Adversarial Robustness of Deep Learning Models

by

Runzhi Tian

A PhD thesis
submitted to the University of Ottawa
in partial fulfillment of the
requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Examining Committee

The following served on the Examining Committee for this thesis.

External Examiner:

Renjie Liao
Professor, Dept. of Electrical and Computer Engineering
University of British Columbia

Internal Examiner(s):

Diana Inkpen
Professor, Dept. of Electrical Engineering and Computer Science
University of Ottawa

Changjian Shui
Professor, Dept. of Electrical Engineering and Computer Science
University of Ottawa

Majid Komeili
Professor, Dept. of Computer Science
Carleton University

Supervisor(s):

Yongyi Mao
Professor, Dept. of Electrical Engineering and Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Deep neural networks (DNNs) have demonstrated remarkable success across various machine learning tasks but remain highly vulnerable to adversarial perturbations. Adversarial training (AT) and its variants aim to enhance robustness by incorporating adversarial examples into training. However, AT often leads to both standard and robust generalization issues, the causes of which remain largely elusive due to the complex learning dynamics involved.

This thesis investigates the learning behavior of AT by analyzing the evolution of perturbation-induced data distributions. Our findings reveal a surprising phenomenon: the distribution induced by adversarial perturbations during AT becomes progressively more difficult to learn. We establish a theoretical explanation for this behavior by deriving a generalization bound that attributes it to the increasing local dispersion of the perturbation operator. Experimental results validate this explanation and further link this deteriorating behavior of the induced distributions to robust overfitting in AT.

To advance the understanding of generalization in adversarial settings, we propose a unified framework for analyzing perturbation-induced loss functions. Within this framework, we introduce a novel stability analysis of AT and derive generalization upper bounds based on the expansiveness properties of adversarial perturbations. These expansiveness parameters appear to not only govern the vanishing rate of the generalization error but also govern its scaling constant. Our analysis attributes robust overfitting in Projected Gradient Descent (PGD)-based AT to the sign function used in PGD attacks, which results in poor expansiveness properties. We further show that similar issues extend to a broader class of PGD-like iterative attack algorithms, highlighting an intrinsic challenge in adversarial training.

By providing theoretical insights and empirical validations, this thesis deepens our understanding of the learning behavior of AT and paves the way for more principled approaches to improving robust generalization.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Deep neural networks (DNNs) have achieved state-of-the-art (SOTA) performance in a wide range of machine learning tasks and application domains. These advancements have been particularly evident in natural language processing [23,58,96], computer vision [28,37, 48,90], and recommendation systems [18,19,39]. Recent progress in large language models (LLMs), especially the emergence of GPT series models [11,44,70], further underscores the potential of DNNs to approach human-level intelligence.

However, despite their remarkable successes, DNNs exhibit notable limitations, particularly in aligning their behavior with human reasoning. One striking example is their vulnerability to adversarial perturbations [34, 92]. In image classification tasks, for instance, adding carefully crafted but visually imperceptible perturbations to input images can significantly alter the predictions of DNNs. In contrast, human perception remains consistent; the classification of perturbed images, as judged by humans, often aligns with that of the original images.

This misalignment between the behaviors of humans and DNNs suggests that DNNs are not yet fully reliable or trustworthy in many safety-critical application scenarios, such as autonomous driving, facial recognition, and medical diagnosis. Recent works [29, 84] clearly demonstrate these security vulnerabilities by constructing adversarial examples in real-world settings. For instance, [29] demonstrates that adding small, carefully designed patches to a stop sign can cause a DNN to misclassify it as a speed limit sign, posing a severe threat in autonomous driving systems. Similarly, [84] designs special adversarial glasses that, when worn, make a DNN misclassify the wearer's identity as another person, exposing weaknesses in facial recognition systems.

In the context of large language models (LLMs), similar security issues also exists.

LLMs are typically trained to reject harmful or unethical requests, such as providing instructions on making a bomb. For example, when a user submits such a query, existing security mechanisms prevent LLMs from generating harmful responses. However, a recent line of work [15, 100, 109, 118] reveals that by carefully modifying the harmful queries, these security mechanisms can be bypassed, eliciting inappropriate or harmful information. These techniques, referred to as "jailbreaking attacks," highlight the vulnerabilities in the security frameworks of LLMs.

These examples emphasize the importance of developing a deeper understanding for the behaviors of DNNs and the need of designing effective methods to address these security concerns.

To defend against adversarial attacks, a substantial body of research [20, 61, 76, 82, 83, 101, 115] has focused on developing revised training algorithms. These algorithms, usually referred to as adversarial training (or AT in this paper), among which the dominant approaches, such as PGD based AT [61], consider incorporating adversarial examples into the training to enhance the model's robustness towards adversarial attacks.

While AT improves a model's robustness against adversarial attacks, it introduces critical challenges related to the model's generalization performance. Specifically, the improved robustness obtained through AT often comes at the cost of reducing the testing accuracy on clean (unperturbed) samples. Additionally, models trained by AT still remain certain level of vulnerability to adversarial attacks on testing data, indicating that the robustness achieved on the training data fails to generalize to the testing data. We here briefly discuss these challenges in the following sections.

1.1 Negative Impacts on Standard Generalization

The seminal work by [94] highlights a significant drawback of adversarial training (AT): its potential to negatively impact a model's generalization performance on clean (unperturbed) data (referred to as "standard generalization"). While AT improves models' robustness against adversarial perturbations, it often leads to reduced classification accuracy on clean testing data. Using a specific theoretical data model, the authors demonstrate that robustness and standard generalization are inherently conflicting objectives, implying an unavoidable trade-off between these two goals.

However, the work of [108] challenged the inevitability of this trade-off. They argued that the conflict identified in [94] is partly due to the large perturbation radius used in their theoretical setting. This setting causes overlaps between perturbed data from different

classes, making the trade-off unavoidable. By contrast, the work of [108] shows that, in practical scenarios with real-world datasets, the perturbation radius is typically small enough to avoid such overlap. Their findings suggest that the trade-off is not inherently unavoidable and that the reduced standard generalization performance might merely be a side effect of AT, rather than an inherent limitation.

The phenomenon observed by [94] has been explored and corroborated in several subsequent studies [42,71,77,107,115,117]. These works further investigate the complex interplay between robustness and standard generalization, shedding light on how factors like data distribution, model capacity, and training strategies influence these two quantities.

1.2 Robust Overfitting

Despite that AT has been shown to have greatly improved the robustness of the learned model against adversarial attacks on the training set, a recent work in [78] has however revealed that models trained by AT may still be vulnerable to adversarial attacks on the unseen data. Specifically, after training, even though the robust error (i.e., error probability in the predicted label for adversarially perturbed instances) is nearly zero on the training set, it may remain very high on the testing set. For example, in Figure 1.1, the AT-trained model achieves nearly zero robust error on the training set of CIFAR-10 [47] (shown by the orange curve), however, its robust error on the testing set is still as high as 51.4% (shown by the blue curve). This significantly contrasts the typical observations in standard training: on CIFAR-10, when the standard error (i.e., the error probability in the predicted label for non-perturbed instances) is nearly zero on the training set, its value on the testing set is only about 4%. This phenomenon, where the robustness of an AT-trained model fails to generalize to the test set, is commonly referred to as robust overfitting.

1.2.1 Empirical works for mitigating robust overfitting

Since its discovery, robust overfitting has attracted significant research attention. A great deal of research effort has been spent on understanding its cause and devising mitigating techniques. The work of [102] and [89] correlate robust overfitting with the sharpness of the minima in the loss landscape and a method flattening such minima is presented as a remedy. Built on a similar intuition, heuristics such as smoothing the weights or the logit output of neural networks are proposed in [16]. The work of [85] suggests that the robust overfitting is related to the curvature of the activation functions and that low

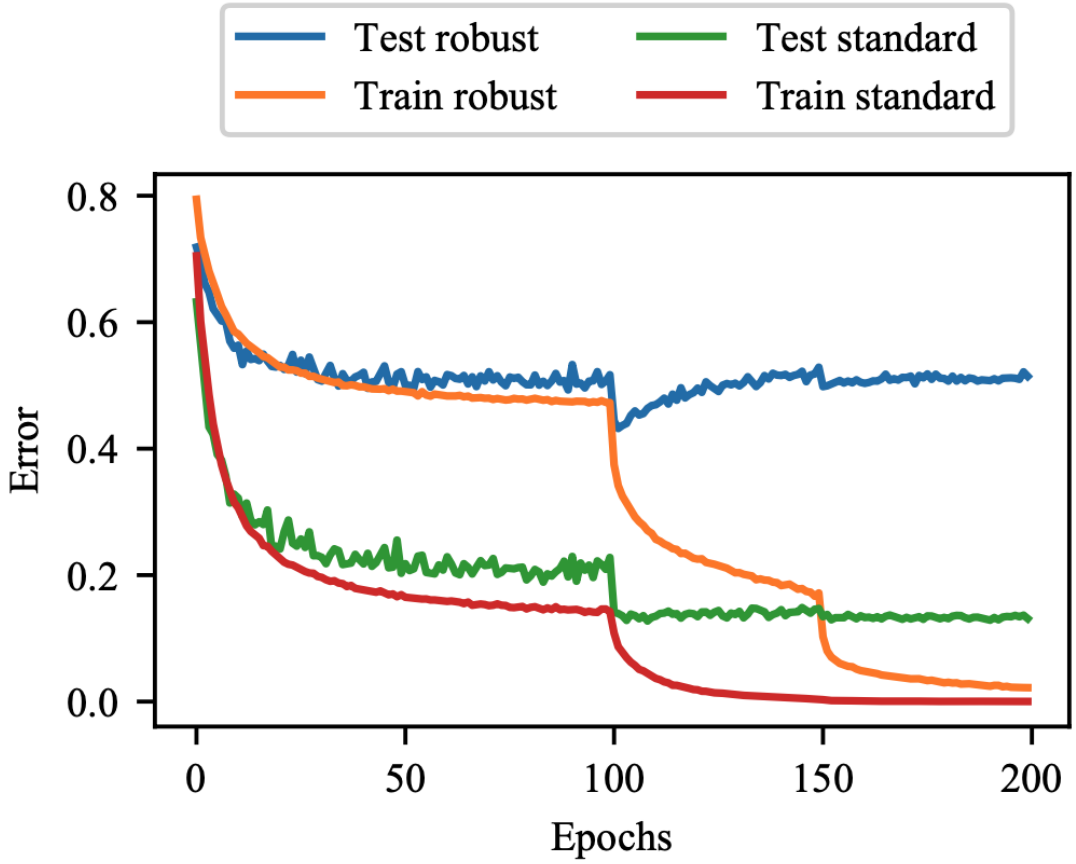


Figure 1.1: [78] Learning curves of a DNN trained by AT on CIFAR-10. The learning rate is decayed at the 100^{th} and 150^{th} training epoch. While the robust training error reaches nearly zero by the end of AT, the robust test error remains as high as 51.4%.

curvature in the activation function appears to improve robust generalization. In [25], the authors observe the existence of label noise in AT and regard it as a source of robust overfitting phenomenon, where the label noise refers to that after adversarial perturbation, the original label may no longer reflect the semantics of the example perfectly. The work of [26] attributes robust overfitting to a memorization effect and label noise in AT, and subsequently proposes a mitigation algorithm based on an analysis of memorization. The authors of [111] observe that in AT, fitting the training examples with smaller adversarial loss tend to cause robust overfitting and propose a heuristic to remove a fraction of the low-loss example during training. In [43], robust overfitting is attributed to the non-smoothness loss used in AT, and the authors propose a smoothing technique as a solution.

Numerous endeavors have been undertaken to address the challenge of robust overfitting with various empirical training algorithms proposed. [4] and [75] provide a comprehensive overview of the latest developments in empirical research in this field.

1.2.2 Theoretical works for understanding robust generalization

Different from standard generalization, robust generalization for deep neural networks —especially on high-dimensional data —appears significantly more challenging. Various work have attempted to understand the reason behind. [81] proves that in simple data models such as the Gaussian and Bernoulli models, robust generalization requires significantly higher sample complexity than standard generalization. The sample complexity of robust generalization has been further analyzed using classical statistical learning tools, including Rademacher complexity [2, 3, 45, 103, 110], VC dimension [63] and algorithmic stability analysis [104, 105], as well as the PAC learning frameworks [21, 24].

Beyond sample complexity, several theoretical perspectives have been explored. The work of [56] analyze robust generalization through the lens of neural network's expressive power, showing that practical models may lack sufficient capacity to achieve low robust test error. The authors in [57] investigate inductive bias of gradient descent for AT, while another line of research connects AT with distributionally robust optimization (DRO) [49,86]. The works of [88] and [13] demonstrate that different AT schemes can be reformulated as special cases in DRO. [7] further show that, under a saddle-point assumption, AT inevitably leads to a larger generalization gap than directly solving empirical risk minimization using adversarially perturbed data.

1.2.3 Limitations of existing works

Encouraging as these progresses are, the current understanding of robust overfitting is still arguably far from being conclusive. For example, as pointed out in [35], the explanations in [26] and [111] appear to conflict to each other: the former attributes the robust overfitting to the model fitting the data with large adversarial loss while the latter claims that fitting the the data with small adversarial loss is the source of robust overfitting. Furthermore, the proposed mitigation techniques so far, although have been shown to improve generalization, only reduce the testing robust error by a few percent. This may imply that robust overfitting can be due to a multitude of sources, the full picture remaining obscure.

It is much more difficult to develop theoretical understanding of the generalization behavior for models obtained from AT, comparing with those from standard training. In that direction, some theoretical works consider the setting where the inner maximization is perfectly solved, e.g., in [3, 110]. However, such settings are invalid for more complex neural networks, where the closed-form solution for the inner maximization is unavailable.

1.3 Thesis Outline

This thesis explores the robust generalization problem, particularly in DNNs trained via AT, from multiple theoretical perspectives, supported by extensive empirical observations. The thesis is structured as follows:

- In Chapter 2, we introduce fundamental concepts related to adversarial perturbations, adversarial training, standard generalization, and robust generalization.
- In Chapter 3, we analyze the adversarial robustness of linear classifiers trained via the support vector machine (SVM) optimization problem. We establish a connection between the SVM loss and the adversarial loss and derive a generalization bound for SVM-trained linear models using classical Rademacher complexity analysis.
- In Chapter 4, we investigate robust generalization in deep learning models by examining the training dynamics of adversarial training. Our experiments reveal a surprising phenomenon: the distribution induced by adversarial perturbations during AT becomes progressively harder to learn. We derive a generalization bound to theoretically analyze the underlying cause of this behavior. Further experiments show that this deterioration in induced distributions correlates with robust overfitting in AT. This chapter provides a new perspective on understanding AT dynamics,

emphasizing their critical role in shaping robust generalization, paving the way for a deeper understanding of robust generalization. The work in this chapter has been published in UAI 2025.

- In Chapter 5, we extend the concept of generalization beyond robust generalization by introducing a framework based on perturbation-induced losses, which encompasses both standard and robust generalization as special cases. This framework also separates the perturbations used in AT from those used in model evaluation, enabling a broader application beyond robust generalization. Building upon this framework, we derive improved generalization bounds using algorithmic stability analysis and validate our theoretical findings through extensive experiments. Our results highlight that the "expansiveness" of perturbation operators in both AT and evaluation has a profound impact on generalization, providing deeper insights into the factors governing robust generalization. This work has been published in ICLR 2025.
- In Chapter 6, we summarize the work in this thesis and discuss potential future works.

