Admix: Enhancing the Transferability of Adversarial Attacks

Xiaosen Wang[1] Xuanran He[2] Jingdong Wang[3] Kun He[1*]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology
[2]Wee Kim Wee School of Communication and Information, Nanyang Technological University
[3]Microsoft Research Asia
xiaosen@hust.edu.cn, xhe015@e.ntu.edu.sg, jingdw@microsoft.com, brooklet60@hust.edu.cn

Abstract

Deep neural networks are known to be extremely vulnerable to adversarial examples under white-box setting. Moreover, the malicious adversaries crafted on the surrogate (source) model often exhibit black-box transferability on other models with the same learning task but having different architectures. Recently, various methods are proposed to boost the adversarial transferability, among which the input transformation is one of the most effective approaches. We investigate in this direction and observe that existing transformations are all applied on a single image, which might limit the adversarial transferability. To this end, we propose a new input transformation based attack method called Admix that considers the input image and a set of images randomly sampled from other categories. Instead of directly calculating the gradient on the original input, Admix calculates the gradient on the input image admixed with a small portion of each add-in image while using the original label of the input to craft more transferable adversaries. Empirical evaluations on standard ImageNet dataset demonstrate that Admix could achieve significantly better transferability than existing input transformation methods under both single model setting and ensemble-model setting. By incorporating with existing input transformations, our method could further improve the transferability and outperforms the state-of-the-art combination of input transformations by a clear margin when attacking nine advanced defense models under ensemble-model setting. Code is available at `https://github.com/JHL-HUST/Admix`.

1. Introduction

A great number of works [7, 2, 1] have shown that deep neural networks (DNNs) are vulnerable to adversarial examples [31, 7], *i.e*. the malicious crafted inputs that are indistinguishable from the legitimate ones but can induce misclassification on the deep learning models. Such vulnerability poses potential threats to security-sensitive applications, *e.g*. face verification [28], autonomous driving [6] and has inspired a sizable body of research on adversarial attacks [22, 2, 21, 4, 16, 5, 38, 18]. Moreover, the adversaries often exhibit transferability across neural network models [25], in which the adversarial examples generated on one model may also mislead other models. The adversarial transferability matters because hackers may attack a real-world DNN application without knowing any information of the target model. However, under white-box setting where the attacker has complete knowledge of the target model, existing attacks [2, 11, 1, 21] have demonstrated great attack performance but with comparatively low transferability against models with defense mechanisms [21, 33], making it inefficient for real-world adversarial attacks.

To improve the transferability of adversarial attacks, various techniques have been proposed, such as advanced gradient calculations [4, 18, 35], ensemble-model attacks [19, 15], input transformations [38, 5, 18, 10] and model-specific methods [36]. The input transformation (*e.g*. randomly resizing and padding, translation, scale *etc*.) is one of the most effective approaches. Nevertheless, we observe that existing methods are all applied on a single input image. Since adversarial attacks aim to mislead the DNNs to classify the adversary into other categories, it naturally inspires us to explore whether we could further enhance the transferability by incorporating the information from other categories.

The *mixup* operation, that linearly interpolates two random images and corresponding labels, is firstly proposed as a data augmentation approach to improve the generalization of standard training [41, 34, 40]. Recently, *mixup* is also used for inference [24] or adversarial training [12, 14] to enhance the model robustness. Since *mixup* adopts the information of a randomly picked image, we try to directly adopt *mixup* to craft adversaries but find that the attack performance decays significantly under white-box setting with little improvement on transferability. To craft highly trans-

*Corresponding author.

ferable adversaries with the information from other categories but not harm the white-box attack performance, we propose a novel attack method called *Admix* that calculates the gradient on the admixed image combined with the original input and images randomly picked from other categories. Unlike *mixup* that treats the two images equally and mixes their labels accordingly, the *admix* operation adds a small portion of the add-in image from other categories to the original input but does not change the label. Thus *Admix* attack could obtain diverse inputs for gradient calculation.

Empirical evaluations on standard ImageNet dataset [26] demonstrate that, compared with existing input transformations [38, 5, 18], the proposed *Admix* attack achieves significantly higher attack success rates under black-box setting and maintains similar attack performance under white-box setting. By incorporating *Admix* with other input transformations, the transferability of the crafted adversaries could be further improved. Besides, the evaluation of the integrated method under the ensemble-model setting [19] against nine advanced defense methods [17, 37, 39, 20, 8, 3, 27, 23] demonstrates that the final integrated method, termed *Admix*-TI-DIM, outperforms the state-of-the-art SI-TI-DIM [18] by a clear margin of 3.4% on average, which further demonstrates the high effectiveness of *Admix*.

2. Related Work

In this section, we provide a brief overview of the adversarial attack methods and the *mixup* family.

2.1. Adversarial Attacks and Transferability

According to the threat model, existing attack methods can be categorized into two settings: a) white-box attack has full knowledge of the threat model, *e.g*. (hyper-)parameters, gradient, architecture, *etc*. b) black-box attack could only access to the model outputs or nothing about the threat model. In this work, we mainly focus on generating highly transferable adversaries without any knowledge of the target model, falling into the black-box setting.

