Improving Transferable Targeted Adversarial Attacks with Model Self-Enhancement

Han Wu* Guanyan Ou* Weibin Wu† Zibin Zheng
School of Software Engineering, Sun Yat-sen University
{wuhan36, ougy3}@mail2.sysu.edu.cn, {wuwb36, zhzibin}@mail.sysu.edu.cn

Abstract

Various transfer attack methods have been proposed to evaluate the robustness of deep neural networks (DNNs). Although manifesting remarkable performance in generating untargeted adversarial perturbations, existing proposals still fail to achieve high targeted transferability. In this work, we discover that the adversarial perturbations' overfitting towards source models of mediocre generalization capability can hurt their targeted transferability. To address this issue, we focus on enhancing the source model's generalization capability to improve its ability to conduct transferable targeted adversarial attacks. In pursuit of this goal, we propose a novel model self-enhancement method that incorporates two major components: Sharpness-Aware Self-Distillation (SASD) and Weight Scaling (WS). Specifically, SASD distills a fine-tuned auxiliary model, which mirrors the source model's structure, into the source model while flattening the source model's loss landscape. WS obtains an approximate ensemble of numerous pruned models to perform model augmentation, which can be conveniently synergized with SASD to elevate the source model's generalization capability and thus improve the resultant targeted perturbations' transferability. Extensive experiments corroborate the effectiveness of the proposed method. Notably, under the black-box setting, our approach can outperform the state-of-the-art baselines by a significant margin of 12.2% on average in terms of the obtained targeted transferability. Code is available at https://github.com/g4alllf/SASD.

1. Introduction

Despite their wide range of applications in real-world scenarios, deep neural networks (DNNs) have shown vulnerability in the face of imperceptible adversarial perturbations [41, 43, 49, 50]. In recent years, numerous transfer attack methods have been proposed to evaluate the robustness of DNNs [1, 42, 44, 48]. Although existing approaches have demonstrated excellent performances on untargeted transfer attacks, it is still challenging to accomplish targeted transfer attacks, which require the attacker to generate perturbations that can mislead the black-box victims to return a target label [22, 30].

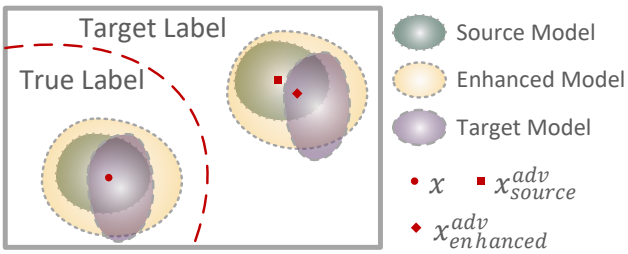


Figure 1. The motivation of the proposed method that enhances the source model's generalization capability to enable more transferable targeted adversarial attacks.

Adversarial perturbations are prone to overfit the source model and manifest limited targeted transferability when the source model's generalization capability is mediocre. Specifically, neural networks can capture features essential for decision-making, with some features being general while the others being model-specific [16]. Intuitively, adversarial examples that primarily manipulate model-specific features are unlikely to transfer well among different models. Working towards a more generalized source model can alleviate the reliance on model-specific features, thus enhancing the transferability of the synthesized adversarial examples [35].

Therefore, we propose to enhance the source model's generalization capability to enable more transferable targeted adversarial attacks. We conceptually illustrate the motivation of our method in Figure 1. Since different models rely on different features to make decisions, and their loss landscapes are often sharp, the transferable targeted adversarial examples are likely to reside away from the local optima of different models. It is thus challenging to transfer the adversarial examples generated by the vanilla

*Equal contribution.
†Corresponding author.

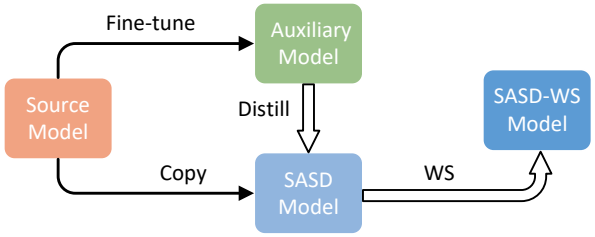


Figure 2. The overall procedure of SASD-WS.

source model to a different target model. A viable solution is to enhance the source model's generalization capability to capture the general features learned by different models. This can be achieved by learning a flatter loss landscape that covers the transferable targeted adversarial examples in the low-loss region.

To this end, we propose a novel model self-enhancement method: SASD-WS, which incorporates **Sharpness-Aware Self-Distillation (SASD)** and **Weight Scaling (WS)** to promote the source model's generalization capability. Then, we can employ the enhanced source model to produce targeted perturbations with better transferability. The overall procedure of our method is summarized in Figure 2. We explain the two major components of the proposed SASD-WS as follows:

SASD. In the warm-up step of SASD, we employ the technique of Sharpness-Aware Minimization (SAM) [6] to fine-tune an auxiliary model, which mirrors the source model's structure. Specifically, during the fine-tuning of the auxiliary model, we simultaneously minimize its loss of sharpness and the cross-entropy loss between its predictions and ground-truth labels. Given the fine-tuned auxiliary model, we then enhance our source model by distilling the auxiliary model into the source model while minimizing the source model's loss of sharpness. As a result, SASD can effectively endow the source model with a flatter loss landscape and improve the resultant targeted perturbations' transferability.