Chapter 2

Background

This chapter provides a brief overview of key background notions related to adversarial perturbations, adversarial training as well as the notion of standard and robust generalization.

2.1 Adversarial Perturbations

Deep neural networks (DNNs), despite their remarkable success across various machine learning tasks, are vulnerable to carefully crafted small perturbations in input data, which can lead to incorrect predictions [34, 92]. Szegedy et al. [92] illustrate this vulnerability with a compelling example (e.g., Figure 2.1): a DNN accurately classifies an image of a panda with 57.7% confidence. However, after the addition of a subtle, specifically designed perturbation, the network misclassifies the image as a "gibbon" with 99.3% confidence. Importantly, the difference between the original and perturbed images is nearly imperceptible to human observers, meaning that the classification result according to humans would remain unchanged. Such a perturbed input data is then termed as an *adversarial example* in [92].

Despite various definitions for adversarial examples exist (e.g., [14,73,92]), in this paper, we consider the following definition for adversarial examples.

Definition 2.1 (Adversarial examples)**.** *Consider classification problems where $\mathcal{X} \subseteq \mathbb{R}^d$ denotes the input space and $\mathcal{Y} := \{1, 2, \cdots, K\}$ denotes the label set. Given an instance-label pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, a perturbation set $T(x)$ related to x, and a classifier $h : \mathcal{X} \to \mathcal{Y}$,*

the adversarial examples of x w.r.t h is the set of instances

$$T_{adv}(x) := \{\hat{x} \in T(x) : h(\hat{x}) \neq y\}$$

Remark 2.1. The definition of adversarial examples depends not only on the given instance and label (x, y) but also on the classifier h. Specifically, under this definition, any instance in the set $T(x)$ that is misclassified by h is considered an adversarial example. Furthermore, if the original instance $x \in T(x)$ and $h(x) \neq y$, x itself is treated as an adversarial example under this definition.

Remark 2.2. The specific form of the perturbation set $T(x)$ varies under different contexts of machine learning problems. In the image classification problems, the most popular and simplest choice of $T(x)$ is $T(x) = \mathbb{B}_\infty(x, \epsilon)$ where

$$\mathbb{B}_\infty(x, \epsilon) := \{\hat{x} \in \mathbb{R}^n : \|\hat{x} - x\|_\infty \leq \epsilon\}$$

denotes an $\infty-$norm ball centered at x with radius ϵ. The perturbation radius ϵ is usually set to small values such that the perturbed instances $\hat{x} \in T(x)$ remain "visually similar" to the original instance x. This ensures that the perturbations are minimal enough to be imperceptible to humans, while still being capable of misleading the model into making incorrect predictions.

Remark 2.3. In image classification problems, choosing $T(x)$ as $\mathbb{B}_\infty(x, \epsilon)$ is not the only option. More generally, one can consider the $p-$norm ball for any $p \in (0, \infty]$, defined as

$$\mathbb{B}_p(x, \epsilon) := \{\hat{x} \in \mathbb{R}^n : \|\hat{x} - x\|_p \leq \epsilon_p\}$$

Thus, one may take $T(x) = \mathbb{B}_p(x, \epsilon_p)$ with a proper choice of perturbation radius ϵ_p for different values of p.

The approaches of constructing adversarial examples are termed as *adversarial attacks*. Various adversarial attack algorithms have been proposed, such as L-BFGS [91], FGSM [33], JSMA [74], Deep-Fool [64] and CPPNEA [69]. Some of the other adversarial attack methods are summarized by [112] in great details. In general, adversarial attacks can be treated as specific attempts to solve the following constrained optimization problem:

$$x_{\text{adv}} = \arg\max_{\hat{x} \in T(x)} \mathbb{I}(h(\hat{x}) \neq y)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function (or the 0-1 loss function). The solution of this problem may not be unique. To enable gradient methods for solving the problem when h

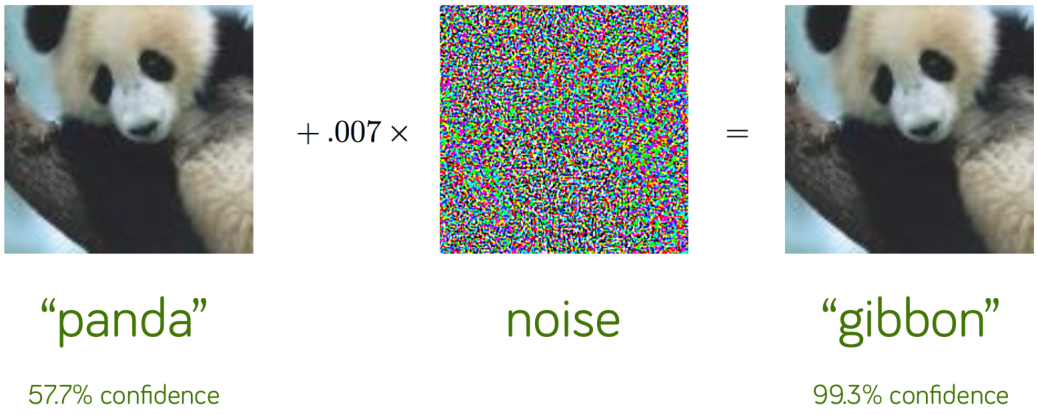


Figure 2.1: An example presented in [33], showing the vulnerability of DNNs to adversarial examples: carefully crafted small modification to the input data causes a neural network to output wrong prediction.

is parameterized by neural networks, the 0-1 loss is usually replaced by a smooth surrogate loss $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ such as the cross-entropy loss and the optimization problem turns into

$$x_{\text{adv}} = \arg \max_{\hat{x} \in T(x)} l(h(\hat{x}), y)$$

When $T(x) = \mathbb{B}_\infty(x, \epsilon)$, this constrained optimization problem can be efficiently solved by projected gradient descend (PGD),[1] which gives the well-known PGD attack [61]. We now briefly introduce the PGD method and then elaborate the well-known PGD attack.

PGD in general. PGD is used to solve constrained optimization problems of the form

$$\min_{x \in T} f(x)$$

where, with a slight abuse of notation, $x \in \mathcal{X}$ represents an optimization variable and $f : \mathcal{X} \to \mathbb{R}$ is a differentiable objective function. The set $T \subseteq \mathcal{X}$ defines the feasible region, imposing constraints on the solutions.

PGD aims to minimize f by iteratively updating x using gradient descent, while ensuring that the updated value of x remains within the feasible set T. This is achieved by

[1] or Projected Gradient Ascent in this context, since we are solving a maximization problem.

projecting the updated point back onto T after each gradient descent step. Specifically, the update rule is

$$x^{\text{new}} = \Pi_T \left(x^{\text{old}} - \lambda \nabla_x f(x^{\text{old}}) \right) \tag{2.1}$$

Here $\Pi_T(z) := \arg\min_{\hat{x} \in T} \|\hat{x} - z\|_2$ denotes the operation of projecting z onto the set T and $\lambda \in \mathbb{R}_+$ is the step size of the gradient descent.

PGD attack [61]. Let the perturbation set be $T(x) = \mathbb{B}_\infty(x, \epsilon)$. Given an instance-label pair (x, y), a classifier h and a differentiable loss function l, the PGD attack proceeds as follows: at the k^{th} iteration where the current perturbed instance is x^k, the next instance x^{k+1} is obtained through the following two steps:

- Update x^k by performing a sign gradient ascent with step size $\lambda \in \mathbb{R}_+$:

$$z = x^k + \lambda \text{sign} \left(\nabla_{x^k} \ell(h(x^k), y) \right) \tag{2.2}$$

  where the sign function is applied element-wisely on the gradients.

- Project z onto $\mathbb{B}_\infty(x, \epsilon)$ to obtain the next perturbed instance x^{k+1}:

$$x^{k+1} = \Pi_{\mathbb{B}_\infty(x,\epsilon)}(z) \tag{2.3}$$

The projection operation in (2.3) has a closed form

$$\Pi_{\mathbb{B}_\infty(x,\epsilon)}(z) = \min \left(\max(z, x - \epsilon), x + \epsilon \right)$$

where $\min(\cdot)$ and $\max(\cdot)$ are applied element-wisely to a vector.

Remark 2.4. Different from the standard PGD method (2.1), the PGD attack updates the perturbed instance using sign gradients, as shown in (2.2). In chapter 5, we will show that the peculiar choice of sign function in the PGD attack appears to impact adversarial training both in terms of (inner) optimization and in terms of generalization. This aspect, which has been largely overlooked in prior research, will be explored in detail in this thesis.

Remark 2.5. In real-world scenarios, the gradients of a model are often inaccessible, and typically only the model's classification outputs can be obtained through queries. Adversarial attacks that rely solely on querying the model's outputs are referred to as *black-box attacks*. These attacks are designed without requiring internal knowledge of the model, such as its architecture, parameters, or gradients.

Conversely, adversarial attacks that have full access to all aspects of the model—including gradients, parameters, and even the training data—are termed *white-box attacks*. These attacks exploit detailed information about the model to craft more precise and effective adversarial examples. PGD attack is a type of white-box attack.

2.2 Adversarial Training

To enhance the robustness of DNNs against adversarial perturbations, a substantial body of research [20, 61, 76, 82, 83, 101, 115] has proposed revised training algorithms. These approaches, commonly referred to as adversarial training (AT), include dominant methods like Projected Gradient Descent (PGD)-based AT [61], which trains models by approximately solving a min-max optimization problem.

In this section, we briefly introduce the adversarial training framework proposed in [61], a foundational work that has significantly influenced subsequent research in this area.

Let $\mathcal{D}$ denote the underlying data distribution over $\mathcal{X} \times \mathcal{Y}$. In the standard classification problem, our ultimate learning target is to find a classifier h in some model class $\mathcal{H}$ that achieves the lowest standard error rate on data sample from $\mathcal{D}$

$$h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathbb{I}\{h(x) \neq y)\}\right]$$

However, the optimal classifier h^* which minimizes the standard error rate, is not necessarily robust to adversarial perturbations. To obtain a classifier that is not only "accurate" on the clean data but also robust to adversarial perturbations, the ultimate learning target then becomes:

$$h^*_{\text{rob}} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\hat{x} \in T(x)} \mathbb{I}\{h(\hat{x}) \neq y)\}\right]$$

With only finite training samples $S := \{(x_i, y_i)\}_{i=1}^n$ drawn from $\mathcal{D}^n$, the training objective of AT in [61] then naturally arises, considering to solve

$$w^* = \arg\min_w \frac{1}{n} \sum_{i=1}^{n} \max_{\hat{x}_i \in T(x_i)} \ell\left(h_w(\hat{x}_i), y_i\right)$$

where parameterized models h_w with parameter w (e.g., DNNs) are considered and the 0-1 loss is replaced by a smooth surrogate loss function l (e.g., the cross-entropy loss).

One of the main challenges for solving this min-max problem for DNNs is that obtaining the closed form of the inner maximization $\max_{\hat{x}_i \in T(x_i)} \ell\left(h_w(\hat{x}_i), y_i\right)$ is intractable. In practice, this maximization is approximated by evaluating the loss value on the adversarial examples $\ell(h_w(x_i^{\text{adv}}), y_i)$ where the adversarial examples x_i^{adv} are generated by some adversarial attack algorithm such as the PGD attack [61]. The min-max training objective is then approximately solved by iterating between generating adversarial examples x_i^{adv} according to current model parameter w and then updating w through stochastic gradient descent to minimize the loss on the adversarial examples. We will provide a more detailed introduction to this procedure in the following chapters.

2.3 Standard Generalization and Robust Generalization

In standard classification problems, we consider the standard empirical risk and population risk as the main performance measurements for a given model h_w.

Definition 2.2 (Standard risks)**.** *Let $\mathcal{D}$ denote a distribution over $\mathcal{X}$ and $\mathcal{Y}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denote a loss function. Let $S = \{(x_i, y_i)\}_{i=1}^n$ be drawn from $\mathcal{D}^n$. We denote the standard population risk $R_{\mathcal{D}}(w)$ and the standard empirical risk $R_S(w)$ of h_w measured by ℓ as*

$$R_{\mathcal{D}}(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(h_w(x), y)\right] \quad \text{and} \quad R_S(w) := \frac{1}{n}\sum_{i=1}^{n} \ell(h_w(x_i), y_i) \tag{2.4}$$

Remark 2.6. We will call the gap $R_{\mathcal{D}}(w) - R_S(w)$ as the *standard generalization gap*, which quantifies the generalization performance of the classifier h_w on the clean data.

We now introduce the robust risks, which are the central performance metrics in classification problems with adversarial perturbations.

Definition 2.3 (Adversarial risks)**.** *Let $\mathcal{D}$ denote a distribution over $\mathcal{X}$ and $\mathcal{Y}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denote a loss function. Let $S = \{(x_i, y_i)\}_{i=1}^n$ be drawn from $\mathcal{D}^n$. Given a perturbation set $T(x)$, we denote the adversarial population risk $R_{\mathcal{D}}^{\text{adv}}(w)$ and the adversarial empirical risk $R_S^{\text{adv}}(w)$ of h_w measured by ℓ as*

$$R_{\mathcal{D}}^{\text{adv}}(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\hat{x}\in T(x)} \ell(h_w(\hat{x}), y)\right] \quad \text{and} \quad R_S^{\text{adv}}(w) := \frac{1}{n}\sum_{i=1}^{n} \max_{\hat{x}_i\in T(x_i)} \ell(h_w(\hat{x}_i), y_i) \tag{2.5}$$

Remark 2.7. The perturbation set $T(x)$ usually includes the original instance x itself (i.e., $x \in T(x)$). Consequently, we have $R_{\mathcal{D}}^{\text{adv}}(w) \geq R_{\mathcal{D}}(w)$ and $R_S^{\text{adv}}(w) \geq R_S(w)$, indicating that models achieving small robust risks also achieve small standard risks simultaneously.

To distinguish this from the standard generalization gap, we call the gap $R_{\mathcal{D}}^{\text{adv}}(w) - R_S^{\text{adv}}(w)$ as the *robust generalization gap*.

In chapter 5, we will demonstrate that both standard risks and robust risks can be unified within a broader framework by defining generalization in terms of perturbation induced losses.

Chapter 3

Robust generalization for linear classifiers

In this chapter, we analyze the adversarial robustness of linear classifiers trained by minimizing the support vector machine (SVM) objective. We show that, with a specific choice of hyperparameters, the SVM loss serves as an upper bound on the adversarial loss. This implies that minimizing the SVM loss yields a linear classifier with a small adversarial loss on the training set. Additionally, we investigate the robust generalization performance of SVM-trained linear models by deriving a generalization upper bound using Rademacher complexity analysis.

3.1 Additional Preliminaries

We first introduce the notion of Rademacher complexity and the generalization bound derived based on this quantity.