Szegedy *et al*. [31] first point out the existence of adversarial examples for DNNs and propose a box-constrained L-BFGS method to find adversarial examples. To accelerate the adversary generation process, Goodfellow *et al*. [7] propose fast gradient sign method (FGSM) to generate adversarial examples with one step of gradient update. Kurakin *et al*. [11] further extend FGSM to an iterative version denoted as I-FGSM that exhibits higher attack success rates. Carlini *et al*. [2] propose a powerful optimization-based method by optimizing the distance between an adversary and the corresponding benign example. Though the above attacks have achieved remarkable attack performance under white-box setting, they often exhibit weak transferability.

Recently, some works focus on generating more transferable adversarial examples, which could be roughly split into four categories, namely ensemble-model attack, momentum-based attack, input transformation based attack and model-specific attack. Liu *et al*. [19] first propose an ensemble-model attack that attacks multiple models simultaneously so as to enhance the transferability. Li *et al*. [15] generate adversarial examples on multiple ghost networks obtained by perturbing the dropout and skip-connection layer. Some works focus on advanced gradient calculation that adopts momentum to generate more transferable adversaries. Dong *et al*. [4] integrate momentum into I-FGSM denoted as MI-FGSM and Lin *et al*. [18] adopt Nesterov's accelerated gradient, to further enhance the transferability.

Several input transformation methods have also been proposed to promote the transferability. Xie *et al*. [38] propose to adopt diverse input pattern by randomly resizing and padding for gradient calculation. Dong *et al*. [5] convolve the gradient with a pre-defined kernel which leads to higher transferability against models with defense mechanism. Lin *et al*. [18] calculate the gradient on a set of scaled images to enhance the transferability. Zou *et al*. [10] propose a three-stage pipeline to generate more transferable adversarial examples, namely resized-diverse-inputs, diversity-ensemble and region fitting. Wu *et al*. [36] find that utilizing more gradient of skip connections rather than the residual modules in ResNet [9] could enhance the transferability.

Note that the ensemble-model attack, momentum based attack, and input transformation based attack could be integrated with each other to achieve higher transferability. The proposed *Admix* falls into the input transformation category, and *Admix* can be combined with other input transformations as well as the other two types of attacks to further boost the transferability.

2.2. The Mixup Family

Zhang *et al*. [41] first propose a novel method called *mixup* to improve the model generalization by interpolating two randomly sampled examples (x, y) and (x', y') with $\lambda \in [0, 1]$ as follows:

$$\tilde{x} = \lambda \cdot x + (1 - \lambda) \cdot x', \;\; \tilde{y} = \lambda \cdot y + (1 - \lambda) \cdot y'. \quad (1)$$

Verma *et al*. [34] extend *mixup* to *manifold mixup* that leverages semantic interpolations as additional training signal, and obtain neural networks with smoother decision boundaries at multiple levels of representation. Yun *et al*. [40] further propose *cutmix* where the patches are cut and pasted among training images and the ground truth labels are also mixed proportionally to the area of patches.

As a powerful data augmentation strategy, *mixup* has also been used to enhance the robustness of deep models. Lamb *et al*. [12] propose *interpolated adversarial training* (IAT) that adopts the adversarial examples processed by *mixup* or *manifold mixup* for training. Pang *et al*. [24] propose *mixup inference* (MI) by mixing the input with other random clean

Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
MI-FGSM	100.0	43.6	42.4	35.7	13.1	12.8	6.2
Mixup	71.8	44.2	41.1	39.0	13.5	13.4	7.2

Table 1: Attack success rates (%) of MI-FGSM and *mixup* transformation. The adversaries are crafted on Inc-v3 model.

sample for inference. Lee *et al*. [14] propose *adversarial vertex mixup* (AVM) by mixing the clean example and adversarial example to enhance the robustness of PGD adversarial training [21]. Laugros *et al*. [13] combine *mixup* and *targeted labeling adversarial training* (TLAT) that interpolates the target labels of adversarial examples with the ground-truth labels.

3. Methodology

In this section, we first provide details of several adversarial attacks for enhancing the transferability to which our method is most related. Then we introduce the proposed *Admix* attack method and highlight the difference between the proposed *admix* operation and the existing *mixup* [41] operation designed for standard training.

3.1. Attacks for Enhancing the Transferability

Let $\mathcal{X}$ be the set of all digital images under consideration for a given learning task, $\mathcal{Y} \in \mathbb{R}$ be the output label space and $\mathcal{B}_\epsilon(x) = \{\bar{x} : \|x - \bar{x}\|_p \leq \epsilon\}$ denote the ℓ_p-norm ball centered at x with radius ϵ. Given a classifier $f(x;\theta) : x \in \mathcal{X} \rightarrow y \in \mathcal{Y}$ that outputs label y for the prediction of input x with model parameters θ, the goal of adversarial attack is to seek an example $x^{adv} \in \mathcal{B}_\epsilon(x)$ that misleads the target classifier $f(x;\theta) \neq f(x^{adv};\theta)$. To align with previous works, we focus on ℓ_∞-norm in this work.

