Universal Adversarial Attack on Attention and the Resulting Dataset DAmageNet

Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang, *Senior Member, IEEE*

Abstract—Adversarial attacks on deep neural networks (DNNs) have been found for several years. However, the existing adversarial attacks have high success rates only when the information of the victim DNN is well-known or could be estimated by the structure similarity or massive queries. In this paper, we propose to *Attack on Attention* (AoA), a semantic property commonly shared by DNNs. AoA enjoys a significant increase in transferability when the traditional cross entropy loss is replaced with the attention loss. Since AoA alters the loss function only, it could be easily combined with other transferability-enhancement techniques and then achieve SOTA performance. We apply AoA to generate 50000 adversarial samples from ImageNet validation set to defeat many neural networks, and thus name the dataset as *DAmageNet*. 13 well-trained DNNs are tested on DAmageNet, and all of them have an error rate over 85 percent. Even with defenses or adversarial training, most models still maintain an error rate over 70 percent on DAmageNet. DAmageNet is the first universal adversarial dataset. It could be downloaded freely and serve as a benchmark for robustness testing and adversarial training.

Index Terms—Adversarial attack, attention, transferability, black-box attack, DAmageNet

1 Introduction

Deep neural networks (DNNs) have grown into the mainstream tools in many fields, thus, their vulnerability has attracted much attention in the recent years. An obvious example is the existence of adversarial samples [1], which are quite similar with the clean ones, but are able to cheat the DNNs to produce incorrect predictions in high confidence. Various attack methods to craft adversarial samples have been proposed, such as FGSM [2], C&W [3], PGD [4], Type I [5] and so on. Generally speaking, when the victim network is exposed to the attacker, one can easily achieve efficient attack with a very high success rate.

Although white-box attacks can easily cheat DNNs, the current users actually do not worry about them, since it is almost impossible to get the complete information including the structure and the parameters of the victim DNNs. If the information is kept well, one has to use black-box attack, which can be roughly categorized into query-based approaches [6], [7], [8] and transfer-based approaches [9], [10], [11]. The former one is to estimate the gradient by querying the victim DNNs. However, until now, the existing query-based attacks still need massive queries, which can be easily detected by the defense systems. Transfer-based attacks rely on the similarity between the victim DNN and the attacked DNN, which serves as the *surrogate model* in a black-box attack, in the attacker's hands. It is expected that white-box attacks on the surrogate model can also invade the victim DNN. Although there are some promising studies recently [12], [13], [14], the transfer performance is not satisfactory and a high attack rate could be reached only when two DNNs have similar structures [15], which however conflicts the aim of black-box attacks.

Black-box adversarial samples that are applicable to vast DNNs need to attack their common vulnerability. Since DNNs are imitating human's intelligence, although DNNs have different structures and weights, they may share similar semantic features. In this paper, we are focusing on the attention heat maps, on which different DNNs have similar results. By attacking the heat maps of one white-box DNN, we could make its attention lose focus and therefore fail in judgement. In fact, some works have been aware of the importance of attention and put the change of heat map as an evidence of successful attacks, see, e.g., [11], [16]. But none of them includes the attention in loss. In our study, we develop an *Attack on Attention* (AoA). AoA has a good white-box attack performance. More importantly, there is high similarity in attention across different DNNs, making AoA highly transferable: replacing the cross-entropy loss by AoA loss increases the transferability by 10 to 15 percent. Combined with some existing transferability-enhancement methods, AoA achieves a state-of-the-art performance, e.g., over 85 percent transfer rate on all 12 black-box popular DNNs in numerical experiments.

Here, we first illustrate one example in Fig. 1. The original image is a "salamander" in ImageNet [17]. By attacking the attention, we generate an adversarial sample, which looks very similar to the original one but with a scattered heat map (in the lower left corner), leading to misclassification. The attack is carried out on VGG19 [18] but other well-trained DNNs on ImageNet also make wrong predictions.

• *The authors are with the Department of Automation, and the Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the MOE Key Laboratory of System Control and Information Processing, Shanghai 200240, P.R. China. E-mail: {sizhe.chen, lstefanie, sunchengjin, jieyang, xiaolinhuang}@sjtu.edu.cn.*

Manuscript received 9 June 2020; revised 14 Sept. 2020; accepted 18 Oct. 2020. Date of publication 23 Oct. 2020; date of current version 4 Mar. 2022.
(Corresponding author: Xiaolin Huang.)
Recommended for acceptance by J. Zhou.
Digital Object Identifier no. 10.1109/TPAMI.2020.3033291

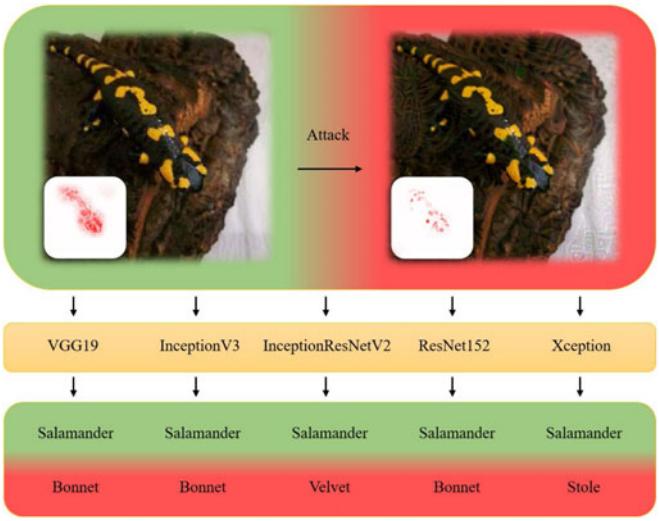

Fig. 1. AoA adversarial sample and its attention heat map (calculated by DenseNet121). The original sample (in ImageNet: image n01629819_15314.JPEG, class No.25) is shown on the left. All well-trained DNNs (listed in the first row) correctly recognize this image as a salamander. The right image is the generated adversarial sample by AoA. The difference between the two images is slight, however, the heat map shown in lower left corner changes a lot, which fools all the listed DNNs to incorrect predictions, as shown in the bottom row.

Since AoA is for common vulnerabilities of DNNs, we successfully generate 50000 adversarial samples that can cheat many DNNs, of which the error rates increase to over 85 percent. We provide these samples in the dataset named as *DAmageNet*. DAmageNet is the first dataset that provides black-box adversarial samples. Those images *DAmage* many neural networks without any knowledge or query. But the aim is not to really damage them, but to point out the weak parts of neural networks and thus those samples are valuable to improve the neural networks by adversarial training [19], [20], robustness certification [21], and so on.

