Towards Defending against Adversarial Examples via Attack-Invariant Features

Dawei Zhou [1 2] **Tongliang Liu** [2] **Bo Han** [3] **Nannan Wang** [1 †] **Chunlei Peng** [4] **Xinbo Gao** [5]

Abstract

Deep neural networks (DNNs) are vulnerable to adversarial noise. Their adversarial robustness can be improved by exploiting adversarial examples. However, given the continuously evolving attacks, models trained on seen types of adversarial examples generally cannot generalize well to unseen types of adversarial examples. To solve this problem, in this paper, we propose to remove adversarial noise by learning generalizable invariant features across attacks which maintain semantic classification information. Specifically, we introduce an adversarial feature learning mechanism to disentangle invariant features from adversarial noise. A normalization term has been proposed in the encoded space of the attack-invariant features to address the bias issue between the seen and unseen types of attacks. Empirical evaluations demonstrate that our method could provide better protection in comparison to previous state-of-the-art approaches, especially against *unseen* types of attacks and *adaptive* attacks.

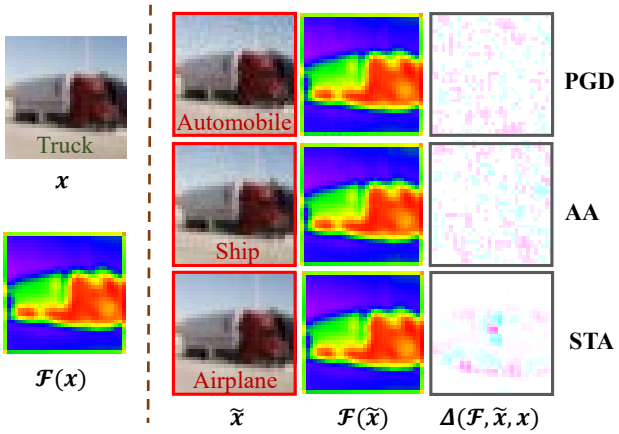

Figure 1. A visual illustration of the natural example (x), adversarial example ($\tilde{x}$), latent feature ($\mathcal{F}(\cdot)$) and attack-specific feature ($\Delta(\mathcal{F},\tilde{x},x) = [\mathcal{F}(\tilde{x}) - \mathcal{F}(x)] \times 10^4$). The latent feature is extracted from the first ReLu layer of the ResNet-110 model (He et al., 2016). Different types of attacks (i.e., PGD (Madry et al., 2018), AA (Croce & Hein, 2020) and STA (Xiao et al., 2018)) generally only modify tiny information and their adversarial examples sufficiently retain invariant features from the natural examples.

1. Introduction

Deep neural networks (DNNs) have been widely utilized in many fields, such as image processing (LeCun et al., 1998; He et al., 2016; Zagoruyko & Komodakis, 2016; Simonyan & Zisserman, 2015; Kaiming et al., 2017) and natural language processing (Sutskever et al., 2014). However, DNNs are found to be vulnerable to adversarial examples which are crafted by adding imperceptible but adversarial noise on natural examples (Goodfellow et al., 2015; Szegedy et al., 2014; Jin et al., 2019; Liao et al., 2018; Ma et al., 2018). The vulnerability of DNNs poses serious risks in many security-sensitive applications such as face recognition (Xu et al., 2020) and autonomous driving (Eykholt et al., 2018).

Existing methods show that the adversarial robustness of target models can be enhanced by exploiting adversarial examples, e.g., employing the adversarial examples as additional training data (Goodfellow et al., 2015; Tramèr et al., 2018; Wu et al., 2020b). However, focusing on the seen types of adversarial examples in the finite training data would cause the defense method to overfit the given types of adversarial noise and lack generalization or effectiveness against unseen types of attacks. Note that there are widespread or even unprecedented types of attacks in the real world. This motivates us to design a defense method that could handle different and unseen types of adversarial examples.

Cognitive science gives us an inspiration to solve this prob-

[1]State Key Laboratory of Integrated Services Networks, School of Telecommunications Engineering, Xidian University [2]Trustworthy Machine Learning Lab, School of Computer Science, The University of Sydney [3]Department of Computer Science, Hong Kong Baptist University [4]State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University [5]Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications. Correspondence to: Nannan Wang <nnwang@xidian.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

lem. Specifically, it shows that we are able to identify human faces even if the faces show different or even unseen expressions, because our brain is good at extracting invariant facial features (Mishkin & Ungerleider, 1982; Kanwisher et al., 1997). Similarly, we human cannot easily distinguish natural examples and adversarial examples, because we focus on the invariant features which represent semantic classification information and ignore the adversarial noise. Note that adversarial examples are designed to retain the invariant features so that we human could not identify the adversarial examples in advance, e.g., by constraining the adversarial noise to be small or non-suspicious (Goodfellow et al., 2015; Gilmer et al., 2018). We name such invariant features as *attack-invariant features* (AIF).

In this paper, we propose an *adversarial noise removing network* (ARN) to restore the natural data by exploiting AIF, which is able to defend against unseen types of adversarial examples. In a high level, we design an autoencoder-based framework, which divides the adversarial noise removing into learning AIF and restoring natural examples from AIF. Specifically, we introduce a pair of encoder and discriminator in an adversarial feature learning manner for disentangling AIF from adversarial noise. The discriminator is devoted to distinguish attack-specific information (e.g., attack type label) from the encoded AIF space, while the encoder aims to learn features which are indistinguishable for the discriminator. By iterative optimization, the attack-specific information will be removed and the invariant features across attacks will be retained.

