Removing Adversarial Noise in Class Activation Feature Space

Dawei Zhou[1], Nannan Wang[1]*, Chunlei Peng[1], Xinbo Gao[2], Xiaoyu Wang[3], Jun Yu[4], Tongliang Liu[5]
[1]Xidian University, [2]Chongqing University of Posts and Telecommunications
[3]The Chinese University of Hong Kong (Shenzhen), [4]University of Science and Technology of China
[5]The University of Sydney
dwzhou.xidian@gmail.com, {nnwang,clpeng}@xidian.edu.cn, gaoxb@cqupt.edu.cn
fanghuaxue@gmail.com, harryjun@ustc.edu.cn, tongliang.liu@sydney.edu.au

Abstract

Deep neural networks (DNNs) are vulnerable to adversarial noise. Pre-processing based defenses could largely remove adversarial noise by processing inputs. However, they are typically affected by the error amplification effect, especially in the front of continuously evolving attacks. To solve this problem, in this paper, we propose to remove adversarial noise by implementing a self-supervised adversarial training mechanism in a class activation feature space. To be specific, we first maximize the disruptions to class activation features of natural examples to craft adversarial examples. Then, we train a denoising model to minimize the distances between the adversarial examples and the natural examples in the class activation feature space. Empirical evaluations demonstrate that our method could significantly enhance adversarial robustness in comparison to previous state-of-the-art approaches, especially against unseen adversarial attacks and adaptive attacks.

1. Introduction

Deep neural networks (DNNs) are known to be vulnerable to adversarial examples. Adversarial examples are maliciously crafted by adding imperceptible but adversarial noise on natural examples [11, 35, 15, 23, 26, 38]. The vulnerability of DNNs poses a potential threat to many decision-critical deep learning applications, such as image processing [21, 14, 47, 33, 17, 27] and natural language processing [34]. Thus, it is important to find an effective defense against adversarial noise.

Previous researches show that adversarial robustness of target models could be enhanced by processing inputs with certain transformations [12, 7, 29, 13, 23]. However, pre-processing based defenses may suffer from the *error amplification effect*, in which small residual adversarial noise is amplified to a large perturbation in internal layers of the target model and leads to misleading predictions [23]. Furthermore, these pre-processing based approaches are shown to be less effective in front of unseen adversarial attacks [39, 6, 45] as the adversarial perturbations of adversarial examples they used may not be the maximum in internal layers (see Section 4).

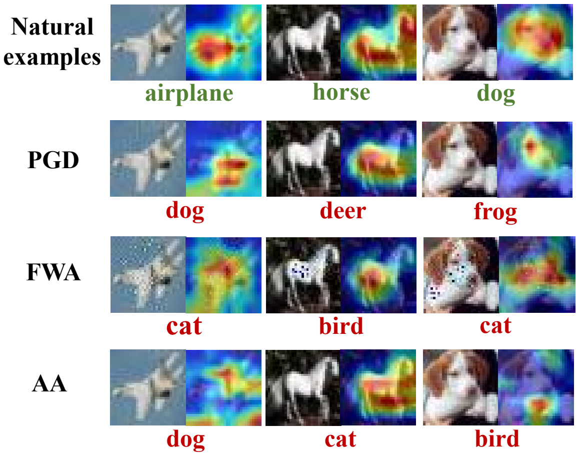

Figure 1. A visual illustration of class activation maps of natural examples and adversarial examples. The adversarial examples are crafted by distinct types of non-targeted attacks, e.g., PGD [28], FWA [39] and AA [6]. Although adversarial noise is imperceptible in pixel level, there exists obvious discrepancies between the class activation maps of natural examples and adversarial examples.

The *class activation mapping* technique [48] gives us an inspiration to solve this problem. Given a classification network, the class activation mapping technique could identify the importance of image regions by projecting back the class weights of the output layer on to the last convolutional features and performing a linear sum of the weighted features [48]. We find that although adversarial noise is imperceptible in pixel level, there exists obvious discrepancies

[*] Corresponding author

between the class activation maps of natural examples and adversarial examples crafted by existing attack approaches (see Figure 1). In addition, the weighted features are in the high-level layer of the network where small residual noise could cause large perturbations. This motivates us to handle the issue of error amplification effect by designing a defense method which focuses on the weighted features called as *class activation features*.

In this paper, we propose an adversarial training mechanism to remove adversarial noise by exploiting class activation features. In a high level, we design a max-min formula in the class activation feature space to learn a denoising model in a self-supervised manner without seen types of adversarial examples and ground-truth labels. Specifically, we first craft adversarial examples by maximally disrupting the class activation features of natural examples. The discrepancies of class activation features make adversarial examples have different prediction results from natural examples. We name such attack as *class activation feature based attack* (CAFA). Then, we train a denoising model, namely *class activation feature based denoiser* (CAFD), to remove adversarial noise. Instead of directly utilizing pixel-level loss functions to train our model, we minimize the distances between the class activation features of the natural examples and the adversarial examples. Finally, an image discriminator is introduced to make restored examples close to the natural examples by enhancing the fine texture details.

Achieved by such self-supervised adversarial training, our defense method could provide more significant protections against unseen types of attacks and adaptive attacks compared to previous defenses, which is empirically verified in Section 4.2. Furthermore, additional evaluations on ablation study and robustness of our model to the perturbation budget in Section 4.3 further demonstrate the effectiveness of our method.

The main contributions in this paper are as follows:

- We find that although adversarial noise is imperceptible in pixel level, it significantly disrupts the class activation features of natural examples. To this end, we design a *class activation features based denoiser* (CAFD) to effectively remove adversarial noise by exploiting class activation features.