Fast Gradient Sign Method (FGSM) [7] crafts adversarial example by adding perturbation in the gradient direction of the loss function $J(x, y; \theta)$ as follows:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y; \theta)),$$

where $\text{sign}(\cdot)$ denotes the sign function and $\nabla_x J(x, y; \theta)$ is the gradient of the loss function w.r.t. x.

Iterative Fast Gradient Sign Method (I-FGSM) [11] is an iterative version of FGSM by adding a small perturbation with step size α in the gradient direction at each iteration:

$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(\nabla_{x^{adv}_t} J(x^{adv}_t, y; \theta)), \;\; x^{adv}_0 = x.$$

Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [4] integrates the momentum term into I-FGSM and exhibits better transferability. The update procedure can be summarized as:

$$g_t = \mu \cdot g_{t-1} + \frac{\nabla_{x^{adv}_t} J(x^{adv}_t, y, ; \theta)}{\|\nabla_{x^{adv}_t} J(x^{adv}_t, y, ; \theta)\|_1},$$
$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(g_t).$$

Diverse Input Method (DIM) [38] is the first input transformation based attack which firstly resizes the input image to an $r \times r \times 3$ image where r is randomly sampled from $[299, 330)$ with a given probability p and pads the resized image into $330 \times 330 \times 3$. Then DIM feeds the transformed image to DNNs for gradient calculation.

Translation-Invariant Method (TIM) [5] calculates the average gradient on a set of translated images for the update. To further improve the efficiency, TIM approximately calculates the gradient by convolving the gradient of the untranslated image with a predefined kernel matrix instead of computing the gradient on a set of images.

Scale-Invariant Method (SIM) [18] discovers the scale invariance property of DNNs and calculates the average gradient over the scaled copies of the input for update:

$$\bar{g}_{t+1} = \frac{1}{m} \sum_{i=0}^{m-1} \nabla_{x^{adv}_t} (J(x^{adv}_t / 2^i, y; \theta)),$$

where m is the number of copies.

3.2. The Admix Attack Method

Lin *et al*. [18] analogize the adversary generation process to the neural model training process and the transferability of crafted adversarial example could be equivalent to the generalization of the trained model. Under such perspective, the input transformation could be treated as data augmentation. Various input transformations have been proposed that could boost the adversarial transferability, however, we observe that all the existing transformations are applied on the single input image. On the other hand, we observe that for standard training, *mixup*, which is a powerful data augmentation strategy by interpolating two randomly sampled examples, can effectively improve the model generalization [41, 32, 40]. This raises an intriguing question, *could we improve the attack transferability by adopting information from other images for the gradient calculation?*

However, as shown in Table 1, we find that directly applying *mixup* for the gradient calculation improves the transferability of crafted adversaries slightly but degrades the attack performance significantly under white-box setting. The main reason might be two-fold. First, there is no difference between x and x' for the *mixup* which might adopt too much information from the add-in image x' for the gradient calculation of the input x and thus provide incorrect direction for update. Second, *mixup* also mixes the labels which introduces the gradient of other category for update when x and x' are not in the same category.

Algorithm 1 The *Admix* Attack Algorithm

Input: A classifier f with loss function J and a benign example x with ground-truth label y

Input: The maximum perturbation ϵ, number of iterations T and decay factor μ

Input: The number of admixed copies m_1 and sampled images m_2, and the strength of sampled image η

Output: An adversarial example $x^{adv} \in \mathcal{B}_\epsilon(x)$

$\alpha = \epsilon/T$; $g_0 = 0$; $\bar{g}_0 = 0$; $x_0^{adv} = x$

for $t = 0 \to T-1$ **do**:

  Randomly sample a set X' of m_2 images from another category

  Calculate the average gradient $\bar{g}_{t+1}$ by Eq. (3)

  Update the enhanced momentum g_t:

$$g_{t+1} = \mu \cdot g_t + \frac{\bar{g}_{t+1}}{\|\bar{g}_{t+1}\|_1}$$

  Update x_{t+1}^{adv} by applying the gradient sign:

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})$$

end for

return $x^{adv} = x_T^{adv}$.

In order to utilize the information of images from other category without harming the white-box attack performance, we propose *admix* operation that admixes two images in a master and slave manner. Specifically, we takes the original image x as the primary image and admixes it with a secondary image x' randomly picked from other category:

$$\tilde{x} = \gamma \cdot x + \eta' \cdot x' = \gamma \cdot (x + \eta \cdot x'), \tag{2}$$

where $\eta = \eta'/\gamma$, $\gamma \in [0, 1]$ and $\eta' \in [0, \gamma)$ control the portion of the original image and the randomly sampled image in the admixed image respectively. In this way, we can assure that the secondary image x' always occupies a smaller portion in $\tilde{x}$. Note that we do not mix the labels, but instead use the original label of x for $\tilde{x}$.

