Adversarial Training for Improving the Robustness of Deep Neural Networks

by

Pengyue Hou

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent System

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Since 2013, Deep Neural Networks (DNNs) have caught up to a human-level performance at various benchmarks. Meanwhile, it is essential to ensure its safety and reliability. Recently an avenue of study questions the robustness of deep learning models and shows that adversarial samples with human-imperceptible noise can easily fool DNNs. Since then, many strategies have been proposed to improve the robustness of DNNs against such adversarial perturbations. Among many defense strategies, adversarial training (AT) is one of the most recognized methods and constantly yields state-of-the-art performance. It treats adversarial samples as augmented data and uses them in model optimization.

Despite its promising results, AT has two problems to be improved: (1) poor generalizability on adversarial data (e.g. large robustness performance gap between training and testing data), and (2) a big drop in model's standard performance. This thesis tackles the above-mentioned drawbacks in AT and introduces two AT strategies.

To improve the generalizability of AT-trained models, the first part of the thesis introduces a representation similarity-based AT strategy, namely self-paced adversarial training (SPAT). We investigate the imbalanced semantic similarity among different categories in natural images and discover that DNN models are easily fooled by adversarial samples from their hard-class pairs. With this insight, we propose SPAT to re-weight training samples adaptively during model optimization, enforcing AT to focus on those data from their hard class pairs.

To address the second problem in AT, a big performance drop on clean data, the second part of this thesis attempts to answer the question: to what extent the robust-

ness of the model can be improved without sacrificing standard performance? Toward this goal, we propose a simple yet effective transfer learning-based adversarial training strategy that disentangles the negative effects of adversarial samples on model's standard performance. In addition, we introduce a training-friendly adversarial attack algorithm, which boosts adversarial robustness without introducing significant training complexity. Compared to prior arts, extensive experiments demonstrate that the training strategy leads to a more robust model while preserving the model's standard accuracy on clean data.

Preface

This thesis is conducted under the supervision of Professor J. Han and Professor X. Li. Some of the work was done in collaboration with M. Zhou and Professor P. Musilek. Chapter 3 of the thesis is the original work of myself and has been submitted to *Association for the Advancement of Artificial Intelligence (AAAI) 2023*, as P. Hou, J. Han, and X. Li, "Self-Paced Adversarial Training", and the article is currently under review. Chapter 4 of the thesis has been published as P. Hou, M. Zhou, P. Musilek, J. Han, and X. Li, "Adversarial Fine-tune with Dynamically Regulated Adversary", IJCNN, IEEE WCCI 2022. M. Zhou and Professor P. Musilek contributed to data collection and manuscripts edits.

Acknowledgements

I would like to thank Professor J. Han and Professor X. Li for their supports. This thesis would not have been possible without their advice and assistance. Their enthusiasm, professionalism and research attitude have greatly inspired me and kept me passionate about my research. I am also grateful to all the professors whom I have had the pleasure to attend their lectures. Their informative lectures offered me solid knowledge to pursue my academic goals.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Deep learning (DL) enables machines to learn from large scale of data and perform human-like tasks. With the recent success of DL in computer vision (CV) and natural language processing (NLP) areas, DL is beginning to provide more convenience and flexibility in our daily lives. Many industrial products have managed to incorporate DL techniques to improve their products, and solve problems that were impossible to address with traditional software engineering. While lots of research has been conducted to further advance the performance of DL algorithms, little attention has been paid on the safety and security aspects of DL until recent years.

Szegedy *et al.* [1] first discovered an intriguing property of neural networks in 2014. Consider a state-of-the-art (SOTA) neural network that generalizes well on the training dataset. They show that by adding some non-random, imperceptible perturbation to a test image can arbitrarily change the output of the network. Such perturbed examples are termed as "adversarial examples". Adversarial examples can be used to perform attacks on machine learning systems and pose security concerns to DL algorithms. As communities increasingly rely on DL techniques in safety-critical applications, it is important to ensure their security and robustness against malicious attacks.

Although AT has shown promising results in improving the adversarial robustness

of DNNs, there are still many unresolved challenges exist in the AT paradigm. (1) Poor generalizability on adversarial samples. Recent work observes AT often suffers from the over-fitting problem, where there is a growing disparity in training and validation robustness performances [2]. (2) Although ATs can improve the robustness of DNNs, they are often accompanied by a decrease in standard accuracy. (3) AT usually uses a specific type of attack to generate adversarial examples during training, this sometimes leads to poor generalizability to some unseen adversarial attacks. (4) AT is an extremely time-consuming training paradigm. This is because AT requires adversarial examples that are generated on-the-fly, and finding representative adversarial examples is hard and often requires multiple iterations of complete back-propagation. This thesis mainly focuses on the first two challenges described above.

1.2 Thesis Scope

The objective of this thesis is to design and train more reliable and robust DNN based image classification models. More specifically, we aim to train DNNs that can correctly classify or detect adversarial examples. Concretely, there are two main contributions described in this thesis:

1. The first main contribution is that we propose a self-paced AT strategy (SPAT) that can significantly boost the robustness of DNNs compared with other SOTA defenses. Our work is inspired by the imbalanced semantic similarity among different classes. More specifically, we notice the fact that DNNs are more easily to be fooled among similar classes, such as cats and dogs. Therefore, we introduced a re-weighting strategy to re-scale the loss of each class based on the difficulty of how easily it can be fooled.

2. The second contribution is to maintain the standard accuracy performance while improving the adversarial robustness of adversarially trained models. To address the trade-off problem between standard accuracy and adversarial robustness,

we propose a training-friendly adversarial attack method to reduce the training complexity of AT. Then we adopt a transfer learning based AT method to allow adversarial robustness smoothly transfer to clean-data pre-trained models without interfering with their standard accuracy performance.

1.3 Thesis Outline

The thesis is outlined as follows. Chapter 2 describes the general background information of adversarial machine learning. Specifically, we introduce why adversarial examples exist, how to craft them, and some existing defense strategies. Chapter 3 presents details of my submitted paper, "Self-Paced Adversarial Training". Chapter 4 presents my published paper: "Adversarial Fine-tune with Dynamically Regulated Adversary". Finally, conclusion of the work and possible directions of future work are presented in Chapter 5.

Chapter 2

Background

2.1 Blind Spots in Neural Networks

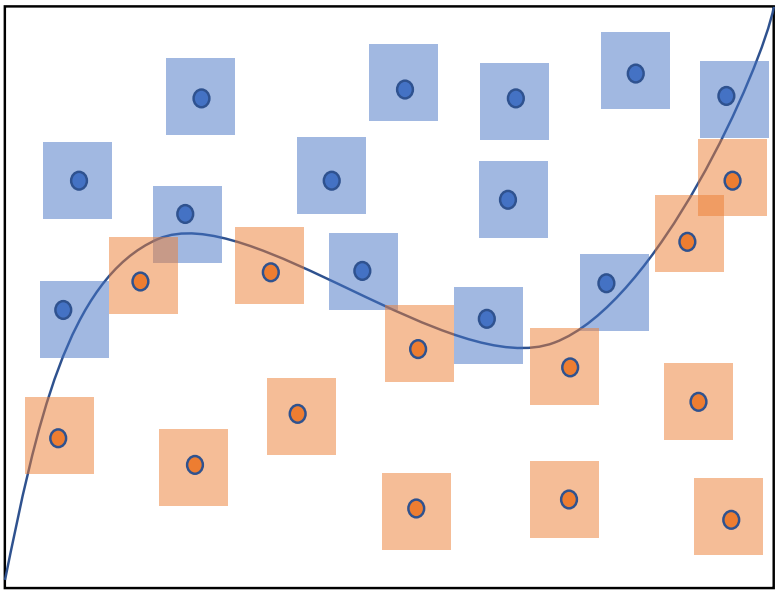


Figure 2.1: A conceptual demonstration of why adversarial examples exist. Although a well-trained classifier can correctly classify data points from different classes, there are still regions close to the data points that will be misclassified.

Neural networks or multi-layer perceptrons (MLPs) are the foundation of deep learning models. The goal of neural networks is to approximate some function f that is highly non-linear and hard to be formulated with traditional mathematics expressions. Neural networks that are used to categorize inputs are called classifiers:

$$y = f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{2.1}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{\theta}$ is the learned parameters of f, and y is the predicted label of $\boldsymbol{x}$.

Neural networks usually consist of an input layer, multiple hidden layers, and an output layer. The process of classifiers making predictions is called **feedfoward**. This is because information from the input layer flows sequentially through hidden layers, and finally to the output layer y. Based on the architecture of feedfoward neural networks, there are lots of successful commercial applications have been developed. For example, Convolution Neural Networks (CNNs) for object detection and recognition in Computer Vision tasks; and recurrent networks, which provide foundations for many natural language processing applications.

Intuitively, adding small perturbations cannot change the object category of an image. However, Szegedy *et al.* show that by adding some carefully calculated noises can cause a well-trained neural network to respond very differently. Previous work has tried to explain the vulnerability of DNNs to adversarial attacks from different perspectives, ranging from the optimization paradigm of supervised learning [1, 3] to discontinuity of neural networks in high dimensional manifold [4–6]. Up until now, there is still no unified theory that can fully explain and capture the behavior of adversarial examples. Therefore, we start by relaxing the problem to a simple linear classification case. Consider a shallow linear neural network with weight vector $\boldsymbol{w}$, the output of the input $\boldsymbol{x}$ is calculated as $y = \boldsymbol{w}^\top \boldsymbol{x}$. And for adversarial input $\boldsymbol{x'} = \boldsymbol{x'} + \boldsymbol{\eta}$, the output is $y' = \boldsymbol{w}^\top \boldsymbol{x'}$. Here $\boldsymbol{\eta}$ is the adversarial perturbation and $\boldsymbol{x'}$ is the adversarial example. We can derive the output interference from adversarial perturbation as:

$$y' - y = \boldsymbol{w}^\top \boldsymbol{\eta} \tag{2.2}$$

Intuitively, we expect the classifier to have little interference as long as $\|\boldsymbol{\eta}\|_\infty < \epsilon$, where ϵ is a regulation term to ensure the noises are small enough to be observed by human eyes. To maximize the interference, we can project $\boldsymbol{\eta}$ to its l_∞ norm so

that $\boldsymbol{\eta} = sign(\boldsymbol{w})$. Assume $\boldsymbol{w}$ is an n-dimensional vector and the average absolute value of $\boldsymbol{w}$ is m. Formally, we can formulate the maximum distortion of adversarial perturbation for a linear classifier as:

$$max(y' - y) = \epsilon mn \tag{2.3}$$

From above analysis, we can conclude that adversarial interference grows linearly with n. The explanation suggests that adversarial examples exist for DNNs with large enough dimensionality.

2.2 How to Craft Adversarial Examples

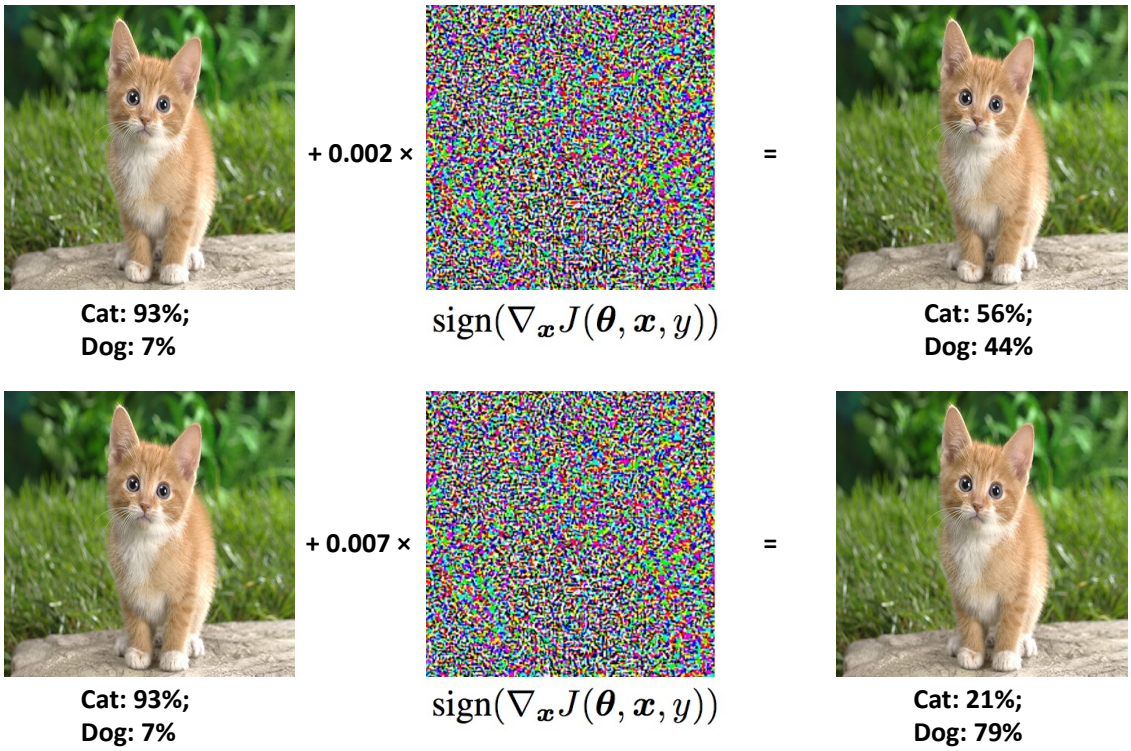


Figure 2.2: An illustration of how FGSM perturbation can interfere with the DNNs. Here we pre-traine a binary classifier to distinguish cats from dogs using a subset of ImageNet. Initially the classifier correctly classy the cat image with 93% confidence. On the top row, we disturb the image with very small noises generated by FGSM. The classifier still can make a correct prediction but with much lower confidence. On the second row, we apply a larger, yet still human-imperceptible perturbation to the image, the prediction of the classifier changes to dog with 79% confidence.

In Section 2.1, we explain why adversarial examples exist and demonstrate a simple method to generate adversarial perturbation for a simple linear classifier. In this section, we formally introduce adversarial attack methods for DNNs with non-linear activation functions. There are two general criteria to characterize adversarial attacks: (1) **White and black box attacks**. Depending on the transparency of the defense model, white box attacks have direct access to the architectures and parameters of the defense model, but black box attacks do not have access to the model information. (2) **Targeted and untargeted attacks**. Both targeted and untargeted attacks aim to perturb inputs so that well-trained models would make incorrect predictions. However, targeted attacks have a more restricted constraint, which is the perturbed inputs need to be misclassified to a specific target class. In the following sections, we give a more detailed explanation of how to generate different types of adversarial attacks and introduce some popular attack strategies in the literature.

2.2.1 White Box Attack

White box attacks allow attackers to access model information directly. Most of the strong adversarial attacks are white box attacks, thus they are commonly used to benchmark adversarial robustness. In this section, we show some SOTA white box attacks that can consistently cause a variety of models to make mistakes.

Fast gradient sign method (FGSM) [4] is the first formally proposed adversarial attack method that directly exploits gradient information of the model.

$$\boldsymbol{x}_{fgsm} = \boldsymbol{x} + \epsilon sign(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \tag{2.4}$$

Here $\boldsymbol{\theta}$ is the parameters of neural network and $\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)$ is the cost to train the classifier. The idea of FGSM is similar to what we explained in Section 2.1, the only difference is that we calculate the sign of back-propagated gradients as the perturbation instead of directly use the sign of weights. FGSM only requires a single-step attack and can be calculated efficiently during back-propagation.