The rest of this paper is organized as follows. In Section 2, we will briefly introduce adversarial attack, especially black-box attack, attention heat map, and several variants of ImageNet. The Attack on Attention is described in detail in Section 3. Section 4 evaluates the proposed AoA along with other attacks and defenses and presents the DAmageNet. In Section 5, a conclusion is given to end this paper.

2 Related Work

2.1 Adversarial Attack and its Defense

Adversarial attacks [22] reveal the weakness of DNNs by cheating it with adversarial samples, which differ from original ones with only a slight perturbation. In the humans' eyes, the adversarial samples do not differ from the original ones, but well-trained networks make false predictions on them in high confidence. The adversarial attack can be expressed as below,

$$\begin{aligned} &\text{find } \Delta x \\ &\text{s.t. } f(x) \neq f(x+\Delta x) \\ &\quad \|\Delta x\| \leq \varepsilon, \end{aligned}$$

where a neural network f predicts differently on the clean sample and the adversarial sample. Even their difference is imperceivable, i.e., Δx is restricted by $||\cdot||$, which could be the ℓ_1-, ℓ_2- or ℓ_∞-norm.

When training a DNN, one updates the weights of the network by the gradients to minimize a training loss. While in adversarial attacks, one alters the image to increase the training loss. Based on this basic idea, there have been many variants on attacking spaces and crafting methods.

For the space to be attacked, most of the existing methods directly conduct attack in the image space [2], [23], [24]. It is also reasonable to attack the feature vector in the latent space [5], [25] or the encoder/decoder [26], [27]. Attack on feature space may produce unique perturbation unlike random noise.

Adversarial attacks could be roughly categorized as gradient-based [2], [4] and optimization-based methods [3], [22]. Gradient-based methods search in the gradient direction and the magnitude of perturbation is restricted to avoid a big distortion. Optimization-based methods usually consider the magnitude restriction in the objective function. For both, the magnitude could be measured by the ℓ_1, ℓ_2, ℓ_∞-norm or other metrics.

To secure the DNN, many defense methods have been proposed to inhibit the adversarial attack. Defense can be achieved by adding adversarial samples to the training set, which is called adversarial training [28], [29], [30]. It is very effective, but consumes several-fold time. Another technique is to design certain blocks in the network structure to prevent attacks or detect adversarial samples [31], [32]. Attack can also be mitigated by preprocessing images before input to the DNN [33], [34], [35], which does not require modification on the pre-trained network.

2.2 Black-Box Attack

When the victim DNNs are totally known, the attacks mentioned above have high success rates. However, it is almost impossible to have access to the victim model in real-world scenarios and thus black-box attacks are required [36], [37], [38]. Black-box attacks rely on either query [6], [7] or transferability [9], [36].

For the query-based approach, the attacker adds a slight perturbation to the input image and observes the reaction of the victim model. By a series of queries, the gradients could be roughly estimated and then one can conduct the attack in the way similar to white-box cases. To decide on the attack direction, attackers adopt methods including Bayes optimization [39], evolutional algorithms [40], meta learning [41] etc. Since the practical DNNs are generally very complicated, good estimation of the gradients needs a massive number of queries, leading to an easy detection by the model owner.

For the transfer-based approach, one conducts white-box attack in a well-designed surrogate model and expects that the adversarial samples remain aggressive to other models. The underlying assumption is that the distance between decision boundaries across different classes is significantly shorter than that across different models [36]. Although a good transfer rate has been recently reported in [12], [13], [14], [42], it is mainly for models in the same family, e.g., InceptionV3 and InceptionV4, or models with the same blocks, e.g., residual blocks [15]. Until now, cross-family transferability of

adversarial samples with small perturbations is limited and there is no publicly available dataset of that.

2.3 Attention Heat Map

In making judgements, humans tend to concentrate on certain parts of an object and allocate attention efficiently. This attention mechanism in human intelligence has been exploited by researchers. In recent studies, methods in natural language process have benefited from the attention mechanism a lot [43]. In computer vision, the same idea has been applied and becomes an important component in DNNs, especially in industrial applications [44].

To attack on attention, we need to calculate the pixel-wise attention heat map, for which network visualization methods [45], [46] are applicable. Forward visualization adopts the intuitive idea to obtain the attention by observing the changes in the output caused by changes in the input. The input can be modified by noise [47], masking [48], or perturbation [49]. However, these methods consume much time and may introduce randomness.

In contrast, backward visualization [48], [50], [51] obtains the heat map by calculating the relevance between adjacent layers from the output to the input. The layer-wise attention is obtained by the attention in the next layer and the network weights in this layer. Significant works include Layer-wise Relevance Propagation (LRP) [52], Contrastive LRP (CLRP) [53] and Softmax Gradient LRP (SGLRP) [54]. These methods extract the high-level semantic attention features for the images from the perspective of the network and make DNNs more interpretable and explainable.

2.4 ImageNet and its Variants

To demonstrate and evaluate our attack, we will modify images from ImageNet as other transfer attacks [12], [13], [14], [42]. ImageNet is a large-scale dataset [17], which contains images of 1000 classes and each has 1300 well-chosen samples. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has encouraged a lot of mile-stone works [18], [55], [56]. Recently, many interesting variants of ImageNet have been developed, including ImageNet-A [57], ObjectNet [58], ImageNet-C, and ImageNet-P [59].

ImageNet-A contains real-world images in ImageNet classes, and they are able to mislead many classifiers to output false predictions. ObjectNet also includes natural images that well-trained models in ImageNet cannot distinguish. Objects in ObjectNet have random backgrounds, rotations and viewpoints. ImageNet-C is produced by adding 15 diverse corruptions. Each type of corruptions has 5 levels from the lightest to the severest. ImageNet-P is designed from ImageNet-C and differs from it in possessing additional perturbation sequences, which are not generated by attack but by image transformations.

The datasets mentioned above are valuable for testing and improving the network generalization capability, but DAmageNet is for the robustness. In other words, samples in the above datasets differ from the samples in ImageNet and the low accuracy is due to the poor generalization. In DAmageNet, the samples are quite similar to the original ones in ImageNet and the low accuracy is due to the over-sensitivity of DNNs.