With the above analysis, we propose an *Admix* attack method to improve the attack transferability, which calculates the average gradient on a set of admixed images $\{\tilde{x}\}$ of the input x by changing the value of γ or picking the add-in image x' from different categories in Eq. (2).

$$\bar{g}_{t+1} = \frac{1}{m_1 \cdot m_2} \sum_{x' \in X'} \sum_{i=0}^{m_1-1} \nabla_{x_t^{adv}} J(\gamma_i \cdot (x_t^{adv} + \eta \cdot x'), y; \theta), \tag{3}$$

where m_1 is the number of admixed images for each x' and X' denotes the set of m_2 randomly sampled images from other categories. Note that when $\eta = 0$, *Admix* will degenerate to SIM [18]. The proposed *Admix* could be integrated with any gradient-based attacks and other input transformation methods except for SIM. We summarize the algorithm of *Admix* integrated into MI-FGSM (denoted as *Admix* without ambiguity in the following) in Algorithm 1.

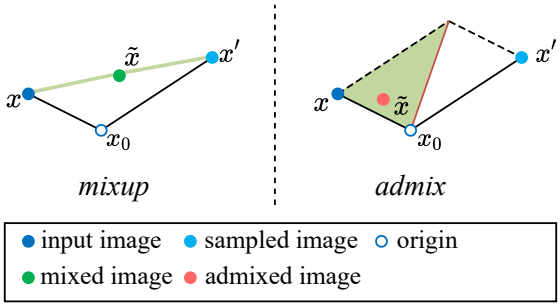


Figure 1: Illustration of the mechanisms in the input space of *mixup* and *admix*. x denotes the input image and x' the randomly sampled image. x_0 denotes the origin where all pixel values are 0s and $\tilde{x}$ is a possible transformed image. The green line and green triangle denotes all the possible transformed images by *mixup* and *admix*, respectively.

3.3. Differences between Admix and Mixup

For the two operations, *admix* and *mixup* [41], they both generate a mixed image from an image pair, x and x'. Here we summarize their differences as follows:

- The goal of *mixup* is to improve the generalization of the trained DNNs while *admix* aims to generate more transferable adversarial examples.
- The *mixup* treats x and x' equally and also mixes the label of x and x'. In contrast, *admix* treats x as the primary component and combines a small portion of x', at the same time maintains the label of x.
- As depicted in Figure 1, *mixup* linearly interpolates x and x' while *admix* does not have such constraint, leading to more diversed transformed images.

4. Experiments

In this section, to validate the effectiveness of the proposed approach, we conduct extensive empirical evaluations on the standard ImageNet dataset [26].

4.1. Experimental Setup

Dataset. We evaluate the proposed method on 1,000 images pertaining to 1,000 categories that are randomly sampled from the ILSVRC 2012 validation set [26] provided by Lin *et al*. [18].

Baselines. We adopt three competitive input transformations as our baselines, *i.e*. DIM [38], TIM [5] and SIM [18] and their combinations, denoted as SI-TIM, SI-DIM and SI-TI-DIM, respectively. All the input transformations are integrated into MI-FGSM [4].

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	DIM	99.0*	64.3	60.9	53.2	19.9	18.3	9.3
	TIM	**100.0***	48.8	43.6	39.5	24.8	21.3	13.2
	SIM	**100.0***	69.4	67.3	62.7	32.5	30.7	17.3
	Admix	**100.0***	**82.6**	**80.9**	**75.2**	**39.0**	**39.2**	**19.2**
Inc-v4	DIM	72.9	97.4*	65.1	56.5	20.2	21.1	11.6
	TIM	58.6	**99.6***	46.5	42.3	26.2	23.4	17.2
	SIM	80.6	**99.6***	74.2	68.8	47.8	44.8	29.1
	Admix	**87.8**	99.4*	**83.2**	**78.0**	**55.9**	**50.4**	**33.7**
IncRes-v2	DIM	70.1	63.4	93.5*	58.7	30.9	23.9	17.7
	TIM	62.2	55.4	97.4*	50.5	32.8	27.6	23.3
	SIM	84.7	81.1	99.0*	76.4	56.3	48.3	42.8
	Admix	**89.9**	**87.5**	**99.1***	**81.9**	**64.2**	**56.7**	**50.0**
Res-101	DIM	75.8	69.5	70.0	98.0*	35.7	31.6	19.9
	TIM	59.3	52.1	51.8	99.3*	35.4	31.3	23.1
	SIM	75.2	68.9	69.0	**99.7***	43.7	38.5	26.3
	Admix	**85.4**	**80.8**	**79.6**	**99.7***	**51.0**	**45.3**	**30.9**

Table 2: Attack success rates (%) on seven models under single model setting with various single input transformations. The adversaries are crafted on Inc-v3, Inc-v4, IncRes-v2 and Res-101 model respectively. * indicates white-box attacks.

Models. We study four popular normally trained models, *i.e*. Inception-v3 (Inc-v3) [30], Inception-v4 (Inc-v4), Inception-Resnet-v3 (IncRes-v3) [29] and Resnet-v2-101 (Res-101) [9] and three ensemble adversarially trained models, *i.e*. ens3-adv-Inception-v3 (Inc-v3$_{ens3}$), ens4-Inception-v3 (Inc-v3$_{ens4}$), ens-adv-Inception-ResNet-v2 (IncRes-v2$_{ens}$) [33]. In the following, we simply call the last three models as adversarially trained models without ambiguity. To further show the effectiveness of *Admix*, we consider nine extra advanced defense models that are shown to be robust against black-box attacks on ImageNet dataset, namely HGD [17], R&P [37], NIPS-r3[1], Bit-Red [39], FD [20], JPEG [8], RS [3], ARS [27] and NRP [23].