Projected gradient decent (PGD) [7] is another gradient based adversarial attack proposed by Madry *et al.* PGD can be viewed as an iterative variant of FGSM and formulated as:

$$\boldsymbol{x}_{\boldsymbol{pgd}}^{\boldsymbol{t+1}} = \boldsymbol{x}_{\boldsymbol{pgd}}^{\boldsymbol{t}} + \epsilon sign(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)) \tag{2.5}$$

where x_{pgd}^0 is randomly perturbed input:

$$\boldsymbol{x}_{\boldsymbol{pgd}}^{\boldsymbol{0}} = \boldsymbol{x} + \boldsymbol{\eta}_{\boldsymbol{random}}, \|\boldsymbol{\eta}_{\boldsymbol{random}}\|_\infty < \epsilon \tag{2.6}$$

PGD uses more number of iterations (t) to find stronger adversarial examples, however, it also requires more computation time compared to FGSM.

C&W attack [8] was proposed by Carlini *et al.* in 2017. C&W attack is an optimization based attack and the overall objective function can be formulated as follow:

$$\text{minimize } (D(x, x')) \text{ such that } argmax f(x') = t \tag{2.7}$$

Here D is the distance metric, $f(x')$ is the output of classifier f, and t represents an arbitrary false class. The exact form of C&W attack varies with different choices of distance metrics and optimization function. The authors showed that with large enough optimization steps, C&W attack can achieve almost 100 percent attack success rate on any undefended model.

Deep Fool [6] is an attack method aiming to find the minimum adversarial perturbation needed to fool a classifier. Specifically, the attack tries to calculate the distance from a normal input x to its nearest decision boundary. Therefore, the adversarial perturbation of Deep Fool is calculated from geometric perspective:

$$\boldsymbol{x}_{\boldsymbol{deepfool}}^{\boldsymbol{t+1}} = \boldsymbol{x}_{\boldsymbol{deepfool}}^{\boldsymbol{t}} - f(\boldsymbol{x}) \frac{\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)}{\|\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y)\|_2} \tag{2.8}$$

where $x_{deepfool}^0$ corresponding to original input x. The number of iterations t is subject to the condition that $f(x_{deepfool}^t) \neq f(x)$.

2.2.2 Black Box Attack

In contrast to white box attacks, black box attacks have limited knowledge to the model, therefore, black box attackers are less effective than the white box counterpart. While the capability of black box attacks is limited, they are the most concerned attacks in real-world scenarios because the model information is usually protected. In this section, we introduce some popular black box strategies.

Transferable attack is a strategy that uses a substitute model to craft adversarial examples. Szegedy *et al.* observed that the same adversarial perturbation can cause a different neural network trained on different subset of the dataset to make mistakes. This observation show that adversarial examples are transferable to unseen neural networks. Inspired by this, some work pre-train a substitute neural network on datasets of the same domain to perform attacks [9–11].

Gradient estimation attack is proposed for tasks that hard to obtain substitute classifiers. Since the attackers only have access to the output results, they choose to use a query feedback mechanism to estimate gradient direction [12–14]. Specifically, the attackers continuously craft the perturbation based on the output results while querying on the model. The attackers usually start with random input and add noises to the input until the output is distorted to an acceptable level.

Gradient free attack [15] is based on the concept of greedy local-search technique. Unlike previous attack methods, gradient free attack does not require gradient information to craft adversarial examples. The proposed attack observes the changes in output by randomly perturbing a local area of the input and determines the importance of pixels based on the change of classification accuracy. Although gradient free attack offers more flexibility to generate adversarial examples, the greedy search method is very time-consuming as it must be conducted pixel by pixel.

2.2.3 Targeted and Untargeted Adversarial Attack

There are two common objectives of adversarial attacks: targeted and untargeted. Untargeted attack aims to push the original input x out of its ground truth class. Targeted attack on the other hand, aims to move x to a specific target class. Therefore, the objective function for **untargeted attack** is:

$$\text{maximize } J(\boldsymbol{\theta}, \boldsymbol{x}, y_i) \text{ subject to } f(x) \neq y_i \tag{2.9}$$

where y_i is the true class for x. For **targeted attack**, the objective function is given by:

$$\text{minimize } J(\boldsymbol{\theta}, \boldsymbol{x}, y_t) \text{ subject to } f(x) = y_t \tag{2.10}$$

where y_t is the target class that we want to move the input to. Targeted attack can be viewed as a more constrained type of untargeted attack as the target space is reduced from n-1 classes to 1 target class. Due to this reason, targeted adversarial examples are harder, and more expensive to generate.

2.3 Adversarial Defenses

The potential threat from adversarial vulnerabilities of DNNs has raised concerns, some defense strategies against adversarial attacks have been proposed from the research communities. However, design adversarial robust DNNs has shown to be a very difficult task, since the research community has not yet arrived at a unified theoretical solution to explain the optimization problem.

To date, most adversarial defenses strategies are developed along three directions: gradient masking ,robust regularization, and adversary detection. Most of the adversarial attack methods in the literature require direct gradient information, or use estimated gradient information to construct adversarial examples. Therefore, a new avenue of research aims to create a defense mechanism by obfuscating gradient information to the attacker, namely gradient masking. While gradient masking can refuse

the attackers to access gradient information of the model, it does not directly improve the robustness of the model itself. Robust regularization on the other hand, aims to fundamentally enhance the robustness of the model without relying on external tools. Finally, adversary detection aims to detect and reject adversarial examples to protect the neural network.

2.3.1 Gradient Masking

Athalye *et al.* [16] characterize gradient masking into three sub-categories. In this section, we introduce these three types of gradient masking strategies by explaining why they are able to help improve the robustness of DNNs and their limitations.

Shattered Gradient refers to defense strategies that are non-differentiable or cause the gradient to be non-exist or incorrect. The intention of this defense is to break local linearity of neural networks so that attackers cannot continuously search for optimal adversarial examples along the gradient direction. The most commonly used and easily deployed non-differentiable defense is input transformation. Guo *et al.* [17] proposed 5 different non-differentiable input transformation methods to counter adversarial examples: image cropping and rescaling, bit-depth reduction, and JPEG compression. They also explore the effectiveness of different combinations of input transformations. Their experimental results indicate that input transformations can effectively protect undefended models without introducing much extra computation loads.

Stochastic Gradient is a strategy that leverages randomization to defend against adversarial attacks. Previous work show that randomization techniques such as drop-out, can be used as effective tools to prevent neural networks from over-fitting. Recently, Dhillon *et al.* [18] show that apply drop-out to each layer can help improve adversarial robustness while only sacrifice little standard classification accuracy. Xie *et al.* [19] propose to use a randomization layer to defend against adversaries. More specifically, the randomization layer first randomly re-scale the input then apply

padding to reconstruct the image to its original size. Such defense strategy is cost-efficient and can largely reduce the success rate of existing adversarial attack methods.

Vanishing & Exploding Gradients method usually deploys generative models to reconstruct the input. PixelCNN [20] uses an auto-encoder-like structure to purify the adversarial perturbations. The authors argue that adversarial examples mainly lie in the low-probability region between the decision boundaries. Thus, PixelCNN aims to project the adversarial examples back to their high-confidence latent regions. Samangouei *et al.* adopt similar idea and apply Generative Adversarial Network as the image reconstructor. Both of them can be viewed as image pre-processing defenses using a sub-neural network. Even this type of defense is differentiable, the generative neural networks are usually deep enough so that the gradient information of the actual classifier is vanished/exploded when perform adversarial attacks.

Although all three type of gradient masking strategies can achieve SOTA performance on specific attacks, they can be easily broken with more general adversarial attacks or black box attacks as shown in [16]. In other word, since gradient masking specifically targets on gradient based attacks, when the attackers do not heavily rely on the information of the gradient, they can easily bypass the defense mechanisms.

2.3.2 Adversarial Training: Robust Regularization

The idea of robust regularization is to build intrinsic robustness into neural networks themselves. Goodfellow *et al.*[4] first propose to use an extra adversarial regulation term combine with the regular cross entropy loss to improve robustness, that is:

$$\mathcal{L}(\theta) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1-\alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon sign(\nabla_x J(\boldsymbol{\theta}, \boldsymbol{x}, y), y)) \tag{2.11}$$

Here α is a hyper-parameter to control the relative loss from adversarial data, and J is cross entropy loss. Such robust regularization method also refer to as "**adversarial training**" and has proven to help model defend against FGSM attack. However, Kurakin *et al.* observe that using FGSM as the adversarial examples during training

can cause over-fitting and label leaking. This is due to the fact that FGSM examples are generated with the information of ground truth label of the inputs. To address this issue, they propose to use iterative FGSM (iFGSM) to hide the information of label. Such training algorithm can also improve the robustness of models against stronger adversarial attacks but is more time-consuming.

Madry *et al.* [7] redefine adversarial training as a saddle point (min-max) optimization problem. They show that even the constituent part of adversarial training is non-convex and non-concave, the underlying optimization problem is overall still tractable. Specifically, they abandon the standard cross entropy loss, and only use PGD adversarial examples during training. Therefore, the min-max game is formulated as:

$$\mathcal{L}(\theta) = \min \mathbb{E}_{(x,y)\sim D} \max_{\delta \in S} [J(\boldsymbol{x_{pgd}}, y; \boldsymbol{\theta})] \tag{2.12}$$

The authors also find that adversarial training favors large capacity models as the decision boundaries of robustness neural networks are far more complicated than their undefended counterparts.

Previous adversarial methods aim to correctly classify adversarial examples, Kannan *et al.* find that a more robust classifier can be found by constraining the logits of neural networks. In other word, as the adversarial perturbations are usually small, it is intuitive to consider the differences in output logits are small too. The authors propose Adversarial Logit Pairing (ALP) to enforce the output logits between clean data and adversarial data to be similar. The overall objective of ALP is:

$$\mathcal{L}(\theta) = J(\boldsymbol{\theta}, \boldsymbol{x}, y) + \lambda L(\boldsymbol{f(x)}, \boldsymbol{f(x')}) \tag{2.13}$$

where L can be different choices of loss to measure the similarity of $f(x)$ and $f(x')$ such as L^2, L^1 or Huber loss. Zhang *et al.* propose Trades [21] which adopt almost identical objective function as ALP. The main difference is that they use KL divergence to measure the similarity between corresponding logits.

Inspired by logit pairing algorithms, a group of research community observe that label information is not required for robustness training. Formally, ALP and Trades enhance model robustness by pushing the output logits to be similar without using the true label to compute losses as shown in Eq. 2.13. Therefore, some work managed to use unlabeled data to further enhance adversarial robustness [22, 23].

2.3.3 Adversarial Example Detection

While previous research has shown that directly imposing adversarial robustness to DNNs is hard, some work suggests deploying detection mechanisms to protect DNNs from adversarial attacks. Grosse *et al.* [24] propose to use an N+1 class training algorithm, where the extra class is used for detecting adversarial examples in arbitrary classes. Hendrycks & Gimpel propose to use PCA to detect adversarial examples from natural data, as they find adversaries place a higher weight on the larger principal components than natural images. Feinman *et al.* observe that the latent distribution of adversarial examples is slightly different from clean data distribution. Therefore, the authors propose a defense called kernel *density estimation*. They use Gaussian Mixture Model to study the distribution of output from the penultimate layer and argue that adversarial examples belong to a different distribution than natural data.

2.3.4 Adversarial Robustness Benchmarks

In the literature of adversarial machine learning, there are several commonly used datasets for benchmarking adversarial robustness. MNIST [25] featuring its simplicity often used to validate the feasibility of the proposed AT strategy. MNIST consists of 10 classes, each representing a number digit from 0 to 9. The whole dataset contains 60,000 training images and 10,000 testing images of size 28×28 pixels. CIFAR-10 and CIFAR-100 are two more complicated classification tasks that are frequently used for benchmarking adversarial robustness. CIFAR datasets consist of real-world image categories such as planes, ships, cats, etc. Compare with MNIST, CIFAR datasets

are more recognized as it has been shown that some AT defenses that perform well on MNIST datasets do not scale on CIFAR datasets [8]. Other than MNIST and CIFAR datasets, TinyImageNet [26] and ImageNet [27] datasets are also used to evaluate adversarial robustness. However, due to the fact AT is a time-consuming training paradigm, the occurrence of ImageNet and other large-scale datasets is not as frequent as CIFAR and MNIST datasets in the literature of AT.

2.4 Discussion & Conclusions

While much effort has been made to discover the best solutions for adversarial robustness, there is still a significant gap to a satisfactory level of security safety. Furthermore, Carlini *et al.* [28] recently show that most existing adversarial example detection method can be easily broken with a more dedicated attack generated by C&W. To date, adversarial training is the most recognized defend method which proven to help model develop robustness against a variety of attacks, but there are still several limitations that are hard to address. Firstly, the training time of adversarial training is significantly longer than vanilla training. This is because adversarial examples have to be generated "on the fly" and most adversarial training strategies use iterative methods to generate them. Secondly, while improving adversarial robustness, it often has the consequence of hurting the standard accuracy. Some work believe that robustness and accuracy are fundamentally against each other [21, 29]. Thirdly, previous work show that adversarial training can easily cause over-fitting [7, 30].

As the research community continues to develop more powerful methods of adversarial attacks, it is becoming increasingly difficult to develop a unified defense solution. Currently, there are still many obstacles needed to be addressed. To allow DL continuously facilitate us in a wider range of applications, we believe that building robust and secure DNN classifiers is as important as enhancing their performance in natural environments.

Chapter 3

Self-Paced Adversarial Training

3.1 Introduction

In existing AT methods, untargeted attacks are widely used in model optimization and evaluation [6, 7, 21, 23, 31–33]. Unlike targeted attacks that aim to misguide a model to a particular class other than the true one, untargeted adversaries don't specify the targeted category and perturb the clean data so that its prediction is away from its true label. In theory, adversarial perturbation in untargeted attacks can be added along arbitrary directions, thus leading to uniformly-distributed false predictions. However, we observe that misclassification statistics after adversarial attacks highly deviate from a uniform distribution over categories. Figure 3.1 presents the misclassification statistics of PDG-attacked dog and cat images, where almost half of dog images are misclassified as cats, and over 40% of the cat images are misclassified as dogs. Considering that cat and dog images share many common features in vision, we raise the following questions:

"Does the unbalanced inter-class semantic similarity lead to the non-uniformly distributed misclassification statistics? If ***yes****, are classification predictions of untargeted adversaries predictable?"*

To answer these questions, this chapter revisits the recipe for generating gradient-based first-order adversaries and surprisingly discovers that untargeted attacks may be targeted! In theory, we prove that adversarial perturbation directions in untar-

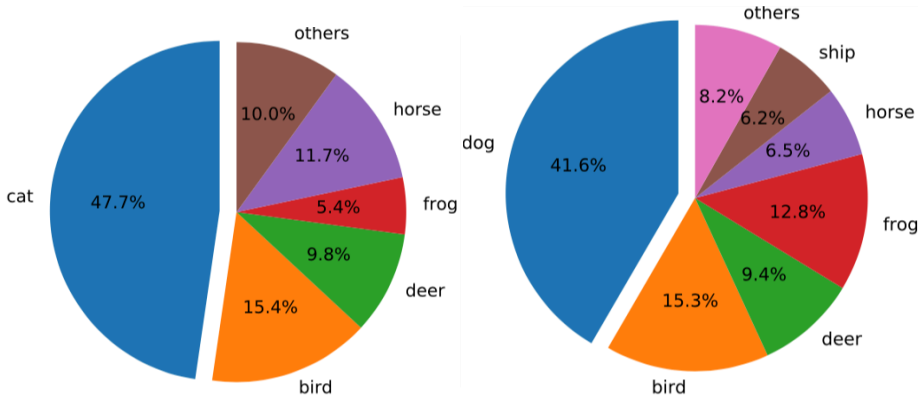
geted attacks are actually biased toward the hard-class pairs of the clean data under attack. Intuitively, semantically-similar classes constitute **hard-class pair (HCP)** and semantically-different classes form **easy-class pair (ECP)**.