- An self-supervised adversarial training mechanism is proposed to train the denoiser. We maximally disrupting the class activation features of natural examples to craft adversarial examples, and use them to train the denoiser for learning to minimize the distances between natural and adversarial examples in the class activation feature space.

- Empirical experiments show that our method could enhance adversarial robustness and it could be transferred across different target models. Particularly, the success rates of unseen attacks and adaptive attacks are reduced significantly in comparison to previous state-of-the-art approaches.

The rest of this paper is organized as follows. In Section 2, we briefly review related work on attacks and defenses. In Section 3, we describe our defense method and present its implementation. Experimental results on different datasets are provided in Section 4. Finally, we conclude this paper in Section 5.

2. Related work

Attacks: Adversarial examples have been shown to mislead DNNs [35] and transfer across different target models [25]. They can be crafted by single-step or multi-step attacks following the direction of adversarial gradients under a L_p norm perturbation budget. Attacks based on this strategy include fast gradient sign method (FGSM) [11], basic iterative attack (BIA) [20], the strongest first-order information based projected gradient descent (PGD) method [28], Carlini and Wagner (CW) method [4], decoupling direction and norm (DDN) method [31] and the autoattack (AA) method [6]. Rather than optimizing the objective function at a single point, the translation-invariance input diversity method (TI-DIM) [9, 46] uses a set of translated images to optimize an adversarial example. In addition, unlike these pixel-constrained attacks which do not consider the semantic or geometric information, spatially-constrained attacks focus on mimicking non-suspicious vandalism via geometry and spatial transformation, e.g., faster wasserstein attack (FWA) [39] and spatial transform attack (STA) [45].

Defenses: Adversarial training (AT) is an extensive strategy for defending adversarial noise [11, 37, 20]. It is dedicated to train a robust model by augmenting the training data with adversarial examples [28]. For example, the defensing against occlusion attacks (DOA) [41] method uses adversarial examples crafted by occlusion attacks to enhance the backbone model's robustness. The channel-wise activation suppressing (CAS) [2] strategy suppresses redundant activations from being activated by adversarial perturbations during the adversarial training process. Adversarial training could improve the accuracy of the target model on adversarial examples, but it typically cannot directly be transferred to other models or tasks.

Pre-processing based methods process the inputs to achieve robustness against adversarial noise. For example, JPEG compression [13] and total variation minimization (TVM) [13] were proposed to remove high-frequency components and small localized changes respectively. Jin *et al.* [15] proposed APE-G to back adversarial examples close to natural examples via a generative adversarial network. A high-level representation guided denoiser (HGD) [23] method was utilized as a pre-processing step to remove adversarial noise. The method in [49] removes adversarial noise by disentangling attack-invariant features from adversarial noise. Different from the above methods, we combine the benefits of adversarial training and pre-processing, and design an denoising model that remove adversarial noise by minimizing the distances between the class activation features of natural and adversarial examples in a self-supervised adversarial training manner.

3. Methodology

3.1. Preliminaries

In this paper, we aim to design a pre-processing based defense which could mitigate the error amplification effect and provide robust protections. The basic intuition behind our defense is to effectively exploit the class activation features of DNNs which could be significantly disrupted by adversarial noise. Towards this end, we design a *class activation features based denoiser* (CAFD) which learns to remove adversarial noise in the class activation feature space. To train our CAFD, we propose a self-supervised adversarial training mechanism without using seen types of adversarial examples and ground-truth labels. As shown in Figure 2, the training procedure can be regarded as a max-min formula and it is expressed as in the following:

For a given natural example x, let Φ_x represent its class activation features obtained from a pretrained deep neural network $\mathcal{P}$. We first craft an adversarial example $\tilde{x}$ by maximally disrupting Φ_x (Section 3.2). Then, at the minimization step, a class activation feature based denoiser $\mathcal{C}$ tries to remove the adversarial noise by minimizing the discrepancies between Φ_x and $\Phi_{\tilde{x}}$, where $\Phi_{\tilde{x}}$ denotes the class activation feature of $\tilde{x}$. In addition, to further enhance fine texture details of restored examples, we introduce an image discriminator $\mathcal{D}$ to play a game with $\mathcal{C}$ (Section 3.3).

3.2. Crafting adversarial examples

Our defense model is trained in an adversarial manner without utilizing seen types of adversarial examples and ground-truth labels. The adversarial examples used for self-supervised training are obtained by *class activation feature based attack* (CAFA). Below, we first outline class activation features and the impact of disrupting class activation features. Then, we describe the procedure of CAFA.

Class activation features: Given a pretrained deep neural network $\mathcal{P}$, the class activation mapping technique [48] projects back class weights of the output layer of $\mathcal{P}$ on to the last convolutional features and performs a linear sum of the weighted features. To be specific, for a given example x, its predicted probability of class c is $p(c|x)$. $c_x = \arg\max_c p(c|x)$ is the predicted class of x. We first use f_x^k to represent the deep feature of x of the $k-th$ channel in the last convolutional layer of $\mathcal{P}$. Then, for class c_x, the weighted feature of the $k-th$ channel is $\phi_x^k = f_x^k \cdot w_x^k$, where w_x^k is the class weight of the $k-th$ channel corresponding to class c_x. Essentially, w_x^k indicates the importance of f_x^k for c_x [48]. By linear summation of all ϕ_x^k, we could get a class activation map of x. We name the weighted features for all K channels as *class activation features* which are denoted by $\Phi_x = \left[\phi_x^1, \phi_x^2, \ldots, \phi_x^K\right]^\top$. Intuitively, the class activation features could be expressed as $\Phi_x = F_x \cdot W_x$, where $F_x{=}\left[f_x^1, f_x^2, \ldots, f_x^K\right]^\top$ and $W_x{=}\left[w_x^1, w_x^2, \ldots, w_x^K\right]^\top$ are the deep features and class weights for all K channels respectively.