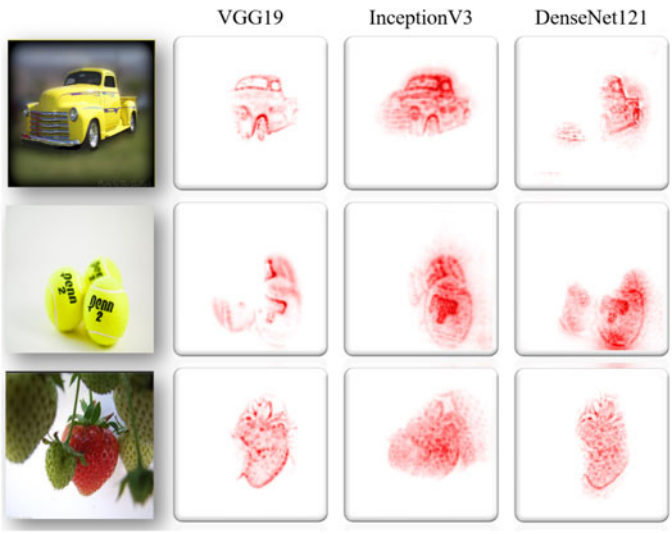

Fig. 2. Attention heat maps for VGG19 [18], InceptionV3 [60], DenseNet121 [56], which are similar even the architectures are different.

3 Attack on Attention (AoA)

To pursue high transferability for black-box attacks, we need to find common vulnerabilities and attack semantic features shared by different DNNs. Attention heat maps for three images are illustrated in Fig. 2, where the pixel-wise heat maps show how the input contributes to the prediction. Even with different architectures, the models have similar attention. Inspired by the similarity across different DNNs, we propose to Attack on Attention (AoA). Different to the existing methods that focus on attacking the output, AoA aims to change the attention heat map.

Let $h(x, y)$ stand for the attention heat map for the input x and a specified class y. $h(x, y_{\text{ori}})$ is a tensor with the dimension consistent to x. The basic idea of AoA is to shift the attention away from the original class, e.g., decrease the heat map for the correct class y_{ori}, as illustrated in Fig. 3. In this paper, we utilize SGLRP [54] to calculate the attention heat map $h(x, y)$, which is good at distinguishing the attention for the target class from the others. There exist of course many other techniques for obtaining the heat map to attack, as long as $h(x, y)$ and its gradient on x could be effectively calculated.

There are several potential ways to change the attention heat maps.

1) Suppress the magnitude of attention heat maps for the correct class $h(x, y_{\text{ori}})$: When the network attention degree on the correct class decreases, attention for other classes would increase and finally exceed the correct one, which leads the model to seek for information on other classes rather than the correct one and thus make an incorrect prediction. We call this design as the following *suppress loss*,

$$L_{\text{supp}}(x) = \|h(x, y_{\text{ori}})\|_1,$$

where $\|\cdot\|_1$ stands for the componentwise ℓ_1-norm.

2) Distract the focus of $h(x, y_{\text{ori}})$: It could be expected that when the attention is distracted from the original regions of interest, the model may lose its

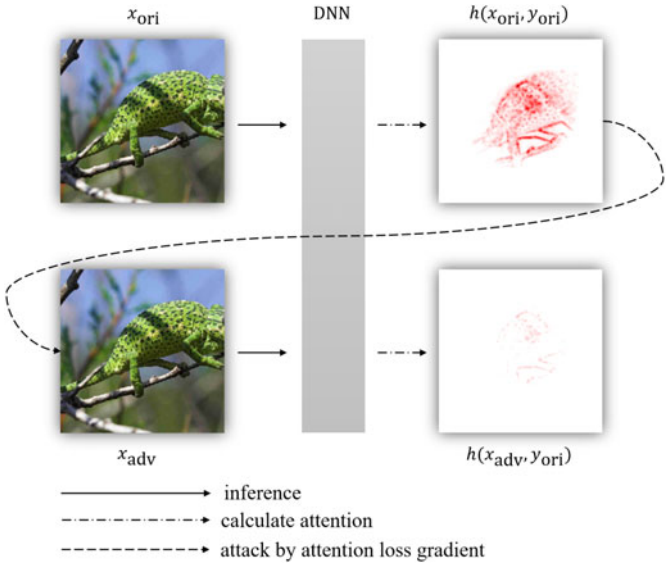

Fig. 3. The design of AoA. AoA calculates the attention heat map by SGLRP after inference. The gradient from the heat map back-propagates to the input and updates the sample iteratively. By suppressing the attention heat map value, one can change the network decision by fooling its focus. Constantly doing this, the produced adversarial sample could beat several black-box models.

capability for prediction. In this case, we do not require the network to focus on information of any incorrect class, but lead it to concentrate on irrelevant regions of the image. The loss could be expressed as the following *distract loss*,

$$L_{\text{dstc}}(x) = -\left\| \frac{h(x, y_{\text{ori}})}{max(h(x, y_{\text{ori}}))} - \frac{h(x_{\text{ori}}, y_{\text{ori}})}{max(h(x_{\text{ori}}, y_{\text{ori}}))} \right\|_1.$$

Here, self-normalization is conducted to eliminate the influence of attention magnitude.

3) Decrease the gap between $h(x, y_{\text{ori}})$ and $h(x, y_{\text{sec}}(x))$, the heat map for the second largest probability: If the attention magnitude for the second class exceeds that for the correct class, the network would focus more on information about the false prediction, which is inspired by CW attack [3]. We call it *boundary loss* and take the following formulation,

$$L_{\text{bdry}}(x) = \|h(x, y_{\text{ori}})\|_1 - \|h(x, y_{\text{sec}}(x))\|_1.$$

The values of attention heat maps vary a lot for different models, so the self-normalization may improve the transferability of adversarial samples. Therefore, rather than L_{bdry}, we can also consider the ratio between $h(x, y_{\text{ori}})$ and $h(x, y_{\text{sec}}(x))$, resulting the following *logarithmic boundary loss*

$$L_{\log}(x) = \log\left(\|h(x, y_{\text{ori}})\|_1\right) - \log\left(\|h(x, y_{\text{sec}}(x))\|_1\right).$$

Now let us illustrate the attack result on the attention heat map by distract loss. In Fig. 4, a clean sample is drawn together with its heat maps away from its original class. Aiming at ResNet50 [55], we minimize L_{dstc} and successfully change the heat map such that the attention is distracted to irrelevant regions (the second right column at the bottom). This common property shared by the attention in different DNNs makes the attack transferable, which is the motivation of attack on attention. The generated adversarial sample is shown in the leftmost in the bottom, which is incorrectly recognized by all the DNNs in Fig. 4. Additionally, we could see that the heat map for VGG19 is much clearer, which might explain the high transferability of its adversarial samples as shown later and in [15].

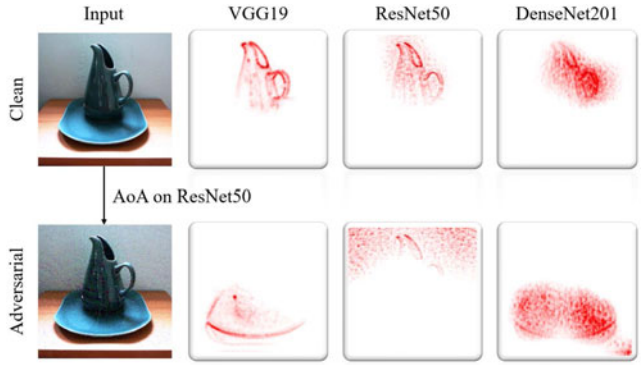

Fig. 4. Minimizing L_{dstc} distracts the attention from the correct ROI to irrelevant regions and similar distraction could be observed for different networks.