Inspired by this intriguing yet far-overlooked aspect of untargeted adversarial attacks, we come up with the conclusion that HCPs deliver the most informatics knowledge for model optimization. Accordingly, we propose explicitly taking the inter-class semantic similarity into account in AT algorithm design and develop a self-paced adversarial training (SPAT) strategy to upweight/downweight hard/easy-class pair loss, encouraging the training procedure to neglect redundant information from easy class pairs. Since HCPs and ECPs may change during model training (depending on the current optimization status), their scaling factors are adaptively updated at their own pace. Such self-paced reweighting offers SPAT more optimization flexibility. In addition, we further incorporate an HCP/ECP consistency term in SPAT and show its effectiveness in boosting model adversarial robustness. Our main contributions are:

- We investigate the cause of the non-uniformly distributed misclassification statistics in untargeted attacks. We find that adversarial perturbations are actually biased by targeted sample's hard-class pairs.

- We introduce a SPAT strategy that takes inter-class semantic similarity into account. Adaptively upweighting hard-class pair loss encourages discriminative feature learning.

- We propose incorporating an HCP/ECP consistency regularization term in adversarial training, which boosts model adversarial robustness by a large margin.



(a) PDG attacks on Dog (b) PDG attacks on Cat

Figure 3.1: Predictions of untargeted adversarial attacks (PGD-20) by a CIFAR-10 vanilla-trained classifier. The standard accuracy of the classifier is 95.53%. After the attacks, almost half of dog images are misclassified as cats (a) and over 40% of the cat images are misclassified as dogs (b).

3.2 Related Work

3.2.1 Adversarial Attack and Defense

The objective of adversarial attacks is to search for human-imperceptible perturbation $\boldsymbol{\delta}$ so that the adversarial sample

$$\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{\delta} \tag{3.1}$$

can fool a model $f(\boldsymbol{x}; \boldsymbol{\phi})$ well-trained on clean data $\boldsymbol{x}$. Here ϕ represent the trainable parameters in a model. For notation simplification, we use $f(\boldsymbol{x})$ to denote $f(\boldsymbol{x}; \boldsymbol{\phi})$ in this chapter.

Adversarial Training uses regulation methods to directly enhance the robustness of classifiers. Such optimization scheme is often referred to as the "min-max game":

$$\underset{\boldsymbol{\phi}}{\operatorname{argmin}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim D}[\max_{\delta \in S} \mathscr{L}(f(\boldsymbol{x'}), \boldsymbol{y})], \tag{3.2}$$

where the inner max function aims to generate efficient and strong adversarial perturbation based on a specific loss function $\mathscr{L}$, and the outer min function optimizes the network parameters $\boldsymbol{\phi}$ for model robustness. Another branch of AT aims to achieve

logit level robustness, where the objective function not only requires correct classification of the adversarial samples, but also encourages the logits of clean and adversarial sample pairs to be similar [21, 23, 31]. Their AT objective functions usually can be formulated as a compound loss:

$$\mathscr{L}(\boldsymbol{\theta}) = \mathscr{L}_{acc} + \lambda \mathscr{L}_{rob} \tag{3.3}$$

where $\mathscr{L}_{acc}$ is usually the cross entropy (CE) loss on clean or adversarial data, $\mathscr{L}_{rob}$ quantifies clean-adversarial logit pairing, and λ is a hyper-parameter to control the relative weights for these two terms. The proposed SPAT in this chapter introduces self-paced reweighting mechanisms upon the compound loss and soft-differentiates hard/easy-class pair loss in model optimization for model robustness boost.

3.2.2 Re-weighting in Adversarial Training

Re-weighting is a simple yet effective strategy for addressing biases in machine learning, for instance, class imbalance. When class imbalance exists in the datasets, the training procedure is very likely over-fit to those categories with a larger amount of samples, leading to unsatisfactory performance regarding minority groups. With the re-weighting technique, one can down-weight the loss from majority classes and obtain a balanced learning solution for minority groups.

Re-weighting is also a common technique for hard example mining. Generally, hard examples are those data that have similar representation but belong to different classes. Hard sample mining is a crucial component in deep metric learning [34, 35] and Contrastive learning [36, 37]. With re-weighting, we can directly utilize the loss information during training and characterize those samples that contribute large losses as hard examples. For example, OHEM [38] and Focal Loss [39] put more weight on the loss of misclassified samples to effectively minimize the impact of easy examples.

In adversarial training, previous studies shows that utilizing hard adversarial sam-

ples promotes stronger adversarial robustness [7, 23, 40, 41]. For instance, MART [23] explicitly apply a re-weighting factor for misclassified samples by a soft decision scheme. Recently, several re-weighting based algorithms have also been proposed to address fairness related issues in AT. [42] adopt re-weighting strategy to address the data imbalance problem in AT and showed that adversarially trained models can suffer much worse performance degradation in under-represented classes. Xu *et al.* [43] empirically showed that even in balanced datasets, AT still suffers from fairness problem, where some classes have much higher performance than others. They propose to combine re-weighting and re-margin for different classes to achieve robust fairness. Zhang *et al.* [44] propose to assign weights based on how difficult to change the prediction of a natural data point to a different class. However, existing AT re-weighting strategies only considered intra-class or inter-sample relationships, but ignored the inter-class biases in model optimization. We propose to explicitly take the inter-class semantic similarity into account in the proposed SPAT strategy and up-weights the loss from hard-class pairs in AT.

3.3 Untargeted Adversaries are Targeted

Untargeted adversarial attacks are usually adopted in adversarial training. In theory, adversarial perturbation in untargeted attacks can be added along arbitrary directions, leading to uniformly-distributed false predictions. However, our observations on many adversarial attacks contradict this. For example, when untargeted adversaries attack images of cats, the resulting images are likely to be classified as dogs empirically. We visualize image embeddings from the penultimate layer of the vanilla-trained model via t-SNE in Figure 3.2. In the figure, the embeddings of dog and cat images are close to each other, which suggests the semantic similarity in their representations. With this observation, we hypothesize that the unbalanced inter-class semantic similarity leads to the non-uniformly distributed misclassification statistics.

In this section, we investigate this interesting yet overlooked aspect of adversarial

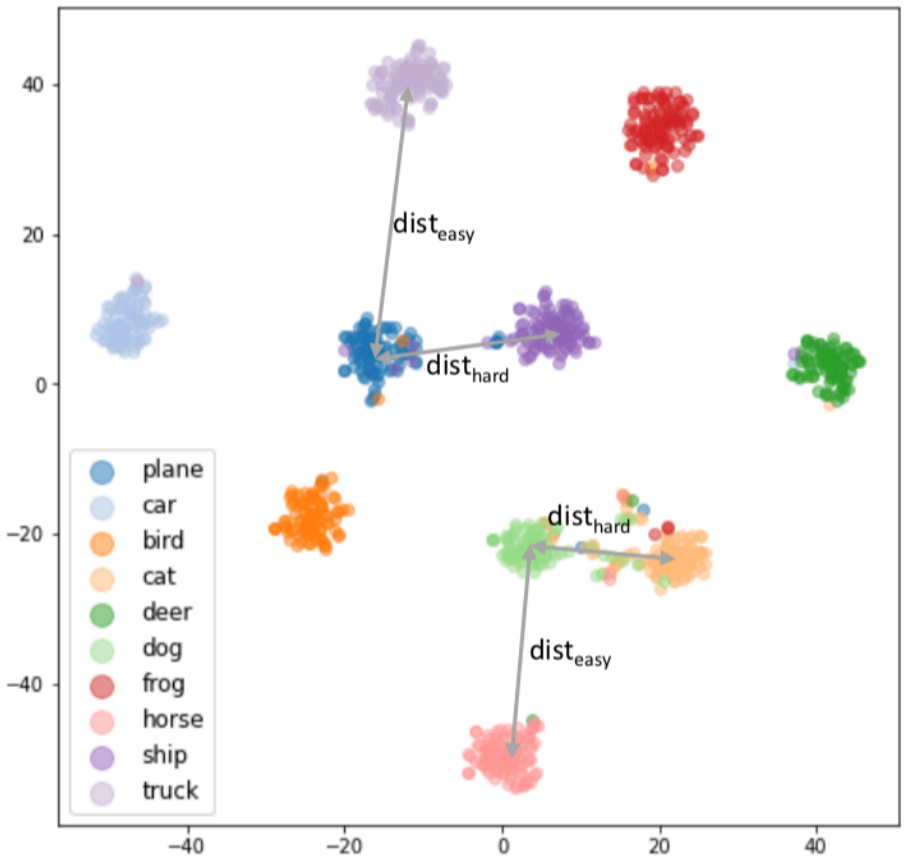


Figure 3.2: t-SNE visualization of 1000 randomly sampled image embeddings from CIFAR-10. Due to the naturally imbalanced semantic similarity, inter-class distance is much smaller for hard-class pairs.

attacks and find that untargeted adversarial examples may be highly biased by their hard-class pairs. The insight in this section directly motivates the proposed self-paced adversarial training for model robustness improvement.

3.3.1 Notations

Given a dataset with labeled pairs $\{\mathscr{X}, \mathscr{Y}\} = \{(x, y)|x \in \mathbb{R}^{c\times m\times n}, y \in [1, C]\}$, a classifier can be formulated as a mapping function $f : \mathscr{X} \to \mathscr{Y}$:

$$f(x) = \mathbb{S}(\boldsymbol{W}^T \boldsymbol{z}_{\boldsymbol{x}}), \tag{3.4}$$

where C is the number of categories, and $\mathbb{S}$ represents the softmax function in the classification layer. We use $\boldsymbol{z}_{\boldsymbol{x}}$ to denote the representation of an input sample x

in the penultimate layer of the model and $\boldsymbol{W} = (\boldsymbol{w_i}, \boldsymbol{w_2}, ..., \boldsymbol{w_C})$ for the trainable parameters (including weights and bias) of the softmax layer. Note that $\boldsymbol{w_i}$ can be considered as the prototype of class i and the production $\boldsymbol{W}^T \boldsymbol{z_x}$ in (3.4) calculates the similarity between $\boldsymbol{z_x}$ and different class-prototype $\boldsymbol{w_i}$. During training, the model f is optimized to minimize a specific loss $\mathscr{L}(f(x), y)$.

In literature, the most commonly used adversarial attacks, such as PGD and its variants, generate adversaries based on first-order derivative information about the network [7]. Such adversarial perturbations can be generally formulated as follows:

$$x' = x + \epsilon g(\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y)), \tag{3.5}$$

where ϵ is the step size to modify the data and $\nabla_{\boldsymbol{x}}$ is the gradient with respect to the input x. We take g to denote any function on the gradient, for example, $g(x) = \|x\|_p$ is the ℓ_p norm.

3.3.2 Bias in Untargeted Adversarial Attacks

The first-order adversarial attacks usually deploy the CE loss between the prediction $f(x)$ and the target y to calculate adversarial perturbations. The CE loss can be formulated as

$$\mathscr{L}(f(x), y) = -log \frac{e^{\boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w_j}^T \boldsymbol{z_x}}} \tag{3.6}$$

For notation simplification in the rest of this chapter, we have $\sigma(\boldsymbol{w_i}^T \boldsymbol{z_x}) = \frac{e^{\boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j=1}^{C} e^{\boldsymbol{w_j}^T \boldsymbol{z_x}}}$.

Lemma 1:

For an oracle model that predicts the labels perfectly on clean data, the gradient of the CE loss with respect to sample x from the i^{th} category is:

$$\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y) = [\sum_{j \neq i}^{C} \sigma(\boldsymbol{w_j}^T \boldsymbol{z_x}) \boldsymbol{w_j}] \nabla_{\boldsymbol{x}} \boldsymbol{z_x}. \tag{3.7}$$

Proof of Lemma 1:

For an oracle model that predicts the labels perfectly on clean data, the gradient of the CE loss with respect to sample x from the i^{th} category is:

$$\nabla_{\boldsymbol{x}}\mathscr{L}(f(x),y)=[\sum_{j\neq i}^{C}\sigma(\boldsymbol{w_j}^T\boldsymbol{z_x})\boldsymbol{w_j}]\nabla_{\boldsymbol{x}}\boldsymbol{z_x},$$

where $\sigma(\boldsymbol{w_i}^T\boldsymbol{z_x})=\frac{e^{\boldsymbol{w_i}^T\boldsymbol{z_x}}}{\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}}$.

Proof:

The CE loss can be formulated as

$$\mathscr{L}(f(x),y)=-log\frac{e^{\boldsymbol{w_i}^T\boldsymbol{z_x}}}{\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}}.$$

Hence,

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\mathscr{L}(f(x),y)&=-\nabla_{\boldsymbol{x}}log\frac{e^{\boldsymbol{w_i}^T\boldsymbol{z_x}}}{\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}}\\
&=-\nabla_{\boldsymbol{x}}[loge^{\boldsymbol{w_i}^T\boldsymbol{z_x}}-log\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}]\\
&=-\nabla_{\boldsymbol{x}}[\boldsymbol{w_i}^T\boldsymbol{z_x}]+\nabla_{\boldsymbol{x}}[log\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}]\\
&=-\nabla_{\boldsymbol{x}}[\boldsymbol{w_i}^T\boldsymbol{z_x}]+\frac{1}{\sum_{k=1}^{C}e^{\boldsymbol{w_k}^T\boldsymbol{z_x}}}\nabla_{\boldsymbol{x}}[\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}]\\
&=-\nabla_{\boldsymbol{x}}[\boldsymbol{w_i}^T\boldsymbol{z_x}]+\frac{1}{\sum_{k=1}^{C}e^{\boldsymbol{w_k}^T\boldsymbol{z_x}}}\cdot\sum_{j=1}^{C}e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}\nabla_{\boldsymbol{x}}[\boldsymbol{w_j}^T\boldsymbol{z_x}]\\
&=-\nabla_{\boldsymbol{x}}[\boldsymbol{w_i}^T\boldsymbol{z_x}]+\sum_{j=1}^{C}\frac{e^{\boldsymbol{w_j}^T\boldsymbol{z_x}}}{\sum_{k=1}^{C}e^{\boldsymbol{w_k}^T\boldsymbol{z_x}}}\nabla_{\boldsymbol{x}}[\boldsymbol{w_j}^T\boldsymbol{z_x}]\\
&=-\nabla_{\boldsymbol{x}}[\boldsymbol{w_i}^T\boldsymbol{z_x}]+\sum_{j=1}^{C}\sigma(\boldsymbol{w_j}^T\boldsymbol{z_x})\nabla_{\boldsymbol{x}}[\boldsymbol{w_j}^T\boldsymbol{z_x}]\\
&=[\sigma(\boldsymbol{w_i}^T\boldsymbol{z_x})-1]\boldsymbol{w_i}^T\nabla_{\boldsymbol{x}}\boldsymbol{z_x}+[\sum_{j\neq i}^{C}\sigma(\boldsymbol{w_j}^T\boldsymbol{z_x})\boldsymbol{w_j}]\nabla_{\boldsymbol{x}}\boldsymbol{z_x}
\end{aligned}$$

For an oracle model that predicts the labels perfectly on clean data, $\sigma(\boldsymbol{w_i}^T \boldsymbol{z_x}) = 1$ for a data from the i^{th} class. Hence, the first term in the proof vanishes. That is,

$$\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y) = [\sum_{j \neq i}^{C} \sigma(\boldsymbol{w_j}^T \boldsymbol{z_x}) \boldsymbol{w_j}] \nabla_{\boldsymbol{x}} \boldsymbol{z_x},$$

Lemma 1 indicates that for a clean data x from the i^{th} category, its first-order adversarial update follows the direction of the superposition of all false-class prototypes $\boldsymbol{w_j}$ for $j \in [1, C], j \neq i$. The weight of the j^{th} prototype $\boldsymbol{w_j}$ in the superposition is $\sigma(\boldsymbol{w_j}^T \boldsymbol{z_x})$. The greater the value of the dot product $\sigma(\boldsymbol{w_j}^T \boldsymbol{z_x})$, the more bias in adversarial perturbations toward the i^{th} category. In an extreme case where only one $\sigma(\boldsymbol{w_k}^T \boldsymbol{z_x})$ is non-zero, the untargeted attack becomes a targeted attack.