Disruptions to class activation features: We note that there exists obvious discrepancies between class activation maps of natural examples and adversarial examples crafted by existing attacks. Since class activation maps is the liner sum of class activation features, the discrepancies indicate that adversarial noise could significantly disrupt the class activation features. This is similar to the phenomenon described in the error amplification effect that residual adversarial noise could cause large perturbations in internal layers of a target model. The reason why we use class activation features to craft adversarial examples is that the disruptions to class activation features could directly impact the effect of misleading the target model. To show this, we conduct a proof-of-concept experiment.

We define the *feature distance* to measure the disruptions to the class activation features of a natural example x as follows:

$$\Delta(x, \tilde{x}) = \delta(\Phi_x, \Phi_{\tilde{x}}), \tag{1}$$

where Φ_x and $\Phi_{\tilde{x}}$ are the class activation features of x and its adversarial example $\tilde{x}$ respectively. $\delta(\cdot)$ denotes a L_2-norm distance metric. As shown in Figure 3, we implement three classic attack methods, BIA [20], PGD [28] and CW [4]. The results show that the fooling rates and the feature distances have the same changing trend. This indicates that the disruptions to class activation features could directly impact the attack effect. Therefore, maximizing the objective in Eq. 1 could craft strong adversarial examples and enable an effective defense model.

Class activation feature based attack: Based on above observations, we design a class activation feature based attack (CAFD) method. Our method aims to find strong adversarial examples in class activation feature space by solving the following optimization problem:

$$\begin{gathered}\max_{\tilde{x}} \Delta(x, \tilde{x}) = \delta(\Phi_x, \Phi_{\tilde{x}}), \\ \text{subject to: } \|x - \tilde{x}\|_\infty \leq \epsilon,\end{gathered} \tag{2}$$

where ϵ denotes the perturbation budget. Our attack method is summarized in Algorithm 1. Given natural examples x, we first initialize adversarial examples $\tilde{x}_0$ as x. Then, we

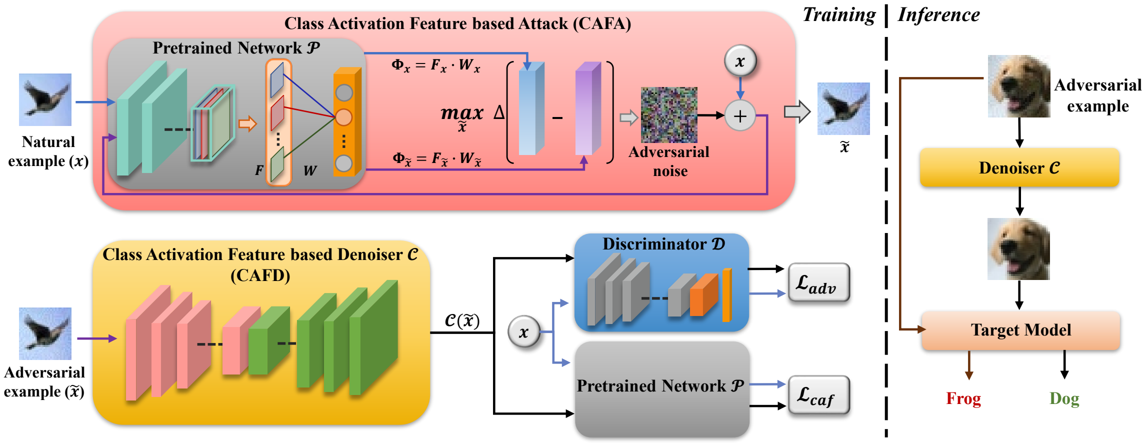
Class Activation Feature based Attack (CAFA)
Pretrained Network $\mathcal{P}$
$\Phi_x = F_x \cdot W_x$
$\max_{\tilde{x}} \Delta$
$\Phi_{\tilde{x}} = F_{\tilde{x}} \cdot W_{\tilde{x}}$
F
W
x
Adversarial noise
Natural example (x)
$\tilde{x}$
Training
Inference
Adversarial example
Denoiser $\mathcal{C}$
Target Model
Frog
Dog
Class Activation Feature based Denoiser $\mathcal{C}$ (CAFD)
Adversarial example ($\tilde{x}$)
$\mathcal{C}(\tilde{x})$
x
Discriminator $\mathcal{D}$
$\mathcal{L}_{adv}$
Pretrained Network $\mathcal{P}$
$\mathcal{L}_{caf}$

Figure 2. A visual illustration of our defense method. The proposed defense learns to remove adversarial noise via a self-supervised adversarial training mechanism. We maximally disrupt the class activation features of natural examples to craft adversarial examples and use them to train the denoiser for learning to bring adversarial examples close to natural examples in the class activation feature space.