Definition 3.1 (Rademacher complexity [5])**.** *Let $\mathcal{F}$ be a family of functions maping from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$. Let σ denote a random variable following the Rademacher distribution, i.e., $\Pr(\sigma = 1) = \Pr(\sigma = -1) = 0.5$. Let $\Sigma := (\sigma_1, \cdots, \sigma_n)$ denote n independent Rademacher random variables. Given a set of n samples $S := \{(x_i, y_i)\}_{i=1}^n$, the empirical Rademacher complexity of $\mathcal{F}$ is defined as*

$$\mathrm{Rad}_n(\mathcal{F}; S) := \mathbb{E}_\Sigma \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i f(x_i, y_i) \right] \tag{3.1}$$

Let P denote the distribution according to which the samples $S := \{(x_i, y_i)\}_{i=1}^n$ are drawn. Then the Rademacher complexity of $\mathcal{F}$ is defined as

$$\mathrm{Rad}_n(\mathcal{F}) := \mathbb{E}_{S\sim P}[\mathrm{Rad}_n(\mathcal{F}; S)] \tag{3.2}$$

Remark 3.1. Rademacher complexity characterize the richness of a family of functions $\mathcal{F}$ by measuring the best correlation between its members and the random noises Σ, reflecting the ability of $\mathcal{F}$ to fit random noises.

Intuitively, smaller Rademacher complexity implies that $\mathcal{F}$ has lower capacity to fit arbitrary patterns, making it less likely to overfit when used to model a given set of training samples. When considering a class of functions $\mathcal{F}$ that are bounded, e.g., $f : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ for any $f \in \mathcal{F}$, this intuition is rigorously formalized by the following result:

$$\mathbb{E}_{S\sim\mathcal{D}^n}\left[\sup_{f\in\mathcal{F}}\left(\mathbb{E}_{(x,y)\sim\mathcal{D}} f(x, y) - \frac{1}{n}\sum_{i=1}^n f(x_i, y_i)\right)\right] \leq 2\mathrm{Rad}_n(\mathcal{F}) \tag{3.3}$$

This upper bound shows that if the Rademacher complexity $\mathrm{Rad}_n(\mathcal{F})$ is small, then every $f \in \mathcal{F}$ will have a small (expected) generalization gap. The proof of (3.3) can be found in [62] chapter 3.1 and [97] chapter 4.2. Building upon (3.3), we have the following well-known generalization upper bound.

Theorem 3.2 (generalization upper bound [5,62])**.** *Let $\mathcal{F}$ be a family of functions maping from $\mathcal{X} \times \mathcal{Y}$ to $[0, C]$. Let $S := \{(x_i, y_i)\}_{i=1}^n$ be drawn i.i.d from $\mathcal{D}$. For any $\eta \in (0, 1)$, with probability at least $1 - \eta$ over drawing S from $\mathcal{D}^n$, every $f \in \mathcal{F}$ satisfy*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}} f(x, y) \leq \frac{1}{n}\sum_{i=1}^n f(x_i, y_i) + 2C\mathrm{Rad}_n(\mathcal{F}; S) + 3C\sqrt{\frac{\log(2/\eta)}{2n}} \tag{3.4}$$

3.2 Theoretical Results

We now consider a binary classification setting with input space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} := \{-1, +1\}$. Let $\mathcal{D}$ denote a distribution on $\mathcal{X} \times \mathcal{Y}$. A linear model with parameters $w \in \mathbb{R}^d$ is denoted by $h_w(x) := w^T x$ and the corresponding linear classifier is defined as $g_w(x) := \mathrm{sgn}(h_w(x))$, where $\mathrm{sgn}(\cdot)$ denotes the sign function.

We consider the hypothesis class $\mathcal{H} := \{h_w : \|w\|_q \leq B\}$, where models are constrained by the $q-$norm of w for some $q \in (0, \infty]$. For any $h_w \in \mathcal{H}$, we analyze its adversarial risks under perturbations from the $p-$norm ball $\mathbb{B}_p(x, \epsilon)$, defined as

$$R_{\mathcal{D}}^{\text{adv}}(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\hat{x}\in\mathbb{B}_p(x,\epsilon)} \mathbb{I}(yh_w(\hat{x}) < 0)\right] \tag{3.5}$$

and

$$R_S^{\text{adv}}(w) := \frac{1}{n}\sum_{i=1}^{n} \max_{\hat{x}_i\in\mathbb{B}_p(x_i,\epsilon)} \mathbb{I}(y_i h_w(\hat{x}_i) < 0) \tag{3.6}$$

where $S := \{(x_i, y_i)\}_{i=1}^n$ is a set of n i.i.d training samples from $\mathcal{D}$ and $\mathbb{I}(\cdot)$ denotes the indicator function. Since $\mathbb{I}(yh_w(x) < 0) = \mathbb{I}(g_w(x) \neq y)$, the adversarial risks in the above equations quantify the classification error rates of h_w (or equivalently g_w) under adversarial perturbations.

Connections to SVM. The support vector machine (SVM) algorithm seeks a classifier that not only fits the training data but also maximizes the geometric margin between classes. Intuitively, a larger margin enhances robustness, as it increases the separation between training samples and the decision boundary. In the following, we show that, for a specific choice of hyperparameters, the SVM objective serves as an upper bound on the adversarial risk. This implies that minimizing the SVM objective inherently promotes robustness.

Given a training set S, the SVM optimization problem is defined as

$$\begin{aligned} &\min_{w,\Xi} \alpha\sum_{i=1}^{n}\xi_i + \beta\|w\|_q \\ &\text{s.t.} \quad \xi_i \geq 1 - y_i h_w(x_i), \quad \xi_i \geq 0 \end{aligned} \tag{3.7}$$

where $\Xi := (\xi_i)_{i=1}^n$ and $\alpha, \beta \in \mathbb{R}_+$ are hyper-parameters, balancing the trade-off between minimizing the slack variables ξ_i and the regularization term $\|w\|_q$. Solving (3.7) is equivalent to minimizing the following loss function

$$L_{\alpha,\beta}(w; S) := \alpha\sum_{i=1}^{n}\phi(y_i h_w(x_i)) + \beta\|w\|_q \tag{3.8}$$

where $\phi : \mathbb{R} \to [0, +\infty)$ is the hinge loss function, defined as

$$\phi(z) := \max(0, 1 - z) \tag{3.9}$$

Now, consider an adversarial perturbation set $\mathbb{B}_p(x, \epsilon)$ where the norm $\|\cdot\|_p$ is the dual norm of $\|\cdot\|_q$, i.e., $\frac{1}{p} + \frac{1}{q} = 1$. Setting $\alpha = 1/n$ and $\beta = \epsilon$ in $L_{\alpha,\beta}(w; S)$, we obtain the following inequality:

$$R_S^{\text{adv}}(w) \leq L_{\frac{1}{n},\epsilon}(w; S) \tag{3.10}$$

This result shows that minimizing the SVM objective simultaneously reduces the empirical adversarial risk, leading to a classifier with inherent robustness on the training set.

We now present the proof of (3.10):

Proof. For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and any w, we have that

$$\begin{aligned}
&\max_{\hat{x} \in \mathbb{B}_p(x,\epsilon)} \mathbb{I}(y h_w(\hat{x}) < 0) && (3.11)\\
\leq &\max_{\hat{x} \in \mathbb{B}_p(x,\epsilon)} \phi(y h_w(\hat{x})) && (3.12)\\
= &\phi\left(\min_{\hat{x} \in \mathbb{B}_p(x,\epsilon)} y h_w(\hat{x})\right) && (3.13)\\
= &\max\left(0, 1 - \min_{\|\delta\|_p \leq \epsilon} y w^T (x + \delta)\right) && (3.14)\\
= &\max\left(0, 1 - y w^T x + \epsilon \|w\|_q\right) && (3.15)\\
\leq &\max\left(0, 1 - y w^T x\right) + \epsilon \|w\|_q && (3.16)
\end{aligned}$$

Averaging over S dervies the inequality

$$R_S^{\text{adv}}(w) \leq L_{\frac{1}{n},\epsilon}(w; S)$$

□

Building upon this connection, we now analyze the robust generalization performance of linear classifiers trained via the SVM algorithm using Rademacher complexity. For brevity, we denote $L_{1/n,\epsilon}$ as L_ϵ for the remainder of this discussion.

Let $\tilde{\phi} : \mathbb{R} \to [0, 1]$ denote the Ramp loss function, written as

$$\tilde{\phi}(z) := \min(1, \phi(z)) \tag{3.17}$$

It follows that, for any x, y and h_w, we have the inequality

$$\mathbb{I}(y h_w(x) < 0) \leq \tilde{\phi}(y h_w(x)) \leq \phi(y h_w(x)) \tag{3.18}$$

Define the function class

$$\tilde{\mathcal{H}} := \{(x, y) \to \tilde{\phi}(yh_w(x)) + \epsilon\|w\|_q : \|w\|_q \leq B\} \tag{3.19}$$

Note that each functions in $\tilde{\mathcal{H}}$ is bounded within $[0, \epsilon B]$.

Based on Theorem 3.2 and the inequalities (3.18), we have the following result:

Proposition 3.3. *Consider the adversarial population risk $R^{\text{adv}}_{\mathcal{D}}(w)$ defined in (3.5). Let $S := \{(x_i, y_i)\}_{i=1}^n$ be a set of samples drawn i.i.d from $\mathcal{D}$. For any $\eta \in (0, 1)$, with probability $1 - \eta$ over sampling S from $\mathcal{D}^n$, every $h_w \in \mathcal{H}$ satisfies*

$$R^{\text{adv}}_{\mathcal{D}}(w) \leq L_\epsilon(w; S) + 2(\epsilon B + 1)\text{Rad}_n(\tilde{\mathcal{H}}; S) + 3(\epsilon B + 1)\sqrt{\frac{\log(2/\eta)}{2n}} \tag{3.20}$$

where $\tilde{\mathcal{H}}$ is defined in (3.19).

The proof of this proposition is deferred to Appendix A. This result suggests that when the empirical Rademacher complexity of $\tilde{\mathcal{H}}$ is small, minimizing the SVM loss $L_\epsilon(w; S)$ yields a linear classifier with a low adversarial population risk with high probability.

We now present an upper bound for the empirical Rademacher complexity $\text{Rad}_n(\tilde{\mathcal{H}}; S)$.

Theorem 3.4. *Let $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leq R\}$ be the input space. Then the empirical Rademacher complexity of $\tilde{\mathcal{H}}$ satisfies the upper bound*

$$\text{Rad}_n(\tilde{\mathcal{H}}; S) \leq \max\left(1, d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{BR}{\sqrt{n}} + \frac{\epsilon B}{2\sqrt{n}} \tag{3.21}$$

We defer the proof of the theorem to Appendix A. It is worth noting that for the perturbation set $\mathbb{B}_p(x, \epsilon)$ with $p \geq 2$, this upper bound does not directly depend on the dimension of the input data.

3.3 Discussion

The robust generalization of linear classifiers has been extensively studied through the lens of Rademacher complexity, with prior works [2,3,45,110] providing upper bounds analogous to Theorem 3.4. However, when it comes to deep neural networks (DNNs), Rademacher complexity has proven to be an overly conservative measure, often leading to vacuous, less meaningful generalization bounds.

A striking example of this limitation is demonstrated by [114], where experiments show that DNNs can memorize random labels, exhibiting extremely high Rademacher complexity. Classical generalization bounds (e.g., Theorem 3.2) suggest that models with high Rademacher complexity are expected to overfit. Yet, deep networks often generalize well in practice despite their high capacity to fit random noise, indicating that Rademacher complexity significantly overestimates the effective complexity relevant to generalization.

Furthermore, analyses based on Rademacher complexity fail to capture the essential role of training dynamics in shaping generalization. These analyses typically assess the worst-case generalization gap across an entire hypothesis class, which may not be representative of the solutions found by specific learning algorithms. In particular, deep neural networks are trained using iterative optimization methods such as stochastic gradient descent (SGD), which introduces implicit biases that favor solutions with better generalization properties. A substantial body of work [59, 67, 68, 87] has shown that the training dynamics of SGD drives models toward solutions that generalize well, even though the hypothesis class itself contains many poorly generalizing solutions.

These limitations motivate a deeper investigation into the robust generalization of deep learning models beyond traditional Rademacher complexity analysis. In particular, training dynamics play a crucial role in shaping robust generalization, especially in adversarial training (AT). In the next chapter, we explore how AT dynamics influences robust generalization.

Chapter 4

Towards Understanding the Dynamic of Adversarial Training

AT may be regarded as stochastic gradient descent (SGD) on an adversarially perturbed version of the training set at each iteration. Specifically, at each gradient descent iteration, each input instance in a training batch is first perturbed to maximize the training loss with respect to the current model parameter, and then gradient descent is performed to update the model parameter. The maximization of the training loss prior to gradient descent is constrained on a maximum allowable perturbation radius; in other words, this maximization is equivalent to an adversarial attack to the model with current parameter setting.

To study the generalization behavior of models learned by AT is challenging, arguably due to this complex dynamics of AT. In particular, this complexity arises from the convoluted interaction between the update of model parameter along AT iterations and the update of the adversarial perturbations in the inner maximization step. More concretely, when the model parameter gets updated, the adversarial perturbation is updated to one that attacks the updated model, and the updated adversarial perturbation in turn governs the next update of the model parameter. It is then conceivable that understanding the generalization behavior of AT requires a deep understanding of the interaction between the model updates and perturbation updates, even "untangling" the convoluted interaction along the training trajectory. This philosophy motivates the work in this chapter.

A key observation in this chapter is the recognition that in each AT iteration, the perturbation operator effectively induces a new data distribution and that the model update may be viewed as the standard training on data drawn from this induced distribution.

Since perturbation in each AT iteration has a small magnitude, the induced distribution is provably close to the original data distribution. However, a surprising finding in this work is that these induced distributions behave distinctively from the original distribution: as AT progresses, they may become increasingly more difficult to learn. The experiments supporting this finding were conducted as follows: for a check point of AT, we extract the perturbation operator and use it to perturb both the training set and test set; we then train a model from scratch on the perturbed training set, using standard training, until the (standard) training error is effectively zero; we then evaluate the learned model on the perturbed testing set to obtain its classification error. We call such an experiment as an "induced distribution experiment" or IDE. When conducting IDE on datasets such as CIFAR-10, we usually observe large testing errors, particularly when the check point is near the end of AT. In fact, on such datasets, the generalization gap for models learned from the induced distribution appears to progressively increase as AT proceeds.

To understand the deteriorating behavior of the induced distribution along AT, we derive a uniform-convergence upper bound of the generation gap for models learned on the induced distributions. The key quantity in the bound is a term we call "local dispersion" of the perturbation operator. Our bound suggests that only when the perturbation operator has small local dispersion, a good generalization guarantee can be obtained for models learned on the distribution induced by the operator. Through experiments, we show that local dispersion is indeed indicative to the generalization gap of models learned on the induced distribution and can be used to explain the deteriorating behavior of the induced distribution along the AT trajectories, as observed in our IDE experiments.

In summary, in this chapter we discover an interesting phenomenon in AT, namely, that the induced distributions by the perturbation operator in AT are progressively more difficult to learn. We prove a generalization bound as a theoretical explanation for this phenomenon and corroborate it with experimental validations. Our results shed new lights in understanding the complex AT dynamics and the interaction therein between model updates and perturbation updates. Although there have been previous works examining AT trajectories, very few actually zoom into the properties of the perturbation operator.

This chapter highlights the importance of investigations in this angle in paving ways towards understanding robust generalization. This importance is further manifested by our additional experimental observation presented at the end of the chapter, where we show that the deteriorating behavior of the induced distributions correlates with robust overfitting.

4.1 Adversarial Training and Induced Distributions

Notations and basic setup. Over any real vector space, we will use $\|\cdot\|_p$ to denote the p-norm and abbreviate the Euclidean norm (i.e., 2-norm) as $\|\cdot\|$. Recall that we use $\mathbb{B}_p(x,\epsilon) := \{t \in \mathbb{R}^d : \|t-x\|_p \leq \epsilon\}$ to denote a $p-$norm ball with radius ϵ and centered at x. In the remainder of this chapter, we consider the $\infty-$norm ball $\mathbb{B}_\infty(x,\epsilon)$ as the perturbation set.