WS. To further boost the generalization capability of the obtained SASD model, we propose WS for efficient model augmentation. Specifically, we apply random pruning to the obtained SASD model to produce numerous pruned models. The resultant ensemble of randomly pruned SASD models can possess better generalization capability than the original SASD model. However, generating adversarial examples with an ensemble model can be computationally expensive. Motivated by the approximation scheme in [17], we efficiently approximate the ensemble of randomly pruned SASD models by scaling the weights of the SASD model. Therefore, WS can be efficiently synergized with SASD to further boost the targeted transferability of the synthesized adversarial examples.

To sum up, the main contributions of this work are:

- We propose a novel model self-enhancement method incorporating two major components: Sharpness-Aware Self-Distillation (SASD) and Weight Scaling (WS). Specifically, SASD distills a fine-tuned auxiliary model into the source model while flattening its loss landscape. WS then performs model augmentation with an efficient approximation. Their resonance promotes the source model's generalization capability and thus enables more transferable targeted adversarial attacks.
- We conduct extensive experiments to validate the superiority of the proposed method. Notably, compared with the state-of-the-art benchmarks, our approach can improve the targeted attack success rate by a significant margin of 12.2% on average under the black-box setting.
- We confirm that our attacks can transfer well to real-world applications, exceeding the state-of-the-art baselines by 8.4% on average. We also visualize the effectiveness of the proposed method from the perspective of the enhanced model's loss landscape, and validate the efficacy of our approach against defense methods.

2. Related Work

2.1. Transferable Adversarial Attack

With the development of transfer attack techniques, there are currently three types of approaches to improve the perturbation's transferability:

Enhancing the source model. As normally trained neural networks perform poorly in targeted attacks, several model enhancement methods have been proposed to improve the source model's capability to conduct targeted transfer attacks. These methods include replacing a single source model with an ensemble of models obtained by random dropout [23] and additional training epochs [9]. Springer et al. [35] instead utilizes adversarial fine-tuning to enhance the source model's ability to synthesize transferable adversarial examples. Our method also attempts to improve adversarial transferability by enhancing the source model. Nevertheless, unlike existing model enhancement-based attacks, we propose to flatten the source model's loss landscape to improve its generalization capability, enabling more transferable targeted attacks.

Augmenting the input data. To boost adversarial transferability, some transfer attacks resort to input augmentation, such as image translation [3], random resizing and padding [46], and mixup [40]. Unlike prior efforts, a recent input augmentation-based attack technique, Spectrum Simulation Attack (SSA), proposes to apply a spectrum transformation to the input [27].

Rectifying the optimization procedure. Instead of greedily perturbing the clean images along the gradient of the cross-entropy loss [20], several attempts propose to im-

prove the optimization procedure to craft more transferable adversarial examples. Straightforward solutions include employing momentum [2] and Nesterov accelerated gradient [25]. Huang et al. [15] instead consider adding the intermediate feature loss into the attack objective to regularize the search of adversarial examples. The state-of-the-art attack of this kind, Reverse Adversarial Perturbation (RAP) [31], seeks adversarial examples located at a region with uniformly low loss values.

Our attack can be conveniently combined with other transfer attacks based on input augmentation and optimization procedure rectification to further enhance the transferability of adversarial attacks.

2.2. Improving DNNs' Generalization Ability

Foret et al. [6] propose Sharpness-Aware Minimization (SAM), which simultaneously minimizes loss values and loss sharpness to improve the model's generalization ability. Hinton et al. [13] introduce knowledge distillation to transfer the knowledge of a teacher model to the student model [45]. Intriguingly, Stanton et al. [36] discover that knowledge distillation can enhance the student model's generalization ability. Building upon Sharpness-Aware Minimization and knowledge distillation methodologies, we propose Sharpness-Aware Self-Distillation (SASD) to integrate Sharpness-Aware Minimization with knowledge distillation, which can further improve the source model's generalization capability. Since a more generalized source model can better capture the general features learned by different models, the produced adversarial examples are more likely to possess better targeted transferability [35].

3. Methodology

3.1. Problem Formulation

We first formulate the targeted transfer attack. Given a labeled image dataset D and a target classifier f_{target}, for an image-label pair $(x, y) \in D$, a targeted transfer attack aims to generate an adversarial example $x_{\text{adv}} = x + \delta$ to mislead the target classifier f_{target} to predict a specific target label $y_{\text{target}} \neq y$. Since the target classifier is black-box, a targeted transfer attack instead derives the adversarial example x_{adv} with a white-box source model f. It is usually accomplished by optimizing x_{adv} to minimize the source model's cross-entropy loss L_{CE} on the target label [8, 51]:

$$\begin{aligned} &\min\ L_{\text{CE}}(f(x+\delta), y_{\text{target}}), \\ &\text{s.t. } ||\delta||_\infty \leq \epsilon. \end{aligned} \tag{1}$$

To ensure that the perturbation δ is imperceptible, attackers should also set a small perturbation budget ϵ. After generating the adversarial example x_{adv} with the source model, a targeted transfer attack directly uses x_{adv} to attack the target classifier, with the goal of $f_{\text{target}}(x_{\text{adv}}) = y_{\text{target}}$. To improve the attack success rate (i.e., the transferability) of the generated targeted perturbations, we propose enhancing the source model's generalization ability, detailed in the following sections.