Attack setting. We follow the attack settings in [4] with the maximum perturbation of $\epsilon = 16$, number of iteration $T = 10$, step size $\alpha = 1.6$ and the decay factor for MI-FGSM $\mu = 1.0$. We adopt the Gaussian kernel with size 7×7 for TIM, the transformation probability $p = 0.5$ for DIM, and the number of copies $m = 5$ for SIM. For a fair comparison, we set $m_1 = 5$ with $\gamma_i = 1/2^i$ as in SIM, and randomly sample $m_2 = 3$ images with $\eta = 0.2$ for *Admix*.

4.2. Evaluation on Single Input Transformation

We first evaluate the attack performance of various single input transformations, namely DIM, TIM, SIM and the proposed *Admix* attack. We craft adversaries on four normally trained networks respectively and test them on all the seven considered models. The attack success rates, *i.e*. the misclassification rates of the corresponding models with adversaries as the inputs, are shown in Table 2. The models we attack are on rows and the models we test are on columns.

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

We can see that TIM exhibits the weakest transferability on normally trained models among four input transformations, but outperforms DIM on adversarially trained models. SIM achieves better transferability than DIM and TIM on both normally trained models and adversarially trained models. Compared with the three competitive baselines, *Admix* achieves much better transferability on all the models and maintains high attack success rates under white-box setting. For instance, both *Admix* and SIM achieve the attack success rates of 100% for white-box attack on Inc-v3 model, however for black-box attack, *Admix* achieves the attack success rates of 82.6% on Inc-v4 model and 39.0% on Inc-v3$_{ens3}$ model while the powerful baseline SIM only achieves 69.4% on Inc-v4 and 30.7% on Inc-v3$_{ens3}$.

4.3. Evaluation on Combined Input Transformation

Lin *et al*. [18] show that combining SIM with TIM and DIM could further boost the transferability of the crafted adversaries. Here we evaluate the generalization of *Admix* to other input transformations. Since SIM is a special case of *Admix*, we compare the attack success rates of TIM and DIM integrated with SIM and *Admix*, denoted as SI-DIM, SI-TIM, SI-TI-DIM, *Admix*-DIM, *Admix*-TIM and *Admix*-TI-DIM respectively. We summarize the results in Table 3.

In general, the transformations combined with *Admix* achieves better transferability than the ones combined with SIM on all models. Taking the adversaries crafted on Inc-v3 model for example, *Admix*-DIM outperforms SI-DIM with a clear margin of 4% $\sim$ 7%, *Admix*-TIM outperforms SI-TIM with a large margin of 8% $\sim$ 12 % and *Admix*-TI-DIM outperforms SI-TI-DIM with a clear margin of 5% $\sim$ 7%. Such remarkable improvements demonstrate the high effectiveness of the proposed method by adopting extra information from other categories for the gradient calculation.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	SI-DIM	98.9*	85.0	81.3	76.3	48.0	45.1	24.9
	Admix-DIM	**99.8***	**90.5**	**87.7**	**83.5**	**52.2**	**49.9**	**28.6**
Inc-v4	SI-DIM	89.3	98.8*	85.6	79.9	58.4	55.2	39.3
	Admix-DIM	**93.0**	**99.2***	**89.7**	**85.2**	**62.4**	**60.3**	**39.7**
IncRes-v2	SI-DIM	87.9	85.1	97.5*	82.9	66.0	59.3	52.2
	Admix-DIM	**90.2**	**88.4**	**98.0***	**85.8**	**70.5**	**63.7**	**55.3**
Res-101	SI-DIM	87.9	83.4	84.0	98.6*	63.5	57.5	42.0
	Admix-DIM	**91.9**	**89.0**	**89.6**	**99.8***	**69.7**	**62.3**	**46.6**

(a) Attack success rates (%) on seven models by SIM and *Admix* integrated with **DIM**.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	SI-TIM	**100.0***	71.8	68.6	62.2	48.2	47.4	31.3
	Admix-TIM	**100.0***	**83.9**	**80.4**	**74.4**	**59.1**	**57.9**	**39.2**
Inc-v4	SI-TIM	78.2	99.6*	71.9	66.1	58.6	55.4	45.1
	Admix-TIM	**87.4**	**99.7***	**82.3**	**77.0**	**68.1**	**65.3**	**53.1**
IncRes-v2	SI-TIM	84.5	82.2	**98.8***	77.4	71.6	64.7	61.0
	Admix-TIM	**90.2**	**88.2**	98.6*	**83.9**	**78.4**	**73.6**	**70.0**
Res-101	SI-TIM	74.2	69.9	70.2	**99.8***	59.5	54.5	42.8
	Admix-TIM	**83.2**	**78.9**	**80.7**	99.7*	**67.0**	**62.5**	**52.8**