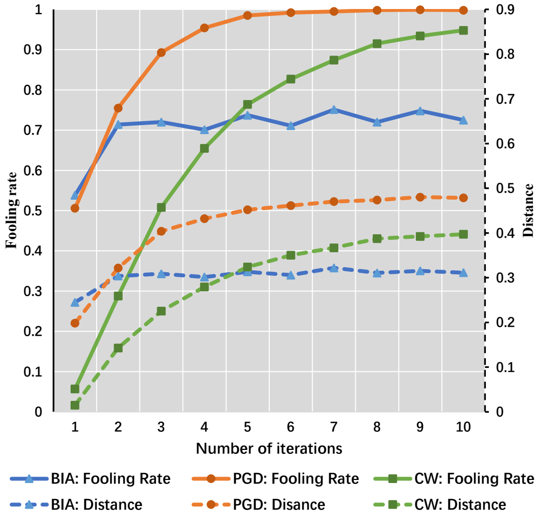

Figure 3. The average feature distances and fooling rate of adversarial examples against a VGG-19 [33] target model on *CIFAR-10*. The adversarial examples are respectively crafted by BIA[20], PGD [28] and CW [4]. It could be seen from the figure that the fooling rates rise synchronously as the feature distances increase.

forward x and $\tilde{x}_t$ to the pretrained deep neural network $\mathcal{P}$. and obtain their class activation features Φ_x and $\Phi_{\tilde{x}_t}$. Next, we compute the feature distance $\Delta(x, \tilde{x}_t)$ and its gradients using Eq. 2 and Eq. 3. Finally, we take the gradients to update $\tilde{x}_t$ and obtain $\tilde{x}_{t+1}$ using Eq. 4 and Eq. 5. By iteratively executing such update procedure, Algorithm 1 could maximize $\Delta(x, \tilde{x})$ and output an adversarial example $\tilde{x}$.

3.3. Removing adversarial noise

We design a class activation feature based denoiser (CAFD) $\mathcal{C}$ to remove adversarial noise. To train the denoiser, we use a hybrid loss function, which consists of a class activation feature loss and an adversarial loss.

Algorithm 1 CAFA: Class Activation Feature based Attack

```
Input: A pretrained deep neural network $\mathcal{P}$, natural example $x$,
    perturbation budget $\epsilon$, number of iterations $T$ and attack step
    size $\alpha$.
Output: An adversarial example $\tilde{x}$ with $\|x - \tilde{x}\|_\infty \leq \epsilon$.
$\tilde{x}_0 \leftarrow x$;
for $t = 0$ to $T - 1$ do
    Forward $x$ and $\tilde{x}_t$ to $\mathcal{P}$, and obtain class activation features
    $\Phi_x$ and $\Phi_{\tilde{x}_t}$;
    Compute the feature distance $\Delta(x, \tilde{x}_t)$ using Eq. 1;
    Compute gradients $w.r.t$ inputs:
        $g_t = \nabla_x \Delta(x, \tilde{x}_t);$    (3)
    Update the adversarial example $\tilde{x}_t$:
        $\tilde{x}_{t+1} = \tilde{x}_t + \alpha \cdot \mathrm{sign}(g_t);$    (4)
    Project $\tilde{x}_{t+1}$ to the vicinity of $x$:
        $\tilde{x}_{t+1} = \mathrm{clip}(\tilde{x}_{t+1}, x - \epsilon, x + \epsilon);$    (5)
end for
return $\tilde{x} = \tilde{x}_T$.
```

Class activation feature loss: Adversarial examples crafted by CAFA directly disrupt the class activation features and thus lead to misleading predictions. In order to effectively protect target models, the denoiser needs to learn to reduce the distances between natural examples and adversarial examples in the class activation feature space. The class activation feature loss could be defined as follows:

$$\mathcal{L}_{caf} = \delta(\Phi_x, \Phi_{\mathcal{C}(\tilde{x})}), \tag{6}$$

where $\Phi_{\mathcal{C}(\tilde{x})}$ denotes the class activation features of the restored example $\mathcal{C}(\tilde{x})$, and $\delta(\cdot)$ denotes the L_2-norm distance metric. Considering that Φ_x is the dot product of deep features F_x and class weights W_x, we could also achieve this optimization goal by jointly reducing the distances of deep features and class weights between the natural example and

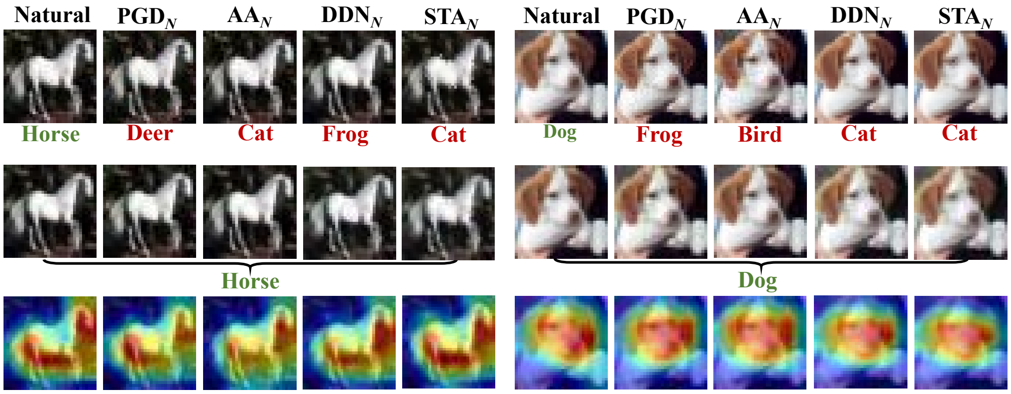

Figure 4. A visual illustration of the defense effect of our model against various types of attacks. (*top:* adversarial examples; *middle:* restored examples; *bottom:* class activation maps of restored examples). Subscripts *"N"* indicates that the corresponding attacks are non-targeted attacks.