Algorithm 1 Sharpness-Aware Self-Distillation

Input: Fine-tuned auxiliary model

Initialization: $n \leftarrow 0$
for every batch in the data loader **do**
  $n \leftarrow n + 1$
  Calculate $L_{\text{distillation}}$ by Equation (3)
  $\epsilon \leftarrow \underset{\|\epsilon\|_2 \leq \rho}{\text{argmax}}\ \epsilon^T \nabla_{\omega_s} L_{\text{distillation}}(\omega_s)$
  $g = \nabla_{\omega_s} L(\omega_s)|_{\omega_s + \epsilon}$
  $\omega_s \leftarrow \omega_s - lr \cdot g$
  if $n \geq n_{\max}$ **then**
    break
  end if
end for

Output: SASD model

3.2. Sharpness-Aware Self-Distillation

Fine-tuning the auxiliary model. We start with a pre-trained source model and fine-tune it on its original training dataset to obtain the auxiliary model. The fine-tuning procedure follows Sharpness-Aware Minimization (SAM), which simultaneously minimizes loss values and loss sharpness [6]. Specifically, in each iteration, we first perform back-propagation based on the cross-entropy loss [7] between the output of the auxiliary model and the ground-truth label. We then modify the auxiliary model's weight ω by adding a perturbation ϵ_m that can maximize the cross-entropy loss:

$$\epsilon_m = \underset{\epsilon_m}{\arg\max}\ L_{\text{CE}}(f(\omega + \epsilon_m, x), y), \tag{2}$$

where $f(\omega, x)$ is the output of a neural network f given the model weight ω and the input x. We perform back-propagation again based on the perturbed weight to minimize the auxiliary model's loss of sharpness.

Distilling the auxiliary model. After obtaining a fine-tuned auxiliary model, we distill it into the source model to enhance its generalization ability. Specifically, we first make the source model's probability output close to the auxiliary model's. Let $q_1 = \phi_1(l_{\text{aux}})$ denote the probability output of the auxiliary model, where ϕ_1 is a differentiable transformation, and l_{aux} is the auxiliary model's logit output. In our SASD, we set $\phi_1(l_{\text{aux}}) = \text{Softmax}(l_{\text{aux}}/\tau)$, where τ is the distillation temperature. The larger the temperature is, the softer the probability distribution will be. Similarly, let $q_2 = \phi_2(l_{\text{source}})$ denote the probability output of the source model, where $\phi_2(l_{\text{source}}) =$

$\mathrm{Softmax}(l_{\mathrm{source}}/\tau)$. To minimize the prediction difference between the auxiliary and source models, we employ the Kullback-Leibler divergence D_{KL} [18] between q_1 and q_2 as the distillation loss:

$$L_{\mathrm{distillation}}(\omega_s) = \mathbb{E}_{(x,y)\in D}\left[D_{\mathrm{KL}}(q_1(x,y), q_2(x,y))\right]. \tag{3}$$

We then simultaneously minimize the distillation loss and its sharpness:

$$\min_{\omega_s} \mathbb{E}_{(x,y)\in D}\left[L(\omega_s)\right], \tag{4}$$

where

$$\begin{aligned} L(\omega_s) &= \max_{\|\epsilon\|_2\leq\rho} L_{\mathrm{distillation}}(\omega_s+\epsilon) \\ &\approx \max_{\|\epsilon\|_2\leq\rho}\left[L_{\mathrm{distillation}}(\omega_s) + \epsilon^T\nabla_{\omega_s} L_{\mathrm{distillation}}(\omega_s)\right]. \end{aligned} \tag{5}$$

Let lr be the learning rate, and $n_{\max}$ be the maximum iteration number. Algorithm 1 summarizes the SASD process.

3.3. Weight Scaling

To further boost the generalization capability of the obtained SASD model, we apply the network pruning method [47] to generate an ensemble of pruned SASD models. Specifically, we randomly prune the convolutional layers of the obtained SASD model with a probability of $1-p$. Then, the weight of a pruned SASD model is:

$$\omega_{\mathrm{pruned}} = \omega \odot (\mathbf{1} - \xi(\omega, 1-p)), \tag{6}$$

where $\mathbf{1}$ denotes the all-ones matrix with the same size as the weight of the SASD model. $\xi(\omega, 1-p)$ indicates the randomly selected weights in the convolutional layers to be pruned. By applying the Hadamard product $\odot$ that represents element-wise multiplication to the weight, we can obtain the pruned model with some weights in the convolutional layers setting to zero.

Inspired by the approximation scheme in [17], we further approximate an ensemble of pruned models with a weight-scaled model. Specifically, let $\overline{f} = \frac{1}{n}\sum_{i=1}^{n} f(\omega_i)$ be the ensemble of models with the same structure but different weights. Let $\overline{\omega} = \frac{1}{n}\sum_{i=1}^{n}\omega_i$ be the average weight of the component networks in the ensemble, and $\eta_i = \omega_i - \overline{\omega}$. For $i \in \{i | 1 \leq i \leq n, i \in \mathbb{N}\}$, we have:

$$\frac{1}{n}\sum_{i=1}^{n}\eta_i = 0, \tag{7}$$

$$\begin{aligned} f(\omega_i, x) &= f(\overline{\omega}+\eta_i, x) \\ &= f(\overline{\omega}, x) + \eta_i^T\nabla_{\overline{\omega}} f(\overline{\omega}, x) + O(\|\eta_i\|_2^2). \end{aligned} \tag{8}$$

Therefore, we can derive that:

$$\begin{aligned} \overline{f} &= \frac{1}{n}\sum_{i=1}^{n} f(\omega_i, x) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[f(\overline{\omega}, x) + \eta_i^T\nabla_{\overline{\omega}} f(\overline{\omega}, x) + O(\|\eta_i\|_2^2)\right] \\ &= f(\overline{\omega}, x) + \frac{1}{n}\sum_{i=1}^{n}\eta_j^T\nabla_{\overline{\omega}} f(\overline{\omega}, x) + O(\|\eta\|_2^2) \\ &= f(\overline{\omega}, x) + O(\|\eta\|_2^2) \approx f(\overline{\omega}, x). \end{aligned} \tag{9}$$

It implies that a single model with the average weight of multiple models can replace the ensemble of these models as long as their weight differences are small enough.