The transferability across different DNNs could be observed not only for the L_{dstc} but also for the other attention-related losses. To compare the above losses' attack performance, we attack on ResNet50 [55] and feed the adversarial samples to other DNNs (see the setting in Section 4 for details). Two attacks on classification loss, namely CW and PGD, are also compared as the baseline. The white-box attack success rates. i.e., the error rates of ResNet50, are all near 100 percent but attacks by different losses have different transferability performance, which is reported in Table 1. The suppress loss and the distract loss have a better transferability than PGD and CW. The logarithmic boundary loss is the best and is hence chosen as the attack target. Moreover, attack on attention could be readily combined with the existing attack on prediction (the cross entropy loss attacked in PGD, denoted by L_{ce}), resulting in the following *AoA loss*,

$$L_{\text{AoA}}(x) = L_{\log}(x) - \lambda L_{\text{ce}}(x, y_{\text{ori}}), \tag{1}$$

where λ is a trade-off between the attack on attention and cross entropy. In this paper, $\lambda = 1000$ is suggested such that the two items have similar variance for different inputs. The combination further increases the transferability, as shown in Table 1.

Basically, the adversarial samples are generated in an update process by minimizing the AoA loss L_{AoA}. Specifically, set $x^0_{\text{adv}} = x_{\text{ori}}$ and the update procedure could be generally described as the following

$$\begin{aligned} x^{k+1}_{\text{adv}} &= \text{clip}_{\varepsilon}\left(x^k_{\text{adv}} - \alpha \frac{g(x^k_{\text{adv}})}{||g(x^k_{\text{adv}})||_1/N}\right), \\ g(x) &= \frac{\partial L_{\text{AoA}}(x)}{\partial x}. \end{aligned} \tag{2}$$

The gradient g is normalized by its average ℓ_1-norm, i.e., $||g(x_k)||_1/N$, where N is the size of the image. Further, to

TABLE 1
Transfer Rate From ResNet50 to Other Neural Networks

Loss/Method	DN121 [56]	VGG19 [18]	RN152 [55]	IncV3 [60]	IncRNV2 [61]	Xception [62]	NASNetL [63]
CW [3]	66.6±1.24%	54.2±4.27%	47.3±4.69%	39.6±2.92%	37.9±4.77%	37.4±2.67%	28.8±2.58%
PGD [4]	67.8±1.83%	54.2±2.56%	46.8±3.71%	38.7±2.25%	35.6±4.21%	37.4±4.08%	28.4±3.17%
$L_{\text{supp}}(x)$	66.8±3.37%	57.2±3.96%	54.8±2.50%	43.9±2.78%	41.6±1.66%	40.9±2.60%	33.0±2.53%
$L_{\text{dstc}}(x)$	67.1±4.04%	56.5±2.28%	55.5±4.15%	45.4±3.77%	40.0±1.82%	41.6±4.07%	31.0±2.17%
$L_{\text{bdry}}(x)$	50.2±5.26%	49.8±4.39%	44.0±4.05%	34.1±3.34%	32.9±3.22%	31.7±1.86%	21.7±1.29%
$L_{\log}(x)$	74.9±3.48%	64.2±4.13%	59.2±4.71%	50.1±2.69%	46.2±3.39%	48.0±4.87%	36.3±3.74%
$L_{\text{AoA}}(x)$	**78.7±2.54%**	**64.9±2.01%**	**63.9±1.98%**	**53.3±2.27%**	**48.9±2.65%**	**50.9±3.01%**	**41.0±2.00%**

keep the perturbations invisible, we restrict our attack by the distance from the original clean sample such that the ℓ_∞ distance does not exceed ε. AoA is different from other attacks merely on the loss. Therefore, transferability-enhancement techniques developed for directly attacking prediction are also applicable to AoA. In fact, with optimization modification [12] or input modification [11], [13], [14], the transfer performance of AoA gets further improved, as numerically verified in Section 4.2. The procedure of AoA is summarized in Algorithm 1.

Algorithm 1. Attack on Attention

Input: AoA loss $L_{\text{AoA}}(x)$, origin sample x_{ori}, ℓ_∞-norm bound ϵ, RMSE threshold η, attack step length α.
Output: adversarial sample x_{adv}

$x^0_{\text{adv}} \leftarrow x_{\text{ori}}$
$N \leftarrow height \times width \times channel$ of x_{ori}
$k \leftarrow 0$
while $RMSE(x_{\text{ori}}, x^k_{\text{adv}}) < \eta$ **do**
  $g = \frac{\partial L_{AoA}(x^k_{\text{adv}})}{\partial x^k_{\text{adv}}}$ $\star$
  $x^{k+1}_{\text{adv}} = \text{clip}_\epsilon(x^k_{\text{adv}} - \alpha \cdot \frac{g}{||g||_1/N})$ $\star\atop\star$
  $k = k + 1$
end while
return x^k_{adv}

$\star$: could be modified for DI [13],SI [14] enhancement.
$\star\atop\star$: could be modified for MI [12],TI [11] enhancement.

Because of its good transferability on attention heat maps, AoA could be used for the black-box attack. The basic scheme is to choose a white-box DNN, which serves as the surrogate model for black-box attacks, to attack by updating (2). The generated adversarial samples tend to be aggressive to other black-box victim models.

4 Experiments

In this section, we will evaluate the performance of our Attack on Attention, especially its black-box attack capability compared to other SOTA methods. Since AoA is a very good black-box attack, it provides adversarial samples that can defeat many DNNs in a zero-query manner. These samples are collected in the dataset DAmageNet. This section will also introduce DAmageNet and report the performance of different DNNs on it. We further test the AoA performance under several defenses and find that AoA is the most aggressive method in almost all the cases.

4.1 Setup

The experiments for AoA are conducted on ImageNet [17] validation set. For attack and test, several well-trained models in Keras Applications [64] are used, including VGG19 [18], ResNet50 [55], DenseNet121 [56], InceptionV3 [60] and so on. We also use other adversarial-trained models (not by AoA, indicated by underline). For preprocessing, Keras preprocessing function, central cropping, and resizing (to 224) are used. The experiments are implemented in TensorFlow [65], Keras [64] with 4 NVIDIA GeForce RTX 2080Ti GPUs.