(b) Attack success rates (%) on seven models by SIM and *Admix* integrated with **TIM**.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	SI-TI-DIM	99.1*	83.6	80.8	76.7	65.2	63.3	46.5
	Admix-TI-DIM	**99.9***	**89.0**	**87.0**	**83.1**	**72.2**	**71.1**	**52.4**
Inc-v4	SI-TI-DIM	87.9	98.7*	83.0	77.7	72.4	68.2	57.5
	Admix-TI-DIM	**90.4**	**99.0***	**87.3**	**82.0**	**75.3**	**71.9**	**61.6**
IncRes-v2	SI-TI-DIM	88.8	86.8	**97.8***	83.9	78.7	74.2	72.3
	Admix-TI-DIM	**90.1**	**89.6**	97.7*	**85.9**	**82.0**	**78.0**	**76.3**
Res-101	SI-TI-DIM	84.7	82.2	84.8	99.0*	75.8	73.5	63.4
	Admix-TI-DIM	**91.0**	**87.7**	**89.2**	**99.9***	**81.1**	**77.4**	**70.1**

(c) Attack success rates (%) on seven models by SIM and *Admix* integrated with **TI-DIM**.

Table 3: Attack success rates (%) on seven models under single model setting with various combined input transformations. The adversaries are crafted on Inc-v3, Inc-v4, IncRes-v2 and Res-101 model respectively. * indicates white-box attacks.

4.4. Evaluation on Ensemble-model Attack

Liu *et al*. [19] have shown that attacking multiple models simultaneously can improve the transferability of the generated adversarial examples. To further demonstrate the efficacy of the proposed *Admix*, we adopt the ensemble-model attack as in [4] by fusing the logit outputs of various models. The adversaries are generated on four normally trained models, namely Inc-v3, Inc-v4, IncRes-v2 and Res-101 using different input transformations and the integrated input transformations respectively. All the ensemble models are assigned with equal weights and we test the transferability of the adversaries on three adversarially trained models.

As shown in Table 4, *Admix* always achieves the highest attack success rates under both white-box and black-box settings no matter for single input transformation or integrated input transformation. Compared with single input transformation, *Admix* achieves higher attack success rate that is at least 6.7% higher than SIM, which achieves the best attack performance among the three baselines. When combined with DIM or TIM, *Admix* outperforms the corresponding baseline with a clear margin of at least 4%. When integrating *Admix* into the combination of DIM and TIM, even though SI-TI-DIM exhibits great attack performance, *Admix* can further improve the baseline for more than 2% on three adversarially trained models. This convincingly demonstrates the high efficacy of adopting the information from other categories to enhance the transferability.

4.5. Evaluation on Advanced Defense Models

To further show the effectiveness of our method, we consider nine extra advanced defense methods, *i.e*. the top-3 defense methods in the NIPS 2017 competition (HGD (rank-1) [17], R&P (rank-2) [37] and NIPS-r3 (rank-3)), three popular input transformation based defenses (Bit-Red [39], FD [20] and JPEG [8]), two certified defenses (RS [3] and ARS [27]) and a powerful denoiser (NRP [23]). The target model for Bit-Red, FG, JPEG and NRP is Inc-v3$_{ens3}$ and the other methods adopt the official models provided in the corresponding papers. From the above evaluations,

Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
DIM	99.4*	97.4*	94.9*	99.8*	58.1	51.1	34.9
TIM	99.8*	97.9*	95.2*	99.8*	62.2	56.8	48.0
SIM	99.9*	99.3*	98.3*	**100.0***	78.8	73.9	59.5
Admix	**100.0***	**99.6***	**99.0***	**100.0***	**85.5**	**80.9**	**67.8**
SI-DIM	**99.7***	98.9*	97.7*	99.9*	85.2	83.3	71.3
Admix-DIM	**99.7***	**99.5***	**98.9***	**100.0***	**89.3**	**87.8**	**79.0**
SI-TIM	**99.7***	99.0*	97.6*	**100.0***	87.9	85.2	80.4
Admix-TIM	**99.7***	**99.1***	**98.1***	**100.0***	**91.8**	**89.7**	**85.8**
SI-TI-DIM	99.6*	**98.9***	97.8*	99.7*	91.1	90.3	86.8
Admix-TI-DIM	**99.7***	**98.9***	**98.3***	**100.0***	**93.9**	**92.3**	**90.0**

Table 4: Attack success rates (%) on seven models under ensemble-model setting with various input transformations. The adversaries are crafted on the ensemble model, *i.e*. Inc-v3, Inc-v4, IncRes-v2 and Res-101. * indicates white-box attacks.

Attack	HGD	R&P	NIPS-r3	Bit-Red	FD	JPEG	RS	ARS	NRP	Average
SI-TI-DIM	91.4	88.0	90.0	75.7	88.0	93.2	69.2	46.4	77.1	79.9
Admix-TI-DIM	**93.7**	**90.3**	**92.4**	**80.1**	**91.9**	**95.4**	**74.9**	**51.4**	**80.7**	**83.3**

Table 5: Attack success rates (%) on nine extra models with advanced defense by SI-TI-DIM and *Admix*-TI-DIM respectively. The adversaries are crafted on the ensemble model, *i.e*. Inc-v3, Inc-v4, IncRes-v2 and Res-101.