Therefore, if the pruning probability $1-p$ is small, the ensemble of infinite pruned SASD models can be approximated by a single SASD model that is weight-scaled by the scaling ratio p (i.e., the SASD-WS model):

$$\begin{aligned} \overline{f} &= \frac{1}{n}\sum_{i=1}^{n} f(\omega_i^{\text{pruned SASD}}, x) \\ &\approx f(\overline{\omega}, x) = f(p\cdot\omega^{\mathrm{SASD}}, x) = f_{\mathrm{SASD-WS}}, \end{aligned} \tag{10}$$

where ω^{SASD} is the weight of the SASD model before pruning.

3.4. Attacking Algorithm

After obtaining the SASD-WS model, we generate targeted adversarial examples with the TI-DI-MI method [2, 3, 46]. Specifically, given the input data x, the source model f, and the target class y_{target}, the perturbation δ_{i+1} can be iteratively updated by:

$$\delta_{i+1} = \mathrm{Clip}_{\delta_i}^{\epsilon_p}\left\{\delta_i + \alpha\cdot\mathrm{sign}\left(g_{i+1}\right)\right\}, \tag{11}$$

$$g_{i+1} = g_i + W\cdot\nabla_\delta \frac{\mathcal{L}\left(\mathrm{DI}(x+\delta_i), y_{\mathrm{target}}; f\right)}{\|\mathcal{L}\left(\mathrm{DI}(x_i+\delta_i), y_{\mathrm{target}}; f\right)\|_1}. \tag{12}$$

α is the step size for attacks. W is the Gaussian kernel for image translation. $\mathcal{L}$ is the attack objective function. DI denotes the input transformation of random resizing and padding. The generated adversarial perturbation is l_∞-norm bounded by the maximum perturbation budget ϵ_p.

4. Experiments

4.1. Experimental Setup

We focus on attacking the image classifiers trained on the ImageNet (ILSVRC 2012) dataset [33]. We perform SASD with the standard training and validation splits of the ImageNet dataset. We apply our method to eight models pre-trained on ImageNet: Inception-v3 [37], Inception-v4 [38], Inception-ResNet-v2 [38], ResNet-50 [11], ResNet-101

Table 1. Targeted transfer attack success rates (%) with a single source model. All methods are combined with the input augmentation technique TI-DI-MI to generate adversarial perturbations. "Pre-trained" means directly using the normally trained model as the source model without modification. "*" indicates white-box cases. We run each method five times and show the mean and standard deviation of the results.

Method	Source Model: Inc-v3				Source Model: Res50			
	→ Inc-v3	→ Res50	→ Dense121	→ VGG16	→ Inc-v3	→ Res50	→ Dense121	→ VGG16
Pre-trained	**99.0 ± 0.0***	2.1 ± 0.1	3.6 ± 0.1	1.9 ± 0.3	7.8 ± 0.3	**98.7 ± 0.1***	64.3 ± 0.9	53.8 ± 0.7
RAP	93.6 ± 0.0*	3.5 ± 0.0	5.4 ± 0.0	3.9 ± 0.0	14.4 ± 0.0	98.5 ± 0.0*	56.2 ± 0.0	52.6 ± 0.0
GhostNet	97.9 ± 0.2*	8.1 ± 0.3	12.8 ± 0.1	6.6 ± 0.4	13.2 ± 0.4	98.0 ± 0.2*	75.0 ± 0.3	68.4 ± 0.6
LGV	55.0 ± 0.2*	40.5 ± 0.3	48.3 ± 0.3	49.5 ± 0.2	54.5 ± 0.8	93.2 ± 0.2*	90.0 ± 0.1	84.0 ± 0.2
SASD-WS	94.6 ± 0.2*	**44.4 ± 0.2**	**57.6 ± 1.1**	**53.8 ± 0.8**	**70.3 ± 0.8**	97.2 ± 0.4*	**93.0 ± 0.2**	**88.7 ± 0.4**
Method	Source Model: Dense121				Source Model: VGG16			
	→ Inc-v3	→ Res50	→ Dense121	→ VGG16	→ Inc-v3	→ Res50	→ Dense121	→ VGG16
Pre-trained	6.0 ± 0.4	39.3 ± 0.6	**98.8 ± 0.2***	33.6 ± 0.5	0.9 ± 0.2	9.8 ± 0.5	12.2 ± 0.5	**95.4 ± 0.3***
RAP	12.0 ± 0.0	41.2 ± 0.0	97.8 ± 0.0*	37.4 ± 0.0	2.1 ± 0.0	9.9 ± 0.0	9.8 ± 0.0	85.8 ± 0.0*
GhostNet	10.4 ± 0.6	49.6 ± 0.8	98.6 ± 0.1*	44.0 ± 0.6	1.1 ± 0.2	13.0 ± 0.1	14.9 ± 0.7	94.9 ± 0.2*
LGV	51.3 ± 0.3	80.8 ± 0.2	92.9 ± 0.2*	75.8 ± 0.2	—	—	—	—
SASD-WS	**61.6 ± 0.6**	**87.5 ± 0.2**	97.7 ± 0.1*	**82.7 ± 0.6**	**16.3 ± 0.7**	**41.9 ± 0.5**	**51.7 ± 0.2**	94.5 ± 0.3*

[11], ResNet-152 [11], DenseNet-121 [14], and VGGNet-16 [34]. All these pre-trained models' weights are publicly accessible[1]. For SASD, we set the distillation temperature $\tau = 1$ and the learning rate $lr = 0.05$ for all source models. We set WS's scaling ratio p to 0.93 for each SASD-WS model.