For the attack performance, we care about two aspects: the success/transfer rate of attack and how large the image is changed. Denote the generated adversarial sample as x_{adv}. The change from its corresponding original image x_{ori} could be measured by the Root Mean Squared Error (RMSE) in each pixel: $d(x_{\text{adv}}, x_{\text{ori}}) = \sqrt{\|x_{\text{adv}} - x_{\text{ori}}\|_2^2/N}$. In the experiments, 200 images are randomly selected from ImageNet validation set and the samples incorrectly predicted by the victim model are skipped as the same setting in [15]. Experiments are repeated 5 times and the overall performance on 1000 samples is reported. All the compared attacks will be fairly stopped when RMSE exceeds $\eta = 7$ and the perturbation is bounded by $\varepsilon = 0.1 * 255$. In this way, the number of iterations is about 10 with step size $\alpha = 2$ as the setting of [42] and other literatures. We alter $\alpha = 0.5$ for MI [12] based on numerical experiments.

4.2 Transferability of AoA

We first compare AoA with popular attacks CW [3] and PGD [4], which aim at classification losses. Specifically, CW uses the hinge loss and PGD uses the cross entropy loss. For CW, a gradient-based update is applied to keep the perturbation small. We carefully tune their parameters, resulting in a better transferability than reported in [15].

We use AoA, CW, and PGD to attack different neural networks, and then feed the generated adversarial samples to different models. The average error rates are reported in Table 2. AoA, CW, and PGD all have a high white-box attack success rate but the transfer performance varies a lot, which depends on both the surrogate model and the victim model. But in all the tested situations, AoA achieves a better black-box attack performance.

The essential difference of AoA from CW/PGD is the attack target. The existing effort on improving attack transferability for CW/PGD is mainly on modifying the optimization process. For example, DI proposes to transform 4 times when calculating gradients with a probability [13]. TI

TABLE 2
Error Rate (Top-1) of Different Attack Baselines

Surrogate	Method	DN121 [56]	IncRNV2 [61]	IncV3 [60]	NASNetL [63]	RN152 [55]	RN50 [55]	VGG19 [18]	Xception [62]
RN50 [55]	CW	66.6±1.24%	37.9±4.77%	39.6±2.92%	28.8±2.58%	47.3±4.69%	100.0±0.00%	54.2±4.27%	37.4±2.67%
	PGD	67.8±1.83%	35.6±4.21%	38.7±2.25%	28.4±3.17%	46.8±3.71%	**100.0±0.00%**	54.2±2.56%	37.4±4.08%
	AoA	**78.4±2.44%**	**49.0±1.87%**	**52.2±2.66%**	**39.6±3.61%**	**63.4±2.63%**	99.9±0.20%	**65.6±2.82%**	**51.1±2.18%**
DN121 [56]	CW	100.0±0.00%	33.5±2.55%	39.5±1.67%	31.9±2.87%	39.6±2.85%	64.6±3.76%	53.2±3.93%	39.4±1.16%
	PGD	100.0±0.00%	34.0±3.49%	41.7±2.38%	31.9±2.87%	41.5±3.21%	68.9±4.76%	55.5±2.28%	41.5±2.30%
	AoA	**100.0±0.00%**	**46.1±2.91%**	**53.5±3.46%**	**46.1±2.44%**	**55.0±2.77%**	**76.7±2.29%**	**64.6±2.18%**	**52.1±2.15%**
IncV3 [60]	CW	31.0±1.95%	22.7±3.01%	100.0±0.00%	21.3±0.60%	26.1±3.62%	42.3±2.01%	40.7±3.34%	33.4±1.56%
	PGD	32.7±2.50%	24.2±2.89%	100.0±0.00%	21.3±1.91%	27.3±2.29%	45.3±1.17%	40.7±3.39%	33.7±3.22%
	AoA	**39.0±1.79%**	**30.2±2.77%**	**100.0±0.00%**	**32.7±1.81%**	**34.0±2.93%**	**52.8±1.69%**	**45.9±3.98%**	**45.1±2.08%**
VGG19 [18]	CW	85.5±0.84%	62.0±1.67%	69.8±1.60%	62.7±1.21%	60.0±1.61%	77.8±2.04%	100.0±0.00%	68.0±2.39%
	PGD	87.1±1.20%	64.1±2.03%	71.8±1.63%	63.9±1.77%	63.1±4.14%	82.5±2.63%	100.0±0.00%	71.9±0.97%
	AoA	**91.4±2.65%**	**73.7±1.29%**	**79.8±1.08%**	**74.2±1.63%**	**73.5±1.05%**	**86.6±1.77%**	**100.0±0.00%**	**81.0±1.30%**
RN152 [55]	CW	42.4±2.52%	36.2±2.32%	35.3±1.66%	25.6±2.24%	100.0±0.00%	57.7±0.81%	46.0±4.06%	31.9±1.77%
	PGD	42.7±3.19%	35.0±2.47%	34.9±2.96%	24.5±3.05%	98.1±0.97%	55.3±2.71%	43.6±3.61%	30.5±4.87%
	AoA	**55.9±2.35%**	**54.2±2.36%**	**49.6±4.21%**	**36.4±2.60%**	**100.0±0.00%**	**71.5±2.57%**	**57.2±3.79%**	**45.6±1.93%**

translates the image for more transferable attack gradients [11]. MI tunes momentum parameter for boosting attacks [12]. SI divides the sample by the power 2 for 4 times to calculate the gradient [14]. Those state-of-the-art transferability-enhancement methods could improve the performance for CW/PGD and are also applicable to AoA.

In Table 3, we report the black-box attack performance when attacking ResNet50 with MI-DI, MI-TI, and SI (all with the hyperparameters suggested by their inventors). We find that SI is very helpful and can prominently increase the error rate for PGD and CW. Applying SI in AoA, denoted as SI-AoA, achieves the highest transfer rate, which is significantly better than other state-of-the-art methods.

4.3 AoA Under Defenses

Our main contribution in this paper is for black-box attack by increasing the transferability. It is not necessary that AoA can break defenses, but indeed, it is interesting to evaluate the attack performance under several defenses. In this experiment, we apply PGD, CW, and AoA, all enhanced by SI to attack ResNet50. We consider defenses that have been verified effective on ImageNet [66]. Those defense methods can be roughly categorized as preprocessing-based and adversarial-training-based, which could be used together.

Preprocessing-based defenses are to eliminate the adversarial perturbation. We use JPEG Compression [33], Pixel Deflection [34], Total Variance Minimization (TVM) [67] with provided parameters. Another idea is to add the randomness to observe the variance of the outputs. For example, Random Smoothing [68] makes prediction by m intermediate images, which is crafted by Gaussian noise from the input image. We choose $m = 100$ and the Gaussian noise scale $\sigma = 0.25 * 255$ here.