Algorithm 2 CAFD: Class Activation Feature based Denoiser

```
Input: Training data $X$, pretrained deep neural network $\mathcal{P}$ and perturbation budget $\epsilon$.
1: repeat
2:     Simple natural example $x$ from $X$;
3:     Craft adversarial example $\tilde{x}$ at the given perturbation budget $\epsilon$ by utilizing Algorithm 1;
4:     Forward-pass $\tilde{x}$ through $\mathcal{C}$ and calculate $\mathcal{L}_{caf}$ (Eq. 6 or Eq. 7);
5:     Forward-pass $\mathcal{C}(\tilde{x})$ through $\mathcal{D}$, then calculate $\mathcal{L}_D$ (Eq. 8) and $\mathcal{L}_{adv}$ (Eq. 9);
6:     Back-pass and update $\mathcal{D}$, $\mathcal{C}$ to minimize $\mathcal{L}_C$ (Eq. 10) and $\mathcal{L}_D$ (Eq. 8);
7: until $\mathcal{C}$ and $\mathcal{D}$ converge.
```

its adversarial example. The class activation feature loss could be modified as follows:

$$\mathcal{L}_{caf} = \delta(F_x, F_{\mathcal{C}(\tilde{x})}) + \delta(W_x, W_{\mathcal{C}(\tilde{x})}), \quad (7)$$

where $F_{\mathcal{C}(\tilde{x})}$ denotes the deep features of $\tilde{x}$ and $W_{\mathcal{C}(\tilde{x})}$ denotes the class weights of $\tilde{x}$. We use Eq. 6 and Eq. 7 respectively to train $\mathcal{C}$ and present their results in Section 4.2. In addition, we empirically observe that removing $\mathcal{L}_{caf}$ loss would result in a significant decrease in the defense effect of the denoiser (See Figure 5).

Adversarial loss: We introduce an image discriminator $\mathcal{D}$ to enhance the fine texture details of restored examples in the manner of a relativistic average generative adversarial network [16]. Compared with a standard generative adversarial network (SGAN), the relativistic average generative adversarial network (RaGAN) is significantly more stable and can generate higher quality examples [16]. In SGAN, the discriminator $\mathcal{D}$ estimates the probability that the input data is real. However, the probability that real data is real should be also decreased simultaneously [16]. RaGAN achieves this property by making discriminator relativistic (i.e., having the output of $\mathcal{D}$ depends on both real and fake (restored) data). To make the relativistic discriminator act more globally, RaGAN further focuses on the average of the relativistic discriminator over random samples of data of the opposing type.

For a given natural example x and its adversarial example $\tilde{x}$ crafted by CAFA, the adversarial loss for $\mathcal{D}$ is defined as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{D}} = &-\log(\sigma(\mathcal{D}(x) - \tau(\mathcal{D}(\mathcal{C}(\tilde{x}))))) \\ &-\log(1 - \sigma(\mathcal{D}(\mathcal{C}(\tilde{x})) - \tau(\mathcal{D}(x)))),\end{aligned} \quad (8)$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\tau(\cdot)$ denotes the mean function. The adversarial loss for $\mathcal{C}$ is represented by,

$$\begin{aligned}\mathcal{L}_{adv} = &-\log(\sigma(\mathcal{D}(\mathcal{C}(\tilde{x})) - \tau(\mathcal{D}(x)))) \\ &-\log(1 - \sigma(\mathcal{D}(x) - \tau(\mathcal{D}(\mathcal{C}(\tilde{x}))))).\end{aligned} \quad (9)$$

Combining the above class activation feature loss and adversarial loss, the overall loss function for $\mathcal{C}$ is given as:

$$\mathcal{L}_{\mathcal{C}} = \lambda_1 \mathcal{L}_{caf} + \lambda_2 \mathcal{L}_{adv}, \quad (10)$$

where λ_1 and λ_2 are positive parameters to trade off each component. The overall procedure is summarized in Algorithm 2. Given training data X, we first simple natural example x from X and craft its corresponding adversarial example $\tilde{x}$ via CAFA. Then, we forward-pass $\tilde{x}$ through the denoiser $\mathcal{C}$ and calculate $\mathcal{L}_{caf}$. Next, we forward-pass $\mathcal{C}(\tilde{x})$ through $\mathcal{D}$ and then calculate $\mathcal{L}_D$ and $\mathcal{L}_{adv}$. Finally, we take a gradient step to update $\mathcal{C}$ and $\mathcal{D}$ to minimize $\mathcal{L}_C$ and $\mathcal{L}_D$. The above steps are repeated until $\mathcal{C}$ and $\mathcal{D}$ converge.

4. Experiments

In this section, we first introduce the datasets, network architectures and training details used in this paper (Section 4.1). Then, we present and analyze the experimental results of defending against unseen types of attacks and adaptive attacks (Section 4.2). Finally, we conduct an ablation

Table 1. Classification error rates (percentage) of the VGG-19 target model against adversarial examples (*lower is better*). CAFD and CAFD$'$ are our defense models corresponding to Eq. 7 and Eq. 6 respectively. The subscript N indicates that the corresponding attack is a non-targeted attack and the subscript T indicates that the corresponding attack is a targeted attack. The compression quality of JPEG is 75 and the weight of TVM is 0.003. The DOA method in this paper uses 7×7 adversarial patches crafted by exhaustive searching to retrain the target model. For each attack, we show the most successful defense with **bold** and the second one with underline.