We consider a classification setting with input space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} := \{1, 2, \cdots, K\}$. We use $\mathcal{D}$ to denote a distribution on $\mathcal{X} \times \mathcal{Y}$ and denote $\mathcal{D}^{\mathcal{X}}$ as the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$. Let Θ be the parameter space of a parameterized model of interest and $h_\phi : \mathcal{X} \to \mathcal{Y}$ denotes a classifier with parameter $\phi \in \Theta$. We again use $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ to represent a loss function. Standard choices of l for classification tasks are 0-1 loss and cross-entropy loss. The classification error of h_ϕ over a sample (x, y) is then measured by $f_\phi(x, y) := \ell(h_\phi(x), y)$.

We rewrite the *standard population risk* and the *standard empirical risk* in (2.4) respectively as

$$R_{\mathcal{D}}(\phi) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f_\phi(x,y)\right] \quad \text{and} \quad R_S(\phi) := \frac{1}{n}\sum_{i=1}^{n} f_\phi(x_i, y_i) \tag{4.1}$$

for a set of n samples $S := \{(x_i, y_i)\}_{i=1}^n$ drawn i.i.d. from $\mathcal{D}$.

The standard generalization performance of the model f_ϕ is then measured by the *standard generalization gap*:

$$\mathrm{GG}_n(\phi, S; \mathcal{D}) := |R_{\mathcal{D}}(\phi) - R_S(\phi)| \tag{4.2}$$

Adversarial perturbations Given any instance-label pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a target model f_ϕ paramterized by ϕ, we define the *ϵ-adversarial perturbation of x with respect to f_ϕ* as

$$\mathcal{Q}_\phi(x, y) := \arg\max_{v \in \mathbb{B}_\infty(x,\epsilon)} f_\phi(v, y) \tag{4.3}$$

Clearly the operator $\mathcal{Q}_\phi$ also depends on the allowable perturbation magnitude ϵ, but we suppress such dependency in our notations throughout this chapter for simplicity.

Adversarial risks Given a data distribution $\mathcal{D}$ and its n i.i.d samples S, the *adversarial risks* $R_{\mathcal{D}}^{\text{adv}}(\phi)$ and the *adversarial empirical risk* $R_S^{\text{adv}}(\phi)$ of a model f_ϕ can be re-written, respectively, as

$$R_{\mathcal{D}}^{\text{adv}}(\phi) := \mathbb{E}_{(x,y)\sim\mathcal{D}} f_\phi(\mathcal{Q}_\phi(x,y),y) \tag{4.4}$$

and

$$R_S^{\text{adv}}(\phi) := \frac{1}{n}\sum_{i=1}^{n} f_\phi(\mathcal{Q}_\phi(x_i,y_i),y_i) \tag{4.5}$$

Adversarial training Given a training set S, at the t^{th} iteration of adversarial training (AT), where the model parameter is ϕ_t, the model parameter is updated, with learning rate η, by

$$\phi_{t+1} = \phi_t - \eta\nabla_{\phi_t}\left[\frac{1}{n}\sum_{i=1}^{n} f_{\phi_t}\left(\mathcal{Q}_{\phi_t}(x_i,y_i),y_i\right)\right] \tag{4.6}$$

Notably, the update equation (4.6) of AT results in a complex dynamics, namely, the update of ϕ causes the update of the perturbation operator $\mathcal{Q}_\phi$, and the update of $\mathcal{Q}_\phi$ in turn influences the next update of ϕ. This complex interaction between the model parameter and the perturbation operator makes analyzing AT trajectories very difficult.

One key perspective of this work is recognizing that at training iteration t, the perturbation operator $\mathcal{Q}_{\phi_t}$ essentially induces a different distribution and that the AT step in (4.6) may be seen as a one-step gradient descent on the standard empirical risk of training data drawn from this induced distribution. We next make this precise.

Perturbation induced distribution Let (X,Y) be drawn from $\mathcal{D}$. Given an adversarial perturbation $\mathcal{Q}_\phi$, the *perturbation induced distribution* (or simply induced distribution) is defined as the joint distribution of $(\mathcal{Q}_\phi(X,Y),Y)$ and is denoted by $\tilde{\mathcal{D}}_\phi$. For a given training set $S = \{(x_i,y_i)\}_{i=1}^n$, denote $\tilde{S}_\phi := \{(v_i,y_i)\}_{i=1}^n$, where $v_i := \mathcal{Q}_\phi(x_i,y_i)$. It is clear that the samples $\tilde{S}_\phi$ are drawn from the induced distribution $\tilde{\mathcal{D}}_\phi$.

Since each perturbed instances $\mathcal{Q}_\phi(x,y)$ lies within a small neighborhood of x (i.e., $\|\mathcal{Q}_\phi(x,y)-x\|_\infty \le \epsilon$), it follows immediately that for any ϕ, the Wasserstein p-distance (denoted by $\mathcal{W}_p(\cdot,\cdot)$) between $\mathcal{D}$ and $\tilde{\mathcal{D}}_\phi$ satisfies

$$\mathcal{W}_p(\tilde{\mathcal{D}}_\phi,\mathcal{D}) \le \epsilon \tag{4.7}$$

for any $p\in[1,+\infty]$. Here the metric, say d, on $\mathcal{X}\times\mathcal{Y}$ by which the Wasserstein distance is defined, is

$$d((x,y),(x',y')) := \|x-x'\|_\infty + d_{\mathcal{Y}}(y,y')$$

where $d_{\mathcal{Y}}$ is an arbitrary metric on $\mathcal{Y}$.

Remark 4.1. Notably, in the context of adversarial training, the maximum perturbation magnitude ϵ is usually small. Then by equation (4.7), the distribution $\tilde{\mathcal{D}}_\phi$ induced by the perturbation operator $\mathcal{Q}_\phi$ during AT is very close to the original data distribution $\mathcal{D}$. However, a surprising observation in this work is that models trained (via standard training) on $\mathcal{D}$ and on $\tilde{\mathcal{D}}_\phi$ may have very different behaviors.

Remark 4.2. It is also worth noting that $R^{\text{adv}}_{\mathcal{D}}(\phi) = R_{\tilde{\mathcal{D}}_\phi}(\phi)$ and $R^{\text{adv}}_{S}(\phi) = R_{\tilde{S}_\phi}(\phi)$ —the adversarial risks of ϕ can be treated as the standard population (resp. empirical) risk of ϕ measured on the induced distribution (resp. the samples drawn from the induced distribution) generated by ϕ.

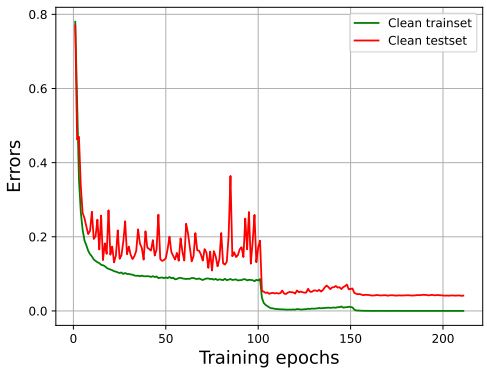
Following the definition of generalization gap in (4.2), the notations $\text{GG}_n(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ and $\text{GG}_n(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ are both well defined, where the former is the *robust generalization gap* of an arbitrary model f_ϕ and the latter is the standard generalization gap of an arbitrary model f_θ measured with respect to a given induced distribution $\tilde{\mathcal{D}}_\phi$ and its samples $\tilde{S}_\phi$.

4.2 Learning on the induced distributions

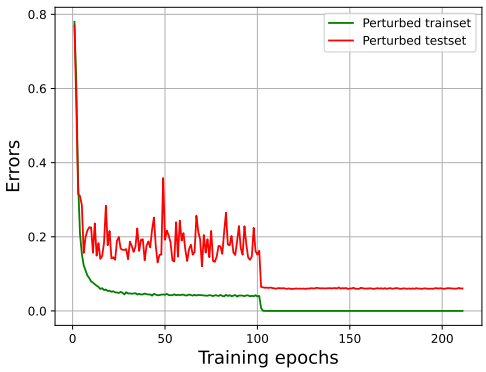
In this section, we experimentally study the problem of learning on the induced distribution $\tilde{\mathcal{D}}_\phi$, where ϕ is the parameter of a model being trained during AT.

Induced distribution experiment Let S and T be the training set and testing set of a classification task. We perform AT for a neural network model using S. Let $\text{AT}(t)$ denote that model's parameter obtained by performing AT for t epochs. For some choice of t, we obtain model parameter $\phi = \text{AT}(t)$. We then perturb S and T using $\mathcal{Q}_\phi$, and obtain the perturbed training and testing datasets $\tilde{S}_\phi$ and $\tilde{T}_\phi$ respectively. A new model (with the same architecture) is then trained from scratch (namely, starting from random initialization of its parameters) on $\tilde{S}_\phi$ **using standard training** and denote the learned model parameter by θ. This model θ is evaluated on $\tilde{T}_\phi$. For the ease of reference we call such an experiment the "induced distribution experiment" (IDE).

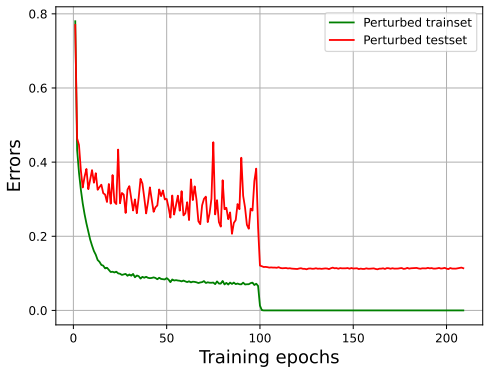
In our IDE experiments, $\mathcal{Q}_\phi$ is taken as the Projected Gradient Descend (PGD) attack [61], which is used both for AT and for generating the perturbed datasets. Other details of the experiments are given below.



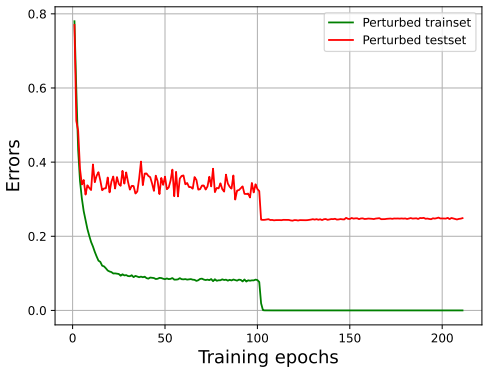
(a) Clean dataset



(b) ϕ=AT(0)



(c) ϕ=AT(80)



(d) ϕ=AT(200)

Figure 4.1: Learning curves of standard training on the clean CIFAR-10 dataset and IDEs w.r.t various ϕ. In each training, the learning rate is decayed at the 100^{th} epoch.

Datasets The experiments are conducted on CIFAR10 and CIFAR100 [47]. We also conduct experiments on a "scaled-down" version of the ImageNet dataset [80], which we call Reduced ImageNet, drawing inspiration from a similar approach in [95] for reduced training complexity. Reduced ImageNet aggregates several subsets of the original ImageNet and comprises 10 classes, each containing 5000 training samples and approximately 1000 testing samples per class. More details concerning this dataset are given in Appendix B.1.

Settings for AT and PGD On CIFAR-10 and Reduced ImageNet we perform AT to train the pre-activation ResNet (PRN) model [38] with 18 and 50 layers respectively. On CIFAR-100 we train the Wide ResNet (WRN) model with 34 layers [113]. We use 5-step PGD with $\epsilon = 4/255$ for Reduced ImageNet and 10-step PGD with $\epsilon = 8/255$ for CIFAR-10 and CIFAR-100 according to [78]. We set $\lambda = 2/255$ on CIFAR10 and CIFAR100, $\lambda = 0.9/255$ on Reduced ImageNet. More details concerning the hyper-parameter settings are given in Appendix B.1.

Experimental results Let $\phi = \text{AT}(0)$ denote a randomly initialized model. Figure 4.1(b)-(d) presents the learning curves of IDEs conducted on the CIFAR-10 datasets for ϕ obtained after AT for different numbers of epochs, while Figure 4.1(a) shows the learning curves of standard training on the clean CIFAR-10 dataset for comparison. The green and red curves respectively represent the training and testing error recorded along the training process. In all cases, the model is trained to achieve zero training error. However, the testing error varies significantly in different IDEs. On the clean dataset, the model attains a testing error as low as 4.13%; A similar performance is observed on the IDE with $\phi = \text{AT}(0)$, where the testing error reaches around 6.06%. In contrast, for $\phi = \text{AT}(80)$, the learned model shows a reduced generalization performance, with the testing error increasing to 11.38%. A more significant rise on the testing error occurs when a model is trained on the perturbed dataset generated by $\phi = \text{AT}(200)$, where the testing error increases to 24.89%. Similar results are also observed on CIFAR-100 and Reduced ImageNet (see Appendix B.3 Figure B.1 and B.2).

For IDE with $\phi = \text{AT}(200)$, a large generalization gap —the gap between the red and green curves —emerges in the early phase of the training (around the 20^{th} training epoch). After the drop of learning rate (at the 100^{th} training epoch), the training error quickly reduces to zero, yet the generalization gap remains nearly unchanged, resulting in a high final testing error. This is in contrast to the learning behavior observed on the clean dataset and the IDE with $\phi = \text{AT}(0)$, where a small generalization gap is established at the early phase of training and is consistently preserved along the training.

These experiments reveal a rather surprising phenomenon: despite $\tilde{\mathcal{D}}_\phi$ being very close to $\mathcal{D}$, the model's learning performance on the induced distribution $\tilde{\mathcal{D}}_\phi$ can be significantly different from that on $\mathcal{D}$. In particular, as AT proceeds, the induced distribution $\tilde{\mathcal{D}}_\phi$ may deteriorate, in the sense that it becomes increasingly more difficult to generalize, as signified by the increasing generalization gap.

Additional related works Existing studies have uncovered intriguing properties of adversarial examples, such as their transferability across different models [34, 72, 93] and their distinct geometric characteristics compared to clean examples [31, 60]. The work in [41] reveals that adversarial examples generated w.r.t a model trained via standard training may still contain useful features. Specifically, they demonstrate that a classifier trained on mislabeled adversarial examples can achieve remarkable generalization performance on unseen clean data. Theoretical explanations for this finding are then provided in [50, 51]. Additionally, the work in [116] presents another intriguing finding that adversarial perturbations for two-layer neural networks with random weights are linearly separable, suggesting structural properties of adversarial perturbations exist.

Unlike [41] and [116], who focus on adversarial examples for models trained via standard training or with random weights, our work explores adversarial examples along AT trajectories, providing new insights into how features of adversarial examples evolve throughout the training process.

4.3 Theoretical analysis

In this section, we provide a theoretical analysis to explain the deteriorating learning behavior of the induced distribution along AT. Specifically, we derive an upper bound for the "worst-case" generalization gap $\sup_{\theta\in\Theta} \mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$.