SI-TI-DIM exhibits the best attack performance among all the baselines under the ensemble-model setting. Thus, we compare the proposed *Admix*-TI-DIM with SI-TI-DIM under the ensemble-model setting as in Sec. 4.4.

We can observe from Table 5 that the proposed *Admix*-TI-DIM achieves higher attack success rates on all the defense models than SI-TI-DIM and outperforms the baseline with a clear margin of 3.4% on average. In general, *Admix*-TI-DIM results in a larger margin compared with SI-TI-DIM when attacking more powerful defense methods. For instance, *Admix*-TI-DIM outperforms the baseline more than 5.7% and 5% on the models trained by randomized smoothing (RS) and adversarially randomized smoothing (ARS) that both provide certified defense.

4.6. Ablation Studies

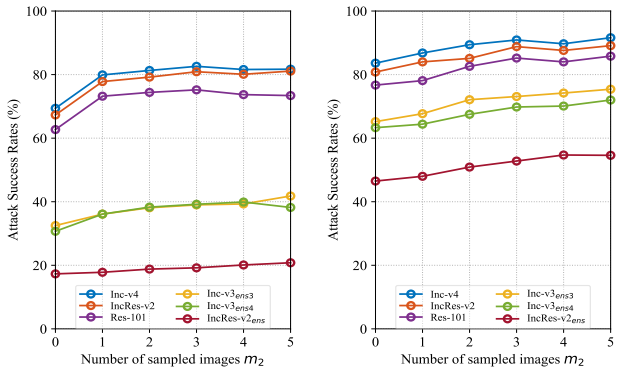
For hyper-parameter m_1, we follow the setting of SIM [18] for a fair comparison, and choose $m_1 = 5$. Here we conduct a series of ablation experiments to study the impact on *Admix* and *Admix*-TI-DIM on two other hyper-parameters, m_2 and η used in experiments.

On the number of sampled images x'. In Figure 2, we report the attack success rates of *Admix* and *Admix*-TI-DIM for various values of m_2 with adversaries crafted on Inc-v3 model, where η is fixed to 0.2. The attack success rates of *Admix* are 100% for all values of m_2 and that of *Admix*-TI-DIM are at least 99.7% under white-box setting. When $m_2 = 0$, *Admix* and *Admix*-TI-DIM degenerate to SIM and SI-TI-DIM respectively, and exhibit the weakest transferability. When $m_2 \leq 3$, the transferability on all models increases when we increase the value of m_2. When $m_2 > 3$, the transferability tends to decrease on normally trained models but still increases on adversarially trained models. Since a bigger value of m_2 indicates a higher computation cost, we set $m_2 = 3$ to balance the computational cost and attack performance.

On the admixed strength of sampled image x'. In Figure 3, we report the attack success rates of *Admix* and *Admix*-TI-DIM for various values of η with adversaries crafted on Inc-v3 model, where m_2 is fixed to 3. The attack success rates of *Admix* and *Admix*-TI-DIM are at least 99.9% and 99.7% for various values of η respectively under white-box setting. When $\eta = 0$, *Admix* and *Admix*-TI-DIM also degenerate to SIM and SI-TI-DIM respectively, which exhibit the weakest transferability. When we increase η, the transferability increases rapidly and achieves the peak when $\eta = 0.2$ for *Admix* on adversarially trained models and $\eta = 0.2$ or $\eta = 0.25$ for *Admix*-TI-DIM on all models. In general, we set $\eta = 0.2$ for better performance.

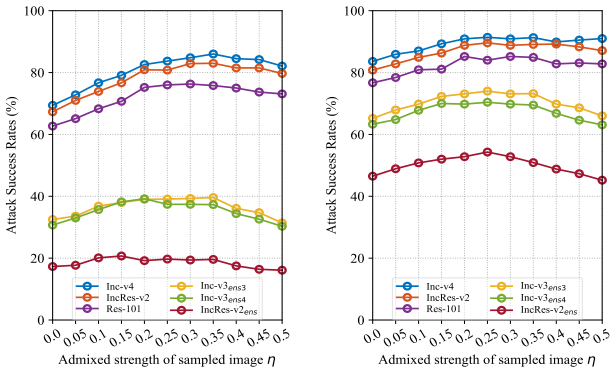
4.7. Discussion

The *admix* operation adds a small portion of the sampled image from other category to the original input but does not mix the labels. To verify the effectiveness of the two strategies, we evaluate the performance of *mixup* without label mixing and *admix* with label mixing for attacks, termed *Mixup*$_{wlm}$ and *Admix*$_{lm}$ respectively. As shown in Figure 4, *Mixup*$_{wlm}$ achieves better attack performance than *Mixup*, which validates our hypothesis that the fact *Mixup* mixes the labels and hence introduces the gradient of other category will weaken the attack performance. We also see that *Admix*$_{lm}$ achieves higher attack success rates than *Mixup* and *Mixup*$_{wlm}$, highlighting the importance that the input image should be dominant in the mixed image. Thus, *Admix* lets the input image be the dominant but does not mix the labels, and achieves the best attack performance.



(a) *Admix* (b) *Admix*-TI-DIM

Figure 2: Attack success rates (%) on the other six models with adversaries crafted by *Admix* and *Admix*-TI-DIM on Inc-v3 model for various number of sampled images, m_2.