We compare the targeted transfer attack performance of our SASD-WS method with the state-of-the-art model enhancement-based attacks: GhostNet [23] and LGV [9]. Although RAP [31] works by rectifying the optimization procedure and does not modify the source model, its motivation is similar to our method. Therefore, we also compare our approach with RAP. We implement the baseline methods following the default settings in their original papers. For LGV, we additionally train the source model on the ImageNet training set for ten epochs [33]. We then obtain four LGV models in each epoch, leading to 40 LGV models for each source model. We cannot apply LGV to VGGNet-16 since the default settings of LGV are not compatible with VGGNet-16. Notably, since LGV and GhostNet transform a single source model into multiple models to generate adversarial examples, we follow their papers to ensemble these models with the longitudinal ensemble method [23]. In contrast, we only transform a single source model into a single SASD-WS model when generating adversarial examples.

Following the practice in the literature [2, 23], we set the step size $\alpha = 2/255$ and the maximum perturbation budget $\epsilon_p = 16/255$. We set the maximum iteration number $t_{\max} = 100$ in the adversarial perturbation generation process. We conduct targeted transfer attacks on NIPS17 targeted adversarial attack competition dataset[2] [21]. All experiments are implemented with PyTorch 2.0.1 and conducted with an NVIDIA GeForce RTX 4090 GPU of 24 GB memory.

4.2. Attacking Performance

Targeted transfer attacks with a single source model. We first evaluate the performance of our method when only a single source model is available. We use the logit loss $L_{\text{Logit}} = -l^{(t)}(x+\delta)$ proposed in [51] as the attack objective function $\mathcal{L}$, where $l^{(t)}(\cdot)$ is the output logit of the source model concerning the target label. Table 1 reports the attack success rate of each method.

As shown in Table 1, our method consistently outperforms the state-of-the-art baselines by a significant margin of **12.2%** on average under the black-box settings. Besides improving the targeted attack success rate under the black-box settings, SASD-WS can also maintain high white-box attack performance.

Targeted transfer attacks with an ensemble of source models. We then test the performance of our method when multiple source models are available, including Inception-v3, Inception-v4, Inc-Res-v2, ResNet-50, ResNet-101, and ResNet-152. For the target models, due to the widespread application of adversarial training techniques [19, 28, 39], we first consider adversarially trained models. We also consider the scenarios where the target and source models are from different structure families. Specifically, we evaluate the effectiveness of the targeted perturbation on Vision

[1] https://github.com/Cadene/pretrained-models.pytorch, https://pytorch.org/vision/stable/models.html

[2] https://github.com/cleverhans-lab/cleverhans/blob/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset/dev_dataset.csv

Table 2. Targeted transfer attack success rates (%) with an ensemble of six source models: Inception-v3, Inception-v4, Inc-Res-v2, ResNet-50, ResNet-101, and ResNet-152. All methods are combined with the input augmentation technique TI-DI-MI to generate adversarial perturbations. "Pre-trained" means directly using the normally trained model as the source model without modification. We run each method five times and show the mean and standard deviation of the results.

Method	Inc-v3$^{adv}_{ens3}$	Inc-v3$^{adv}_{ens4}$	IncRes-v2$^{adv}_{ens}$	ViT-B/16	ViT-L/16	CLIP (RN50)	CLIP (ViT-B/32)	CLIP (ViT-L/14)
Pre-trained	0.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0	3.1 ± 0.3	1.5 ± 0.2	16.3 ± 0.7	2.0 ± 0.1	3.2 ± 0.4
RAP	1.7 ± 0.0	1.5 ± 0.0	0.1 ± 0.0	6.0 ± 0.0	3.0 ± 0.0	17.7 ± 0.0	5.1 ± 0.0	4.8 ± 0.0
GhostNet	0.7 ± 0.1	1.1 ± 0.2	0.2 ± 0.0	9.1 ± 0.5	5.7 ± 0.1	33.0 ± 0.5	5.3 ± 0.2	10.7 ± 0.1
LGV	8.1 ± 0.2	6.7 ± 0.2	1.7 ± 0.1	19.6 ± 0.0	13.0 ± 0.2	$\mathbf{59.4 \pm 0.3}$	22.4 ± 0.2	28.4 ± 0.3
SASD-WS	$\mathbf{16.6 \pm 0.2}$	$\mathbf{13.5 \pm 0.3}$	$\mathbf{5.3 \pm 0.1}$	$\mathbf{34.1 \pm 0.2}$	$\mathbf{25.1 \pm 0.2}$	58.7 ± 0.5	$\mathbf{24.2 \pm 0.2}$	$\mathbf{34.7 \pm 0.2}$

Table 3. Targeted transfer attack success rates (%) of the adversarial examples generated by the normally trained ResNet-50 and that enhanced by our SASD and WS, respectively. We run each test five times and show the mean and standard deviation of the results.