Note that the adversarial examples used in the training procedure are often biased because the widespread types of attacks in the real world are very diverse. For example, as shown in Figure 1, the autoattack (AA) method based adversarial example (Croce & Hein, 2020) and the projected gradient descent (PGD) method based adversarial example (Madry et al., 2018) are similar; while they are quite different from the spatial transform attack (STA) based adversarial example (Xiao et al., 2018). If we do not handle the bias problem, the learned AIF may work well for the attacks or similar types of attacks used in the training procedure, but may have poor generalization ability for some unseen types of attacks whose perturbations are significantly diverse from those in seen types of attacks (Li et al., 2018; Makhzani et al., 2015). To address the bias issue, we impose a normalization term in the encoded space of AIF to match the feature distribution of each type of attack to a multivariate Gaussian prior distribution (Makhzani et al., 2015; Kingma & Welling, 2014). By this design, the learned AIF is expected to generalize well to widespread unseen types of attacks.

To restore the original natural examples from AIF, a decoder is trained by minimizing the gap between the synthesized examples and the natural examples in the pixel space. Achieved by jointly optimizing the encoder and decoder for learning AIF, our ARN could provide more superior protection against unseen types of attacks compared to previous methods. This will be empirically verified on pixel-constrained and spatially-constrained attacks in Section 4.2. Furthermore, additional evaluations on cross-model defenses and adversarial example detection in Section 4.3 further show the effectiveness of ARN. The main contributions in this paper are as follows:

- Adversarial examples typically have shared invariant features even if they are crafted by unseen types of attacks. We propose an *adversarial noise removing network* (ARN) to effectively remove adversarial noise by exploiting *attack-invariant features* (AIF).
- To handle the bias issue of the adversarial examples available in the training procedure, we design a normalization term in the encoded AIF space to enhance its generalization ability to unseen types of attacks.
- Empirical experiments show that our method presents superior effectiveness against both pixel-constrained and spatially-constrained attacks. Particularly, the success rates of unseen types of attacks and adaptive attacks are reduced 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 against both pixel-constrained and spatially-constrained attacks are provided in Section 4. Finally, we conclude this paper in Section 5.

2. Related work

Attacks: The seminal work of Szegedy *et al.* (2014) first proposed adversarial examples that can mislead DNNs. Adversarial examples can be crafted by adding adversarial noise following the direction of adversarial gradients. Attacks based on this strategy include fast gradient sign method (FGSM) (Goodfellow et al., 2015), the strongest first-order information based projected gradient descent (PGD) method (Madry et al., 2018), the Jacobian-based saliency map attack (JSMA) method (Papernot et al., 2016). The autoattack (AA) method (Croce & Hein, 2020) forms a parameter-free, computationally affordable and user-independent ensemble of attacks. The adversarial noise crafted by these attacks is typically bounded by a small norm-ball $\|\cdot\|_p \leq \epsilon$, so that their adversarial examples can be perceptually similar to natural examples. In addition, optimization-based attacks, such as Carlini and Wagner (CW) method (Carlini & Wagner, 2017b) and decou-

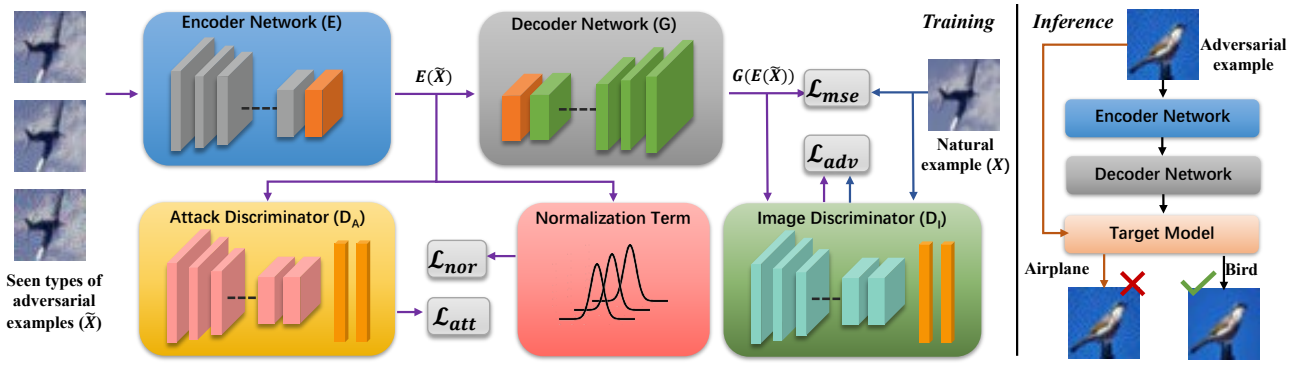

Figure 2. A visual illustration of our *adversarial noise removing network* (ARN). Our main idea is to restore natural examples by exploiting invariant features. ARN is composed of an encoder network and a decoder network. The encoder network learns attack-invariant features (AIF) via an adversarial feature learning mechanism and a normalization term. The decoder is trained to restore natural examples from AIF via a pixel similarity metric and an image discriminator.

pling direction and norm (DDN) method (Rony et al., 2019), minimize the adversarial noise as part of the optimization objectives. The above attacks directly modify the pixel values on the whole sample without considering semantics of objectives, e.g., shape and posture. They are named as *pixel-constrained attacks*. In addition, there are also *spatially-constrained attacks* which focus on mimicking non-suspicious vandalism via geometry and spatial transformation or physical modifications. These attacks include faster wasserstein attack (FWA) (Wu et al., 2020a), spatial transform attack (STA) (Xiao et al., 2018) and robust physical perturbations (RP2) (Eykholt et al., 2018).