To investigate if the values of $\sigma(\boldsymbol{w_j}^T \boldsymbol{z_x})$ is equal or not, we let $v_j = \|\boldsymbol{w_j}\|_2$ and $s = \|\boldsymbol{z_x}\|_2$ be the Euclidean norm of the weight and data embedding. Then (3.7) in Lemma 1 can be rewritten as $\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y) = [\sum_{j \neq i}^{C} \sigma(v_j s \cos(\boldsymbol{\theta}_j)) \boldsymbol{w_j}] \nabla_{\boldsymbol{x}} \boldsymbol{z_x}$, where $\cos(\boldsymbol{\theta}_j)$ measures the angle between the two vectors $\boldsymbol{w_j}$ and $\boldsymbol{x_z}$. Here, we discussed two conditions.

Condition 1:

We regulate $v_j = 1$ and thus convert the CE loss to the normalized cross entropy (NCE) loss in Lemma 1. Recently, many studies show that NCE loss encourages a model to learn more discriminative features [45–47]. Furthermore, such hypersphere embedding boosts adversarial robustness [41]. When we follow NCE's regularization and enforce $v_j = 1$, (3.7) in Lemma 1 is further simplified to

$$\nabla_{\boldsymbol{x}} \mathscr{L}(f(x), y) = [\sum_{j \neq i}^{C} \sigma(s \cos(\boldsymbol{\theta}_j)) \boldsymbol{w_j}] \nabla_{\boldsymbol{x}} \boldsymbol{z_x}, \tag{3.8}$$

Since $\sigma()$ is a monotonically increasing function, the adversarial update direction is significantly biased by large $\cos(\boldsymbol{\theta}_j)$. It is noteworthy that $s \cos(\boldsymbol{\theta}_j)$ quantifies the projection of a data representation $\boldsymbol{x_z}$ onto the j^{th} class prototype $\boldsymbol{w_j}$, which reflects the inter-class similarity between z_x and a specific false-class prototype. Therefore,

this chapter defines the false classes associated with a higher $\cos(\boldsymbol{\theta}_j)$ as the **hard-class pairs** of data x; contrastively, the false classes with large $\boldsymbol{\theta}_j$ as the **easy-class pairs**. With this context, we conclude that the adversarial perturbations introduced by the NCE loss are dominated by those **hard** classes with smaller inter-class distances from the true data category.

Condition 2:

We relax the condition $v_j = 1$ and extend our discovery to a generic CE loss. Though v_j can be any value in theory, we empirically find that their values are quite stable and even for all j as shown in Table 3.1. With these observations, we conclude that untargeted adversaries are actually targeted; Furthermore, the virtual targeted categories are its hard-class pairs.

Models	ResNet-18	ResNet-34	ResNet-50
v_0	1.113	1.036	1.072
v_1	1.134	1.063	1.106
v_2	1.103	1.023	1.055
v_3	1.082	1.005	1.032
v_4	1.113	1.037	1.068
v_5	1.096	1.023	1.057
v_6	1.122	1.048	1.083
v_7	1.121	1.046	1.085
v_8	1.127	1.051	1.096
v_9	1.121	1.046	1.088

Table 3.1: Weight norms of the softmax layer in CE-trained models on CIFAR-10

Figure 3.3 illustrates a geometric interpretation of our discovery in a simple triplet classification setting, with $y = \{-1, 0, 1\}$. We assume the latent representation of class -1 is closer to class 0 (hard class pair) and class 1 is farther from class 0 (easy

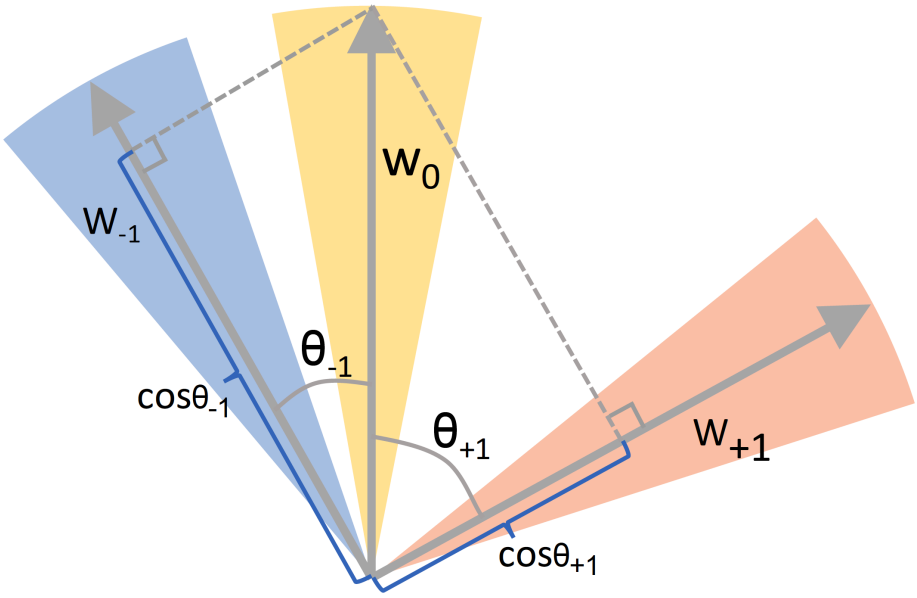


Figure 3.3: A geometric interpretation of our discovery about untargeted attacks. Different colors represent different classes and W_i is the prototype vector for class i. According to Lemma 1, the overall attack direction for class 0 will be dominated by class -1.

class pair). Since $cos(\boldsymbol{\theta}_{-1}) > cos(\boldsymbol{\theta}_{+1})$, The attack direction of samples from class 0 is dominated by class -1. Therefore, the data from class 0 is adversarially modified towards class -1.

3.4 Self-Paced Adversarial Training

Our discovery in Section 3.3.2 motivates the innovation of our re-weighting strategy in the proposed SPAT in twofold.

- From the perspective of learning robust, discriminative features. Compared to adversaries from hard-class pairs having similar semantic representations, easy-class pairs contribute less to model optimization. Encouraging a model to learn HCP samples facilitates the model to extract good features.

- From the perspective of adversarial defense of untargeted attacks. Thanks to

the discovered targeted property of untargeted attacks, we know that many clean data are adversarially modified toward their hard-class pairs. With this prior knowledge of untargeted attacks, one can improve models' robustness by learning HCP adversaries in AT.

With above considerations, our self-paces strategy proposes to up-weights training sample's hard-class pair loss in adversarial training.

Specifically, following prior arts in adversarial training, the proposed SPAT strategy adopts a compound loss:

$$\mathscr{L}^{SPAT} = \mathscr{L}^{sp}_{acc} + \lambda \mathscr{L}^{sp}_{rob} \tag{3.9}$$

where λ is the trade-off parameter for the accuracy and robustness terms. Notably, we introduces distinct up-weighting policies in $\mathscr{L}^{sp}_{acc}$ and $\lambda\mathscr{L}^{sp}_{rob}$, which encourages the model learning from hard-class pairs.

3.4.1 Self-Paced Accuracy Loss

According to our empirical observations and theoretical analysis in Section 3, untargeted attacks are prone to generate adversaries from hard-class pairs. We argue that a model with stronger HCP discrimination capability would be more robust against adversarial attacks. To this end, we propose up-weighting HCP loss and down-weighting ECP loss in model training to facilitate discriminative representation learning.

As analysis in Section 3.2, $\cos(\boldsymbol{\theta}_j)$ evaluates the representation similarity between $\boldsymbol{z_x}$ and the prototype vector $\boldsymbol{w_j}$ of the j^{th} class. Ideally, for a data from the i^{th} category, we target $\cos(\boldsymbol{\theta}_j) = \delta(i-j)$, where $\delta(x)$ is the Dirichlet identity function. Toward this goal, we monitor the values of $\cos(\boldsymbol{\theta}_j)$ and take them as metrics to adaptively re-weight training samples in adversarial training.

Formally, we propose to reshape the NCE loss by the self-paced modulating factors g^t and g^f_j:

$$\mathscr{L}^{sp}_{acc} = -\log\left(\frac{e^{g^t \boldsymbol{w_i}^T \boldsymbol{z_x}}}{\sum_{j\neq i}^{C} e^{g^f_j \boldsymbol{w_j}^T \boldsymbol{z_x}} + e^{g^t \boldsymbol{w_i}^T \boldsymbol{z_x}}}\right), \tag{3.10}$$

where $\|\boldsymbol{w_i}\|_2 = 1$ and $\|\boldsymbol{z_x}\|_2 = s$ [45]. For a sample with true label i, the true-class modulating gain g^t and false-class weights g_j^f are defined as

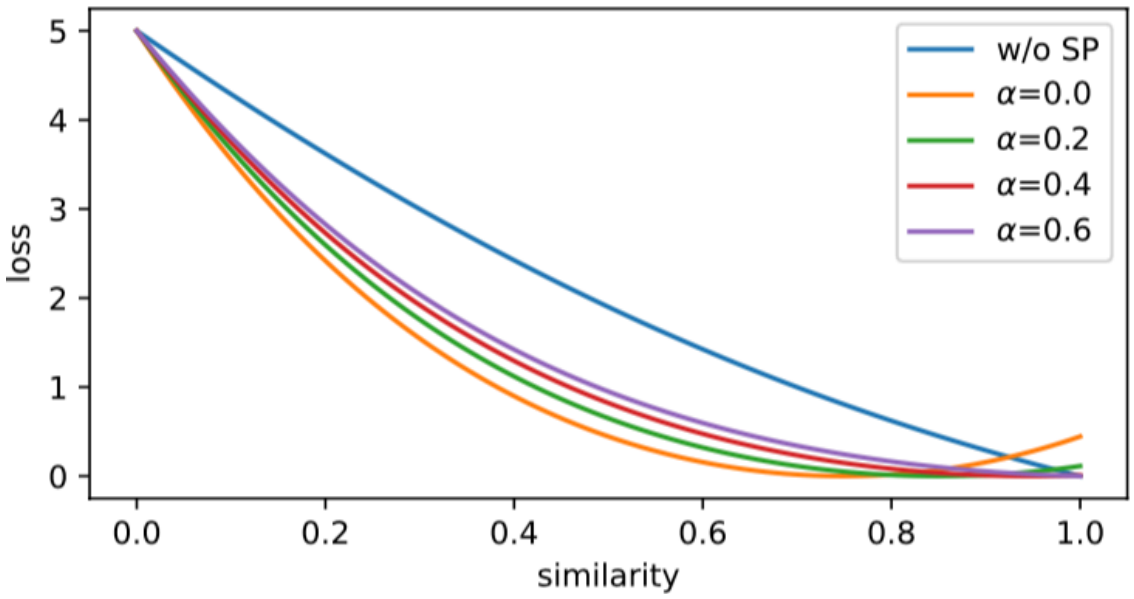
$$\begin{cases} g^t = 1 - \cos(\boldsymbol{\theta}_i) + \beta \\ g_j^f = \cos(\boldsymbol{\theta}_j) + \beta \end{cases} . \tag{3.11}$$

β is a smoothing hyper-parameter to avoid $g^t = 0$ and $g_j^f = 0$. This study adopts the NCE loss, rather than the CE loss, in $\mathscr{L}_{acc}^{sp}$ for the following reasons. NCE is a hypersphere embedding. Compared to the CE loss, the directional embedding encourages a model to learn more discriminative features [45–47]. Recent study in [41] further shows that deploying NCE in adversarial training boosts model's robustness against various attacks. It is noteworthy that our ablation study shows that the proposed self-paced modulating mechanism not only boosts model robustness with the NCE loss, it also improves model performance with the CE loss.

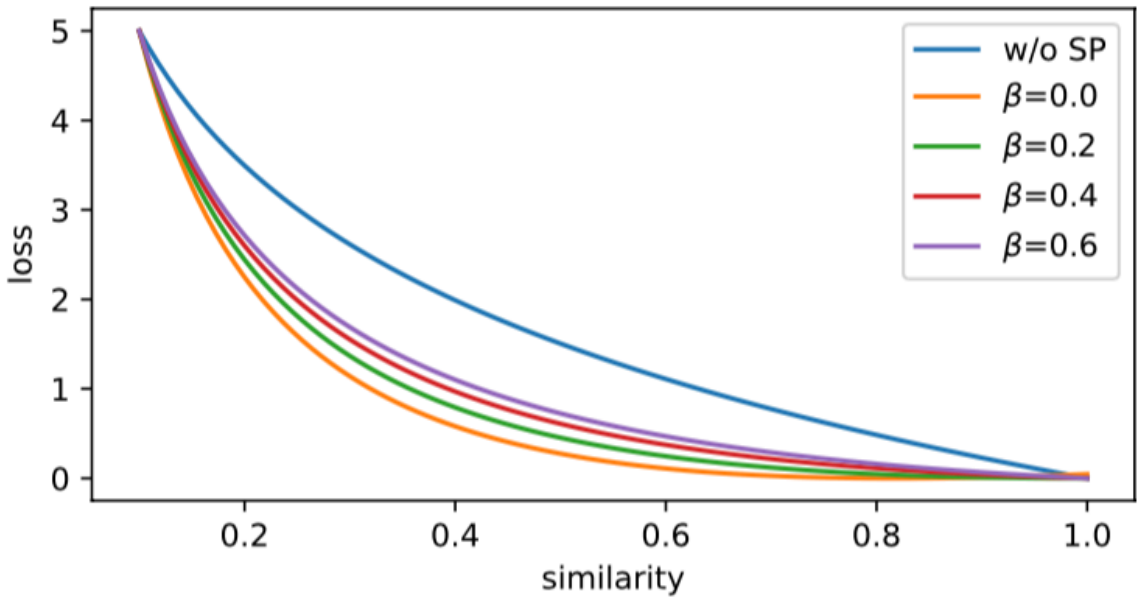
Intuitively, the introduced self-paced modulating factors amplify the loss contribution from hard-class pairs, meanwhile down-weight easy-class pair loss. Specifically, according to (3.11), a data from the i^{th} category is associated with large g^t and g_j^f when its representation z_x is far away from its true-class prototype vector $\boldsymbol{w_i}$ while close to a false-class prototype $\boldsymbol{w_i}$. In this scenario, z_x and a false-class prototype vector $\boldsymbol{w_j}$ constitutes a hard-class pair and both g^t and g_j^f amplify the loss in (3.10), encouraging the model to learn a better representation. On the other hand, when z_x and a false-class prototype vector $\boldsymbol{w_j}$ constitutes an easy-class pair with small $\cos(\boldsymbol{\theta}_j)$, g_j^f is small and thus reduces the ECP contributions to model optimization.

