adversarial transformation network by adversarial learning, and requiring the generated adversarial samples to withstand the adversarial transformation network. Moreover, our strategy can be conveniently combined with other transfer-based attacks to further promote their performance. Extensive experiments corroborate the superiority of our approach on synthesizing transferable adversarial samples against both state-of-the-art undefended and defended models. Therefore, our attack can serve as a strong benchmark to evaluate future defenses.

Acknowledgment

We thank anonymous reviewers for their valuable comments. The work described in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14210920 of the General Research Fund and CUHK 3133150, RIF R5034-18).

References

[1] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision*, pages 158–174. Springer, 2018. 2

[2] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018. 1, 2

[3] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv:1902.06705*, 2019. 6

[4] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320, 2019. 2, 3, 7

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1, 2, 3, 7

[6] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3, 6, 7

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2, 3

[8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 4

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 3, 4, 7

[10] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493, 2019. 1, 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 770–778, 2016. 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *The European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016. 1, 6

[13] Zhichao Huang and Tong Zhang. Black-box adversarial attack with transferable model-based embedding. In *International Conference on Learning Representations*, 2020. 2

[14] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. ComDefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 7

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105. Curran Associates, Inc., 2012. 1

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR Workshop*, 2017. 1, 5, 7

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 3, 7

[19] Yingwei Li, Song Bai, Cihang Xie, Zhenyu Liao, Xiaohui Shen, and Alan L Yuille. Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses. In *European Conference on Computer Vision*, 2020. 2

[20] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. 3, 7

[21] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. 2, 3, 6, 7, 8

[22] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 82–92, 2019. 1

[23] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature Distillation: DNN-Oriented JPEG compression against adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 7

[24] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13548–13557, 2020. 2, 3, 4

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 1, 4, 7

[26] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018. 1

[27] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence O (1/kˆ 2). In *Doklady an ussr*, volume 269, pages 543–547, 1983. 3

[28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 1, 2

[29] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. 3

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. 6

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 4, 5

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *The Thirty-first AAAI Conference on Artificial Intelligence*, 2017. 6

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE International Conference on Computer Vision (ICCV)*, 2016. 6

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[35] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 1

[36] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 7

[37] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1161–1170, 2020. 1

[38] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Towards global explanations of convolutional neural networks with concept attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8652–8661, 2020. 1

[39] Weibin Wu, Hui Xu, Sanqiang Zhong, Michael R Lyu, and Irwin King. Deep validation: Toward detecting real-world corner cases for deep neural networks. In *Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 125–137. IEEE, 2019. 1

[40] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018. 3, 7

[41] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019. 3

[42] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2, 3, 4, 7

[43] Hui Xu, Zhuangbin Chen, Weibin Wu, Zhi Jin, Sy-yen Kuo, and Michael Lyu. NV-DNN: Towards fault-tolerant DNN systems with N-version programming. In *The 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 44–47. IEEE, 2019. 1

[44] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *The European Conference on Computer Vision (ECCV)*, 2018. 1

[45] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018. 2, 3, 4