	Defenses	**Attacks**								
		None	DDN$_N$	TI-DIM$_N$	PGD$_N$	PGD$_T$	AA$_N$	STA$_N$	STA$_T$	FWA$_N$
SVHN	DOA	**6.37**	**7.48**	**37.84**	38.96	32.90	39.09	**17.03**	19.10	65.89
	AT	7.16	10.49	40.10	31.27	25.78	32.56	20.90	21.10	**45.26**
	JPEG	9.78	11.25	92.04	95.67	86.60	97.56	37.81	33.65	93.07
	TVM	10.01	21.37	88.94	94.53	75.17	95.84	27.76	23.66	96.61
	APE-G	10.40	11.92	89.45	93.20	83.40	81.92	43.75	42.37	86.57
	HGD	10.12	10.99	90.83	57.35	45.00	62.25	40.43	36.53	67.44
	CAFD	7.65	14.90	58.57	**14.64**	**10.63**	**13.57**	19.92	**18.48**	58.07
	CAFD$'$	9.79	16.34	61.49	15.91	13.46	14.87	22.19	22.06	60.82
CIFAR-10	DOA	7.82	10.42	28.53	47.81	32.37	47.43	**17.51**	18.13	50.65
	AT	10.34	18.07	31.78	31.77	30.17	30.61	21.16	20.69	39.70
	JPEG	13.32	16.10	38.68	51.41	50.34	58.84	47.76	47.35	85.27
	TVM	9.65	23.15	39.46	68.79	56.71	66.46	30.76	31.90	90.81
	APE-G	8.18	11.75	34.19	78.08	62.31	76.65	24.08	21.40	76.66
	HGD	**7.64**	**9.18**	35.50	46.87	31.18	45.73	19.96	18.27	46.87
	CAFD	8.90	9.24	**26.57**	**12.79**	**10.58**	**11.80**	18.19	**17.17**	**35.59**
	CAFD$'$	8.95	9.32	29.85	15.45	12.67	14.27	19.03	18.10	39.49

study and an evaluation of the robustness of our model to the perturbation budget, to further show the effectiveness of our defense method (Section 4.3). The code is available at `https://github.com/dwDavidxd/CAFD`.

4.1. Experiment setup

Datasets: We verify the effective of our defense method on two popular benchmark datasets, i.e., *SVHN* [30] and *CIFAR-10* [19]. *SVHN* and *CIFAR-10* both have 10 classes of images, but the former contains 73,257 training images and 26,032 test images, and the latter contains 60,000 training images and 10,000 test images. Images in the two datasets are all regarded as natural examples. Adversarial examples for evaluating defense models are crafted by applying state-of-the-art attacks. These attacks inlcude: (i) pixel-constrained attacks, i.e., PGD [28], CW [4], AA [6], DDN [31] and TI-DIM [9, 46]. (ii) spatially-constrained attacks, i.e., STA [45] and FWA [39]. Pixel-constrained attacks generally manipulate the pixel values directly by leveraging the L_p norm distance for penalizing adversarial noise, while spatially-constrained attacks focus on mimicking non-suspicious vandalism via spatial transformation and physical modifications [10, 41].

Network architectures: We use three network architectures to perform classification tasks on *SVHN* and *CIFAR-10*, i.e., a VGG-19 architecture [33], a ResNet-50 architecture [14] and a Wide-ResNet architecture [47]. The depth and widen factor in the Wide-ResNet architecture are set to 28 and 20 respectively. The architecture of our defense is a DUNET architecture [23]. It consists of multiple basic blocks and each block contains a 3×3 convolutional layer, a batch normalization layer and a rectified linear unit. Our image discriminator is a VGG style discriminator [36, 22], it consists of a fully connected layer and three convolutional blocks containing convolutional layers followed by a batch normlization layer and a leaky ReLU activation function.

Training details: For fair comparison, all experiments are conduced on four NVIDIA RTX 2080 GPUs, and all methods are implemented by PyTorch. We use the implementation codes of PGD, DDN, CW and STA methods in the *advertorch toolbox* [8] and the author's implementation codes of AA, TI-DIM and FWA methods. The default perturbation budget ϵ is set to $8/255$ for both *SVHN* and *CIFAR-10*. The VGG-19, ResNet-50 and Wide-ResNet networks are used as target models and the VGG-19 network is also utilized as the pretrained network $\mathcal{P}$. Learning rates for target models is 10^{-2} on *SVHN* and 10^{-1} on *CIFAR-10*. All these networks are pretrained and remain fixed. The denoiser $\mathcal{C}$ and the discriminator $\mathcal{D}$ are optimized using Adam [18]. Their learning rates are initially set to 10^{-3} and decay to 2.7×10^{-5} when the training loss converges. The positive parameters λ_1 and λ_2 are set to 10^2 and 5×10^{-3} on *SVHN* and 10^3 and 5×10^{-3} on *CIFAR-10*.

4.2. Defense Results

Defending against unseen types of attacks: We use adversarial examples crafted by non-targeted L_2 norm CW to train previous defense models, and select non-targeted L_∞ norm PGD, targeted L_∞ norm PGD, non-targeted L_2 norm DDN, non-targeted L_2 norm CW, non-targeted AA,

Table 2. Classification error rates (percentage) of distinct target models with CAFD (*lower is better*). We transfer the CAFD defense trained based on VGG-19 to ResNet-50 and Wide-ResNet.

Attack	Target Model				
	VGG-19	ResNet-50		Wide-ResNet	
	CAFD	None	CAFD	None	CAFD
SVHN					
PGD$_N$	14.64	100	11.01	97.71	21.33
PGD$_T$	10.63	100	10.62	93.85	14.14
DDN$_N$	14.90	99.98	15.53	100	16.60
AA$_N$	13.57	100	18.80	97.20	23.17
STA$_N$	19.92	99.87	22.34	96.79	23.51
STA$_T$	18.48	99.71	21.91	96.63	24.87
CIFAR-10					
PGD$_N$	12.79	100	18.86	100	20.18
PGD$_T$	10.58	100	13.19	99.91	12.45
DDN$_N$	9.24	99.99	9.25	100	7.84
AA$_N$	11.8	100	16.25	100	17.23
STA$_N$	18.19	100	17.51	99.99	18.16
STA$_T$	17.17	99.97	17.07	99.66	17.93

non-targeted TI-DIM, non-targeted STA, targeted STA and non-targeted FWA as unseen types of attacks to craft adversarial examples for evaluating defense models. The details of these attacks could be found in appendix A. Figure 4 shows that our method is effective to remove strong adversarial noise. Quantitative analysis in Table 1 demonstrates that our method achieves more robust performance, e.g., reducing the success rate of AA$_N$ from 30.61% to 11.80% compared to previous state-of-the-art. The adversarial examples and restored examples are shown in appendix B.