3.4.2 Self-Paced Robustness Loss

The robustness loss term in AT encourages a model to generate the same label to both clean data x and their adversarial samples x'. Intuitively, given a robust representation model, x and x' should share the same hard-class pairs and easy-class pairs. From our analysis in Section 3.2, such an HCP/ECP consistency constraint on x and



(a) Effects of α to KL loss



(b) Effects of β to CE loss

Figure 3.4: The loss value with respect to the cosine similarity between the true class vector and the predicted vector (S_p) for a binary classification task. We propose to use the SP modulating factor (w_{sp}) to adjust the relative weights between easy samples and hard samples. For those samples that can be correctly classified with high confidence (similarity$>$ 0.6), the relative losses are reduced. Therefore, enforcing the training process to focus more on hard class pairs.

x' can be formulated as:

$$\cos(\boldsymbol{\theta}_j) \approx \cos(\boldsymbol{\theta'}_j), \forall j. \tag{3.12}$$

$\boldsymbol{\theta'}_j$ is the angle between $z_{x'}$ and a prototype vector $\boldsymbol{w_j}$ in the softmax layer of a model.

In prior arts, KL divergence is a widely used as a surrogate robust loss in AT [21, 23]. It quantifies the difference between predicted logits on clean data and its adversarial version:

$$KL(f(x)\|f(x')) = \sum_{i=1}^{C} f_i(x) log \frac{f_i(x)}{f_i(x')}. \tag{3.13}$$

Though the KL divergence measures the logit similarity from the point of view of statistics, it doesn't impose the aforementioned HCP/ECP consistency constant in (3.12) on model optimization.

In this study, we propose a new regularization factor, $L_{inc}(x, x')$, to penalize HCP/ECP inconsistency in model robustness training. With simple math, (3.12) can be converted into a more intuitive expression: $f_j(x) \approx f_j(x')$ for all j. To accommodate the two inconsistency conditions, $f_j(x) \gg f_j(x')$ and $f_j(x) \ll f_j(x'))$, within one formula, we propose the use of $[log\frac{f_j(x)}{f_j(x')}]^2$ to quantify the HCP/ECP inconsistency between x and x with respect to a specific class j. Another benefit of the square operation is its amplification effect on large values, which encourages the model to satisfy the HCP/ECP consistency constraint. Instead of accumulating all inconsistency penalties direction, we follow the statistic perspective of computing KL divergence and the new regularization factor is formulated as

$$L_{inc}^{sp}(x, x') = \sum_{j}^{C} [f_j(x) log \frac{f_j(x)}{f_j(x')}]^2. \tag{3.14}$$

Therefore, our new robustness loss is

$$\mathscr{L}_{rob}^{sp} = \alpha KL(f(x)\|f(x')) + L_{inc}^{sp}(x, x'), \tag{3.15}$$

where α is a hyper-parameter to balance the two robustness terms.

The pseudocode of the proposed SPAT algorithm is presented in Algorithm 1.

Algorithm 1 Self-Paced Adversaral Training

Input: Number of training data N, batch size m, number of iterations for inner optimization K, maximum perturbation ϵ, step sizes η_1 and η_2, classifier parameterized by θ
Output: Robust classifier f_θ
for $s = 1...\ N/m$ **do**
    read mini-batch $\{\boldsymbol{x_1}, ..., \boldsymbol{x_m}\}$ from training data
    for $i = 1...\ m$ **do**
        $\boldsymbol{x_i'} = \boldsymbol{x_i} + 0.001 * \boldsymbol{\mathcal{N}(0, I)}$ ▷ $\mathcal{N}(0, I)$ is the standard normal distribution
        for $j = 1...\ K$ **do**
            $\boldsymbol{x_i'} = \Pi_{\mathscr{B}(\boldsymbol{x_i}, \epsilon)}(x_i' + \eta sign(\nabla_{\boldsymbol{x}} \mathscr{L}_{rob}^{sp}(\boldsymbol{x_i}, \boldsymbol{x_i'}; \boldsymbol{\theta}))$ ▷ Π is the projection operator
        end for
    end for
    $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta_2 \sum_i^m \nabla_{\boldsymbol{\theta}} \mathscr{L}^{SPAT}(\boldsymbol{x_i}, \boldsymbol{y_i}, \boldsymbol{x_i'}; \boldsymbol{\theta})$
end for

3.5 Experiments

In this section, we first evaluate SPAT on two popular benchmark datasets, MNIST and CIFAR10, in both white-box and black-box settings. Then we conduct a comprehensive empirical study on the proposed SPAT, providing an in-depth analysis of the method. A comparison study with state-of-the-art AT methods is presented.

3.5.1 Robustness Evaluation under Different Attacks

In this section, we evaluate the robustness of SPAT on two benchmarks, MNIST and CIFAR10, under various attacks.

Experimental settings:

For MNIST, we use a simple 4-layer-CNN followed by three fully connected layers as the classifier. We apply 40-step PGD to generate adversaries in training, with $\epsilon = 0.3$ and step size of 0.01. We train the models for 80 epochs with the learning rate of 0.01. Since MNIST is a simple dataset, three classical attacks, FGSM [4], PGD-20 [7], and C&W with l_∞ [28], are deployed in our white-box and black-box settings.

On CIFAR-10, adversarial samples used in ATs are generated by 10-step PGD, with

defense	Clean	FGSM	PGD-20	C&W
Madry's	99.15	97.22	95.51	95.66
ALP	98.79	97.31	95.85	95.50
TRADES	99.10	97.42	96.22	96.01
MART	98.89	97.70	96.24	96.33
SPAT	**99.21**	**98.12**	**96.64**	**96.57**

Table 3.2: White box robustness accuracy(%) on MNIST

defense	Clean	FGSM	PGD-20	C&W
Madry's	99.15	97.06	96.00	96.88
ALP	98.79	97.23	96.13	97.32
TRADES	99.10	97.27	96.88	97.03
MART	98.89	97.68	96.73	97.20
SPAT	**99.21**	**97.80**	**97.27**	**97.40**

Table 3.3: Black box robustness accuracy(%) on MNIST.

$\epsilon = 8/255$ and step size of $\epsilon/4$. The rest training setup is the same as in section 3.5.2. Since CIFAR-10 is a more complex dataset, we further include four stronger attacks in this experiment, which are PGD-100, MIM [48], FAB [49], and AutoAttack (AA) [50]. All attacks are bounded by the l_∞ box with the same maximum perturbation $\epsilon = 8/255$.

Baselines:

SOTA defense methods including Madry's [7], ALP [31], TRADES [21], MART [23], GAIRAT [44], and MAIL-AT [**liu2021probabilistic**] are evaluated in this comparison study. We follow the default hyperparameter settings presented in the original papers. For instance, $\lambda = 6$ in TRADES and 5 in MART. For ALP, we set the weight for logit paring as 0.5.

defense	Clean	FGSM	PGD-20	PGD-100	MIM-20	FAB	C&W	AA
Madry's	84.35	54.23	46.70	45.73	47.03	47.67	48.62	46.90
ALP	**85.21**	54.07	46.19	44.78	46.55	47.60	48.80	46.44
TRADES	82.12	56.49	51.82	50.21	51.25	48.21	49.96	47.32
MART	83.08	60.19	54.87	52.97	53.91	48.62	**51.23**	47.87
GAIRAT	83.14	60.03	54.85	52.68	53.44	37.11	40.73	35.90
MAIL-AT	83.80	61.33	55.06	53.26	54.57	45.55	48.67	44.32
SPAT	84.26	**62.27**	**59.56**	**58.57**	**59.07**	**48.74**	50.56	**48.33**

Table 3.4: White box robustness accuracy(%) on CIFAR-10 with ResNet-18.

defense	Clean	FGSM	PGD-20	PGD-100	MIM-20	FAB	C&W	AA
Madry's	84.35	79.84	80.35	80.91	80.12	81.93	79.98	82.02
ALP	**85.21**	80.07	81.20	81.04	80.77	82.46	81.33	83.01
TRADES	82.12	79.98	80.69	80.80	80.24	81.71	80.55	81.91
MART	83.08	81.50	82.31	82.89	82.04	83.02	82.97	83.06
GAIRAT	83.14	79.92	80.40	80.61	80.22	82.49	82.43	82.69
MAIL-AT	83.80	81.22	82.16	82.37	81.96	83.10	82.38	83.36
SPAT	84.26	**82.54**	**83.45**	**83.33**	**82.95**	**84.23**	**83.96**	**84.25**

Table 3.5: Black box robustness accuracy(%) on CIFAR-10 with ResNet-18.

White-Box Robustness:

Table. 3.2 and Table 3.4 report the white-box robustness performance on MNIST and CIFAR-10, respectively. SPAT achieves the highest robustness in all 4 attacks on MNIST and 6 out of 7 on CIFAR-10. The only exception is the l_∞ C&W attack which directly optimizes the difference between correct and incorrect logits [7]. Notice that the optimization function of the C&W attack (l_∞ version) is the same as the objective function (boosted cross entropy) for MART which makes the rest defense strategies in an unfair position. Even so, SPAT is only 0.67% less robust than MART under C&W attack. We shown in Appendix that the proposed SPAT also works well with larger models such as WideResNet-34.

Black-Box Robustness:

In the black-box attack setting, since adversaries do not access the model architecture and parameters, adversarial samples are crafted on a naturally trained model and transferred to the evaluated models. Here we use a naturally trained LENET-5 [25] and ResNet101 for adversarial sample generation, whose natural accuracy is 98.94% and 95.53% on MNIST and CIFAR-10 respectively.

Table. 3.3 and Table 3.5 report the white-box robustness performance on MNIST and CIFAR-10, respectively. Since the features for MNSIT is simple and linear, we notice for certain cases the black box attacks are even stronger than the white box attacks. For example, white box FGSM attack is weaker than its black box counterpart on all defenses. On the CIFAR10 dataset, while all models reach much higher robustness accuracy compared to white box attacks, SPAT again achieves the top performance. It is worth noting that the weakest attack (FGSM) has the highest black box transferability, while the strongest attack method, AutoAttack, has almost no effect on the SPAT trained model (from 84.26% to 84.25%).

In addition, our experimental results on CIFAR-10C in Appendix suggest that the model trained by SPAT is also robust to naturally image corruptions.

3.5.2 Breaking Down SPAT

To gain a comprehensive understanding of SPAT, three sets of ablation experiments are conducted: (1) Sensitivity to hyper-parameters, (2)Remove the SP factors in the SPAT loss, and (3) Replacing NCE with CE in SPAT.

Experimental Setup:

We use ResNet-18 [51] as our classifier for CIFAR-10 dataset. Our experimental settings follow prior arts in [21, 23]. All models in this ablation study are trained 100 epochs with SGD and batch size is 128. The initial learning rate is set as 0.1 and decays by 10 times at 75^{th} and 90^{th} epoch. At AT training stage, we use 10-step PGD to generate adversarial samples, with $\epsilon = 8/255$, step size $= \epsilon/4$, and $\lambda = 6$. For evaluation, we apple 20-step PGD to generate attack data, with $\epsilon = 8/255$, step size $= \epsilon/10$. The default hyper-parameter in all experiments are $s = 5$ and $\alpha = \beta = 0.2$, unless otherwise specified.

Sensitivity of Hyper-parameters

SPAT has three newly introduced hyper-parameters, s and α in $\mathscr{L}_{acc}^{sp}$ and β in $\mathscr{L}_{rob}^{sp}$. Table 1 presents the sensitivity of these hyper-parameters on CIFAR-10 dataset and shows their impacts on model accuracy and robustness. The best performance metrics are highlighted in bold. Similar to NCE[41, 45], the scale factor s in SPAT regulates the length of embeddings. From Table 3.6c, a larger s leads to higher robustness but lower accuracy. This is because a larger s indicates a larger spherical embedding space and thus samples from different classes can be distributed more discretely. However, the relatively-sparse sample distribution in the large embedding space increases the difficulty of classification. α and β are parameters up-weighting hard-class pair loss in SPAT. As shown in Table 3.6a and 3.6b, appropriately choosing α and β can boost model robustness with little accuracy degradation.

α	Clean	PGD-20
0.0	**84.66**	58.32
0.2	84.26	59.56
0.4	83.60	60.11
0.6	83.01	**60.57**

(a) Varying α in $\mathscr{L}_{rob}^{sp}$

β	Clean	PGD-20
0.0	**85.03**	57.88
0.2	84.26	59.56
0.4	83.81	**59.64**
0.6	82.66	57.62

(b) Varying β in $\mathscr{L}_{acc}^{sp}$

s	Clean	PGD-20
1	**87.57**	49.52
3	86.16	55.77
5	84.26	59.56
8	82.54	60.24
10	81.24	**61.02**

(c) Varying s in $\mathscr{L}_{acc}^{sp}$

Table 3.6: Hyper-parameter sensitivity in SPAT. If unspecified, the default values are: $s = 5, \alpha = \beta = 0.2$.

Analysis of SP:

Table 3.7 records the performance when removing the proposed self-paced factors in the SPAT loss function. Note, when removing SP weights in the accuracy loss, we let $g^t = g_j^f = 0$ and the proposed self-paced NCE loss becomes the original NCE loss. As indicated in Table 3.7, removing the SP mechanism from either robustness loss or accuracy loss leads to substantial performance degradation. In particular, the introduced self-paced robustness term encourages the model to follow the HCP/ECP consistency constraint, which contributes to a larger margin of robustness improvement.

loss functions	Clean	PGD-20
$L_{nce}^{sp} + \lambda L_{rob}^{sp}$	**84.26**	**59.56**
$L_{nce} + \lambda L_{rob}^{sp}$	82.49	58.74
$L_{nce}^{sp} + \lambda L_{rob}$	84.01	56.14
$L_{nce} + \lambda L_{rob}$	83.33	54.58

Table 3.7: Removing SP factors from SPAT.

loss functions	Clean	PGD-20
$L_{nce}^{sp} + \lambda L_{rob}^{sp}$	**84.26**	**59.56**
$L_{ce}^{sp} + \lambda L_{rob}^{sp}$	82.86	53.55
$L_{ce} + \lambda L_{rob}$	82.12	51.82

Table 3.8: Replacing NCE with CE in SPAT.

Analysis of NCE in SPAT:

This study introduces the self-paced modulation factors upon the NCE loss. Table 3.8 compares model performance when we replace NCE with either the CE loss or a self-paced CE loss (by relaxing normalization $v_j = 1$). The normalization regularization in NCE boosts both model robustness and standard accuracy. In addition, incorporating the self-paced factors into the CE loss also improves model performance. This observation validates our innovation of up-weighting hard-class pair loss in model optimization.

3.5.3 Case Study: Naturally Corrupted Perturbation

Adversarial attack is the most extreme scenario for evaluating the robustness of models. Unlike adversarial attacks, naturally-corrupted data, such as blurring, compression, defocusing, *etc*, do not require model information to generate noises and can be seen as a type of generic black-box attack. In this experiment, we explore the potential of SPAT on such naturally-corrupted data. We apply the SPAT-trained

ResNet-18 in Section 5.2 on the corrupted CIFAR-10 dataset (*CIFAR-10-C* [52]). In CIFAR-10-C, the clean CIFAR-10 data are processed to mimic various image distortions under harsh conditions. Table 3.9 presents classification accuracy on the CIFAR-10-C dataset. Results show that the SPAT-trained model exhibits stronger robustness to different types of corruption.

defense	PGD	TRADES	MART	SPAT
Blur	73.23	72.43	73.35	**74.71**
Contrast	78.68	76.05	76.59	**78.39**
Fog	46.38	45.74	45.19	**49.34**
Frost	70.59	64.65	70.39	**71.36**
Snow	73.95	70.08	74.92	**75.02**
jpeg	81.09	79.09	80.42	**81.73**
Saturate	80.23	78.68	80.51	**81.92**
Defocus	77.66	76.05	76.59	**78.39**

Table 3.9: Accuracy (%) of different corruption types in CIFAR10-C.