Defenses: Adversarial training (AT) is a widely used strategy for defending against adversarial noise by augmenting the training data with adversarial examples, such as PGD based adversarial training method (AT_{PGD}) (Madry et al., 2018) and defending against occlusion attacks (DOA) (Wu et al., 2020b) method. In addition, input processing based methods have also been proposed to defend against attacks. They aim to pre-process input data for mitigating the aggressiveness of adversarial noise. For example, Jin *et al.* (2019) proposed APE-G to back adversarial examples close to natural examples via a generative adversarial network. Liao *et al.* (2018) utilized a high-level representation guided denoiser (HGD) as a pre-processing step to remove adversarial noise. HGD used the class labels predicted by a target model to supervise the training of an end-to-end denoiser. Compared with the above defenses, we design an input processing based model that remove adversarial noise by learning attack-invariant features, instead of directly relying on learning a function which maps seen types of adversarial examples to the perceptual space of natural examples. In addition, the method in (Xu et al., 2017) shows that reducing the color bit depth of an adversarial example could reduce its attack success rate, but the method may make processed examples lose some useful natural features. Our method brings adversarial examples close to the natural examples without causing human-observable loss. The defense in (Xie et al., 2019) focuses on denoising the perturbations in the feature maps on internal layers of the target model by modifying the target models' architectures. Differently, our method aims to disentangle natural features from adversarial noise, and use the natural features to generate clean examples. Our defense is a pre-processing based defense, which dose not require the knowledge of the target models and could provide cross-model protection.

3. Adversarial noise removing network

3.1. Preliminaries

In this paper, we aim to design a defense which could provide robust protection against widespread unseen types of attacks. The basic intuition behind our defense is to effectively exploit the invariant features. To this end, we propose the *adversarial noise removing network* (ARN) which eliminates adversarial noise by learning *attack-invariant features* (AIF). As shown in Figure 2, our ARN divides the remove of adversarial noise into two steps. The first step is to learn AIF from input examples via an encoder network E. The second one is to restore natural examples from AIF via a decoder network G. The encoder network E is trained by exploiting an attack discriminator D_A and a normalization term, while the decoder network G is trained to minimize the gap between synthetic examples and natural examples via utilizing a pixel similarity metric and an image discriminator D_I.

Our defense model can be expressed as $G(E(\tilde{X})$, where $E(\cdot)$ represents the process of learning AIF from input adversarial examples $\tilde{X}$. We use $\tilde{X}_k = [\tilde{x}_{k_1}, \tilde{x}_{k_2}, \ldots, \tilde{x}_{k_N}]^\top$

to denote adversarial data for the k-th type of attack, where k_N is the number of adversarial examples crafted by the k-th type of attack. The natural data are denoted by $X = [x_1, x_2, \ldots, x_N]^\top$, where N is the number of the natural examples.

3.2. Learning Attack-invariant Features

We propose a hybrid objective function to train our ARN to learn AIF. The objective function consists of two terms that we explain below:

Adversarial feature learning: To remove attack-specific information by disentangling invariant features from adversarial noise, we distinguish the different types of adversarial noise in the resulting encoded feature space. More precisely, we introduce an attack discriminator D_A to form an adversarial feature learning mechanism with the encoder network E. Given a set of K seen types of adversarial examples $\tilde{X} = \{\tilde{X}_1, \tilde{X}_2, \cdots, \tilde{X}_K\}$, D_A takes the encoded features $E(\tilde{X})$ as inputs and predicts attack types. The attack type of adversarial examples crafted by the k-th attack is embodied as the attack-specific label $Y_k^p = \left[y_{k_1}^p, y_{k_2}^p, \ldots, y_{k_N}^p\right]^\top$, where $y_{k_n}^p = [\xi_1, \xi_2, \ldots, \xi_K]^\top$ is a one-hot vector and ξ_i equals one when $i = k$ and zero otherwise. Based on the predictions about attack-specific labels, D_A could reflect whether the encoded features contain attack-specific information. The objective function of D_A is derived as follows:

$$\mathcal{L}_{D_A} = -\frac{1}{K}\sum_{k=1}^{K} Y_k^p \cdot \log(\sigma(D_A(E(\tilde{X}_k)))), \quad (1)$$

where σ denotes the softmax layer.

In contrast, E aims to remove the attack-specific information and make the learned encoded features indistinguishable for D_A. That is to say, as the adversary of D_A, E is devoted to confuse D_A from correctly predicting the attack-specific label and pushes its prediction close to the attack-confused label $Y_\zeta^p = \left[y_{\zeta_1}^p, y_{\zeta_2}^p, \ldots, y_{\zeta_N}^p\right]^\top$, where $y_{\zeta_n}^p = [1/K, 1/K, \ldots, 1/K]^\top$ is a K-dimensional constant vector. As a result, the objective of E is as follows:

$$\mathcal{L}_{att} = -\frac{1}{K}\sum_{k=1}^{K} Y_\zeta^p \cdot \log(\sigma(D_A(E(\tilde{X}_k)))). \quad (2)$$

Normalization term: Since the widespread types of attacks in the real world are very diverse, the adversarial examples used in the training procedure are often biased. Although the above adversarial feature learning mechanism could effectively defend against an unseen type of attack that is similar to seen types of attacks, this bias issue may lead a risk that the learned AIF has poor generalization ability for some unseen types of attacks whose perturbations are significantly different from those in seen types of attacks. For example, as shown in Figure 1, adversarial noise crafted by pixel-constrained PGD and AA looks similar, while adversarial noise crafted by spatially-constrained STA presents significant difference from them.

To address the bias issue, inspired by previous studies (Makhzani et al., 2015; Larsen et al., 2016; Kingma & Welling, 2014), we introduce a normalization term in the encoded space of AIF to decrease the undesirable risk. Specifically, the feature distribution of each type of attack $P_k(E(\tilde{X}_k))$ is matched to a multivariate Gaussian prior distribution $\mathcal{N}(0, I)$ through utilizing the *Jensen-Shannon Divergence* (JSD). The normalization could make the encoded features of different types of adversarial examples have similar distributions, which is beneficial for robustly restoring natural examples in Section 3.3. The JSD measure is the average of *Kullback-Leibler divergences* between each distribution and the average distribution $\bar{P}$, which is formulated as $JSD\left(P_1, \cdots, P_K\right) = \frac{1}{K}\sum_{k=1}^{K} KL\left(P_k \| \bar{P}\right)$. In our method, the distribution of each encoded feature is expected to be similar to the uniform prior distribution. We replace $\bar{P}$ by $\mathcal{N}(0, I)$. The objective function of this normalization term is derived as:

$$\mathcal{L}_{nor} = JSD\left(P_1, \cdots, P_K\right) = \frac{1}{K}\sum_{k=1}^{K} KL\left(P_k \| \mathcal{N}\right). \quad (3)$$

3.3. Restoring Natural Examples

A hybrid objective function is also used to restore natural examples from AIF. The object consists of following two terms:

Pixel similarity metric: Adversarial noise could be viewed as the delicately crafted special noise. The widely used metric for image denoising or reconstruction would be able to achieve satisfactory results for generating examples close to natural examples (Jin et al., 2019). Therefore, we apply the *mean square error* (MSE) metric in the pixel space:

$$\mathcal{L}_{mse} = \sum_{k=1}^{K} \|G(E(\tilde{X}_k)) - X\|_2^2, \quad (4)$$

where $\|\cdot\|_2$ is the L_2 norm.