Cross-model defense results: In order to evaluate the cross-model defense capability of our method, we transfer our CAFD model to other classification models, i.e., ResNet-50 and Wide-ResNet. Results in Table 2 present that our method significantly removes adversarial noise crafted by various unseen types of attacks against ResNet-50 and Wide-ResNet. The classification error rates of the ResNet-50 and Wide-ResNet target models are relatively similar to those of the VGG-19 target model, which demonstrates that our method could provide effective cross-model protections. The adversarial examples and restored examples are shown in appendix B.

Defending against adaptive attacks: An adaptive attack can access the leaked defense. In this case, the attacker uses the knowledge of the defense and is only restricted by the threat model [1, 3]. We study the following three difficult scenarios: (i) The attacker knows the defense and uses BPDA [1] to bypass it. (ii) The attacker gains a copy of the defense and combine it with the original target model into a new target model. Then, the attacker perform a white-box attack on the new target model (iii) The attacker does not directly access the defense but train a similar local defense model to craft adversarial examples in a gray-box manner.

Table 3. Classification error rates (percentage) under scenarios where defense are leaked. (*lower is better*). Defense models APE-G$'$ and HGD$'$ are trained based on adversarial examples crafted by non-targeted PGD with iteration number 20. 'It-τ' means that the maximum number of attack iterations is controlled to be τ.

BPDA			
Target	Attack	Defense	Error rate
APE-G$'$+VGG-19	PGD$_N$ (It-10)	APE-G$'$	98.32
HGD$'$+VGG-19	PGD$_N$ (It-10)	HGD$'$	79.50
CAFD+VGG-19	PGD$_N$ (It-10)	CAFD	47.74
APE-G$'$+VGG-19	PGD$_N$ (It-20)	APE-G$'$	99.12
HGD$'$+VGG-19	PGD$_N$ (It-20)	HGD$'$	85.04
CAFD+VGG-19	PGD$_N$ (It-20)	CAFD	51.25
White-box adaptive attack			
Target	Attack	Defense	Error rate
APE-G+VGG-19	DDN$_N$	APE-G	97.85
HGD+VGG-19	DDN$_N$	HGD	97.10
CAFD+VGG-19	DDN$_N$	CAFD	93.23
APE-G$'$+VGG-19	PGD$_N$	APE-G$'$	98.86
HGD$'$+VGG-19	PGD$_N$	HGD$'$	98.13
CAFD+VGG-19	PGD$_N$	CAFD	95.18
Gray-box adaptive attack			
Target	Attack	Defense	Error rate
APE-G$'$+VGG-19	DDN$_N$	APE-G	11.07
HGD$'$+VGG-19	DDN$_N$	HGD	10.93
CAFD$'$+VGG-19	DDN$_N$	CAFD	10.90
APE-G$'$+VGG-19	PGD$_N$	APE-G	81.19
HGD$'$+VGG-19	PGD$_N$	HGD	65.03
CAFD$'$+VGG-19	PGD$_N$	CAFD	56.76

In the BPDA scenario, the defense models APE$'$and HGD$'$ are trained based on adversarial examples crafted by non-targeted PGD. We use non-targeted PGD with iteration numbers 10 and 20 to attack defense models in the BPDA manner. As shown in Table 3, our method shows significant gains, i.e., the classification error rates are reduced by 49.85% and 39.83% on average compared to APE and HGD. In the white-box adaptive attack scenario, our method presents a slight reduction in the error rates. Since the defense models are completely leaked to the attacker, defense models' protection capabilities are destroyed under this scenario, which prompts us to strengthen the defense against such attacks in the future. In the gray-box adaptive attack scenario, we use APE-G$'$, HGD$'$ and CAFD$'$ as the local defense models to craft adversarial examples. Our method shows competitive performance against DDN$_N$ and obtains better experimental results against PGD$_N$.

4.3. Further Evaluations

Ablation study: Figure 5 shows the ablation study on *CIFAR-10*. We respectively remove the adversarial loss $\mathcal{L}_{adv}$ and the class activation feature loss $\mathcal{L}_{caf}$ to investigate their impacts on our model. Removing $\mathcal{L}_{adv}$ slightly reduces the classification accuracy rates because some fine

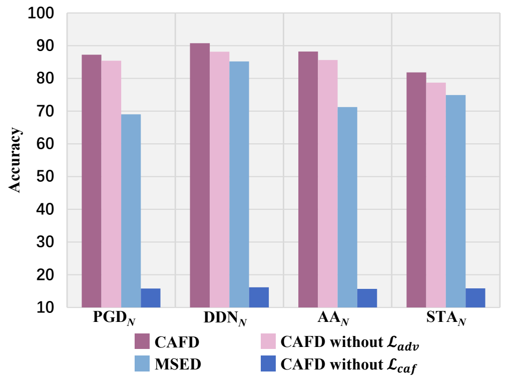

Figure 5. Ablation study on *CIFAR-10*. The figure shows the classification accuracy rates (percentage) of VGG-19 (*higher is better*). The performance of our method without $\mathcal{L}_{caf}$ is significantly affected, which indicates the importance of the class activation feature loss. MSED denotes the defense model which is trained by using a pixel-wise mean square error loss and the adversarial loss.

texture details would be lost. The performance of our method without $\mathcal{L}_{caf}$ is significantly affected, which indicated the importance of the class activation feature loss. We train a similar defense model named as MSED by using a pixel-wise mean square error loss and the adversarial loss instead of class activation feature loss. Compared to our defense model, the MSED does not provide sufficient protections against these attacks.