3.6 Conclusion and Future work

In this chapter, we studied an intriguing property of untargeted adversarial attacks and concluded the direction of first-order gradient-based attack is largely influenced by its hard-class pairs. With this insight, we introduced a self-paced adversarial training strategy and proposed up-weighting hard-class pair loss and down-weighting easy-class pair loss in model optimization. Such an online re-weighting strategy on hard/easy-class pairs encouraged the model to learn more useful knowledge and disregard redundant, easy information.Extensive experiment results show that SPAT can significantly improve the robustness of the model compared to state-of-the-art AT strategies.

In the future, on one hand, we plan to apply the hard/easy-class pair re-weighting

principles to recently proposed AT algorithms, and explore the potential improvement by differentiating hard/easy-class pairs in AT. On the other hand, we plan to investigate "true" untargeted adversarial attacks so that the adversarial perturbations are less predictable.

Chapter 4

Adversarial Fine-tune with Dynamically Regulated Adversary

4.1 Introduction

To improve model adversarial robustness, many defense strategies, such as data augmentation, gradient masking, adversarial example detection, and adversarial training have been proposed with the aim of finding countermeasures to protect DNNs [7, 48, 53, 54]. Particularly, adversarial training is widely recognized as the most effective solution. It incorporates adversarial data in model training and helps build model robustness to adversarial attacks. Despite its success in improving model robustness to adversarial data, state-of-the-art (SOTA) adversarial training strategies have been observed to cause model performance degradation [7, 21, 22, 41]. For instance, SOTA adversarial training methods such as TRADES [21] loses about 10% standard accuracy for a 50% adversarial robustness improvement on CIFAR-10 image set. This observation has led to a discussion of the relationship between adversarial robustness and standard generalization (e.g., classification accuracy), with a central debate on whether accuracy and robustness are intrinsically in conflict. Some studies, for example, Tsipras *et al.* [55] and Raghunathan *et al.* [29], claim that the trade-off of model accuracy and adversarial robustness is unavoidable due to the nature of DNNs. In contrast, Raghunathan *et al.* [29, 56] argue that the robustness-accuracy trade-off could disappear with unlimited data. Yang *et al.* [57] present a theoretical analysis, as

well as a proof-of-concept example, showing that this trade-off is not inherent and arguing that the observed accuracy-robustness trade-off is introduced by limitations in current adversary training methods. More recently, Xie *et al.* [19] leverage adversarial samples to improve model accuracy by introducing an auxiliary batch normalization layer particularly designed for adversarial samples. However, this study does not discuss if the model robustness is improved or not. To summarize, although adversarial training improves model robustness against adversarial attacks, how to achieve this goal without trading off model accuracy on clean data in adversarial training is still an open question and remains under-explored.

It should be noted that a major loss of standard performance on clean data is unacceptable in many applications that might cause severe consequences. Instead, the standard performance is more valued than model robustness against malicious adversarial attacks. Such applications include medical diagnosis, autonomous surgical robotics, etc. This leads to the question: To what extent can we boost model robustness without sacrificing standard performance? Unlike prior adversarial training strategies that allow model performance degradation on clean data, we investigate if it is feasible to improve adversarial performance without any standard performance loss. Specifically, this chapter proposes a pre-train based adversarial training strategy, where adversarial training is applied to clean-data (vanilla-trained) models. To prevent catastrophic forgetting in model refining, we follow a replay-based strategy and maintain the 1:1 clean-adversarial data ratio in model refinement. To further reduce adversarial samples' negative impacts on standard accuracy, we incorporate novel dynamically regulated adversarial (DRA) samples in model refinement. Unlike most adversarial attacks that add adversarial noise to every image pixel, DRA searches for highly stimulated adversarial features and generates adversarial samples accordingly. Such strategy in adversarial sample generation enforces the model refinement to learn descriptive but non-robust features. Extensive experimentation shows that the proposed adversarial training strategy improves model adversarial robust-

ness with a large margin, but without standard performance loss. The contributions of this study are summarized as follows.

- We propose a simple, yet generic adversarial training strategy that improves adversarial robustness by a large margin without sacrificing standard accuracy. The proposed method is particularly useful for applications where standard performance is more valued than adversarial robustness.
- We further introduce an unbounded adversarial attack method, namely DRA. It introduces smaller image distortions and facilitates the adversarial refinement to focus on the learning of the most descriptive but non-robust features.
- We show that our adversarially trained models exhibit robustness not only to adversarial samples; but also against naturally corrupted images, which suggests its potential for real-world applications.

The rest of this chapter is organized as follows. Section 4.2 presents a brief review of related works. Section 4.3 formulates the target problem and specifies our motivations. The technical details of the proposed method are elaborated in Section 4.4. Section 4.5 presents extensive experiments and discussions, then followed by conclusions in section 4.6.

4.2 Related Work

Since Madry's MinMax optimization is prone to over-fit adversarial samples, many adversarial training methods mix clean data and adversarial data in training.

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim D}[L(x, y; \theta) + \max_{\delta \in S} L(x + \delta, y; \theta)]. \tag{4.1}$$

Note, in (4.1), adversarial examples are leveraged to regularize the vanilla training on clean data. Recently, Raghunathan *et al.* [29] introduce robust self-training (RST) to balance the vanilla and adversarial loss by a regularization parameter $\beta > 0$.

$$\mathcal{L}(\theta, \beta) = \mathbb{E}_{(x,y)\sim D}[L(x, y; \theta) + \beta \max_{\delta \in S} L(x + \delta, y; \theta)]. \tag{4.2}$$

Another regularized adversarial training strategy, TRADES [21], is proposed to boost model robustness following the Locally-Lipschitz smoothness constraint.

$$\begin{aligned} \mathcal{L}(\theta, \lambda) &= \mathbb{E}_{(x,y)\sim D}[L(f_\theta(x), y) \\ &+ 1/\lambda \cdot \max_{\delta \in S} L(f_\theta(x+\delta), f_\theta(x))], \end{aligned} \tag{4.3}$$

where f_θ denotes the training model parameterized by θ. Unlike RST computing an adversarial loss between the prediction $f_\theta(x+\delta)$ and label y as the regularization term in (4.2), TRADE regularizes the training by calculating $L(f_\theta(x+\delta), f_\theta(x))$ from a pair of clean sample x and its adversarial version $x+\delta$. The regularization parameter λ determines the trade-off between accuracy and robustness in the overall optimization. It is worth to notice that a small regularization parameter helps the model emphasize more on accuracy over robustness which is closely aligned to our objective of adversarial training. However, we show later in Section V that even very small value of λ^{-1} cannot guarantee models to achieve comparable accuracy to vanilla trained models.

Pre-training is a popular training framework that can help reduce training time or improve accuracy performance for fine-tuning downstream tasks. Jeddi *et al.* [58] start with a clean data pre-trained model and fine-tune with PGD adversarial training with the aim of reducing the time cost and overfitting issue [30]. Hendrycks *et al.* [59] adversarially pre-train their model on a downsampled ImageNet and apply adversarial fine-tuning which can significantly improve model robustness on CIFAR datasets compared with adversarial training from scratch. Chen *et al.* [36] show that self-supervised pre-training such as Selfie [60], Jigsaw [61], also lead to better robustness than traditional adversarial training. In contrast to previous adversarial pre-training strategies, our approach mainly focuses on maintaining accuracy and treats adversarial robustness as an added bonus. More specifically, we incorporate a novel adversary generating method (DRA) to facilitate our goal by reducing the learning complexity of adversarial training.

4.3 Primitives

4.3.1 Problem Formulation

In this study, we focus on improving model robustness to imperceptible, in-distribution adversarial samples defined by Fawzi *et al.* [62]. Briefly, assume that clean data follows a distributing D, in-distribution adversarial samples (x') can also be roughly described within D.

To answer the question: to what extent we can boost model robustness without sacrificing standard performance, we formulate the problem as

$$\begin{aligned} \mathcal{L}(\theta) = \mathbb{E}_{(x,y)\sim D}[\max_{\delta \in S} L(x+\delta, y; \theta)] \\ \text{s.t. } \mathbb{E}_{(x,y)\sim D}[L(x,y;\theta)] \leq \mathbb{E}_{(x,y)\sim D}[L(x,y;\theta_{std})], \end{aligned} \tag{4.4}$$

where $\theta_{std} = \underset{\theta}{\text{argmin}} \mathbb{E}_{(x,y)\sim D}[L(x,y;\theta)]$ represents the model with parameter θ_{std} that yields the minimized standard loss. Comparing the new problem in (4.4) with previous adversarial training methods, the new regularization term in (4.4) explicitly defines the behavior of the model: to improve adversarial robustness without the loss of model's standard performance.

4.3.2 Motivation

To answer the question in (4.4), we re-examine SOTA adversarial training strategies and obtain an interesting observation. Yang *et al.* [57] show that many real image sets, such as MNIST, CIFAR-10, SVHN and Restricted ImageNet, are r-separated, with the smallest inter-category sample distance being no smaller than $2r$. Furthermore, their empirical separation distance is 3x-7x larger than the typical adversarial perturbation constraint ϵ adopted in prior arts, i.e. $\epsilon < r$. In theory, any r-separated dataset has more than one classifier that are both accurate and robust up to perturbations of size r. This r-separated claim aligns well with adversarial robustness experiments on the MNIST dataset in previous adversarial training studies [7, 21, 41].

Table 4.1: Performance of adversarial training methods on MNIST and CIFAR-10. Adversarial accuracy is evaluated with PGD attacks.

	MNIST ($\epsilon = 0.3$)		CIFAR-10 ($\epsilon = 8/255$)	
	$\mathcal{A}_{std}$	$\mathcal{A}_{rob}$	$\mathcal{A}_{std}$	$\mathcal{A}_{rob}$
Vanilla	99.3%	0.3%	93.0 %	0.0%
Madry's [7]	99.2%	95.6%	87.3%	47.0%
Trades($1/\lambda = 1$) [21]	99.3%	94.1%	86.6%	44.3%
Trades($1/\lambda = 6$) [21]	99.3%	96.0%	81.2%	53.5%
MART [41]	99.1%	96.2%	83.4%	52.8%

However, as summarized in Table 4.1 where the standard performance and adversarial performance are denoted by $\mathcal{A}_{std}$ and $\mathcal{A}_{rob}$ respectively, on CIFAR-10 which is a more complicated dataset, previous adversarial training often leads to around 10% standard accuracy drop for models to gain desired adversarial robustness.

From the above observation, we hypothesize that the observed trade-off between adversarial robustness and standard accuracy is due to the limitation of model capacity. The problem of image classification on the MNIST dataset is relatively easy and the models adopted in previous adversarial training studies have enough capacity to jointly benefit from standard and adversarial samples. However, for complex problems such as classification on CIFAR-10 dataset, conventional adversarial training in 3.2 increases the difficulty in model optimization due to the noisy representation of adversarial perturbations. In another word, learning both standard features and adversarial features (i.e. those non-robust, yet highly predictive patterns [3]) is beyond the capacity of those models (e.g. ResNet-18, ResNet-50, etc.).

Under the hypothesis that model capacity is not enough to simultaneously learn standard and adversarial features, we value clean-data accuracy over adversarial robustness. Thus, we made two modifications to existing adversarial training strategies

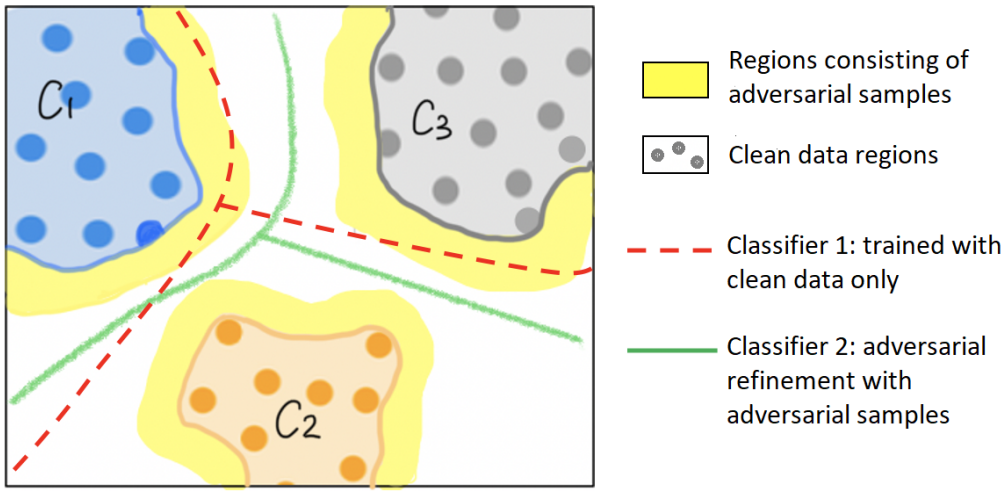


Figure 4.1: A conceptual illustration of our adversarial fine-tune method. Classifier 1 is the clean data pre-trained classifier which is accurate but not robust to adversarial samples. Our adversarial fine-tuning method seeking for both accurate and robust classifier by pushing the classifier 2 out of the yellow adversarial regions defined by $\delta \in S$.

to tackle the problem formulated in (4.4). Briefly, we propose a heuristic transfer learning based adversarial training strategy that starts with a clean data pre-trained model which already has strong generalizability on clean data. To further reduce the learning complexity, we incorporate easy-to-learn adversarial samples in the adversarial fine-tuning. Please refer to the next section for details of the proposed method.

4.4 Methodology

4.4.1 Transfer Adversarial Training

SOTA adversarial training methods usually train a robust model from scratch using either adversarial data only in (3.2) or a combination of clean and adversarial data following (4.1-4.3). However, none of them guarantee that the standard performance will be preserved. Indeed, due to the highly non-convex loss surface in model optimization, optimizing both targets simultaneously may be at odds with each other [55].

To solve the primal conditioned optimization problem in (4.4), we apply the Karush-

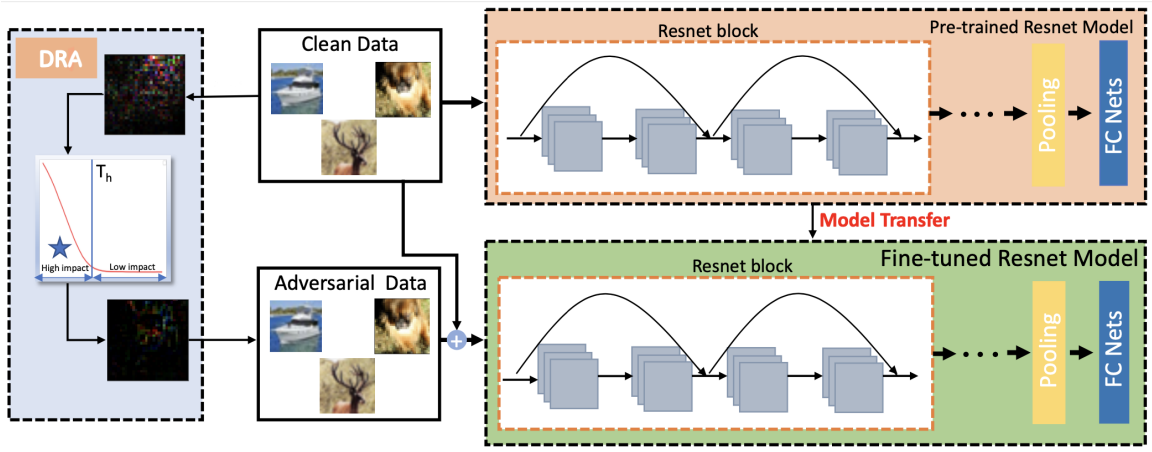


Figure 4.2: Systematic diagram of the proposed adversarial transfer adversarial training strategy, where we use various ResNets as the backbone. From left to right: DRA, a proposed method to generate adversarial samples. It filters out negligible adversarial noises and reduces adversarial training complexity. The orange block and green block represent standard training and robust training, respectively. in standard training, we train a model with θ_{std} on clean data that yields high standard performance. Then the robust training aims to find a better θ in the vicinity of θ_{std} to boost adversarial robustness without standard performance loss.