Method	$\rightarrow$ Inc-v3	$\rightarrow$ Dense121	$\rightarrow$ VGG16
Pre-trained	7.7 ± 0.3	63.7 ± 0.1	56.2 ± 0.5
WS	35.0 ± 0.5	84.7 ± 0.2	77.8 ± 0.4
SASD	46.5 ± 0.5	91.2 ± 0.2	86.7 ± 0.1
SASD-WS	$\mathbf{70.3 \pm 0.8}$	$\mathbf{93.0 \pm 0.2}$	$\mathbf{88.7 \pm 0.4}$

Transformers (ViTs) [4] and CLIP models [32]. ViTs apply the transformer models to visual data and leverage the self-attention mechanism to process data patches. CLIP models utilize both the image and text encoders to encode and calculate the similarity of the image-text pairs. They can produce the similarity score for each label. In our experiments, we use Inc-v3$^{adv}_{ens3}$ [39], Inc-v3$^{adv}_{ens4}$ [39], and IncRes-v2$^{adv}_{ens}$ [39] as the target adversarially trained models. We consider ViT-B/16 and ViT-L/16 as the target ViTs. We target CLIP models with ResNet50, ViT-B/32, and ViT-L/14 as the image encoders, respectively. We use the cross-entropy loss as our attack objective function $\mathcal{L}$.

We report the results in Table 2. It shows that SASD-WS can generate more transferable targeted perturbations than the other baseline methods, with a significant improvement of **6.6%** on average. While most baselines can generate perturbations that can manifest a certain degree of targeted transferability, RAP performs poorly under the ensemble source model setting compared with the other baseline methods, with an average attack success rate of only 8.2%. We believe that the poor targeted attack performance of RAP under the ensemble source model setting is due to the ensemble model's lack of a uniformly low-loss region. Therefore, it highlights the need to enhance the source model to enable more transferable targeted adversarial attacks. Besides, our method's high targeted attack success rates against different target models, including adversarially trained models, reveal the potential security risk of DNNs and call for more effort to improve DNNs' robustness against adversarial perturbations.

Table 4. Targeted transfer attack success rates (%) when using different SASD variants: (1) SASD without fine-tuning the auxiliary model (SASD w/o FT Aux), (2) SASD without distilling the auxiliary model (SASD w/o D), (3) SASD via the commonly used knowledge distillation method (SASD w/ KD), and (4) our proposed SASD. We run each test five times and show the mean and standard deviation of the results.

Method	$\rightarrow$ Inc-v3	$\rightarrow$ Dense121	$\rightarrow$ VGG16
SASD w/o FT Aux	32.2 ± 0.3	87.1 ± 0.0	81.3 ± 0.1
SASD w/o D	29.6 ± 0.5	87.7 ± 0.2	79.5 ± 0.3
SASD w/ KD	38.5 ± 0.4	90.7 ± 0.1	84.9 ± 0.2
Our SASD	$\mathbf{46.5 \pm 0.5}$	$\mathbf{91.2 \pm 0.2}$	$\mathbf{86.7 \pm 0.1}$

4.3. Ablation Study

Validating the contribution of SASD and WS. We validate the contribution of two major components in our method: SASD and WS. Specifically, following the setting of Table 1, we compare the targeted attack success rates of the adversarial examples generated by the normally trained ResNet-50 and that enhanced by our SASD and WS, respectively. The results are shown in Table 3. We can see that both SASD and WS can improve the pre-trained source models' capability to launch targeted transfer attacks. Besides, combining SASD and WS can further enhance the source model to generate more transferable targeted adversarial perturbations.

Validating the design of SASD. We validate the design of SASD by examining SASD's two steps: fine-tuning the auxiliary model and distilling the auxiliary model. Specifically, for examining the first step of SASD, we conduct SASD without fine-tuning the auxiliary model. In other words, we conduct the second step of SASD directly with a pre-trained source model. For examining the second step of SASD, we conduct SASD without distilling the auxiliary model. In other words, we directly use the fine-tuned auxiliary model to generate adversarial examples. Besides, we also consider distilling the auxiliary model via the commonly used knowledge distillation method [13]. Specif-

Table 5. Targeted transfer attack success rates (%) against different defense methods. All methods are combined with the TI-DI-MI technique to generate adversarial perturbations. We run each test five times and show the mean and standard deviation of the results.

Method	JPEG	HGD	NRP	NoisyMix	AugMix
Pre-trained	99.7 ± 0.0	1.4 ± 0.1	3.7 ± 0.0	0.4 ± 0.0	0.7 ± 0.0
RAP	88.7 ± 0.0	0.5 ± 0.0	14.3 ± 0.0	1.0 ± 0.0	0.5 ± 0.0
GhostNet	$\mathbf{99.9 \pm 0.0}$	1.4 ± 0.0	12.4 ± 0.6	2.4 ± 0.0	0.8 ± 0.0
LGV	99.4 ± 0.1	7.6 ± 0.2	2.0 ± 0.3	6.5 ± 0.1	1.4 ± 0.0
SASD-WS	99.8 ± 0.0	$\mathbf{23.5 \pm 0.1}$	$\mathbf{17.9 \pm 0.5}$	$\mathbf{8.0 \pm 0.1}$	$\mathbf{2.0 \pm 0.0}$

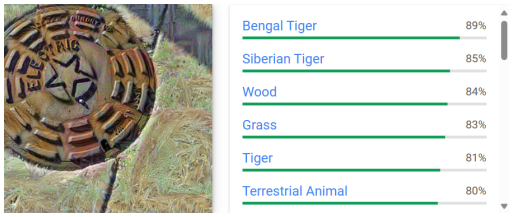


Figure 3. Successful targeted adversarial examples on Google Cloud Vision. The target class is tigers.

ically, the commonly used knowledge distillation method combines the cross-entropy loss with respect to the true label and the Kullback-Leibler divergence with respect to the prediction of the teacher model as the distillation loss. Following the setting of Table 1, we examine the performance of targeted adversarial perturbations generated by the above SASD variants. The results are shown in Table 4. When employing the normal knowledge distillation method, we distill the auxiliary model for ten epochs to maximize the attack performance. Our SASD instead only needs less than one epoch distillation. Nevertheless, our SASD method can still surpass the normal knowledge distillation method. Besides, the SASD model's performance improves when employing a fine-tuned auxiliary model, and distilling the auxiliary model performs better than directly using the fine-tuned auxiliary model. Therefore, the experimental results validate the design of our SASD.