Assumption 4.3 (Anchored data model)**.** We assume that underlying the data distribution $\mathcal{D}$, there is a latent distribution, or "anchor distribution", $\mathcal{D}_*$ on $\mathcal{X}\times\mathcal{Y}$. $\mathcal{D}_*$ is specified by its marginal $\mathcal{D}_*^{\mathcal{X}}$ on $\mathcal{X}$ and a classifier $h^*:\mathcal{X}\to\mathcal{Y}$ (which assigns every x drawn from $\mathcal{D}_*^{\mathcal{X}}$ a label in $\mathcal{Y}$). The data distribution $\mathcal{D}$ of interest is a "smoothed" version of $\mathcal{D}_*$ as follows: Draw an "anchor" variable T from $\mathcal{D}_*^{\mathcal{X}}$. Then draw a noise ρ independent of T from a distribution π (on $\mathbb{R}^d$) with zero mean and a finite variance in each dimension (recall that $\mathcal{X}$ is a subset of $\mathbb{R}^d$) —we assume the variance in each dimension is small. The distribution of $(T+\rho, h^*(T))$ is then the distribution $\mathcal{D}$.

Remark 4.4. In this anchored data model, the true input variable X is treated as a noise-perturbed version of an anchor variable $T \sim \mathcal{D}_*^{\mathcal{X}}$. Such an assumption is widely used in various machine learning contexts, for example, in the VAE model [46] where the reconstruction loss adopts the square error loss . On the other hand, the assumption that $X = T + \rho$ share the same label as T is sensible, since one expects that within small neighborhood of T, the class label remains unchanged.

Given a model class $\mathcal{F} := \{f_\theta : \theta \in \Theta\}$, we now study its generalization performance w.r.t the induced distributions. Specifically, we will derive an upper bound for the generalization gap $\mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ for all $\theta \in \Theta$. As it turns out, a key quantity governing the upper bound is a local property of the perturbation map $\mathcal{Q}_\phi$ that induces $\tilde{\mathcal{D}}_\phi$.

Definition 4.1 (Local dispersion)**.** *For any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we define the local dispersion $\tilde{\gamma}_\phi(x, y)$ of the perturbation mapping $\mathcal{Q}_\phi$ at (x, y) as*

$$\tilde{\gamma}_\phi(x, y) := \mathbb{E}_{\rho,\rho'} \|\mathcal{Q}_\phi(x + \rho, y) - \mathcal{Q}_\phi(x + \rho', y)\|_2^2. \tag{4.8}$$

where ρ and ρ' are drawn independently from π.

Remark 4.5. We refer to this quantity as the *local dispersion* of $\mathcal{Q}_\phi$, as it measures how far apart the operator $\mathcal{Q}_\phi$ disperses two noise-perturbed versions of (x, y). In fact, one may verify that $\tilde{\gamma}_\phi(x, y)$ can be expressed as

$$\tilde{\gamma}_\phi(x, y) = 2 \cdot \mathrm{Trace}\left(\mathrm{COV}_\rho(\mathcal{Q}_\phi(x + \rho, y))\right) \tag{4.9}$$

where ρ is drawn from π and $\mathrm{COV}_\rho(\mathcal{Q}_\phi(x + \rho, y))$ denotes the covariance matrix. That is, $\tilde{\gamma}_\phi(x, y)$ also measures the how far $\mathcal{Q}_\phi$ spreads a randomly perturbed version of (x, y). We defer the proof of (4.9) to Appendix B.2.1.

One may argue intuitively that smaller local dispersion of $\mathcal{Q}_\phi$ may allow the model to generalize better when learning on the distribution $\tilde{\mathcal{D}}_\phi$: consider an instance (T, Y) drawn from the anchor distribution $\mathcal{D}_*$, and two observed data points $(T + \rho, Y)$ and $(T + \rho', Y)$ (with ρ and ρ' drawn independently from π). Suppose that $(T + \rho, Y)$ is included in the training set and $(T + \rho', Y)$ is included in the testing set. When the local dispersion is small, the perturbed version of the training point $(\mathcal{Q}_\phi(T + \rho, Y), Y)$ and that of the testing point $(\mathcal{Q}_\phi(T + \rho', Y), Y)$ (both of which are realizations from $\tilde{\mathcal{D}}_\phi$) are close, allowing the model's prediction on the latter to behave similarly as that on the former.

We now rigorously formalize this intuition, under the following assumptions.

- (Lipchitzness of f_θ over $\mathcal{X}$) For any $y \in \mathcal{Y}$ and any $\theta \in \Theta$, $|f_\theta(x, y) - f_\theta(x', y)| \leq \beta\|x - x'\|_2$ for $\forall x, x' \in \mathcal{X}$.
- (Boundedness) $\sup_{x,y \in \mathcal{X}\times\mathcal{Y}} |f_\theta(x, y)| = B < \infty$ for any $\theta \in \Theta$.

The generalization gap (4.2) then has the following uniform convergence result:

Lemma 4.1. *Consider the model class $\mathcal{F}$ where each $f_\theta \in \mathcal{F}$ satisfies the above boundedness condition. For any ϕ (or $\tilde{\mathcal{D}}_\phi$), with probability $1 - \tau$ over drawing $\tilde{S}_\phi$ from $\tilde{\mathcal{D}}_\phi$, we have*

$$\sup_{\theta\in\Theta} \mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi) \leq \mathbb{E}_{\tilde{S}_\phi \sim \tilde{\mathcal{D}}_\phi^m} \sup_{\theta\in\Theta} \mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi) + 2B\sqrt{\frac{\log \frac{1}{\tau}}{2m}} \tag{4.10}$$

The proof of the lemma is deferred to Appendix B.2.2. Building upon lemma 4.1, we now derive an upper bound for $\mathbb{E}_{\tilde{S}_\phi \sim \tilde{\mathcal{D}}_\phi^m} \sup_{\theta\in\Theta} \mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ where the local dispersion of $\mathcal{Q}_\phi$ plays a role.

Theorem 4.6. *Consider the model class $\mathcal{F}$ where each $f_\theta \in \mathcal{F}$ satisfies the above Lipchitzness and boundedness conditions. Consider the data distribution $\mathcal{D}$ which satisfies the assumptions 4.3. Let $\tilde{\mathcal{D}}_\phi$ denote the induced distribution of $\mathcal{D}$, generated by a perturbation $\mathcal{Q}_\phi$. We have*

$$\mathbb{E}_{\tilde{S}_\phi \sim \tilde{\mathcal{D}}_\phi^m} \sup_{\theta\in\Theta} \mathrm{GG}_m(\theta, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi) \leq \frac{2\beta}{\sqrt{m}} \sqrt{\mathbb{E}_{(x,y)\sim\mathcal{D}_*} \tilde{\gamma}_\phi(x, y)} + \frac{2(\beta\sqrt{d}\epsilon + B)}{\sqrt{m}} \tag{4.11}$$

We leave the proof of the Theorem in Appendix B.2.3. Combining (4.11) with (4.10) immediately gives an upper bound for the generalization gap (4.2) that applies for any $\theta \in \Theta$.

Remark 4.7. The derivation of Theorem 4.6 is based on a modification of the Rademacher complexity analysis. It worth noting that any direct application of Rademacher complexity to establish a learning bound requires certain restriction on the hypothesis class $\mathcal{F}$, thus suffering from a loss of generality.

The theorem suggests that the generalization gap of any f_θ w.r.t to the distribution $\tilde{\mathcal{D}}_\phi$ is affected by the expected local dispersion (ELD) $\mathbb{E}_{\mathcal{D}_*} \tilde{\gamma}_\phi(x, y)$ of $\mathcal{Q}_\phi$ and that a small generalization gap can be uniformly attained—for every $f_\theta \in \mathcal{F}$ —with high probability when ELD $\mathbb{E}_{\mathcal{D}_*} \tilde{\gamma}_\phi(x, y)$ is small.

An interpretation of this theorem is that the learning difficulty of the induced distribution $\tilde{\mathcal{D}}_\phi$ may be attributed to the ELD $\mathbb{E}_{\mathcal{D}_*}\tilde{\gamma}_\phi(x,y)$ of the perturbation operator $\mathcal{Q}_\phi$. But since the theorem only provides an upper bound, such an interpretation is only valid if the upper bound in the theorem is indicative of the true generalization gap. We next report experimental measurements to show this is indeed the case.

4.4 Experimental Validation

We conducted experiments to estimate the ELD of $\mathcal{Q}_\phi$ for $\phi = \mathrm{AT}(t)$ with various t values along the AT trajectory. Note that the expectation here is over the distribution $\mathcal{D}_*$, from which no samples are available. However, due to the relationship between $\mathcal{D}^{\mathcal{X}}$ and $\mathcal{D}_*^{\mathcal{X}}$, namely that $\mathcal{D}^{\mathcal{X}}$ is merely a slightly smoothed version of $\mathcal{D}_*^{\mathcal{X}}$ (since π has small variances), one expects that when we draw x from $\mathcal{D}^{\mathcal{X}}$, $\mathcal{D}^{\mathcal{X}}(x) \approx \mathcal{D}_*^{\mathcal{X}}(x)$ with high probability. As a consequence, $\mathbb{E}_{\mathcal{D}_*}\tilde{\gamma}_\phi(x,y) \approx \mathbb{E}_{\mathcal{D}}\tilde{\gamma}_\phi(x,y)$ with high probability. But the latter can be estimated using the i.i.d. samples from $\mathcal{D}$. This gives us the following estimation formula for ELD:

$$\mathbb{E}_{\mathcal{D}^*}\tilde{\gamma}_\phi(x,y) \approx \frac{1}{m}\sum_{i=1}^{m}\tilde{\gamma}_\phi(x_i,y_i),$$

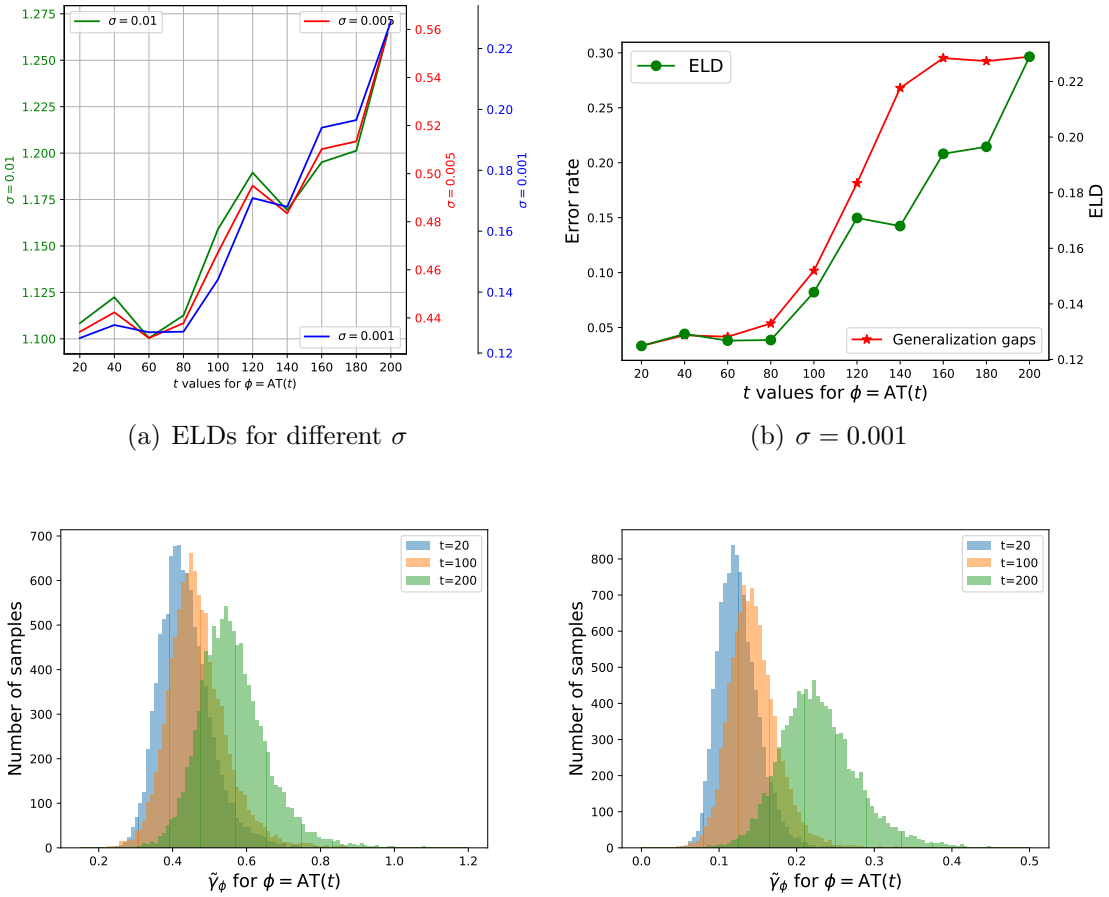
where $\{(x_i,y_i)\}_{i=1}^m$ are drawn from $\mathcal{D}$.

Estimating the local dispersion $\tilde{\gamma}_\phi(x_i,y_i)$ requires the knowledge of π, which is unfortunately unavailable to us. In our experiments, we take π as a spherical Gaussian, with variance in each dimension equal to σ^2. Various values of σ^2 are considered in our experiments.

The estimation of each $\tilde{\gamma}_\phi(x_i,y_i)$ is done by Monte-Carlo approximation via sampling 250 pairs of (ρ,ρ') from π. The expectation in (4.8) is then approximated using the sample mean.

Same trend of ELD estimated from different σ Figure 4.2(a) show that the estimated ELD values with $\phi = \mathrm{AT}(t)$ using $\sigma = 0.001, 0.005, 0.01$ respectively. In the figure, the three curves, each corresponding to a different σ value, have very similar trend. In fact, when adjusting the range of vertical axes, the three curves closely align with each other.

ELD as an indicator of generalization gap Figure 4.2 (b) presents the generalization gaps of the models learned on various $\tilde{\mathcal{D}}_\phi$ (red curve) and the estimated ELD values of the



(a) ELDs for different σ

(b) $\sigma = 0.001$

(c) $\sigma = 0.005$

(d) $\sigma = 0.001$

Figure 4.2: Local dispersion measured on the CIFAR-10 test set. (a) ELDs estimated using different σ values. For different choice of σ, the estimated ELDs fall within different ranges. To clearly compare the trends of ELD across different σ, we plot all estimations in the same graph and position their respective vertical axes on the sides of the figure. (b) ELD (green curve) of $\mathcal{Q}_\phi$ for different ϕ in comparison to the generalization gap achieved on $\tilde{\mathcal{D}}_\phi$. (c) and (d): histograms of $\tilde{\gamma}_\phi(x, y)$ for three distinct ϕ.

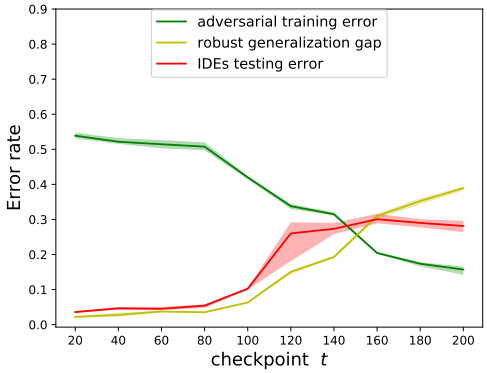
corresponding $\mathcal{Q}_\phi$ (green curve). In the experiments, we set $\sigma = 0.01$ for ELD estimation. In each IDE, the model is trained to achieve zero training error, hence the generalization gaps in Figure 4.2 (b) correspond directly to the testing errors of the learned models. As shown in the figure, when the ELD of $\mathcal{Q}_\phi$ is small, the model learned on the corresponding $\tilde{\mathcal{D}}_\phi$ tends to achieve a smaller generalization gap. This empirical observation aligns with the theoretical findings in Theorem 4.6. The positive correlation between the red and green curve in 4.2(b) suggests that the local dispersion of the perturbation operator significantly affects the generalization performance of the models learned on the induced distribution. This also validates the usefulness of Theorem 4.6, corroborating ELD as an indicator of the generalization gap for the induced distributions.