Kuhn-Tucker (KKT) approach and obtain a dual unconstrained optimization problem by introducing a KKT coefficient λ,

$$\begin{aligned} \mathcal{L}(\theta, \lambda) &= \mathbb{E}_{(x,y)\sim D}[\max_{\delta \in S} L(x+\delta, y; \theta)] \\ &+ \lambda \mathbb{E}_{(x,y)\sim D}[L(x, y; \theta) - L(x, y; \theta_{std})]]. \end{aligned} \tag{4.5}$$

Therefore, the solution to the dual problem

$$\min_{\theta} \max_{\lambda, \lambda > 0} \mathcal{L}(\theta, \lambda) \tag{4.6}$$

is identical to the solution of the primal problem in (4.4). Note that to solve the problem in (4.6), we need the margin value $\mathbb{E}_{(x,y)\sim D}[L(x, y; \theta_{std})]$ which is fixed before the adversarial training described in (4.6). That is, a model with θ_{std} is already obtained for high standard performance. Therefore, instead of training a robust model with θ from scratch, we introduce a transfer learning strategy to solve (4.6) and propose to search a new θ from θ_{std}, as the standard performance is unlikely to severely change in the small vicinity of θ_{std}. Fig. 4.1 visualize the conceptual idea

of the propose adversarial training strategy on a r-separated dataset. Classifier 1 is vanilla-trained over clean data. It is accurate but nor adversarial robustness. By pushing the classification boundary out of the yellow adversarial regions defined by $\delta \in S$, the obtained classifier 2 has the identical standard performance with improved adversarial robustness.

The detailed systematic diagram of the proposed adversarial training strategy is depicted in Fig. 4.2. Specifically, the proposed training strategy divides the training into two phases: vanilla standard training and adversarial robust training. In standard training, we exploits clean data to train an accurate model. The standard training has two benefits. First, it provides the margin value in (4.6). Second, due to inherent transfer learning property, the downstream adversarial robust training is more cost-efficient than SOTA adversarial training strategies that often require large training loads and long training time to handle the complexity introduced by adversarial features.

With the clean-data pre-trained model, adversarial samples are incorporated in the model refinement phase. To prevent model catastrophic forgetting in model fine-tuning, we follow the replay-based strategy and let the network iteratively update upon clean and adversarial samples.

Unlike conventional fine-tuning tends to unfreeze several outer layers to preserve the knowledge learned from the source task, we argue that it is important to update all parameters in robust training stage. Briefly, adversarial noises propagate through each layer in the model and aggregate into large prediction distortions. All layers of a robust model must contribute to the defend against adversarial attacks.

4.4.2 Dynamically Regulated Adversary in Adversarial Training

Adversarial training is usually referred to as the "MinMax" optimization game, where the adversarial samples are significant contributors. In fact, aggressive adversarial at-

tacks are preferred in adversarial training, because models trained on aggressive adversarial attacks are more resistant to weaker adversaries. Therefore previous studies have usually adopted the PGD attack in adversarial training.

A good adversarial attack approach for adversarial training should be aggressive but without greatly increasing training complexity. Although PGD is an aggressive solution for introducing noise, we argue that PDG is not the best candidate for adversarial training in the sense that it introduces excessive noisy information in model robustness training. More specifically, PGD uses a Sign() clipping method to project adversarial noises onto the L_∞ ball. It treats all image pixels equally and applies the same noise injection strategy to all pixels, regardless of their contribution to the prediction results. We argue that the treatment of adding adversarial noise to all pixels in PGD significantly increases the training complexity. We will show in the experimentation that the proposed DRA attack is a better candidate than PGD in adversarial training and facilitates various adversarial training.

This work introduces a novel gradient-based attack method, namely DRA attack. Unlike PGD, which treats image pixels equally, DRA distinguishes important pixels from others by aggressively modifying only those image features that are highly predictive, but non-robust. In this way, DRA adversaries enforce the robustness training by focusing more on these predictive features, thus helping to improve model's robustness against adversarial attacks. Specifically, DRA quantifies pixel significance by the gradient of the loss function with respect to its pixel value. A large gradient value suggests that the pixel contributes more to the image prediction. In this regard, DRA abandons the $Sign(\cdot)$ method so that it can smoothly search for the optimal adversarial samples along the gradient. In addition, when generating adversarial samples, DRA uses a more resilient distance metric L_1 to bound the adversarial noise instead of using the L_∞ constraint.

Fig. 4.3 presents a visual comparison of the DRA and PGD adversaries. Compared to PGD, DRA introduces stronger noise in the "cat" region, which is the most predic-

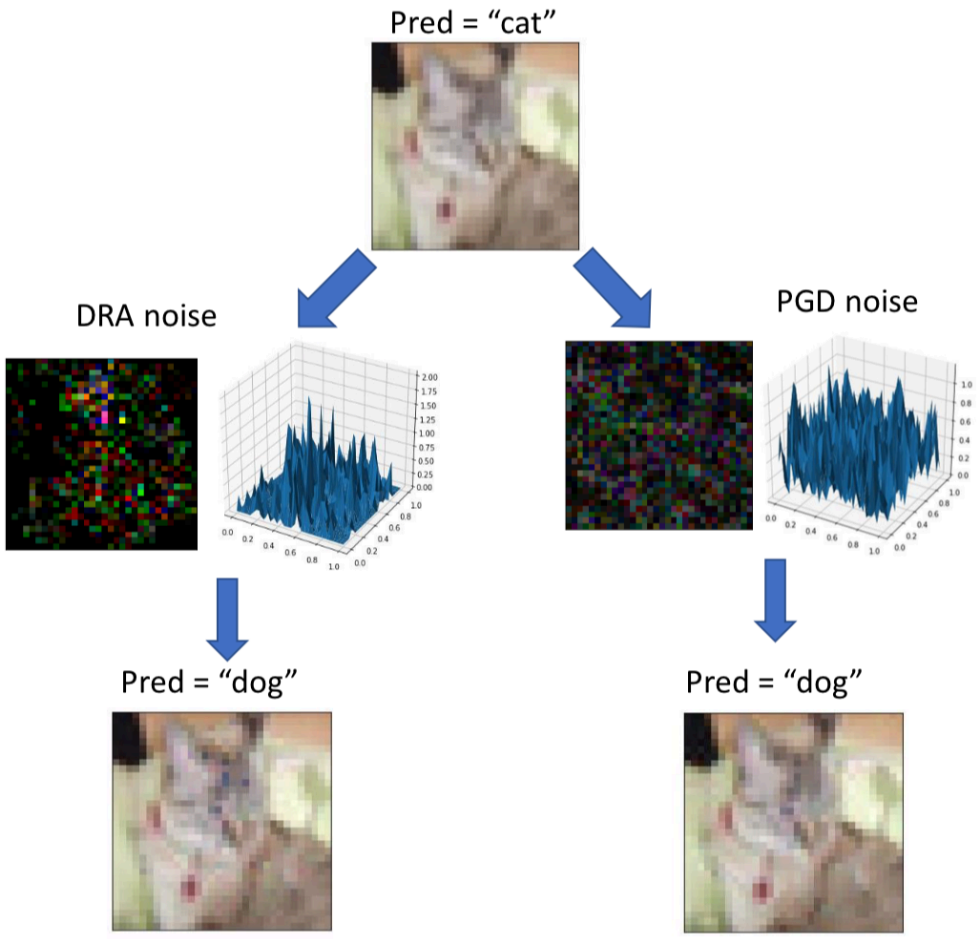


Figure 4.3: A visual comparison of PGD (left) and DRA (right) adversaries. They are both able to fool DNNs with imperceptible noises, however, the overall noise budget of DRA is smaller. Furthermore, DRA focuses on highly discriminate pixels (patterns that contribute most to final prediction outcomes), where PGD equally distributes adversarial noises over the whole image. As we can see from the example, DRA noises align well with the salient map of the cat.

tive pattern in the image. We claim that the lower noise level in the image background makes the learning more complex. We show in the experimental section that DRA samples help various training strategies achieve higher adversarial robustness than PDG for the same adversarial aggressiveness.

4.5 Experiments

In this section, we present extensive experiments to evaluate the accuracy-robustness performance of the proposed adversarial training strategy on MNIST and CIFAR-10

datasets.

4.5.1 Experimental Setup

In the proposed adversarial training strategy, we first train a model on clean data only so that the model has high standard performance. The model is then fine-tuned with clean and DRA samples. In both training processes, we utilize image augmentation including random cropping and random flipping. In addition, the image batch size is set to 128, SGD with momentum to 0.9 and weight decay to $2e-4$ for model optimization.

On the MNIST dataset, we follow the TRADES study and use a simple CNN model with 2 convolutional layers and 2 fully connected layers. We set $\epsilon = 0.1$, $p = 2/3$ and iterate 20 times to generate DRA samples for 50 epochs of robust fine-tuning.

On the CIFAR-10 dataset, we use ResNet as the backbone model and perform 60 epochs of adversarial fine-tune in the robust training phase. Adversarial samples used for our robust training are generated in 5 iterations. The noise constraint is linearly decayed from 2 to 0.5 in the first 50 epochs, and only clean images are fine-tuned in the last 10 epochs.

For evaluation purposes, adversarial samples are generated by PGD under various noise constraints (e.g. $\epsilon = 0.1, 0.3$ out of 1 for MNIST images and $\epsilon = 2, 5, 8$ out of 255 for CIFAR-10) in 20 iterations with step size $\epsilon/20$, unless otherwise specified. We compare our DRA fine-tuned model with TRADES [21] on both datasets using relatively small regularization parameter ($1/\lambda$). We find that this setting is consistent with our objective, as the small regularization parameter intuitively enforces TRADES to primarily optimize the clean data accuracy while "trading off" a small amount of adversarial robustness.

Table 4.2: Accuracy-Robustness performance against PGD attacks on MNIST. Note, the target is to improve adversarial robustness without sacrificing standard performance.

Model	natural	$\text{PGD}_{\epsilon=0.1}$	$\text{PGD}_{\epsilon=0.3}$
Vanilla trained	99.3%	88.3%	18.6%
Trades($1/\lambda = 1$)	99.3%	98.4%	93.5%
Trades($1/\lambda = 0.5$)	99.3%	97.9%	92.1%
Trades($1/\lambda = 0.1$)	99.4%	97.2%	90.8%
Ours	99.3%	98.0%	95.9%

Table 4.3: Accuracy-Robustness performance against PGD attacks on CIFAR-10, with ResNet-50 as the backbone model.

Model	natural	$\text{PGD}_{\epsilon=2}$	$\text{PGD}_{\epsilon=5}$	$\text{PGD}_{\epsilon=8}$
Vanilla trained	93.7%	45.0%	15.9%	5.6%
Trades($1/\lambda = 0.05$)	91.4%	53.5%	25.8%	8.6%
Trades($1/\lambda = 0.01$)	92.6%	49.4%	19.2%	6.3%
Ours	93.8%	64.9%	31.0%	10.9%

4.5.2 Accuracy-Robustness Performance

The numerical results on the MNIST are shown in Table 4.2, where "natural" indicates the standard performance of the model on clean data and "vanilla trained" indicates that the model is trained with clean data only. On the MNIST dataset, both TRADES and our method maintain high standard accuracy while improving model's adversarial robustness; the proposed method obtains 2% to 5% higher robustness than TRADES with various settings.

For the more complex dataset, CIFAR-10, Table 4.3 shows the results with ResNet-50 as the backbone. Unlike the MNIST dataset, ResNet-50 with the proposed adversarial training strategy significantly outperforms TRADES in terms of adversarial robustness. Note that in our experiment, the maximal ϵ of DRA in adversarial training is 2, which corresponds to ϵ =4 to 5 in PDG attacks. We believe that training

Table 4.4: Accuracy-Robustness performance against PGD attacks on CIFAR-10, with various backbone models.

Models	natural	$\text{PGD}_{\epsilon=2}$	$\text{PGD}_{\epsilon=5}$	$\text{PGD}_{\epsilon=8}$
ResNet-18 (vanilla)	93.1%	44.1%	14.6%	5.8%
	(-0.7%)	(+16.3%)	(+10.9%)	(+2.7%)
ResNet-18 (ours)	92.4%	60.4%	25.5%	8.5%
ResNet-34 (vanilla)	93.3%	46.6%	15.3%	4.5%
	(-0.4%)	(+14.8%)	(+12.6%)	(+4.4%)
ResNet-34 (ours)	92.9%	61.4%	27.9%	8.9%
ResNet-50 (vanilla)	93.7%	45.0%	15.9%	5.6%
	(+0.1%)	(+19.9%)	(+15.1%)	(+5.3%)
ResNet-50 (ours)	93.8%	64.9%	31.0%	+10.9%
ResNet-101 (vanilla)	93.7%	46.9%	15.9%	6.8%
	(+0.1%)	(+22.9%)	(+16.3%)	(+7.4%)
ResNet-101 (ours)	93.8%	69.8%	32.3%	+14.2%

the model with a large DRA ϵ would further improve the robustness.

In addition, we vary the backbone models for the CIFAR-10 experiments and report the accuracy-robustness performance in Table 4.4. First, for clean CIFAR-10 image classification, vanilla-trained models and our adversarial fine-tuned models achieve comparable performance in all examined backbone models. In particular, ResNet-50 and ResNet-101 can even outperform their vanilla training counterparts in terms of clean data accuracy. Second, to defend against PGD attacks, we observe that models with larger capacity are able to boost their adversarial robustness to a larger margin, which supports our hypothesis discussed in section 3.

4.5.3 Effect of DRA on Adversarial Training

Generation of strong adversarial samples (with higher attack success rates) is a critical factor in adversarial training. We investigate the impact of different types of adversarial samples in three different adversarial training strategies: Madry's [7], TRADES [21], and ours. Concretely, we replace PGD samples with DRA data in Madry's, and TRADES. Similarly, we use PDG in the proposed method and compare its performance with DRA. We note that DRA is a stronger adversary than PGD for the same ϵ conditions due to the presence of soft-bounded constraint. To make a fair comparison, we choose different ϵ for them so that DRA-generated samples and PGD-generated samples can have similar attack strengths. Specifically, we set $\epsilon = 1$ for DRA and $\epsilon = 2.55$ for PGD, as both settings resulted in a robust accuracy of 44.5% on vanilla-trained ResNet-50.

Table. 4.5 reports CIFAR-10 classification performance with ResNet-50 as the backbone model. We observe that in all three settings, DRA is more beneficial for improving model robustness. Furthermore, the clean data accuracy of DRA trained models is higher than that of the PGD trained models using Madry's and our method. Since the loss function of TRADES (Eqn.4.3) is essentially designed to improve robustness through trading accuracy, we believe it is reasonable to assume that the standard accuracy of the DRA trained model is slightly lower than that of the model trained with PGD. In short, Table. 4.5 with numerical results validate our hypothesis that DRA is a better adversary that benefits model's adversarial training.