Adversarial learning in pixel space: As noted in (Zhao et al., 2017), the decoder network based on MSE tends to synthesize blurry textures, which would lead to incorrect classification in the target model. To overcome the limitation, we introduce an image discriminator D_I to form an adversarial mechanism in the pixel space with the decoder network G. D_I is trained to identify natural examples X as true data and identify synthesized examples as false data.

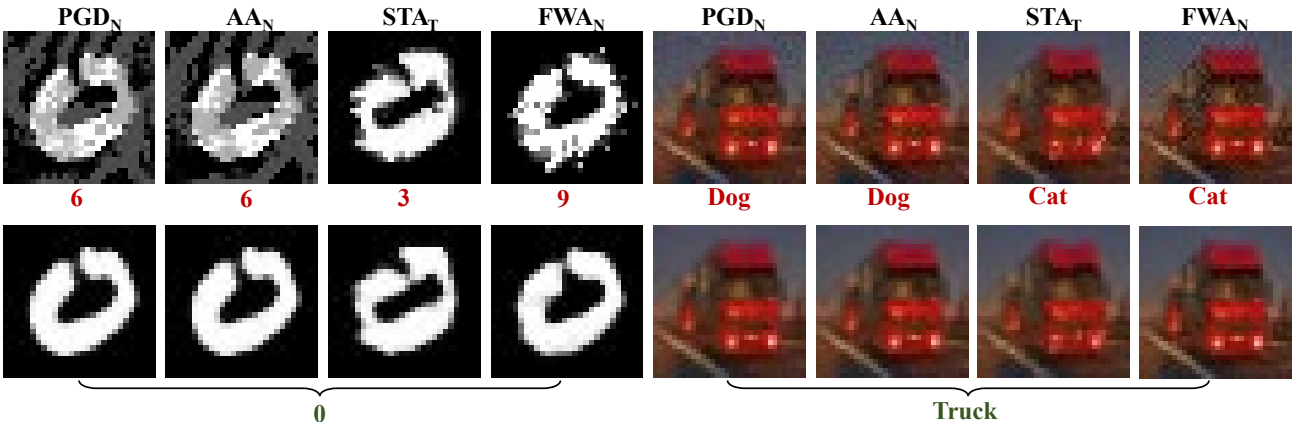

Figure 3. A visual illustration of the performance of our model against various attacks. (*top:* adversarial examples; *bottom:* restored examples). Subscripts "N" and "T" respectively indicate that the corresponding attacks are non-target and target attacks. PGD_N is the seen type of attack while other attacks are regarded as unseen types of attacks.

We define the objective function of D_I as:

$$\mathcal{L}_{D_I} = \sum_{k=1}^{K}[\log(D_I(G(E(\tilde{X}_k)))) + \log(1 - D_I(X))]. \quad (5)$$

The adversarial objective function of G is calculated as:

$$\mathcal{L}_{adv} = -\sum_{k=1}^{K}\log(D_I(G(E(\tilde{X}_k)))). \quad (6)$$

3.4. Implementation

In order to make the encoded features invariant to different types of attacks and retain sufficient semantic classification information, we learn AIF by jointly optimizing the encoder network E and the decoder network G. The overall objective function for E is the combination of attack-invariant loss, normalization term loss and MSE loss:

$$\mathcal{L}_E = \mathcal{L}_{mse} + \lambda_1\mathcal{L}_{att} + \lambda_2\mathcal{L}_{nor}, \quad (7)$$

where λ_1 and λ_2 are positive parameters to trade off each component. The overall objective function for G is given as:

$$\mathcal{L}_G = \mathcal{L}_{mse} + \theta\mathcal{L}_{adv}, \quad (8)$$

where θ is a trade-off parameter. Details of λ_1, λ_2, and θ are given in Section 4.1.

The overall procedure is summarized in Algorithm 1. Given natural examples X and adversarial examples $\tilde{X}$, we first sample a mini-batch X_d and $\tilde{X}_d$ from X and $\tilde{X}$ respectively. Then, we forward-pass $\tilde{X}$ through E to obtain encoded features $E(\tilde{X})$ and calculate $\mathcal{L}_{D_A}$ (Eq. 1), $\mathcal{L}_{att}$ (Eq. 2) and $\mathcal{L}_{nor}$ (Eq. 3). Next, we forward-pass $E(\tilde{X})$ through G and calculate $\mathcal{L}_{mse}$ (Eq. 4), $\mathcal{L}_{adv}$ (Eq. 5) and $\mathcal{L}_{D_I}$ (Eq. 5). Finally, we take a gradient step to update E, G, D_A and D_I to minimize $\mathcal{L}_E$ (Eq. 7), $\mathcal{L}_G$ (Eq. 8), $\mathcal{L}_{D_A}$ (Eq. 1) and $\mathcal{L}_{D_I}$ (Eq. 5). The above operations are repeated until E and G converge.