4.4. Further Analysis

Transfer attacks against defense methods. In this part, we evaluate SASD-WS's targeted attack performance against several adversarial defense methods and common corruption defense methods, including JPEG [10], HGD [24], NRP [29], NoisyMix [5], and AugMix [12]. We use the same experimental settings as those in the ensemble targeted and untargeted attack experiments with an ensemble of source models (Inception-v3 [37], Inception-v4 [38], Inception-ResNet-v2 [38], ResNet-50 [11], ResNet-101 [11], ResNet-152 [11]).

Table 5 presents the results of the targeted transfer attack. We can see that SASD-WS can effectively generate effective perturbations against these defense methods in targeted attack settings. Even though for defense methods like NoisyMix and AugMix, most of the tested adversarial attack methods failed to achieve successful targeted attack, SASD-WS can still generate targeted perturbations with superior transferability compared with other baseline methods and achieves a nearly 100% attack success rate in attacking JPEG. These results show from another perspective that our approach can generate adversarial perturbations with higher transferability.

Table 6. Comparison of the average l_2-norm across 100 images for (1) the logit output difference of the ensemble of pruned models and the corresponding WS model, and (2) the standard deviation among the logit outputs of pruned models in the ensemble. Results for the ensemble of 20/50/100 pruned models are reported.

	20	50	100
Output Difference	30.71	29.54	29.57
Standard Deviation	26.87	26.66	26.36

Attacking real-world applications. We launch targeted transfer attacks against Google Cloud Vision to verify the practical applicability of the proposed method. Similar to previous experiments, we use an ensemble of six source models as the source model. We randomly sample 500 images from the NIPS17 targeted adversarial attack competition dataset as the test set to generate adversarial perturbations. Following the experimental settings in [26], when the labels with similar semantic meanings to the target label appear in the predicted labels of Google Cloud Vision, we regard the attack as successful. Although Google Cloud Vision will return the labels with probabilities more than 50.0%, we only consider the top-5 labels as the predicted labels.

Experimental results show that our method can achieve a targeted transfer attack success rate of 40.2%. In contrast, the baseline methods, LGV, GhostNet, RAP, and Pre-trained, can only attain a targeted transfer attack success rate of 31.8%,17.0%, 4.6%, and 3.6%, respectively. Figure 3 presents some of the adversarial examples generated by our method.

Validating WS. We further validate the effectiveness of our Weight Scaling (WS) method in simulating an ensemble of source models. Intuitively, if the output of the pruned models' ensemble is close to that of the corresponding WS model, the difference between their outputs should be within the standard deviation among the pruned models' outputs in the ensemble. Therefore, we calculate the l_2-norm value of the difference between the logit outputs of the pruned models' ensemble and the corresponding WS model. We also compute the l_2-norm value of the standard deviation among the pruned models' logit outputs in the ensemble. The results are shown in Table 6. According to the three-sigma rule, we can see that the output difference be-

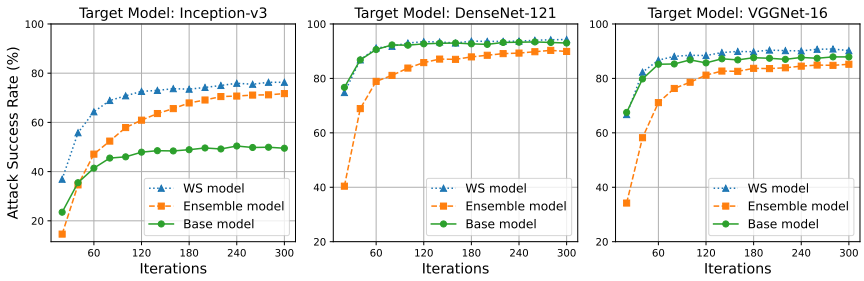
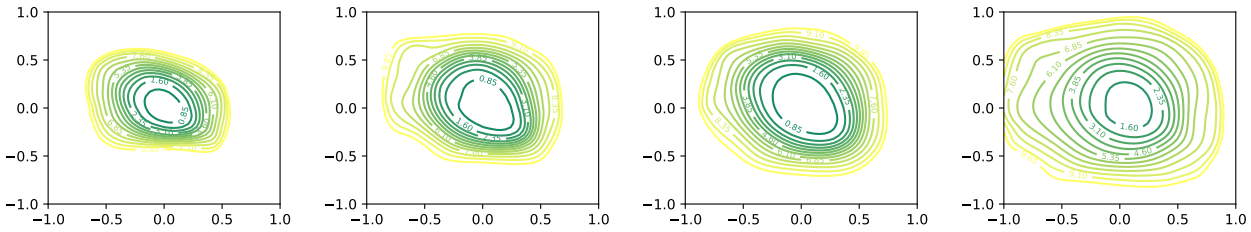


Figure 4. Targeted transfer attack success rates (%) of the base model (SASD ResNet-50), the longitudinal ensemble of pruned base models, and the WS version of the base model (SASD-WS ResNet-50).