4.5.4 Ablation on DRA Hyperparameter Setting

Our DRA attack algorithm filters out unimportant pixels by a pre-fixed percentage and applies adversarial noises only to important image features that contribute to the final prediction. In this ablation study, we investigate the effect of hyperparameter settings (e.g. the value of significant feature percentage p and noise budget ϵ in Algorithm 4.4.2) on CIFAR-10 dataset. Specifically, CIFAR-10 images are in size

Table 4.5: ResNet-50 trained with PGD Vs. DRA on CIFAR-10

Model	Clean data	$\text{PGD}_{\epsilon=2}$	$\text{PGD}_{\epsilon=5}$	$\text{PGD}_{\epsilon=8}$
Madry's + PGD	88.8%	66.1%	30.9%	11.0%
	(+0.6%)	(+0.6%)	(+0.7%)	(+1.5%)
Madry's + DRA	89.4%	66.7%	31.6%	12.5%
TRADES + PGD	87.9%	65.0%	34.3%	12.1%
	(-0.7%)	(+3.3%)	(+4.8%)	(+1.8%)
TRADES + DRA	87.2%	68.3%	39.1%	13.9%
ours + PGD	90.6%	59.2%	27.0%	10.1%
	(+0.7%)	(+0.8%)	(+1.9%)	(-0.3%)
ours + DRA	91.3%	60.0%	28.9%	9.8%

of $3 \times 32 \times 32$. We treat the values of red, green and blue independently and thus obtain $N = 3072$ pixel values per image. To comprehensively study this problem, we vary the values of p and ϵ and report DRA's attack success rates on CIFAR-10 vanilla-trained ResNet-18, ResNet-34 and ResNet-50.

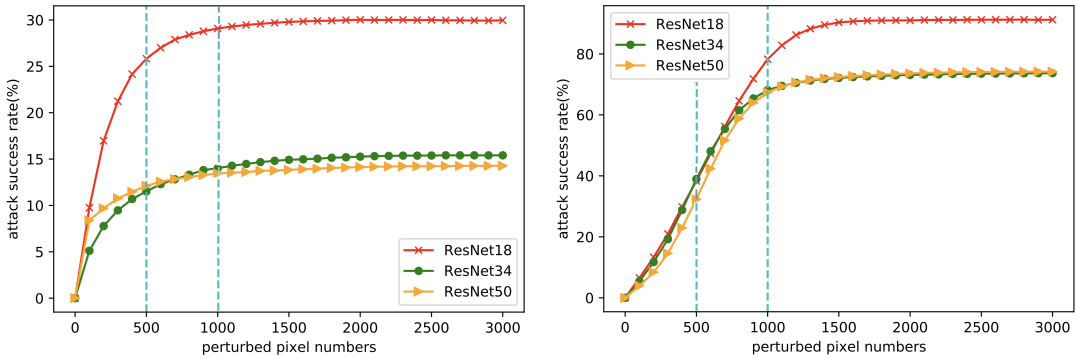
Fig. 4.4 reports adversarial attack success rate versus the percentage p out of the 3072 pixels. With the settings $\epsilon = 1$, we note that the successful attack rate of DRA grows linearly in the most significant 500 (i.e. $p = 1/6$) pixels and saturates in about 1000 ($p = 1/3$) pixels. For $\epsilon = 2$, the successful attack rate also almost saturates at approximately 1000 ($p = 1/3$) pixels.

To further investigate the impact of DRA thresholds on the overall adversarial training method, we train ResNet-50 with different values of p and report the performance in Table. 4.6. We notice that a higher threshold value p does contribute to better model robustness, however, the improvement becomes marginal as $p > 1/3$. In particular, increasing p from 1/2 to 1 only leads to less than 0.5% robustness increment for PGD attacks, but causes the downgrade of natural accuracy by 0.8%. These

Table 4.6: Ablation on DRA hyperparameter settings: adversarial training performance versus different threshold value p.

p	natural	$\text{PGD}_{\epsilon=2}$	$\text{PGD}_{\epsilon=5}$	$\text{PGD}_{\epsilon=8}$
0	93.7%	45.0%	15.9%	5.6%
1/6	91.8%	58.5%	28.0%	9.2%
1/3	91.3%	60.0%	28.9%	9.8%
1/2	90.4%	61.4%	29.1%	10.0%
1	89.6%	61.9%	29.3%	10.1%

results also support our claim that introducing too much unnecessary noise in the images complicates model optimization and tends to lead to a standard performance loss. Similar to the results in Fig. 4.4, Table 4.6 indicates that $p = 1/3$ to $1/2$ is a good setting for generating adversarial samples in the proposed method.



(a) $\epsilon = 1$ (b) $\epsilon = 2$

Figure 4.4: Ablation on DRA hyperparameter settings: DRA's attack success rates versus the significant feature percentage p on CIFAR-10 vanilla-trained ResNet models. A marginal improvement on attack success rate is observed when $p > 1/3$.

4.5.5 Evaluation on Naturally Corrupted Images

Corrupted data with naturally occurring perturbations and distributed shifts pose challenges to model generalization. In this experiment, we explore the potential of DRA fine-tuned models on corrupted data. We train a ResNet-50 model with our adversarial fine-tuning strategy and evaluate its performance on the corrupted

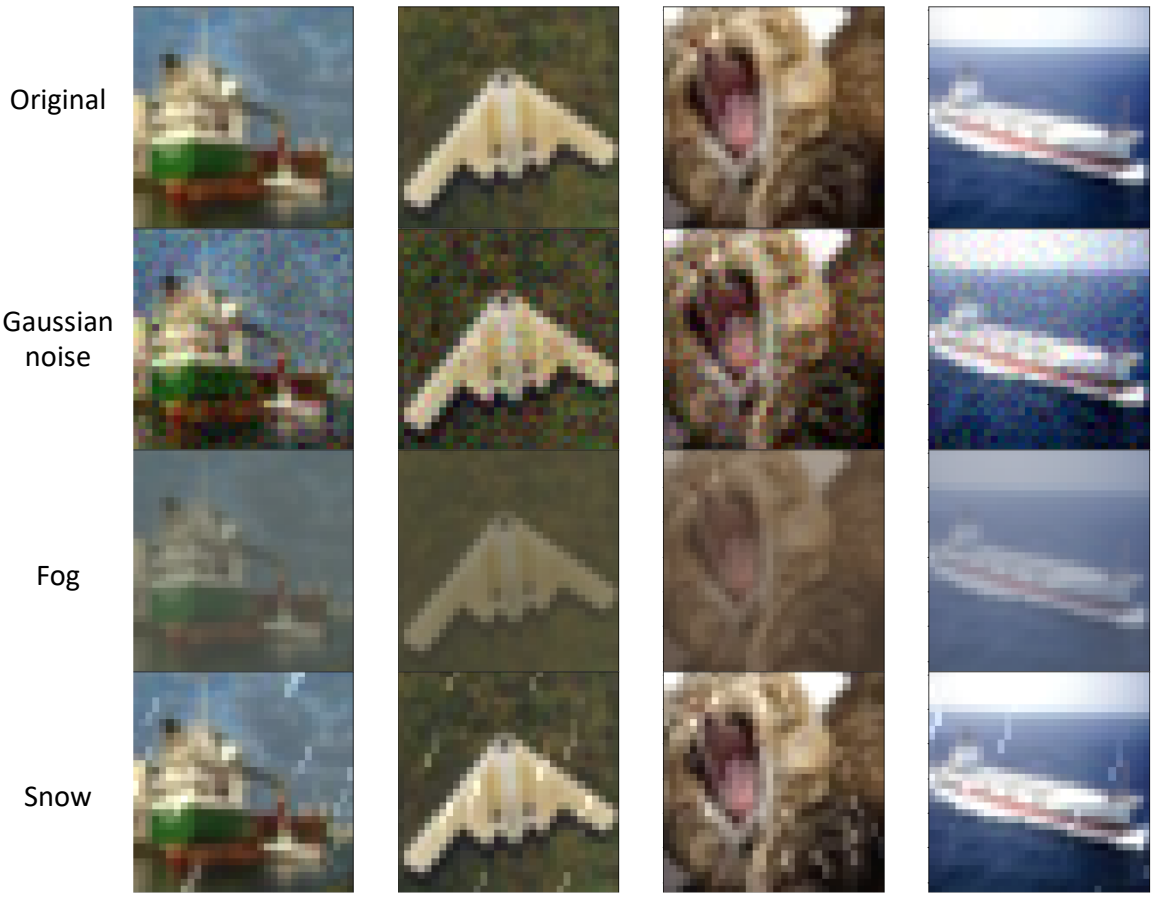


Figure 4.5: Corrupted image samples in *Cifar-10-C* [52].

CIFAR-10 dataset (*CIFAR-10-C* [52]). In CIFAR-10-C, the clean CIFAR-10 data are processed to mimic various image distortions under harsh conditions. Table 4.7 presents classification accuracy on the CIFAR-10-C dataset with the DRA refined model and the vanilla trained model. Results show that the DRA fine-tuned model exhibits stronger robustness to different types of corruption.

4.6 Conclusion

In this chapter, we aim to tackle a unique problem in adversarial training: improving the adversarial robustness of a model without sacrificing the standard performance. We explicitly formulate the problem and propose a cost-efficient adversarial training strategy. It decomposes adversarial training into two phases: standard training and

Table 4.7: The accuracy of DRA trained ResNet-50 vs. Vanilla trained ResNet-50 over different corrupted types.

Corrupted Type	ResNet-50 (DRA Trained)	ResNet-50 (Vanilla trained)
Snow	85.37%	84.10%
Frost	84.4%	81.41%
Zoom_blur	84.97%	82.13%
Motion_blur	76.89%	74.08%
JPEG compression	89.51%	84.51%
Gaussian noise	75.85%	50.92%

robust training. In addition, we introduce a training-friendly adversary to further benefit adversarial training. Extensive experimentation on MNIST and CIFAR-10 datasets suggest that the proposed adversarial training strategy serves better for the target objective.

Chapter 5

Conclusions, & Future Work

5.1 Conclusions

In the past few years, Deep Learning has achieved great success in numerous industrial applications. While many researchers continue to develop more advanced and superior algorithms, limited attention has been paid to the security and safety aspects of deep learning. That is, can we trust DNNs to make decisions for us in environments where safety or security is highly concerned? Recently, a newly proposed research area namely adversarial machine learning aims to discover if DNNs can be easily attacked or fooled. Since 2014, a variety of adversarial attack methods have been proposed to trick DNNs, but there are few methods proposed that can successfully defend against them. In this thesis, we propose two novel adversarial training methods to improve the adversarial robustness of DNNs.

In the first phase of the thesis, we discover a bias pattern from the semantic similarity between different classes that exist in most SOTA adversarial training algorithms. Inspired by this discovery, we propose Self-Paced Adversarial Training (SPAT) to balance the semantic bias in untargeted adversarial training. Experimental results show that SPAT significantly boosts adversarial robustness of DNNs compared to SOTA defenses. The second phase of the thesis is inspired by the trade-off paradigm of traditional adversarial training. That is, while many defense methods help improve adversarial robustness, they can also hurt the standard accuracy of clean data.

Therefore, we propose a more efficient adversarial training strategy to disentangle this negative effect. Extensive experiment results show that our method can help DNNs jointly benefit from adversarial data and clean data.

5.2 Future Work

With the growing interest in adversarial machine learning from the research community, there are stronger and more concealed adversarial attack methods have been proposed each period of time, which means that training an optimal robust and secure DNN model is becoming harder. We believe that certified adversarial robustness cannot be fully achieved unless we possess a deeper understanding and interpretation of deep learning models themselves as well as their optimization methods.

Bibliography

[1] C. Szegedy *et al.*, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[2] A. Pedraza, O. Deniz, and G. Bueno, "On the relationship between generalization and robustness to adversarial examples," *Symmetry*, vol. 13, no. 5, p. 817, 2021.

[3] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[6] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, IEEE, 2017, pp. 39–57.

[9] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[10] Y. Shi, S. Wang, and Y. Han, "Curls & whey: Boosting black-box adversarial attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6519–6527.

[11] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4312–4321.

[12] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[13] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," *arXiv preprint arXiv:1807.04457*, 2018.

[14] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *International Conference on Machine Learning*, PMLR, 2018, pp. 2137–2146.

[15] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," *arXiv preprint arXiv:1612.06299*, 2016.

[16] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*, PMLR, 2018, pp. 274–283.

[17] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.

[18] G. S. Dhillon *et al.*, "Stochastic activation pruning for robust adversarial defense," *arXiv preprint arXiv:1803.01442*, 2018.

[19] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.

[20] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," *arXiv preprint arXiv:1701.05517*, 2017.

[21] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*, PMLR, 2019, pp. 7472–7482.

[22] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

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

[24] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[28] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 3–14.

[29] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang, "Adversarial training can hurt generalization," *arXiv preprint arXiv:1906.06032*, 2019.

[30] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*, PMLR, 2020, pp. 8093–8104.

[31] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv:1803.06373*, 2018.

[32] A. Shafahi *et al.*, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[33] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[34] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*, Springer, 2015, pp. 84–92.

[35] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[37] P. Khosla *et al.*, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 661–18 673, 2020.

[38] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[40] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[41] T. Pang, X. Yang, Y. Dong, K. Xu, J. Zhu, and H. Su, "Boosting adversarial training with hypersphere embedding," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7779–7792, 2020.

[42] W. Wang, H. Xu, X. Liu, Y. Li, B. Thuraisingham, and J. Tang, "Imbalanced adversarial training with reweighting," *arXiv preprint arXiv:2107.13639*, 2021.

[43] H. Xu, X. Liu, Y. Li, A. Jain, and J. Tang, "To be robust or to be fair: Towards fairness in adversarial training," in *International Conference on Machine Learning*, PMLR, 2021, pp. 11 492–11 501.

[44] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli, "Geometry-aware instance-reweighted adversarial training," *arXiv preprint arXiv:2010.01736*, 2020.

[45] H. Wang *et al.*, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[46] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[47] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[48] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[49] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *International Conference on Machine Learning*, PMLR, 2020, pp. 2196–2205.

[50] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*, PMLR, 2020, pp. 2206–2216.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[52] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019.

[53] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[54] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2018.

[55] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *arXiv preprint arXiv:1805.12152*, 2018.

[56] D. Stutz, M. Hein, and B. Schiele, "Disentangling adversarial robustness and generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6976–6987.

[57] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," *Advances in neural information processing systems*, vol. 33, pp. 8588–8601, 2020.

[58] A. Jeddi, M. J. Shafiee, and A. Wong, "A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning," *arXiv preprint arXiv:2012.13628*, 2020.

[59] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[60] T. H. Trinh, M.-T. Luong, and Q. V. Le, "Selfie: Self-supervised pretraining for image embedding," *arXiv preprint arXiv:1906.02940*, 2019.

[61] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*, Springer, 2016, pp. 69–84.

[62] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *Conference on Neural Information Processing Systems*, Conference on Neural Information Processing Systems, 2018, 1186–1195.

[63] J.-B. Alayrac, J. Uesato, P.-S. Huang, A. Fawzi, R. Stanforth, and P. Kohli, "Are labels required for improving adversarial robustness?" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[64] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2712–2721.

[65] R. Zhai *et al.*, "Adversarially robust generalization just requires more unlabeled data," *arXiv preprint arXiv:1906.00555*, 2019.

[66] L. Fan, S. Liu, P.-Y. Chen, G. Zhang, and C. Gan, "When does contrastive learning preserve adversarial robustness from pretraining to finetuning?" *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[67] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 199–16 210, 2020.

[68] M. Kim, J. Tack, and S. J. Hwang, "Adversarial self-supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2983–2994, 2020.

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

[70] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[71] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[72] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv preprint arXiv:1702.04267*, 2017.

[73] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[74] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5764–5772.

[75] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 935–942.

[76] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[77] S. Chen, Z. He, C. Sun, J. Yang, and X. Huang, "Universal adversarial attack on attention and the resulting dataset damagenet," in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2020.

[78] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations*, 2017.

[79] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*, Springer, 2016, pp. 499–515.

[80] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, vol. 2, 2006, pp. 1735–1742.

[81] Y. L. Cacheux, H. L. Borgne, and M. Crucianu, "Modeling inter and intra-class relations in the triplet loss for zero-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 333–10 342.

[82] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.