To further demonstrate the effectiveness of the proposed CAFA for improving the adversarial robustness, we show the results of the proposed models trained using adversarial examples crafted by PGD and CW in appendix C. In addition, we also present the results of previous defense methods when using adversarial examples crafted by PGD as adversarial training data in appendix C. The results show that using CAFA can achieve a great defense performance and improve the generalization of the defense against unseen types of attacks.

Robustness of our model to the perturbation budget: To explore the robustness of our defense model to the perturbation budget ϵ, we set the L_∞ norm perturbation budget ϵ within the range of $(6/255, 16/255)$ and craft adversarial examples via non-targeted PGD, targeted PGD and non-targeted AA. As shown in Figure 6, our defense model maintains a relatively high accuracy rate when the adversarial noise is constrained within $(6/255, 12/255)$. This indicates that our model is suggested to defense against attacks with ϵ less than $12/255$ when the perturbation budget of CAFA is $8/255$. When the perturbation budget continues to increase, the protection effect would reduce significantly.

5. Conclusion

In this paper, we aim to design a defense that could mitigate the error amplification effect, especially in the front of unseen types of attacks. Inspired by the observation of the discrepancies between the class activation maps of adversarial and natural examples, we propose a self-supervised adversarial training mechanism to remove adversarial noise in a class activation feature space. Specifically, we first use *class activation feature based attack* to craft adversarial examples. Then, we train a *class activation feature based denoiser* to minimize the distances between the adversarial and natural examples in the class activation feature space for removing adversarial noise. Experimental results demonstrate that our defense could provide protections against unseen types of attacks. In future, we can extend this work in the following aspects. First, we need to strengthen the defense against white-box adaptive attacks. Second, we can use the class weights in the internal layers via gradient-based methods, e.g., Grad-CAM [32] and Grad-CAM++ [5]. Third, we can use the strategies in the filed of label noise [24, 44, 43, 37, 42, 40] to improve the adversarial robustness of the target model against the adversarial noise.

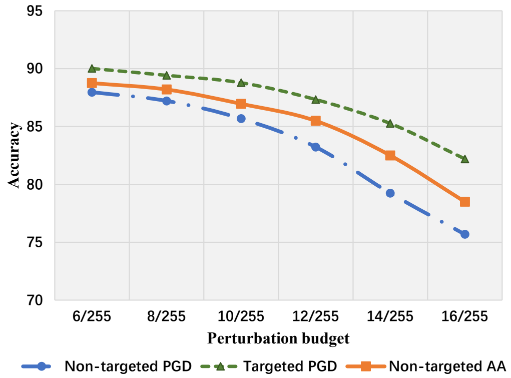

Figure 6. Classification accuracy rates (percentage) of our defense model against adversarial examples with distinct perturbation budget (*higher is better*). We select three strong attacks for this evaluation and set the L_∞ norm perturbation budget ϵ within the range of $(6/255, 16/255)$.

Acknowledgement: NNW and XBG were supported by the National Key Research and Development Program of China under Grant 2018AAA0103202 and the National Natural Science Foundation of China under Grants 61922066, 61876142, 62036007, 61772402, Grant 62050175. CLP was supported by Guangxi Natural Science Foundation Program under Grant 2021GXNSFDA075011. JY was supported by the Anhui Province Key Research and Development Program (202104a05020007) and USTC Research Funds of the Double First-Class Initiative (YD2350002001). TLL was supported by Australian Research Council Project DE-190101473. The authors thank the reviewers and the meta-reviewer for their helpful and constructive comments on this work.

References

[1] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[2] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing, 2021.

[3] Nicholas Carlini and David Wagner. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017.

[4] 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.

[5] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[7] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

[8] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

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

[10] Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

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

[12] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[13] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

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

[15] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. APE-GAN: adversarial perturbation elimination with GAN. In *International Conference on Acoustics, Speech and Signal Processing*, pages 3842–3846, 2019.

[16] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[17] He Kaiming, Gkioxari Georgia, Dollar Piotr, and Girshick Ross. Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP:1–1, 2017.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

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

[24] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[25] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[26] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.

[27] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, and Feng Lu. Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, 110:107332, 2021.

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations*, 2018.

[29] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[31] Jérôme Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019.

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2020.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations*, 2015.

[34] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*, pages 3104–3112, 2014.

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

[36] Xintao Wang, Ke Yu, Kelvin C.K. Chan, Chao Dong, and Chen Change Loy. Basicsr. `https://github.com/xinntao/BasicSR`, 2020.

[37] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

[38] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020.

[39] Kaiwen Wu, Allen Houze Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10377–10387, 2020.

[40] Songhua Wu, Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Nannan Wang, Haifeng Liu, and Gang Niu. Class2simi: A noise reduction perspective on learning with noisy labels. In *International Conference on Machine Learning*, pages 11285–11295. PMLR, 2021.

[41] Tong Wu, Liang Tong, and Yevgeniy Vorobeychik. Defending against physically realizable attacks on image classification. In *8th International Conference on Learning Representations*, 2020.

[42] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

[43] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

[44] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *arXiv preprint arXiv:1906.00189*, 2019.

[45] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *6th International Conference on Learning Representations*, 2018.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[47] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Richard C. Wilson, Edwin R. Hancock, and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference 2016*, 2016.

[48] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[49] Dawei Zhou, Tongliang Liu, Bo Han, Nannan Wang, Chunlei Peng, and Xinbo Gao. Towards defending against adversarial examples via attack-invariant features. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12835–12845, 2021.