Adversarial training is to re-train the neural networks by adversarial samples. In [69], InceptionV3adv and Inception-ResNetV2adv are designed and [32] proposes ResNetXt101-denoise with denoising blocks in architectures to secure the model.

Table 4 gives the comprehensive black-box attack performance under defenses. Generally speaking, the preprocessing-based defenses decrease the error rate for about 5 to 10 percent and SI-AoA maintains the highest transfer rate. Adversarial-trained models (indicated by underlines in tables) exhibit a strong robustness to attacks, including SI-AoA (but still, it is better than SI-PGD, SI-CW). That means although samples generated by SI-AoA are different to others, the distribution can still be captured by adversarial training. Developing adversarial attacks that can defeat adversarial training is

TABLE 3
Error Rate (Top-1) of Transfer Attacks on ResNet50

Method	DN121 [56]	IncRNV2 [61]	IncV3 [60]	NASNetL [63]	RN152 [55]	RN50 [55]	VGG19 [18]	Xception [62]
CW	66.6±1.24%	37.9±4.77%	39.6±2.92%	28.8±2.58%	47.3±4.69%	100.0±0.00%	54.2±4.27%	37.4±2.67%
MI-DI-CW	66.9±1.91%	39.4±4.03%	42.9±1.59%	32.3±3.83%	50.2±4.74%	99.8±0.24%	57.9±3.40%	39.9±2.92%
MI-TI-CW	63.4±3.35%	42.0±3.33%	44.6±1.02%	33.7±1.96%	51.6±3.77%	99.7±0.24%	60.2±2.80%	40.6±2.40%
SI-CW	80.3±1.86%	46.4±2.22%	51.6±2.60%	38.3±3.53%	63.9±1.50%	99.9±0.20%	66.5±1.67%	48.8±3.70%
PGD	67.8±1.83%	35.6±4.21%	38.7±2.25%	28.4±3.17%	46.8±3.71%	100.0±0.00%	54.2±2.56%	37.4±4.08%
MI-DI-PGD	70.5±1.30%	43.3±3.33%	45.8±2.58%	35.7±3.53%	55.9±3.68%	99.5±0.00%	62.1±1.93%	43.3±2.42%
MI-TI-PGD	68.6±0.97%	44.6±2.18%	49.5±1.30%	38.0±1.00%	54.2±1.99%	99.3±0.51%	64.2±2.29%	45.3±1.72%
SI-PGD	81.2±1.63%	48.7±1.91%	53.0±0.95%	38.6±2.06%	66.1±2.46%	100.0±0.00%	69.5±2.10%	49.1±1.59%
AoA	78.4±2.44%	49.0±1.87%	52.2±2.66%	39.6±3.61%	63.4±2.63%	99.9±0.20%	65.6±2.82%	51.1±2.18%
MI-DI-AoA	74.1±1.02%	50.4±2.92%	52.0±3.32%	44.2±3.39%	58.7±3.59%	99.8±0.24%	66.4±4.20%	50.6±3.01%
MI-TI-AoA	79.2±1.21%	58.7±4.27%	62.5±3.52%	52.2±3.23%	67.5±2.76%	99.8±0.40%	75.3±2.89%	58.9±1.56%
SI-AoA	**90.5±0.89%**	**64.6±2.71%**	**66.1±3.89%**	**57.9±2.20%**	**78.8±1.75%**	**100.0±0.00%**	**80.4±2.73%**	**64.6±3.07%**

TABLE 4
Error Rate (Top-1) Under Defenses (ResNet50 as the Surrogate Model)

Victim	Method	None	JPEG [33]	Pixel [34]	Random [70]	TVM [67]	Smooth [68]
DN121 [56]	SI-CW	80.3±1.86%	64.9±2.40%	67.2±2.20%	64.5±3.99%	70.2±1.63%	60.0±2.26%
	SI-PGD	81.2±1.63%	65.1±1.24%	66.4±0.58%	64.0±3.44%	69.7±1.29%	60.0±2.26%
	SI-AoA	**90.5±0.89%**	**81.0±3.32%**	**82.1±2.85%**	**78.0±3.70%**	**83.7±3.14%**	**63.4±2.35%**
IncRNV2 [61]	SI-CW	46.4±2.22%	38.0±2.17%	38.3±0.93%	40.3±3.04%	41.0±1.64%	31.7±2.19%
	SI-PGD	48.7±1.91%	39.8±0.93%	39.3±0.75%	40.0±3.11%	42.1±0.86%	31.8±1.70%
	SI-AoA	**64.6±2.71%**	**56.7±1.72%**	**58.2±3.91%**	**57.8±4.37%**	**59.5±2.63%**	**34.6±3.24%**
IncV3 [60]	SI-CW	51.6±2.60%	43.2±3.39%	42.7±2.98%	46.2±2.34%	46.1±3.47%	33.5±4.73%
	SI-PGD	53.0±0.95%	44.8±3.33%	45.0±2.98%	47.9±3.09%	48.3±3.23%	32.6±5.66%
	SI-AoA	**66.1±3.89%**	**62.3±3.87%**	**62.4±4.12%**	**62.9±2.67%**	**64.1±3.79%**	**37.5±6.18%**
NASNetL [63]	SI-CW	38.3±3.53%	31.3±3.09%	32.4±4.12%	35.2±2.93%	34.0±4.57%	23.7±3.68%
	SI-PGD	38.6±2.06%	30.8±3.59%	31.5±2.92%	34.3±4.07%	34.6±2.96%	23.5±3.35%
	SI-AoA	**57.9±2.20%**	**49.2±3.71%**	**53.0±4.01%**	**52.7±3.93%**	**53.0±3.32%**	**29.3±2.80%**
RN152 [55]	SI-CW	63.9±1.50%	51.4±1.91%	51.6±1.85%	48.9±3.85%	56.6±1.56%	41.2±5.28%
	SI-PGD	66.1±2.46%	52.8±2.56%	54.1±1.53%	51.5±3.39%	58.4±1.83%	40.2±4.81%
	SI-AoA	**78.8±1.75%**	**70.3±3.56%**	**72.8±4.49%**	**67.1±2.82%**	**75.6±3.93%**	**44.2±5.07%**
RN50 [55]	SI-CW	99.9±0.20%	98.5±0.84%	98.7±0.81%	89.5±2.59%	99.6±0.49%	93.4±0.94%
	SI-PGD	100.0±0.00%	99.1±0.49%	99.4±0.58%	90.8±1.33%	99.6±0.37%	92.4±1.71%
	SI-AoA	**100.0±0.00%**	**99.9±0.20%**	**99.8±0.40%**	**95.6±2.13%**	**99.9±0.20%**	**94.1±1.20%**
VGG19 [18]	SI-CW	66.5±1.67%	60.7±4.27%	60.6±3.20%	62.9±4.07%	63.3±5.09%	89.8±1.89%
	SI-PGD	69.5±2.10%	62.8±3.54%	61.4±4.92%	65.7±3.80%	65.2±4.25%	89.6±1.73%
	SI-AoA	**80.4±2.73%**	**77.7±4.43%**	**78.5±3.77%**	**77.1±4.52%**	**79.8±4.04%**	**89.9±2.18%**
Xception [62]	SI-CW	48.8±3.70%	40.6±3.81%	40.9±3.71%	44.0±2.92%	44.7±3.23%	36.5±4.38%
	SI-PGD	49.1±1.59%	40.8±3.59%	43.0±4.02%	43.5±3.89%	44.7±3.37%	37.1±3.35%
	SI-AoA	**64.6±3.07%**	**57.6±3.26%**	**58.4±1.80%**	**61.1±3.89%**	**59.0±2.65%**	**40.9±4.52%**
IncV3adv [69]	SI-CW	31.2±1.29%	33.8±2.50%	35.0±3.35%	38.1±3.73%	37.0±4.27%	96.5±1.44%
	SI-PGD	31.5±3.08%	34.3±3.44%	35.8±2.99%	39.2±3.14%	38.4±2.85%	96.2±1.13%
	SI-AoA	**53.7±2.25%**	**52.7±2.20%**	**54.9±3.15%**	**55.1±2.78%**	**56.2±2.71%**	**96.2±1.16%**
IncRNV2adv [69]	SI-CW	26.4±1.59%	27.4±2.03%	27.6±2.63%	30.1±4.78%	28.2±3.66%	81.7±3.74%
	SI-PGD	26.1±1.98%	27.9±0.86%	28.5±2.51%	29.7±3.64%	29.8±0.93%	81.5±3.47%
	SI-AoA	**44.0±1.52%**	**44.2±3.23%**	**46.2±3.71%**	**48.0±4.55%**	**47.0±2.30%**	**82.3±3.16%**
RNXt101den [32]	SI-CW	18.0±3.13%	18.2±3.11%	18.2±3.33%	44.4±3.69%	18.1±3.22%	70.4±2.26%
	SI-PGD	18.2±2.87%	18.5±2.88%	18.9±3.17%	44.6±3.46%	18.4±3.31%	70.5±2.09%
	SI-AoA	**18.7±3.01%**	**19.2±2.71%**	**19.1±2.97%**	**44.6±3.48%**	**19.0±2.88%**	**70.5±2.26%**