(a) Pre-trained ResNet-50 (b) Fine-tuned ResNet-50 (c) SASD ResNet-50 (d) SASD-WS ResNet-50

Figure 5. The change of the loss landscapes with the application of our method. The fine-tuned ResNet-50 is the auxiliary model adopted in our SASD method.

tween the pruned models' ensemble and the corresponding WS model falls into the three-sigma range of the internal variability of the pruned models in the ensemble. Therefore, the output of the WS model is not an outlier among the pruned models' outputs, which implies that the output of the WS model is close to the ensemble model.

Moreover, we compare the targeted attack performance of a single weight-scaled SASD model and the ensemble of pruned SASD models. Specifically, we generate adversarial perturbations by employing a longitudinal ensemble of pruned SASD models with a pruning rate of $1 - p = 0.07$. Therefore, we set our WS method's scaling ratio $p = 0.93$. Figure 4 shows the targeted transfer attack success rates of three types of source models with SASD ResNet-50 as the base model. According to the attack performance, we can see that our WS method can reasonably approximate the ensemble of pruned models.

Visualizing the effectiveness of SASD-WS. Our SASD-WS attempts to promote the source model's generalization capability to enable more transferable targeted adversarial attacks. To visualize the effectiveness of our SASD-WS on improving the source model's generalization capability, we depict the change of the loss landscapes by applying our method in Figure 5. We can see that with the application of our approach, the loss landscape of the resultant enhanced model becomes flatter, which can help to capture general features learned by different models [6, 16, 35]. Therefore, the enhanced model is more likely to cover the transferable targeted adversarial examples in its low-loss region, facilitating more transferable targeted adversarial attacks.

5. Conclusion

This work presents a novel model self-enhancement method, incorporating Sharpness-Aware Self-Distillation (SASD) and Weight Scaling (WS) to enhance the source model's capability to generate more transferable targeted perturbations. Extensive experiments show that our method can significantly outperform the state-of-the-art approaches. Notably, under the black-box setting, we can surpass the state-of-the-art baselines by a significant margin of 12.2% on average. We also validate the design of our method by several ablation studies and the practical applicability of the proposed method by attacking a popular real-world application. We believe that our approach can serve as a strong benchmark when evaluating the robustness of DNNs and an effective patch when conducting adversarial training.

Acknowledgment

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 62206318).

References

[1] Yang Deng, Weibin Wu, Jianping Zhang, and Zibin Zheng. Blurred-dilated method for adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 58613–58624, 2023. 1

[2] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 3, 4, 5

[3] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2, 4

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6

[5] N Benjamin Erichson, Soon Hoe Lim, Winnie Xu, Francisco Utrera, Ziang Cao, and Michael W Mahoney. NoisyMix: Boosting model robustness to common corruptions. *arXiv preprint arXiv:2202.01263*, 2022. 7

[6] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020. 2, 3, 8

[7] Irving John Good. Rational decisions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 14(1):107–114, 1952. 3

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 3

[9] Martin Gubri, Maxime Cordy, Mike Papadakis, Yves Le Traon, and Koushik Sen. LGV: Boosting adversarial example transferability from large geometric vicinity. In *European Conference on Computer Vision*, pages 603–618, 2022. 2, 5

[10] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4, 5, 7

[12] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020. 7

[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 6

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 5

[15] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2019. 3

[16] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019. 1, 8

[17] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 2018. 2, 4

[18] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22 (1):79–86, 1951. 4

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. 5

[20] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations Workshop*, 2017. 2

[21] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231, 2018. 5

[22] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 641–649, 2020. 1

[23] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via Ghost Networks. In *AAAI Conference on Artificial Intelligence*, pages 11458–11465, 2020. 2, 5

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

[25] 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. 3

[26] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. 7

[27] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *European Conference on Computer Vision*, pages 549–566, 2022. 2

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 5

[29] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 262–271, 2020. 7

[30] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. On generating transferable targeted perturbations. In *IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2021. 1

[31] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *Advances in Neural Information Processing Systems*, pages 29845–29858, 2022. 3, 5

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 6

[33] 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*, 115(3): 211–252, 2015. 4, 5

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

[35] Jacob Springer, Melanie Mitchell, and Garrett Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. In *Advances in Neural Information Processing Systems*, pages 9759–9773, 2021. 1, 2, 3, 8

[36] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? In *Advances in Neural Information Processing Systems*, pages 6906–6919, 2021. 3

[37] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 4, 7

[38] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, pages 4278–4284, 2017. 4, 7

[39] 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*, 2018. 5, 6

[40] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *IEEE/CVF International Conference on Computer Vision*, pages 16158–16167, 2021. 2

[41] 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 *IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 125–137, 2019. 1

[42] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1158–1167, 2020. 1

[43] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2020. 1

[44] Weibin Wu, Yuxin Su, Michael R Lyu, and Irwin King. Improving the transferability of adversarial samples with adversarial transformations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9024–9033, 2021. 1

[45] Weibin Wu, Jianping Zhang, Victor Junqiu Wei, Xixian Chen, Zibin Zheng, Irwin King, and Michael R Lyu. Practical and efficient model extraction of sentiment analysis APIs. In *International Conference on Software Engineering*, pages 524–536, 2023. 3

[46] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2, 4

[47] Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, page 022022, 2019. 4

[48] Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14993–15002, 2022. 1

[49] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16415–16424, 2023. 1

[50] Jianping Zhang, Yung-Chieh Huang, Weibin Wu, and Michael R Lyu. Towards semantics- and domain-aware adversarial attacks. In *International Joint Conference on Artificial Intelligence*, pages 536–544, 2023. 1

[51] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In *Advances in Neural Information Processing Systems*, pages 6115–6128, 2021. 3, 5