Algorithm 1 ARN: Adversarial Noise Removing Network

Input: Natural examples X and adversarial examples $\tilde{X}$.
repeat
1: Sample a mini-batch X_d and $\tilde{X}_d$ from X and $\tilde{X}$ respectively.
2: Forward-pass $\tilde{X}_d$ through E to obtain encoded features $E(\tilde{X}_d)$ and calculate $\mathcal{L}_{D_A}$ (Eq. 1), $\mathcal{L}_{att}$ (Eq. 2) and $\mathcal{L}_{nor}$ (Eq. 3).
3: Forward-pass $E(\tilde{X}_d)$ through G to restore natural examples and calculate $\mathcal{L}_{mse}$ (Eq. 4), $\mathcal{L}_{adv}$ (Eq. 6) and $\mathcal{L}_{D_I}$ (Eq. 5).
4: Back-pass and update E, G to minimize $\mathcal{L}_E$ (Eq. 7) and $\mathcal{L}_G$ (Eq. 8).
5: Update D_A and D_I to minimize $\mathcal{L}_{D_A}$ (Eq. 1) and $\mathcal{L}_{D_I}$ (Eq. 5).
until E and G converge.

4. Experiments

In this section, we first introduce the datasets used in this paper (Section 4.1). We next show and analyze the experimental results of defending against pixel-constrained and spatially-constrained attacks on visual classification tasks, especially against adaptive attacks (Section 4.2). Finally, we conduct additional evaluations on the cross-model defense, ablation study and adversarial detection to further show the effectiveness of our ARN (Section 4.3). The code is avail-

Table 1. Classification error rates (percentage) against adversarial examples crafted by pixel-constrained attacks on *MNIST* and *CIFAR-10* (*lower is better*). 'ϵ'' means the raised *perturbation budget* ϵ of corresponding attack, it is set to 0.4 for *MNIST* and 0.05 for *CIFAR-10*. '7×7' denotes the size of sticker used by DOA. For each attack we show the most successful defense with **bold** and the second result with underline.

	DEFENSE	ATTACKS								
		NONE	PGD_N	PGD_T	CW_N	DDN_N	AA_N	JSMA_T	$\text{PGD}_{N\epsilon'}$	$\text{AA}_{N\epsilon'}$
LENET	NONE	**0.64**	100	100	100	100	100	100	100	100
	AT_{PGD}	1.19	9.63	8.38	6.42	5.91	12.60	28.59	54.34	60.06
	$\text{DOA}_{7\times7}$	6.27	65.23	38.84	11.48	10.53	68.49	19.81	86.76	92.51
	APE-G_{PP}	1.57	8.76	3.20	2.34	2.15	12.40	36.49	34.86	46.72
	APE-G_{DP}	1.73	10.39	5.81	2.93	1.91	15.26	38.04	37.33	49.38
	HGD_{PP}	1.36	1.89	1.30	1.67	1.54	2.43	50.62	75.79	90.34
	HGD_{DP}	1.18	2.56	1.91	1.79	1.23	3.30	53.73	78.95	93.76
	ARN_{PP}	1.16	**1.85**	**1.29**	**1.45**	1.28	**2.38**	**16.75**	**15.27**	**26.84**
	ARN_{DP}	1.11	1.91	1.80	1.53	**1.22**	2.97	17.81	17.63	29.74
RESNET	NONE	**7.67**	100	100	100	99.99	100	100	100	100
	AT_{PGD}	12.86	51.02	49.68	50.17	49.19	53.66	44.59	59.09	61.65
	$\text{DOA}_{7\times7}$	9.82	89.03	73.96	24.11	49.29	97.52	**23.26**	96.83	97.75
	APE-G_{PP}	23.08	44.38	39.09	23.18	32.39	60.09	39.10	79.34	87.16
	APE-G_{DP}	24.23	45.96	41.50	27.43	24.73	64.82	41.67	83.19	89.92
	HGD_{PP}	10.41	39.44	23.03	13.26	16.02	42.34	38.65	57.97	58.41
	HGD_{DP}	9.42	41.62	25.30	12.46	**10.04**	43.45	43.63	58.63	59.86
	ARN_{PP}	8.21	**38.66**	**20.43**	**11.47**	14.64	**38.94**	35.49	**49.45**	**52.64**
	ARN_{DP}	8.18	40.28	22.87	12.24	10.17	41.27	36.23	52.87	55.91

able at https://github.com/dwDavidxd/ARN.

4.1. Experiment setup

Datasets: We verify the effective of our method on three popular benchmark datasets, i.e., *MNIST* (LeCun et al., 1998), *CIFAR-10* (Krizhevsky et al., 2009), and *LISA* (Jensen et al., 2016). *MNIST* and *CIFAR-10* both have 10 classes of images, but the former contains 60,000 training images and 10,000 test images, and the latter contains 50,000 training images and 10,000 test images. To alleviate the problem of imbalance and extremely blurry data in *LISA*, we picked 16 best quality signs with 3,509 training images and 1,148 test images from a subset which contains 47 different U.S. traffic signs (Eykholt et al., 2018; Wu et al., 2020b). Adversarial examples are crafted by applying state-of-the-art attacks. These attacks can be divided into two categories: (i) *Pixel-constrained attacks*, i.e., non-target L_∞ norm PGD (PGD_N), target L_∞ norm PGD (PGD_T), non-target DDN (DDN_N), non-target L_2 norm CW (CW_N), non-target AA (AA_N) and target JSMA (JSMA_T). (ii) *Spatially-constrained attacks*, i.e., non-target STA (STA_N), target STA (STA_T), non-target FWA (FWA_N) and non-target RP_2 (RP_N).