Increasing dispersiveness along AT Since in our experiments ϕ is obtained at different AT epochs, the upward trend in the green curve of Figure 4.2(b) and that of all the three curves in 4.2(a) suggest that performing AT for more iterations tends to make the perturbation operator $\mathcal{Q}_\phi$ increasingly dispersive. To further illustrate this trend, Figure 4.2 (c) and (d) respectively plot the histograms of $\tilde{\gamma}_\phi(x, y)$ for $\phi = \text{AT}(20), \text{AT}(100), \text{AT}(200)$, estimated using different σ values. As shown on both figures, the histograms shift progressively to the right as AT is performed for more iterations, indicating that the perturbation operator $\mathcal{Q}_\phi$ becomes more locally dispersive as ϕ evolves in AT. Similar experimental results are also observed on CIFAR-100 and Reduced ImageNet (see Appendix B.3 Figure B.3 and B.4).

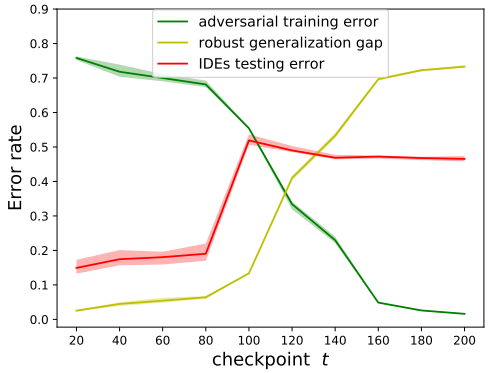
Summary From Theorem 4.6 and these experiments, one may conclude that the deteriorating learning performance on the induced distribution along the AT trajectory can be attributed to the progressive increase of local dispersions of the perturbation operators. It remains unclear what causes perturbation operators in AT to become increasingly dispersive. Nonetheless, this study may shed new lights in understanding the complex dynamics of AT. In particular, we show next that the induced distribution deteriorating along the AT trajectory is correlated with robust overfitting.

4.5 Correlation with Robust generalization

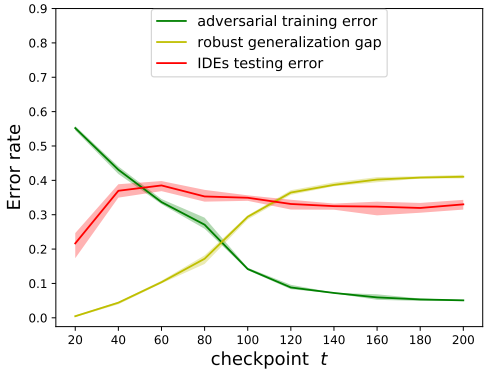
We now explore if the (standard) generalization performance of models learned on the induced distribution $\tilde{\mathcal{D}}_\phi$ along the AT trajectory has any connection to the *robust generalization* performance of ϕ on original data distribution $\mathcal{D}$.



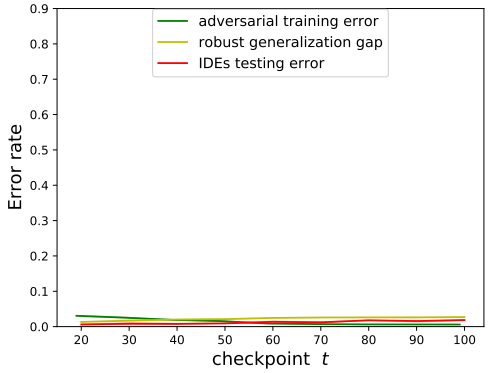
(a) CIFAR-10



(b) CIFAR-100



(c) Reduced ImageNet



(d) MNIST

Figure 4.3: Robust generalization gap of $\phi = \mathrm{AT}(t)$ in comparison to the IDE results w.r.t ϕ. The trend of the red curves matches that of the yellow curves in each sub-figures, demonstrating a compelling correlation between these two quantities.

We conduct extra IDEs for ϕ collected along AT at various epochs and compare the IDE testing errors with the robust generalization performance of the corresponding ϕ. AT and each IDE are repeated five times with different random seeds.

The experimental results on CIFAR-10 and CIFAR-100 are shown in Figure 4.3(a) and (b), where the green and yellow curves respectively report the adversarial training error and the robust generalization gap of ϕ (i.e., $R_S^{\text{adv}}(\phi)$ and $\text{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$). The two curves illustrate a phenomenon known as robust overfitting [78]: after a certain point in AT, the robust generalization gap steadily increases while the adversarial training error constantly decreases. The red curves in the figures depict the standard testing errors achieved in each IDEs (i.e., $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ with θ learned on $\tilde{S}_\phi$). Notably, a significant rise in the IDE testing error is observed when ϕ is taken between AT(80) and AT(120), increasing from 3.6% to 27.68% for CIFAR-10 and from 19% to 48.99% for CIFAR-100. Furthermore, this shift coincides with the onset of robust overfitting, where a significant rise in $\text{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ is also observed.

These results further demonstrate that $\tilde{\mathcal{D}}_\phi$ becomes harder to learn as AT progresses. More importantly, it shows that the appearance of this deteriorating induced distribution is closely linked to the onset of the robust overfitting phenomenon, revealing a correlation between the two. This correlation is further demonstrated by experimental results on Reduced ImageNet (see Figure 4.3 (c)), where robust overfitting emerges at an earlier training stage and simultaneously a rise in $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ occurs. This increment in $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ is also substantial, with an averaged error of 21.65% at AT(20) elevating to 38.52% at AT(60).

Our experiments on MNIST [53] (see Figure 4.3 (d)) exhibits a scenario where a good robust generalization is achieved. Experimental settings on MNIST are shown in Appendix B.1. Interestingly, a small testing error $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ is maintained throughout the evolution of $\tilde{\mathcal{D}}_\phi$ with the absence of robust overfitting. Figure 4.4 shows results from additional experiments on CIFAR-10. In these experiments, we perform AT with different levels of weight decay to control the robust generalization gap. Subsequently, IDEs are conducted for each such variant of AT. In Figure 4.4, each distinct color corresponds to a different weight decay factor utilized in AT. Within each color category, the dashed curves and the corresponding solid lines represent, respectively, $\text{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$ and $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ with ϕ trained by that specific AT variant. From these results, we see that increasing the weight decay factor results in a notable reduction in the $\text{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$, while conversely, decreasing the weight decay factor leads to the opposite effect. This is shown by the downward shift in the dashed curves across the three color categories. More noteworthy is a clear synchronization observed between each pair of dashed and solid curves (of the same color), with lower dashed curves consistently corresponding to lower solid curves in the same color

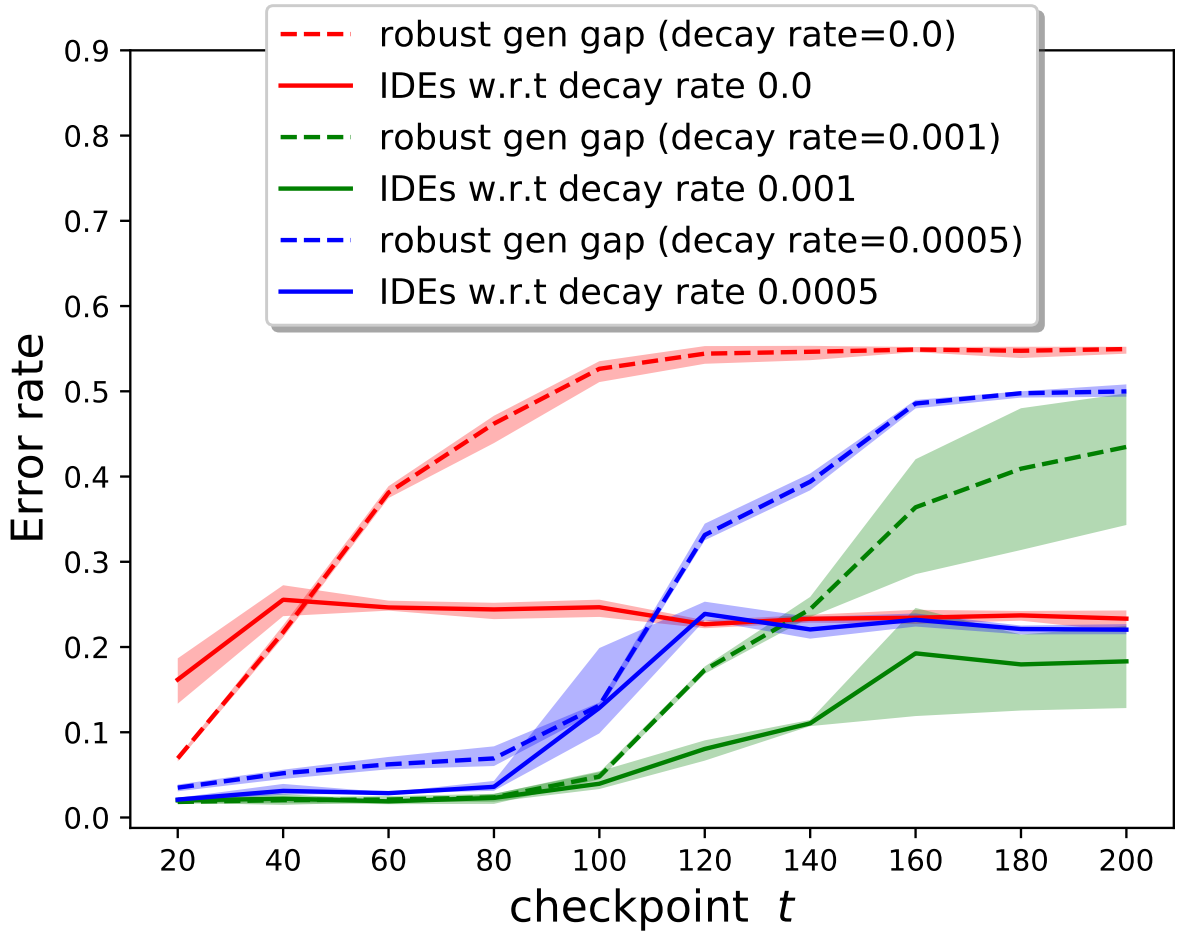


Figure 4.4: AT with various weight decay rates and the test error achieved in IDEs for each of the AT variants. The blue curves are reproduced from Figure 4.3(a), serving as a reference for a clear comparison. The results further solidify the correlation between the robust generalization and the generalization performance on the induced distribution.

category.

All these results suggest a strong correlation between $R_{\tilde{\mathcal{D}}_\phi}(\theta)$ and the robust generalization gap $\mathrm{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi)$. Although by construction, the robust generalization gap is written by $\mathrm{GG}_m(\phi, \tilde{S}_\phi; \tilde{\mathcal{D}}_\phi) = |R_{\tilde{\mathcal{D}}_\phi}(\phi) - R_{\tilde{S}_\phi}(\phi)|$ due to that $R^{\mathrm{adv}}_{\mathcal{D}}(\phi) = R_{\tilde{\mathcal{D}}_\phi}(\phi)$ and $R^{\mathrm{adv}}_{S}(\phi) = R_{\tilde{S}_\phi}(\phi)$, such a correlation is still quite surprising. This is because the learning of the parameter θ has been started from a completely random initialization and one would not expect the resulting parameter θ is linked to the parameter ϕ in any obvious way, despite that the latter contributes to shaping the distribution $\tilde{\mathcal{D}}_\phi$.

A novel observation in this work, this correlation is certainly curious in its own right and deserves further investigation. At this point, it has at least highlighted the impact of the dynamics of AT on robust overfitting, beyond the static quantities, such as loss landscape, while also paving a way for developing deeper understanding of how AT results in robust overfitting.

4.6 Summary

In this chapter, we show that the distribution induced by the perturbation operator in AT may deteriorate along the trajectory of AT. In particular, we observe experimentally that as AT progresses, the induced distribution may become harder to learn. Our theoretical analysis suggests that a key factor governing this increasing difficulty of learning is the local dispersion of the perturbation operator that induces the distribution. Experimental results confirm that as AT proceeds, the perturbation becomes more dispersive, validating our theoretical results. Additionally, we empirically observed a correlation between the deteriorating behavior of the induced distributions with robust overfitting.

The novel observations and our theoretical explanation presented in this chapter contribute to better understanding the complex dynamics of AT. Unraveling this complexity is arguably essential to understanding robust generalization in AT.

Chapter 5

Algorithmic Stability of Adversarial Training

In this chapter, we define generalization with respect to perturbation-induced losses, a framework that encompasses both standard generalization and robust generalization as special cases. Building upon this framework, we present a novel stability analysis of adversarial training (AT) and prove generalization upper bounds in terms of an expansiveness property of adversarial perturbations used during training and used for evaluation. These expansiveness parameters appear to not only govern the vanishing rate of the generalization error but also govern its scaling constant. Our bound attributes the robust overfitting in PGD-based adversarial training to the sign function used in the PGD attack, resulting in a bad expansiveness parameter. The peculiar choice of sign function in the PGD attack appears to impact adversarial training both in terms of (inner) optimization and in terms of generalization, as shown in this work. This aspect has been largely overlooked to date. Going beyond the sign-function based PGD attacks, we further show that poor expansiveness properties exist in a wide family of PGD-like iterative attack algorithms, which may highlight an intrinsic difficulty in adversarial training.

Notations and Basic Setup We consider the standard setting of supervised learning, where the training samples are instance-label pairs, (x_i, y_i)'s, drawn i.i.d from an underlying data distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Here the input space $\mathcal{X}$ is $\mathbb{R}^d$ and the label space $\mathcal{Y}$ is finite. We restrict to parameterized models, e.g., neural networks, in which the model parameter w lives in a subset $\mathcal{W}$ of some real vector space. We use $f(w, x, y)$ to denote the loss value of (x, y) under model parameter w, where a standard choice of loss function

(e.g. 0-1 loss, cross-entropy loss, etc.) is absorbed in f. For example, $f(w, x, y)$ can be the cross-entropy loss of the a neural network with parameter w on sample (x, y).

The central object of this study is adversarial training, which allows the learned model to resist adversarial attacks. Each adversarial attack (or adversarial perturbation) on input x is assumed to live in an ∞-norm ball $\mathbb{B}_\infty(x, \epsilon) := \{t \in \mathbb{R}^d : \|t - x\|_\infty \leq \epsilon\}$ with radius ϵ and centered at x. For a vector $x \in \mathbb{R}^d$, $x[i]$ denotes the i^{th} coordinate of x.