interesting but out of our scope. Random smoothing generally has a low error rate but its inference time is much longer than other methods, generally m times and hence it is not a fair comparison. In our experiment, random smoothing seems not to work well on adversarial-trained models, sometimes even oppositely, which is also interesting but in the field of defenses.

4.4 DAmageNet

The above experiments verify that AoA has a promising transferability, which then makes it possible to generate adversarial samples that are able to beat many well-trained DNNs. An adversarial dataset will be very useful for evaluating robustness and defense methods. To establish an adversarial dataset, we use SI-AoA to attack VGG19 to generate samples from all 50000 samples from ImageNet validation set. Since the original images come from ImageNet training set and the adversarial samples are going to cheat neural networks, we hence name this dataset as DAmageNet.

DAmageNet contains 50000 adversarial samples and could be downloaded from http://www.pami.sjtu.edu.cn/Show/56/122. The samples are named the same as the original ones in ImageNet validation set. Accordingly, users could easily find the corresponding samples as well as their labels. The average RMSE between samples in DAmageNet and those in ImageNet is 7.23. In Fig. 5, we show several image pairs in ImageNet and DAmageNet.

To the best of our knowledge, DAmageNet is the first adversarial dataset, which can be used to evaluate model robustness and defenses. As an example, we use several well-trained models to recognize the images in DAmageNet. Several neural networks strengthened by adversarial training are considered as well. The error rate (top-1) is reported in Table 5. The models are from Keras Application and the test error may differ from original references. One could observe that i) all the listed 13 undefended models are not robust: DAmageNet increases the error rate of all 13 undefended models to over 85 percent; ii) the 5 listed adversarial-trained models have a slightly better performance and the error rate is over 70 percent; iii) DAmageNet resists 4 tested defenses with almost no drop on the error rate compared to other methods; iv) feature denoising model shows promising robustness but simply combining it with preprocessing-based defence does not work well.

Fig. 5. Samples in ImageNet and DAmageNet. The images on the left are original samples from ImageNet. The images on the right are adversarial samples from DAmageNet. One could observe that these images look similar and human beings have no problem to recognize them as the same class.

TABLE 5
Error Rate (Top-1) on ImageNet and DAmageNet

	No defense		Defenses on DAmageNet			
Victim	ImageNet [17]	DAmageNet	JPEG [33]	Pixel [34]	Random [70]	TVM [67]
VGG16 [18]	38.51	99.85	99.67	99.70	99.19	99.76
VGG19 [18]	38.60	99.99	99.99	99.99	99.96	99.99
RN50 [55]	36.65	93.94	91.88	92.48	92.52	93.08
RN101 [55]	29.38	88.13	85.44	86.23	86.12	87.06
RN152 [55]	28.65	86.78	83.93	84.83	84.71	85.68
NASNetM [63]	27.03	92.81	90.42	91.43	90.31	91.86
NASNetL [63]	17.77	86.32	83.31	84.87	84.91	85.53
IncV3 [60]	22.52	89.84	87.82	89.01	88.49	89.59
IncRNV2 [61]	24.60	88.09	85.01	85.95	89.04	86.79
Xception [62]	21.38	90.57	88.53	89.77	86.03	90.32
DN121 [56]	26.85	96.14	93.96	94.85	93.82	95.30
DN169 [56]	25.16	94.09	91.72	92.78	91.78	93.36
DN201 [56]	24.36	93.44	90.52	91.71	90.86	92.45
IncV3adv [69]	22.86	82.23	82.03	83.35	82.88	83.95
IncV3advens3 [71]	24.12	80.72	80.35	81.68	81.57	82.36
IncV3advens4 [71]	24.45	79.26	78.86	79.96	79.76	80.8
IncRNV2adv [69]	20.03	76.42	75.71	76.85	76.86	77.73
IncRNV2advens [71]	20.35	70.70	71.09	72.32	73.32	73.04
RNXt101den [32]	32.20	35.40	36.27	36.65	55.53	36.21

5 Conclusion

To improve the transferability of adversarial attack, we are the first to attack on attention and achieve a great performance on the black-box attack. The high transferability of AoA relies on the semantic features shared by different DNNs. AoA enjoys a significant increase in transferability when the traditional cross entropy loss is replaced with the attention loss. Since AoA alters the loss only, it could be easily combined with other transferability-enhancement methods, e.g., SI [14], and achieve a state-of-the-art performance.