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, JSMA and STA in the *advertorch toolbox* (Ding et al., 2019) and the implementation codes of RP2, FWA and AA provided by their authors. We set default perturbation budget $\epsilon = 0.3$ and $\epsilon = 8/255$ for *MNIST* and *CIFAR-10* respectively. More details about the attack approaches can be found in appendix A. Learning rates for the encoder network, the decoder network, the attack discriminator and the image discriminator are all set to 10^{-4}, with the value of $\lambda_1 = 10^{-1}$, $\lambda_2 = 10^{-2}$, $\theta = 10^{-1}$ for *MNIST*, the value of $\lambda_1 = 10^2$, $\lambda_2 = 10^1$, $\theta = 10^2$ for *CIFAR-10* and *LISA*. In addition, we consider the following deep neural network architectures as the target models:

- **MNIST:** The LetNet-5 architecture (LeNet) (LeCun et al., 1998) embedded in the advertorch toolbox is used for the *MNIST* digits recognition task.
- **CIFAR-10:** The ResNet-110 (ResNet) architecture (He et al., 2016), the Wide-ResNet (WRN) architecture (Zagoruyko & Komodakis, 2016) and the VGG-19 (VGG) architecture (Simonyan & Zisserman, 2015) are utilized for the classification task on *CIFAR-10*. The depth and widen factors in WRN are set to 28×20.
- **LISA:** We use the LISA-CNN architecture defined in (Eykholt et al., 2018) for the traffic sign recognition task according to the previous work (Wu et al., 2020b). The convolutional neural network contains three convolutional layers and one fully connected layer.

4.2. Defense Results

Defending against pixel-constrained attacks: We select two attacks as seen types of attacks to craft adversarial examples, and use them together with natural examples as training data to train defense models. The other attacks are regarded

Table 2. Classification error rates (percentage) against adversarial examples crafted by spatially-constrained attacks (*lower is better*). 'ϵ'' is set to 0.4 for *MNIST* and 0.05 for *CIFAR-10*. AP_i denotes different adversarial patches crafted by RP_2 (Eykholt et al., 2018). We show the most successful defense with **bold** and the second result with underline.

	Defense	Attacks				
		None	STA_N	STA_T	FWA_N	$\text{FWA}_{N\epsilon'}$
LENET	None	**0.64**	100	100	98.56	99.91
	AT_{PGD}	1.19	21.55	31.49	42.41	61.55
	$\text{DOA}_{7\times7}$	6.27	13.54	19.24	**23.29**	**40.16**
	APE-G_{PP}	1.57	16.57	21.40	33.95	51.81
	APE-G_{DP}	1.73	20.18	25.63	37.50	58.16
	HGD_{PP}	1.36	21.32	36.41	50.43	71.12
	HGD_{DP}	1.18	25.01	38.47	52.84	75.35
	ARN_{PP}	1.16	**9.08**	**13.73**	25.79	43.76
	ARN_{DP}	1.11	10.14	14.51	28.50	47.63
RESNET	None	**7.67**	100	100	99.83	99.98
	AT_{PGD}	12.86	44.86	44.60	40.32	49.37
	$\text{DOA}_{7\times7}$	9.82	38.33	28.02	49.69	62.00
	APE-G_{PP}	23.08	47.19	36.46	42.79	50.53
	APE-G_{DP}	24.23	49.93	37.51	45.26	57.61
	HGD_{PP}	10.41	42.89	31.97	37.67	43.41
	HGD_{DP}	9.42	49.52	36.06	35.95	42.87
	ARN_{PP}	8.21	**36.81**	**23.62**	**24.17**	**31.89**
	ARN_{DP}	8.18	37.74	26.90	27.10	33.06
		None	AP_1	AP_2	AP_3	AP_4
CNN	None	**0.86**	55.46	62.07	61.21	56.03
	AT_{PGD}	3.16	50.29	43.68	56.03	33.62
	APE-G	3.43	8.33	5.43	21.56	24.71
	$\text{DOA}_{9\times5}$	2.59	18.39	6.90	25.86	8.91
	$\text{DOA}_{7\times7}$	5.17	16.95	11.49	19.83	10.06
	ARN	2.31	**5.46**	**3.74**	**6.90**	**6.03**

as unseen types of attacks to evaluate the generalization ability of defense models. Considering that the L_2 norm distance between adversarial examples and natural examples varies greatly across different attacks, which may influence the performances of models, we construct two different combinations of seen types of attacks: (i) the target PGD and the non-target PGD ("defense_{PP}"). (ii) the non-target DDN and the non-target PGD ("defense_{DP}"). Figure 3 demonstrates that our ARN is effective to remove strong adversarial noise. Quantitative analysis in Table 1 represents that our ARN achieves better robust performance, especially reducing the success rate of JSMA_T from 50.62% to 16.75% compared to previous state-of-the-art. Moreover, our ARN shows a significant improvement in defending against attacks with greater perturbation budgets (i.e., $\epsilon = 0.4$ for *MNIST* and $\epsilon = 0.05$ for *CIFAR-10*).

Defending against spatially-constrained attacks: In addition to pixel-constrained attacks, some attacks focus on mimicking non-suspicious vandalism via spatial transformation and physical modifications (Gilmer et al., 2018; Wu et al., 2020b). We evaluate the robustness of above defense models against STA_T, STA_N and FWA_N on *MNIST* and *CIFAR-10*. As shown in Table 2, our method achieves more effective defense and has better robustness. In particularly, our ARN significantly reduces the success rate of STA_T from 36.41% to 13.73% in comparison to previous state-of-the-art, which has outstanding performance against pixel-constrained attacks. In order to further remove the spatially-constrained adversarial noise, we train our ARN by using adversarial examples crafted by PGD_N and STA_N. The fooling rates of STA_N, STA_T and FWA_N on *MNIST* are decreased from 9.08%, 13.73% and 25.79% to 6.52%, 7.94% and 16.32%, while the fooling rates of PGD_N and PGD_T are remained at 3.66% and 2.51% respectively. In addition, for protecting the target model on *LISA*, defense models are trained based on two seen types of *adversarial patches (AP)* crafted by RP_2, i.e., AP_1 and AP_2. Our ARN also achieves better performance on defending against unseen types of adversarial patches. The restored images are shown in appendix B.

Leaked defenses: We study the following three different scenarios where defenses are leaked:

(i) An attacker knows the per-processing defense model and directly uses white-box adaptive attacks (Carlini & Wagner, 2017a) to break it. In this scenario, the attacker gains a copy of the trained defense model. The architecture and model parameters of the pre-processing model are both leaked to the attacker.