5.1 Perturbation Induced Loss

Let J be a function mapping $\mathcal{W} \times \mathcal{X} \times \mathcal{Y}$ to $\mathcal{X}$ satisfying $J(x; y, w) \in \mathbb{B}_\infty(x, \epsilon)$. Then $J(x; y, w)$ may be regarded as a perturbation of x by a magnitude of up to ϵ (under ∞-norm). We then define the *perturbation J induced loss* or simply *J-loss* by

$$f_J(w, x, y) := f\left(w, J(x; y, w), y\right) \tag{5.1}$$

Specially, let $J^*(x; y, w) := \arg\max_{\hat{x} \in \mathbb{B}_\infty(x,\epsilon)} f(w, \hat{x}, y)$, and $J^{\text{id}}(x; y, w) := x$. We will call $f_J(w, x, y)$ as the *robust loss* when $J = J^*$ and call it as the *standard loss* when $J = J^{\text{id}}$ and $f_J(w, x, y)$ reduces to $f(w, x, y)$. For simplicity we will denote the robust loss f_{J^*} by f^* and simply write the standard loss as $f(w, x, y)$. We will soon encounter other forms of J-loss.

Remark 5.1. The motivations for considering perturbations J other than J^* stems from practical scenarios where J^* is usually unavailable for DNNs and is often approximated using other perturbation methods, such as multi-step PGD perturbation. Considering specific forms of J allows us to analyze the impact of the perturbations used in AT on the generalization performance of the model.

Generalization w.r.t the induced loss Let the training set $S = \{(x_i, y_i)\}_{i=1}^n$ be drawn from $\mathcal{D}^n$. Consider a learning algorithm A, which when applied on S gives rise to a learned model parameter $w = A(S)$. The population risk and empirical risk w.r.t J-loss are defined respectively as:

$$R_{\mathcal{D}}[A(S); J] := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f_J(A(S), x, y)\right] \quad \text{and} \quad R_S[A(S); J] := \frac{1}{n}\sum_{i=1}^n f_J(A(S), x_i, y_i) \tag{5.2}$$

Remark 5.2. Notably w entails randomness, due to the random sampling of S and the possible intrinsic randomness in A. The risks $R_{\mathcal{D}}[A(S); J]$ and $R_S[A(S); J]$ are therefore random variables in this setting.

Remark 5.3. Let $w = A(S)$. We notice that $R_{\mathcal{D}}[w; J]$ and $R_S[w; J]$ reduce to the standard risks when $J = J^{\mathrm{id}}$ and turn into the robust risks when $J = J^*$. The risks defined based on the J-loss then includes the standard risks and the robust risks as special cases.

We consider the expected generalization gap w.r.t the J-loss, defined as

$$\mathrm{GG}_n(J, A) := \mathbb{E}_{S,A}\left[R_{\mathcal{D}}[A(S); J] - R_S[A(S); J]\right] \tag{5.3}$$

where expectation over A refers to averaging over the intrinsic randomness in A. Specially, we will call $\mathrm{GG}_n(J^{\mathrm{id}}, A)$ and $\mathrm{GG}_n(J^*, A)$ respectively the *standard generalization* gap and the *robust generalization* gap of the algorithm A.

Remark 5.4. Note that the definitions in (5.2) and (5.3) applies to arbitrary choice of learning algorithm A which may not be related to AT or the perturbation J. In the next chapter, we will analyze the generalization gap (5.3) when A is taken as the AT algorithms introduced below.

A motivating example In this work, we aim to understand the underlying causes of the robust overfitting phenomenon observed in [78] and to provide a better theoretical explanation for it. The work of [78] shows that on the CIFAR-10 dataset [47], the model trained by AT using 10-step PGD attack is still vulnerable to *the same* 10-step PGD attack on the testing set. We extend this experiment and show that robust overfitting persists even when AT uses a 3-step PGD attack instead. Specifically, in Figure 5.1, we train a model using AT with a 3-step PGD and measure the error of the model against 3-step PGD attack as well as its standard error on the test set along the training process. We observe that the model trained by AT with 3-step PGD is still vulnerable to the same 3-step PGD attack on the testing set. After the first learning rate decay (the 100^{th} epoch), the testing error w.r.t the 3-step PGD starts to rise, similar to the observations in [78].

In this example, there is no reason to treat the 3-step PGD as J^*. Therefore, theoretical works that consider the setting where J^* is used in AT, such as those in [3, 104, 106, 110], do not fully explain the robust overfitting observed in our experiments. This gap motivates our introduction of generalization w.r.t an induced loss, a framework that examines the generalization behavior of AT under a broad range of perturbations beyond J^*. We believe our proposed framework could provide a more comprehensive theoretical foundation for understanding robust overfitting as observed in practice.

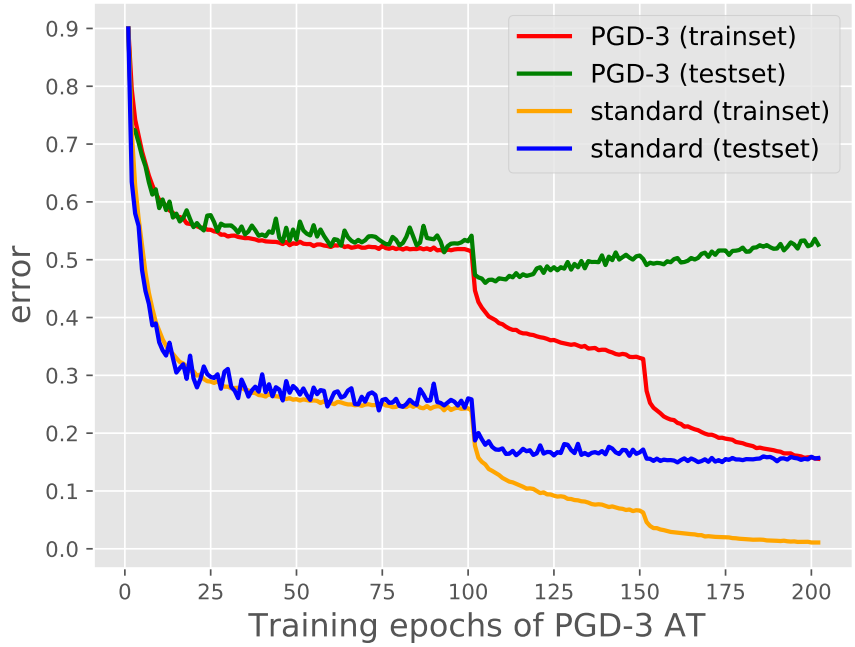


Figure 5.1: The learning curve of a model trained by AT on CIFAR-10 with 3-step PGD. The standard error as well as the error against the same 3-step PGD attack are measured during AT on both the training and testing sets. The step size for PGD and the perturbation radius w.r.t the $\infty-$norm are respectively set to 7/255 and 8/255. The learning rate is decayed at the 100^{th} and the 150^{th} epoch.

5.2 AT Algorithms and PGD Attacks

We consider the following iterative AT algorithm. At each iteration of AT, it first draw a training sample $(x_{i_t}, y_{i_t}) \in S$ and then updates the model parameter w_t according to

$$x_t^{\text{adv}} = \pi(x_{i_t}; y_{i_t}, w_t) \tag{5.4}$$

$$w_{t+1} = w_t - \tau_t \nabla_{w_t} f(w_t, x_t^{\text{adv}}, y_{i_t}) \tag{5.5}$$

Here $\tau_t \in \mathbb{R}_+$ denotes the step size of the gradient descend at the iteration t. The index i_t of the training samples is drawn uniformly and independently (across t) from $\{1, 2, \ldots, n\}$, and $\pi(x_{i_t}; y_{i_t}, w_t)$ denotes a certain perturbation of x_{i_t} within $\mathbb{B}_\infty(x_{i_t}, \epsilon)$.

Remark 5.5. Note that when the perturbation π in (5.4) is chosen as the identity map J^{id}, the AT algorithm reduces to the standard stochastic gradient descend (SGD) algorithm. The standard SGD can therefore be treated as a special case of AT.

Remark 5.6. We note that ideally π should be $J^*(x_{i_t}; y_{i_t}, w_t)$ but in practice it is only an approximation of it due to the difficulty in acquiring the exact solution. Additionally and more critically, we note that, despite that both π and J refer to perturbations, the two notions in this paper may be completely different. Specifically, J induces the J-loss, which is used as a performance metric (evaluated either on the training set or on the testing set), whereas π denotes the perturbation operation applied during adversarial training. Although in some cases π is taken as J or is related to J, there are scenarios in which π and J are completely decoupled, for example, when we perform adversarial training but choose to evaluate the model using the standard loss, i.e., using J^{id}-loss. In the later sections, we will see more cases in which J and π are completely different.

Remark 5.7. Note that although x_t^{adv} is also a function of w_t, we consider that the derivative operator in (5.5) does not go through π, an option consistent with the standard AT implementation as in [61, 78].

As we may look into various choices of π in AT algorithms, we use A_π to denote an AT algorithm, emphasizing its dependence on π. Under such notations, we may even consider "mis-matched generalization gap", namely, $\text{GG}_n(J, A_\pi)$ with $J \neq \pi$, for example, $J = J^{\text{id}}$ and π is a particular adversarial perturbation.

The PGD attack Associated with any $(x, y) \in \mathbb{R}^d \times \mathcal{Y}$ and any weight parameter $w \in \mathcal{W}$, we define and one-step PGD map $T_{x,y,w}$ by

$$T_{x,y,w}(x') = \Pi_{\mathbb{B}_\infty(x,\epsilon)} \left[x' + \lambda G \left(\nabla_{x'} f(w, x', y) \right) \right] \tag{5.6}$$

Here x' is any point in $\mathbb{R}^d$, G is a mapping from $\mathbb{R}^d$ to $\mathbb{R}^d$, possibly taking various forms, which we will specify momentarily, λ is another step size, and $\Pi_{\mathbb{B}_\infty(x,\epsilon)} : \mathbb{R}^d \to \mathbb{B}_\infty(x,\epsilon)$ denotes the projection onto the set $\mathbb{B}_\infty(x,\epsilon)$, namely, $\Pi_{\mathbb{B}_\infty(x,\epsilon)}(x') = \arg\min_{\tilde{x}\in\mathbb{B}_\infty(x,\epsilon)} \|\tilde{x} - x'\|_2$.

The K-step PGD attack π^{PGD} is then defined as the K-fold compositions of the (same) mapping $T_{x,y,w}$:

$$\pi^{\mathrm{PGD}}(x;y,w) := T^K_{x,y,w}(x) := \left(\underbrace{T_{x,y,w} \circ T_{x,y,w} \circ \ldots T_{x,y,w}}_{K \text{ times}}\right)(x) \tag{5.7}$$

Specially, we call the AT algorithm as the *PGD-based AT* when $\pi = \pi^{\mathrm{PGD}}$ in (5.4).

Remark 5.8. The choice of the mapping G lacks a unified criterion and is largely heuristic. In [61] and several other empirical studies (see [1, 27, 78, 99, 101, 102]), the mapping G is commonly taken as the sign function, applied element-wise on the gradients. In contrast, some theoretical works (e.g., [12, 22, 32]), simply adopt G as the identity map. To the best of our knowledge, no formal theoretical guidance currently exists for selecting G in the PGD attack.

In section 5.6, we will show that the choice of G, this peculiar and largely overlooked building block in PGD, in fact has non-negligible impact on the generalization performance of PGD-based AT.

5.3 Preliminaries for Algorithmic Stability Analysis

Uniform stability was first introduced by the landmark work of [9]. An influential work by [36] adapts this framework to analyze the uniform stability of SGD with smooth loss functions, explaining the effectiveness of SGD in training neural networks. Since then, many studies have built upon [36] to develop stability bounds for SGD with non-smooth losses (e.g., [6,55]). Data-dependency in stability analysis is introduced in [52], and uniform stability for more sophisticated variants of SGD is also studied (e.g., [17,65]). Additionally, works such as [30, 54] have explored algorithmic stability in general minimax problems. These studies are more closely related to generative adversarial networks (GANs), rather distant from the standard settings of adversarial training.

A line of works use uniform stability to analyze the generalization of AT. In [106], the adversarial loss is assumed convex and non-smooth and AT is regarded as standard SGD

on this loss, whereby an existing generic bound for non-smooth loss in [6] is invoked for analysis. As pointed out in [104], the bound obtained in [106] is independent of the specific choice of loss function used for training and insufficient to reflect the difference between AT and SGD observed in practice. The work of [104] argues that the adversarial loss is approximately smooth and derive bounds based on the stability framework of SGD in [36]. The work of [98], built upon [104], extends the analysis of AT to the data-dependent stability framework in [52]. But the bounds obtained in both [104] and [98] do not vanish with sample size.

To overcome these limitations and shed new light in understanding robust overfitting, we present in this chapter novel stability analysis for the generalization of models learned using AT with arbitrary perturbations. We will first introduce related notions in stabiliy analysis and then present our analysis towards the generalization gap (5.3) by exploiting the tool of uniform stability [9].

Definition 5.1 (Uniform stability w.r.t J-loss)**.** *Let $S \simeq S'$ denotes two datasets that each contains n samples but differ in at most one. A randomized algorithm A is $\rho-$uniformly stable w.r.t J-loss, if*

$$\Delta_n(J, A) := \sup_{S \simeq S'} \sup_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \mathbb{E}_A[f_J(A(S), x, y) - f_J(A(S'), x, y)] \leq \rho \tag{5.8}$$

Here the expectation is taken over the possible intrinsic randomness in A.

Remark 5.9. This definition is in fact a modification of the Definition 2.1 in [36], where we specifically use J-loss to measure the performance of an algorithm A.

Theorem 2.2 in [36] shows that uniform stability implies generalization in expectation. This result can be easily extended to the uniform stability w.r.t J-loss due to that the analysis [36] applies to arbitrary loss functions, including the $J-$loss defined in the previous chapter.

Lemma 5.1. *For any perturbation J and any algorithm A,*

$$\mathrm{GG}_n(J, A) \leq \Delta_n(J, A) \tag{5.9}$$

For completeness, we present the proof of this lemma here.

Proof. Recall that $\mathrm{GG}_n(J, A) := \mathbb{E}_{S,A}\left[R_{\mathcal{D}}[A(S); J] - R_S[A(S); J]\right]$ and $S := \{(x_1, y_1), \cdots, (x_n, y_n)\}$. Denote $S^{(i)} := \{(x_1, y_1), \cdots, (\hat{x}_i, \hat{y}_i), \cdots, (x_n, y_n)\}$ as a copy of S with the i^{th} sample in S

replaced by some sample $(\hat{x}_i, \hat{y}_i)$. Additionally, let $\hat{S} := \{(\hat{x}_1, \hat{y}_1), \cdots, (\hat{x}_n, \hat{y}_n)\}$. We have that

$$\mathbb{E}_{S,A} R_S[A(S); J] \tag{5.10}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{S,A} f_J(A(S), x_i, y_i) \tag{5.11}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{S,A} \mathbb{E}_{(\hat{x}_i, \hat{y}_i) \sim \mathcal{D}} f_J(A(S^{(i)}), \hat{x}_i, \hat{y}_i) \tag{5.12}$$

$$= \mathbb{E}_{S,A} \mathbb{E}_{\hat{S}} \left[\frac{1}{n} \sum_{i=1}^{n} f_J(A(S^{(i)}), \hat{x}_i, \hat{y}_i) \right] \tag{5.13}$$

We also have

$$R_{\mathcal{D}}[A(S); J] := \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[f_J(A(S), x, y) \right] \tag{5.14}$$

$$= \mathbb{E}_{\hat{S}} \left[\frac{1}{n} \sum_{i=1}^{n} f_J(A(S), \hat{x}_i, \hat{y}_i) \right] \tag{5.15}$$

Based on (5.13) and (5.15), we have

$$\mathrm{GG}_n(J, A) = \mathbb{E}_{S,A} \mathbb{E}_{\hat{S}} \left[\frac{1}{n} \sum_{i=1}^{n} f_J(A(S), \hat{x}_i, \hat{y}_i) \right] - \mathbb{E}_{S,A} \mathbb{E}_{\hat{S}} \left[\frac{1}{n} \sum_{i=1}^{n} f_J(A(S^{(i)}), \hat{x}_i, \hat{y}_i) \right] \tag{5.16}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\hat{S}} \mathbb{E}_{S,A} \left[f_J(A(S), \hat{x}_i, \hat{y}_i) - f_J(A(S^{(i)}), \hat{x}_i, \hat{y}_i) \right] \tag{5.17}$$

$$\leq \sup_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \sup_{S \simeq S'} \mathbb{E}_A [f_J\left(A(S), x, y\right) - f_J\left(A(S'), x, y\right)] \tag{5.18}$$

$$= \Delta_n(J, A) \tag{5.19}$$

□

Assumptions on f In this chapter, we will consider the family of f that are Lipschitz and gradient-Lipschitz with respect to both x and w in the following sense: there exist positive constants $L_{\mathcal{X}}$, $L_{\mathcal{W}}$, $\Gamma_{\mathcal{X}}$ and β such that for any $y \in \mathcal{Y}$, any $x, x' \in \mathcal{X}$ and any $w, w' \in \mathcal{W}$

$$|f(w', x', y) - f(w, x, y)| \leq L_{\mathcal{X}} \|x - x'\| + L_{\mathcal{W}} \|w - w'\| \tag{5.20}$$

$$\|\nabla_{w'} f(w', x', y) - \nabla_w f(w, x, y)\| \leq \Gamma_{\mathcal{X}} \|x - x'\| + \beta \|w - w'\| \tag{5.21}$$

With the Lipschitz condition of f, the uniform stability w.r.t f_J can be related with the notion of the uniform argument stability (UAS), a notion coined in [6], as well as an "expansiveness" property of J, which we will soon define.