(a) *Admix* (b) *Admix*-TI-DIM

Figure 3: Attack success rates (%) on the other six models with adversaries crafted by *Admix* and *Admix*-TI-DIM on Inc-v3 model for various strength of the sampled image, η.

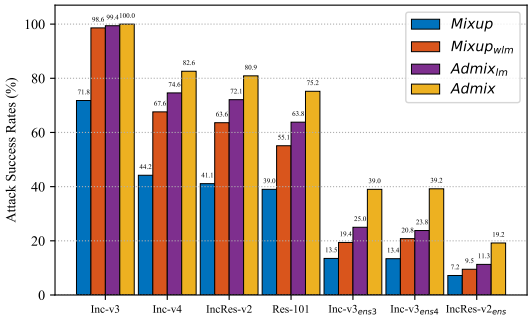


Figure 4: Attack success rates (%) on seven models with adversaries crafted by *Mixup*, *Mixup*$_{wlm}$, *Admix*$_{lm}$ and the proposed *Admix* on Inc-v3 model.

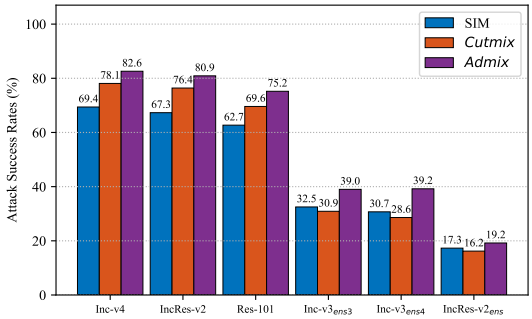


Figure 5: Attack success rates (%) on the other six models with adversaries crafted by SIM, *Cutmix* and the proposed *Admix* on Inc-v3 model.

We further provide a brief discussion on why *Admix* helps craft more transferable adversarial examples. *Admix* adds a small portion of the image from other class that moves the data point towards other class, *i.e*. closer to the decision boundary. We hypothesize that *Admix* utilizes the data points closer to the decision boundary for gradient calculation so that it could obtain more accurate update direction. Similar strategy has also been used by NI-FGSM [18] which looks ahead for the gradient calculation. To verify this hypothesis, we adopt *Cutmix* input transformation [40], which randomly cuts a patch of input image and pastes a patch from another image designed for standard training, with the same procedure as *Admix*. Note that *Cutmix* is not an interpolation of two images and does not guarantee the data point to be closer to the decision boundary. As shown in Figure 5, *Cutmix* exhibits better transferability on normally trained models but lower transferability on adversarially trained models when compared with SIM, and both *Cutmix* and SIM exhibit much lower transferability than *Admix*. This indicates that adopting information from other categories cannot always enhance the transferability and validate the hypothesis.

5. Conclusion

In this work, we propose a novel input transformation method called *Admix* to boost the transferability of the crafted adversaries. Specifically, for each input images, we randomly sample a set of image from other categories and admix a minor portion for each sampled image into the original image to craft a set of diverse images but using the original label for the gradient calculation. Extensive evaluations demonstrate that the proposed *Admix* attack method could achieve much better adversarial transferability than the existing competitive input transformation based attacks while maintaining high attack success rates under white-box setting. In our opinion, the *admix* operation is a new paradigm of data argumentation for adversarial learning in which the admixed images are closer to the decision boundary, offering more transferable adversaries. We hope our *Admix* attack that adopts information from other categories will shed light on potential directions for adversarial attacks.

Acknowledgements: This work is supported by National Natural Science Foundation (62076105) and Microsft Research Asia Collaborative Research Fund (99245180).

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *International Conference on Machine Learning (ICML)*, 2018.

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

[3] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning (ICML)*, 2019.

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

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

[6] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Models. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015.

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

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[10] Zou. Junhua, Zhisong Pan, Junyang Qiu, Xin Liu, Ting Rui, and Wei Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. *European Conference on Computer Vision (ECCV)*, 2020.

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations (ICLR), Workshop Track Proceedings*, 2017.

[12] Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 95–103, 2019.

[13] Alfred Laugros, Alice Caplier, and Matthieu Ospici. Addressing neural network robustness with mixup and targeted labeling adversarial training. *European Conference on Computer Vision (ECCV) Workshops*, 2020.

[14] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan L. Yuille. Learning transferable adversarial examples via ghost networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 11458–11465, 2020.

[16] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *International Conference on Machine Learning (ICML)*, 2019.

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

[18] 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 (ICLR)*, 2020.

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

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

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

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[23] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 262–271, 2020.

[24] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. *International Conference on Learning Representations (ICLR)*, 2020.

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

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large

scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115(3):211–252, 2015.

[27] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11292–11303, 2019.

[28] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, 2016.

[29] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2818–2826, 2016.

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

[32] Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5486–5494, 2018.

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

[34] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, 2019.

[35] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[36] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations (ICLR)*, 2020.

[37] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *International Conference on Learning Representations (ICLR)*, 2018.

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

[39] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *Network and Distributed System Security Symposium (NDSS)*, 2018.

[40] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.

[41] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations (ICLR)*, 2018.