By SI-AoA, we generate DAmageNet, the first dataset containing samples with a small perturbation and a high transfer rate (an error rate over 85 percent for undefended models and over 70 percent for adversarial-trained models). DAmageNet provides a benchmark to evaluate the robustness of DNNs by elaborately-crafted adversarial samples.

AoA has found the common vulnerability of DNNs in attention. Also, attention is just one semantic feature and attacking on other semantic features shared by DNNs is also promising to have good transferability.

ACKNOWLEDGMENTS

This work was partially supported by National Key Research Development Project (No. 2018AAA0100702, 2019YFB1311503) and National Natural Science Foundation of China (No. 61977046, 61876107, U1803261). The authors are grateful to the anonymous reviewers for their insightful comments.

REFERENCES

[1] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015, Art. no. 20.
[3] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.
[5] S. Tang, X. Huang, M. Chen, C. Sun, and J. Yang, "Adversarial attack type I: Cheat classifiers by significant changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, 2019, doi: 10.1109/TPAMI.2019.2936378.
[6] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," in *Proc. 32nd Adv. Neural Inf. Process. Syst.*, 2019.
[7] A. Ilyas, L. Engstrom, and A. Madry, "Prior convictions: Black-box adversarial attacks with bandits and priors," in *Proc. 7th Int. Conf. Learn. Representations*, 2019.
[8] Y. Guo, Z. Yan, and C. Zhang, "Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 3820–3829.
[9] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.
[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1765–1773.
[11] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4312–4321.
[12] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.
[13] C. Xie *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2730–2739.
[14] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," in *Proc. 8th Int. Conf. Learn. Representations*, 2020.
[15] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 644–661.
[16] T. Zhang and Z. Zhu, "Interpreting adversarially trained convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 7502–7511.
[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[19] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 2096–2030, 2016.
[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2107–2116.
[21] A. Sinha, H. Namkoong, and J. Duchi, "Certifiable distributional robustness with principled adversarial training," in *Proc. Int. Conf. Learn. Representations*, 2018, Art. no. 29.
[22] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.
[23] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.
[24] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
[25] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," in *Proc. Advances Neural Inf. Process. Syst.*, 2018, pp. 8322–8333.
[26] S. Baluja and I. Fischer, "Adversarial transformation networks: Learning to generate adversarial examples," 2017, *arXiv: 1703.09387*.
[27] J. Han *et al.*, "Once a man: Towards multi-target attack via learning multi-target adversarial network once," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5158–5167.
[28] T. Miyato, A. M. Dai, and I. J. Goodfellow, "Adversarial training methods for semi-supervised text classification," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.
[29] S. Sankaranarayanan, A. Jain, R. Chellappa, and S. N. Lim, "Regularizing deep networks using efficient layerwise adversarial training," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4008–4015.
[30] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Painless adversarial training using maximal principle," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 227–238.
[31] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1778–1787.
[32] C. Xie, Y. Wu, L. V. D. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 501–509.
[33] Z. Liu *et al.*, "Feature distillation: DNN-oriented JPEG compression against adversarial examples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 860–868.
[34] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8571–8580.
[35] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image super-resolution as a defense against adversarial attacks," *IEEE Trans. Image Process.*, vol. 29, pp. 1711–1724, 2020.
[36] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," early access, 2016.
[37] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.
[38] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2137–2146.
[39] B. Ru, A. Cobb, A. Blaas, and Y. Gal, "Bayesopt adversarial attack," in *Proc. Int. Conf. Learn. Representations*, 2020.
[40] L. Meunier, J. Atif, and O. Teytaud, "Yet another but more efficient black-box adversarial attack: Tiling and evolution strategies," 2019, *arXiv: 1910.02244*.
[41] J. Du, H. Zhang, J. T. Zhou, Y. Yang, and J. Feng, "Query-efficient meta attack to deep neural networks," in *Proc. 8th Int. Conf. Learn. Representations*, 2019.
[42] D. Wu, Y. Wang, S. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," in *Proc. Int. Conf. Learn. Representations*, 2019.
[43] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[44] W. Samek, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, vol. 11700, Berlin, Germany: Springer, 2019.
[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
[46] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.
[47] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[48] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[49] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nat. Methods*, vol. 12, no. 10, 2015, Art. no. 931.

[50] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.
[51] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[52] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, 2015, Art. no. e0130140.
[53] J. Gu, Y. Yang, and V. Tresp, "Understanding individual decisions of cnns via contrastive backpropagation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 119–134.
[54] B. K. Iwana, R. Kuroki, and S. Uchida, "Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 4176–4185.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[56] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[57] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," 2019, *arXiv: 1907.07174*.
[58] A. Barbu *et al.*, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 9453–9463.
[59] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. Int. Conf. Learn. Representations*, 2019.
[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[61] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.
[62] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[63] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.
[64] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://keras.io
[65] M. Abadi *et al.* "TensorFlow: Large-scale mchine learning on heterogeneous systems," 2015, *arXiv:1603.04467*, software available from tensorflow.org.
[66] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 3–14.
[67] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.
[68] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 1310–1320.
[69] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. 5th Int. Conf. Learn. Representations*, 2017.
[70] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, "Mitigating adversarial effects through randomization," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.
[71] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

Sizhe Chen received the BS degree from Shanghai Jiao Tong University, Shanghai, China, in 2020. He is now working toward the master's degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His research interests include model security, robust learning, and interpretability of DNN.

Zhengbao He is a senior student with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. He is now doing research with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His research interests include adversarial attack and deep learning.

Chengjin Sun received the BS degree from Nanjing University, Nanjing, China, in 2018. She is now working toward the master's degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. Her research interests include adversarial robustness for deep learning.

Jie Yang received the PhD degree from the Department of Computer Science, Hamburg University, Hamburg, Germany, in 1994. Currently, he is a professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., National Science Foundation, 863 National High Technique Plan), had one book published in Germany, and authored more than 300 journal papers. His major research interests include object detection and recognition, data fusion and data mining, and medical image processing.

Xiaolin Huang (Senior Member, IEEE) received the BS degree in control science and engineering, and the BS degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2006, and the PhD degree in control science and engineering from Tsinghua University, Beijing, China. From 2012 to 2015, he worked as a postdoctoral researcher with ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that he was selected as an Alexander von Humboldt fellow and working in Pattern Recognition Lab, the Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. From 2016, he has been an associate professor with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. In 2017, he was awarded by "1000-Talent Plan" (Young Program). His current research interests include machine learning and optimization, especially for robustness and sparsity of both kernel learning and deep neural networks.