(ii) An attacker trains a similar pre-processing defense model and then take the combination of the known pre-processing model and the original target model as a new target model to craft adversarial examples via gray-box adaptive attacks. We use APE-G_{PP} and HGD_{PP} as the known pre-processing models to craft adversarial examples via different types of attacks.

(iii) An attacker can utilize BPDA (Athalye et al., 2018) strategy to bypass the pre-processing defense. Specifically, BPDA is different from the attack strategy which directly computes the gradient of the defense model $g(\cdot)$ and the target model $f(\cdot)$. If the knowledge of $g(\cdot)$ is inaccessible or if $g(\cdot)$ is neither smooth nor differentiable, $g(\cdot)$ cannot be backpropagated through to generate adversarial examples with a white-box attack that requires gradient signal. BPDA can approximate $\nabla_x f(g(x))$ by evaluating $\nabla_x f(x)$ at the point $g(x)$. This allows an attacker to compute gradients and therefore mount a white-box attack. BPDA is widely used to bypass pre-processing defenses. It can be used to explore whether an adversary can precisely approximate the gradient of the defense model for implementing white-box attacks. We combine BPDA with PGD_N to evaluate our defense model.

As shown in Table 3 and Table 4, experimental results

Table 3. Classification error rates (percentage) against white-box and gray-box adaptive attacks (*lower is better*) on *MNIST*. "TAR" denotes the target attack and "NON-TAR" denotes the non-target attack. "ITE-τ" means that the maximum number of iterations is controlled to be τ and "ϵ-τ" means that the perturbation budget is set to τ. "L" denotes the original target model.

TARGET LENET (L)	ATTACK	DEFENSE	ITE-40	
			ϵ-0.3	ϵ-0.5
APE-G+L	PGD_T	APE-G	99.84	100
HGD+L	PGD_T	HGD	62.50	100
ARN+L	PGD_T	ARN	58.52	99.95
APE-G+L	PGD_T	ARN	1.49	3.80
HGD+L	PGD_T	ARN	2.56	10.65
TARGET LENET (L)	ATTACK	DEFENSE	TAR	NON-TAR
APE-G+L	CW	APE	99.90	98.07
APE-G+L	CW	ARN	1.39	1.28
HGD+L	DDN	HGD	100	100
HGD+L	DDN	ARN	1.29	1.42

Table 4. Classification error rates (percentage) against BPDA (*lower is better*). "P+B" denotes the hybrid attack of PGD_N and BPDA (Athalye et al., 2018). "L" and "R" denote the original target models on *MNIST* and *CIFAR-10* respectively.

MNIST: LENET (L)				
TARGET	ATTACK	DEFENSE	ITE-40	ITE-100
APE-G+L	P+B	APE-G	72.01	72.75
ARN+L	P+B	ARN	24.65	24.70
CIFAR-10: RESNET-110 (R)				
TARGET	ATTACK	DEFENSE	ITE-40	ITE-100
APE-G+R	P+B	APE-G	89.06	89.51
ARN+R	P+B	ARN	60.47	60.75

present that our defense model achieves positive gains in these challenging settings compared to other pre-processing defenses. For example, the classification error rates against BPDA and white-box PGD_T are decreased by 66% and 24% on average respectively. This may be due to the attack-invariant features being more robust against adversarial noise under the constraints of small perturbation budgets. The adversarial examples crafted by adaptive attacks and their restored examples are shown in appendix C.

4.3. Further Evaluations

Cross-model defense results: In order to evaluate the cross-model defense capability of our ARN, we transfer the ARN_{PP} model used for ResNet to other classification models, i.e., WRN and VGG. Results in Table 5 present that our ARN effectively removes adversarial noise crafted by various unseen types of attacks against WRN and VGG, which demonstrates that our ARN could provide generalizable cross-model protection.

Table 5. Classification error rates (percentage) of different target models with ARN (*lower is better*) on *CIFAR-10*. ARN is trained by using adversarial examples crafted against ResNet, and then is applied to WRN and VGG.

ATTACK	TARGET MODEL				
	RESNET	WRN		VGG	
	ARN	NONE	ARN	NONE	ARN
PGD_N	38.66	100	33.38	100	36.15
PGD_T	20.43	99.91	23.73	99.66	24.20
CW_N	11.47	100	9.92	100	10.39
DDN_N	14.64	100	9.20	100	9.57
AA_N	38.94	100	36.94	100	37.59
STA_N	36.81	100	29.46	100	30.47
STA_T	23.62	99.95	22.28	99.96	21.46
FWA_N	24.17	94.37	24.23	95.21	23.31

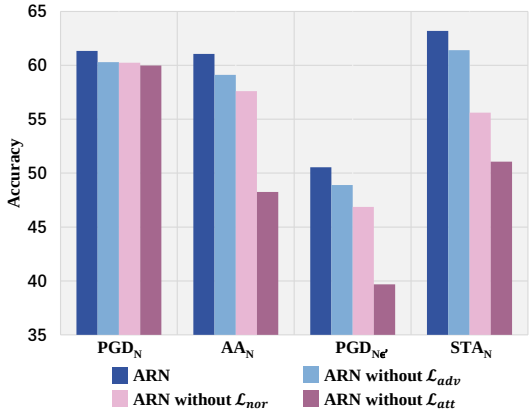

Figure 4. Ablation study. The figure shows the classification accuracy rates (percentage) of ResNet against different attacks (*higher is better*) on *CIFAR-10*. The performance of ARN against unseen STA_N is significantly affected when $\mathcal{L}_{nor}$ is dropped. ARN trained without $\mathcal{L}_{att}$ has poor robust against unseen types of attacks i.e., AA_N, $PGD_{N\epsilon'}$ and STA_N.

Ablation: Figure 4 shows the ablation study on *CIFAR-10*. We respectively remove the pixel adversarial loss $\mathcal{L}_{adv}$, the normalization term loss $\mathcal{L}_{nor}$ and the attack-invariant loss $\mathcal{L}_{att}$ to investigate their impacts on our ARN. We use PGD_N and PGD_T as seen types of attacks to train ARN. Removing $\mathcal{L}_{adv}$ slightly reduces the classification accuracy rates. The performance of ARN against STA_N is significantly affected when $\mathcal{L}_{nor}$ is dropped. ARN trained without $\mathcal{L}_{att}$ no longer learns AIF and hence loses its superior generalizable ability to unseen types of attacks.

Adversarial examples detection: Local intrinsic dimensionality (Lid) method (Ma et al., 2018) could distinguish between adversarial examples and natural examples by revealing the essential difference between them. In this way, we can evaluate our ARN from the perspective of detecting adversarial examples. A binary classifier is first trained to

distinguish between positive examples (adversarial examples) and negative examples (natural examples). Then, we take the clean examples restored by our ARN as positive examples and input them to the classifier. The classifier presents low recall rates, i.e., 1.26% for *MNIST* and 8.48% for *CIFAR-10*, which reflects that restored examples are almost indistinguishable from natural examples. This demonstrates that our ARN could effectively remove adversarial noise.

Discussion on the number of seen types of attacks: In the above experiments, we choose two attacks as seen types of attacks to train the pre-processing model. Of course, the ideal number of seen types of attacks is not fixed. We think that the ideal number of seen types of attacks is related to the diversity of unseen types of attacks that may appear in a practical scenario. Specifically, If the number of unseen types of attacks is small or all unseen types of attacks are similar (e.g., CW_N and CW_T), using one strong seen types of attacks (e.g., PGD_N) may be ideal. If the unseen types of attacks are quite different (i.e., CW_N, AA_N and FWA_N), we can use more seen types of attacks (e.g., PGD_N and STA_N) to train our defense model for providing robust protection. The selected seen types of attacks are expected to approximately cover the unseen types of attacks. In the real world, attacks are continuously evolving. The new attacks may have obvious discrepancies with previous attacks, and thus pose potential threats to the defense model. We could update the seen types of attacks and retrain the defense model to enhance the model' s adversarial robustness.

5. Conclusion

In this paper, we focus on designing a pre-processing model for adversarial defense against different unseen types of attacks. Inspired by cognitive science researches on the human brain, we propose an *adversarial noise removing network* to restore natural examples by exploiting *attack-invariant features*. Specifically, we introduce an adversarial feature learning mechanism to disentangle invariant features from adversarial noise. A normalization term is proposed in the encoded space of the invariant features to address the bias issue between the seen and unseen types of attacks. By minimizing the gap between the synthesized examples and natural examples, our method could restore natural examples from attack-invariant features. Experimental results demonstrate that our proposed model presents superior effectiveness against both pixel-constrained and spatially-constrained attacks, especially for unseen types of attacks and adaptive attacks. In future, we can extend the work in the following aspects. First, we can try to leak our defense model to attacks during the training process for improving the defense effective against adaptive attacks. Second, we can use the feedback of a target model (e.g. predictions of class labels) to train our defense model for further improving classification accuracy rates. Third, we can combine the pre-training model with recently proposed robust target model (Liu & Tao, 2015; Xia et al., 2019; 2020; Wang et al., 2019; Xia et al., 2021) to explore the robustness of the combined model against noisy data.

Acknowledgements

DWZ was supported by the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University. TLL was supported by Australian Research Council Project DE-190101473. NNW, CLP 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 62036007, 61922066, 61876142, 61772402, 61806152. BH was supported by the RGC Early Career Scheme No. 22200720, NSFC Young Scientists Fund No. 62006202 and HKBU CSD Departmental Incentive Grant. The authors thank the reviewers and the meta-reviewer for their helpful and constructive comments on this work.

References

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

Carlini, N. and Wagner, D. Magnet and" efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017a.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 Ieee Symposium on Security and Privacy (sp)*, pp. 39–57. IEEE, 2017b.

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

Ding, G. W., Wang, L., and Jin, X. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Gilmer, J., Adams, R. P., Goodfellow, I., Andersen, D., and Dahl, G. E. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Jensen, M. B., Philipsen, M. P., Møgelmose, A., Moeslund, T. B., and Trivedi, M. M. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17 (7):1800–1815, 2016.

Jin, G., Shen, S., Zhang, D., Dai, F., and Zhang, Y. APE-GAN: adversarial perturbation elimination with GAN. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 3842–3846, 2019.

Kaiming, H., Georgia, G., Piotr, D., and Ross, G. Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP:1–1, 2017.

Kanwisher, N., McDermott, J., and Chun, M. M. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of neuroscience*, 17(11):4302–4311, 1997.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *International Conference on Learning Representations*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48 of *Workshop and Conference Proceedings*, pp. 1558–1566, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In *Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.

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

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S. N. R., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations*, 2018.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Mishkin, M. and Ungerleider, L. G. Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys. *Behavioural brain research*, 6(1):57–77, 1982.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations*, 2015.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Neural Information Processing Systems*, pp. 3104–3112, 2014.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I. J., Boneh, D., and McDaniel, P. D. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Wu, K., Wang, A. H., and Yu, Y. Stronger and faster wasserstein adversarial attacks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 10377–10387, 2020a.

Wu, T., Tong, L., and Vorobeychik, Y. Defending against physically realizable attacks on image classification. In *8th International Conference on Learning Representations*, 2020b.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? *arXiv preprint arXiv:1906.00189*, 2019.

Xia, X., Liu, T., Han, B., Wang, N., Gong, M., Liu, H., Niu, G., Tao, D., and Sugiyama, M. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

Xiao, C., Zhu, J., Li, B., He, W., Liu, M., and Song, D. Spatially transformed adversarial examples. In *6th International Conference on Learning Representations*, 2018.

Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P.-Y., Wang, Y., and Lin, X. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pp. 665–681. Springer, 2020.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Wilson, R. C., Hancock, E. R., and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference 2016*, 2016.

Zhao, S., Song, J., and Ermon, S. Towards deeper understanding of variational autoencoding models. 2017.