Definition 5.2 (Uniform Argument Stability (UAS) [6])**.** *We define the UAS parameter $\delta_n(A)$ of an algorithm A as*

$$\delta_n(A) := \sup_{S \simeq S'} \mathbb{E}_A \|A(S) - A(S')\| \tag{5.22}$$

Definition 5.3 (c-expansiveness)**.** *For any given $c \geq 0$, we define the c-expansiveness of a perturbation J as*

$$q_c(J) := \sup_{(x,y)} \sup_{w,w' : \|w - w'\| > c} \frac{\|J(x;y,w) - J(x;y,w')\|}{\|w - w'\|} \tag{5.23}$$

Remark 5.10. We note that such a notion of expansiveness reduces to a Lipschitz condition when $c = 0$. It measures the sensitivity of an operator to the perturbation of its input, sharing similarity with the Lipschitz condition but provide extra benefit when analyzing operators whose Lipschitz constant is unbounded. When taking $c > 0$, this expansiveness, however, excludes measuring sensitivity for perturbation with magnitude lower than c. This consideration is motivated by the fact that in practice, extremely small perturbation do not arise. Additionally, this expansiveness behaves nicely, i.e., being bounded, even for non-continuous operators, such as those defined via the sign function, to arise later in this chapter.

Relating uniform stability with UAS For any given S and S' differing by only one element and every $c^* \geq 0$, let $Q(S, S'; c^*)$ denote the probability (under the probability measure induced by the randomness in A) that $\|A(S) - A(S')\| < c^*$. Specifically, let S_* and S'_* denote two training sets with $S_* \simeq S'_*$ which achieve the supremum in the definition of $\Delta_n(J, A)$ in (5.8). We write $Q(c^*)$ in place of $Q(S_*, S'_*; c^*)$ for simplicity.

Lemma 5.2. *If the loss function f satisfies the Lipschitz condition (5.20), then for any $c^* \geq 0$,*

$$\Delta_n(J, A) \leq (L_{\mathcal{W}} + q_{c^*}(J) L_{\mathcal{X}}) \delta_n(A) + L_{\mathcal{X}} Q(c^*) \cdot 2\epsilon\sqrt{d} \tag{5.24}$$

The proof of this lemma is deferred to Appendix C.1. In the remainder of this chapter, we will use this bound to analyze the generalization of adversarial training (AT) algorithms. We will show, for most cases, that this bound vanishes with sample size n by choosing a judicious choice of c^*. The only case in which a vanishing bound is not attainable is sign-PGD based AT, where the bound converges to a constant. This may reveal some intrinsic difficulty in generalization for such AT algorithm.

5.4 Main Results

In this section, we present our main results. Recall that A_π denotes the AT algorithm defined in (5.4) and (5.5). We have the following upper bound for the UAS of A_π.

Theorem 5.11. *Suppose that f satisfies the conditions (5.20) and (5.21). If we run A_π for T steps with step sizes $\tau_t \leq \frac{1}{\beta}$, there exist a constant $c > 0$ such that we have*

$$\delta_n(A_\pi) \leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} (2 + q_c(\pi)\Gamma_{\mathcal{X}}/\beta)^t \tag{5.25}$$

We defer the proof of the theorem to Appendix C.1. With the upper bound of the UAS, an upper bound for the mismatched generalization gap can be immediately derived according to (5.9) and (5.24) as below:

Theorem 5.12. *Under the condition of Theorem 5.11, for any $c^* \geq 0$, there exists a constant $c > 0$, such that*

$$\mathrm{GG}_n(J, A_\pi) \leq (L_{\mathcal{X}} q_{c^*}(J) + L_{\mathcal{W}}) \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} (2 + q_c(\pi)\Gamma_{\mathcal{X}}/\beta)^t + L_{\mathcal{X}} Q(c^*) \cdot 2\epsilon\sqrt{d} \tag{5.26}$$

Remark 5.13. The bound in (5.26) also includes as a special case the "matched" generalization gap $\mathrm{GG}_n(J, A_J)$, where the perturbation used in adversarial training is identical to that defining performance metric, as is typical in the adversarial training literature. Beyond the Lipschitz and smoothness conditions of f, the expansiveness parameters of π and J turn out to also influence the generalization of AT algorithms, as suggested in the generalization bound (5.26). This has been overlooked by the previous stability analysis as in [98, 104, 106].

The behavior of the bound in (5.26) clearly depends on $Q(c^*)$. We now show that with additional conditions, one can choose a c^* to either remove the term containing $Q(c^*)$ or make $Q(c^*)$ also vanish with n.

For example, if the perturbation J has bounded Lipschitz constant q^*, that is $q_{c^*}(J) \leq q^* < \infty$ for any $c^* \geq 0$, then taking $c^* = 0$ simply results in the following bound that vanishes as $\mathcal{O}(1/n)$.

$$\mathrm{GG}_n(J, A_\pi) \leq (L_{\mathcal{X}} q^* + L_{\mathcal{W}}) \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} (2 + q_c(\pi)\Gamma_{\mathcal{X}}/\beta)^t \tag{5.27}$$

On the other hand, if the second moment of the random variable $\|A(S_*) - A(S'_*)\|$ has a fast vanishing rate with n, one can choose c^* to decay with n at a judicious choice of rate, pushing $Q(c^*)$ to vanish faster than $1/n$, resulting in the bound in the following form

$$\mathrm{GG}_n(J, A_\pi) \leq (L_{\mathcal{X}} q_{c^*}(J) + L_{\mathcal{W}}) \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} (2 + q_c(\pi)\Gamma_{\mathcal{X}}/\beta)^t + o(1/n) \tag{5.28}$$

We defer the proof of (5.28) to Appendix C.1.

Convex loss and strongly convex loss When f is further assumed to be convex or strongly convex, a tighter UAS upper bound can be attained.

Theorem 5.14. *Suppose that $f(\cdot, x, y)$ is convex for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and satisfies the conditions (5.20) and (5.21). If we run A_π for T steps with step sizes $\tau_t \leq \frac{1}{\beta}$, we have*

$$\delta_n(A_\pi) \leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} (1 + q_c(\pi)\Gamma_{\mathcal{X}}/\beta)^t \tag{5.29}$$

If we further assume $f(\cdot, x, y)$ is $\mu-$strongly convex, we have

$$\delta_n(A_\pi) \leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} \left(1 - \frac{\mu}{2\beta} + \Gamma_{\mathcal{X}} q_c(\pi)/\beta\right)^t \tag{5.30}$$

As shown, performing AT using convex loss functions results in a tighter upper bound compared to the non-convex functions. When f is strongly convex, the bound can be tightened again. In fact, in the strongly convex case, if $q_c(\pi)$ is small enough, the UAS upper bound can be made independent with the number of iteration T.

Corollary 5.1. *Suppose that f is $\mu-$strongly convex and satisfies the conditions (5.20) and (5.21). Suppose that $q_c(\pi) < \mu/(2\Gamma_{\mathcal{X}})$ and we run A_π for T steps with step sizes $\tau_t \leq \frac{1}{\beta}$, we have*

$$\delta_n(A_\pi) \leq \frac{4L_{\mathcal{W}}}{n(\mu - 2q_c(\pi)\Gamma_{\mathcal{X}})} \tag{5.31}$$

The proofs of Theorem 5.14 and Corollary 5.1 are deferred to Appendix C.1.

Remark 5.15. Notably, when π is chosen as the identity map, we have $q_c(\pi) = 0$ and A_π reduces to the standard SGD algorithm. In this case, our UAS upper bounds matches the bounds in [36] up to constants.

5.5 Comparison with existing UAS bounds for AT

The work in [30] derives UAS bounds for the AT-like algorithm (refer to as GDmax in their paper) under the assumption that f is strongly concave in $\mathcal{X}$. Our work goes beyond this restricted setting and derive UAS bounds without this assumption. In [106], the stability of AT is analyzed by treating AT as standard SGD with an adversarial loss (i.e., f^*) and invoke the generic bound in [6] for non-smooth losses, while assuming f^* to be non-smooth. The non-smoothness is however not quantitatively characterized in their work; additionally since the bound in [6] is developed for SGD with any non-smooth convex functions, it fails to explain the notable difference between SGD and AT observed in practice.

The assumptions used in our proof looks similar to those in [104] and [98]. However, the UAS bounds proposed in [104], as well as the bound in [98], which extends from [104], include terms that do not vanish with increasing sample size. Our bounds derived based on the proposed new framework and by considering the expansiveness parameter, overcome this limitation, vanishing with the sample size.

We now compare our work to [104] and [98] in details. Since the work of [98] is built upon the framework in [104], we here only presents the connections and differences between [104] and our work.

Summary of generalization bounds in [104] First we would like to note that the problem setting in this paper includes the setting in [104] as a special case. Specifically, the generalization gap discussed in [104] corresponds to the generalization gap $\mathrm{GG}_n(J^*, A_{J^*})$ defined in this paper, where the perturbations in both $J-$loss and the AT algorithm are taken as the optimal adversarial perturbation J^*.

This work and [104] both take the Lipschitzness and smoothness conditions of the standard loss f as the starting point, but derive generalization bounds from different perspectives: the work in [104] defines and proposes to study the $\eta-$approximate smoothness of the adversarial loss (f^* in our notation) and derive generalization bounds based on this quantity. This work defines the notion of $c-$expansiveness of the perturbation operator (e.g., J^*) and show how this quantity affects generalization performance of AT.

For completeness, we here present the definition of $\eta-$approximate smoothness, rewrite the Definition 4.1 of [104] using the notations in this paper.

Definition 5.4 ($\eta-$approximate smoothness [104])**.** *A loss function f_J is called $\eta-$approximately $\beta-$gradient Lipschitz if there exists $\beta > 0$ and $\eta > 0$ such that for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and for any $w_1, w_2 \in \mathcal{W}$ we have*

$$\|\nabla f_J(w_1, x, y) - \nabla f_J(w_2, x, y)\| \leq \beta\|w_1 - w_2\| + \eta \tag{5.32}$$

The work in [104] then derives generalization bounds for loss functions that are $\eta-$approximately smooth. For example, after replacing the notations in [104] with ours, Theorem 5.1 of [104] shows that if f_J is $\eta-$approximately $\beta-$gradient Lipschitz, convex in w for all (x, y) and the standard loss f satifies the same Lipschitz condition in (6) of this paper (or Assumption 4.1. in [104]), then their bound in Theorem 5.1 becomes

$$\mathrm{GG}_n(J, A_J) \leq \frac{L_{\mathcal{W}}}{\beta}\eta T + \frac{2L_{\mathcal{W}}^2}{n\beta}T$$

The authors of [104] show that the adversarial loss f^* satisfies η-approximately β-gradient Lipschitz with $\eta = 2\Gamma_{\mathcal{X}}\epsilon$ so that the generalization bound above gives their generalization bound for adversarial training. In their determination of the η parameter, they have assumed that the standard loss f satisfies certain Lipschitz and smoothness condition; this condition is effectively equivalent to the condition (5.21) in this paper.

It is worth noting that the generalization bounds derived based on the approximate smoothness parameter η contain a term unrelated to the sample size n because of the independence of η on n.

The limitation of the framework in [104] We would like to note that when the standard loss f satisfies the Assumption 4.1 in [104] (or condition (5.21) in this paper), in fact every $J-$loss (for any arbitrary J, including but not limited to J^*) is $2\Gamma_{\mathcal{X}}\epsilon-$approximately smooth. To see this:

$$\begin{aligned}
&\|\nabla_{w_1} f_J(w_1, x, y) - \nabla_{w_2} f_J(w_2, x, y)\| \\
=&\|\nabla_{w_1} f(w_1, J(x; y, w_1), y) - \nabla_{w_2} f(w_2, J(x; y, w_2), y)\| && (5.33)\\
\leq&\beta\|w_1 - w_2\| + \Gamma_{\mathcal{X}}\|J(x; y, w_1) - J(x; y, w_2)\| && (5.34)\\
\leq&\beta\|w_1 - w_2\| + \Gamma_{\mathcal{X}}(\|J(x; y, w_1) - x\| + \|x - J(x; y, w_2)\|) && (5.35)\\
\leq&\beta\|w_1 - w_2\| + 2\Gamma_{\mathcal{X}}\epsilon && (5.36)
\end{aligned}$$

where inequality (5.34) follows from Assumption 4.1 in [104]. Inequality (5.35) and (5.36) are derived by using the triangle inequality and the condition that $\|J(x; y, w) - x\| \leq \epsilon$ for any $w \in \mathcal{W}$.

Due to the fact that all the $J-$losses have the same approximate smoothness parameter η, the generalization bounds derived for different $J-$loss, based on the framework in [104], will be the same. This type of generalization bound ignores the influence of the perturbations used in AT on generalization and it is therefore unable to explain the experimental observations in the following Sections where different choices of perturbations indeed have distinct impact on generalization.

Difference of our approach from [104] In this paper, we depart from the approach of [104], which ignores the specific properties of perturbation J, and take a different route which considers the impact of J measured via its expansiveness parameter. Our approach allows us to analyze how different perturbations used in AT affect its generalization performance. Our bounds, derived based on the expansiveness parameter, also avoid having the non-vanishing term (like the first term in Theorem 5.1 of [104]) when the expansiveness parameter is finite. Only in the case when the expansiveness parameter is unbounded, our results are similar to [104], where the generalization bound contains a non-vanishing term.

The UAS parameter of AT characterizes the gap $\|w - w'\|$ where $w = A(S)$ and $w' = A(S')$ are the model parameters produced by the AT algorithm on two nearly identical datasets $S \simeq S'$. Intuitively, the difference between w and w' arises from the single different example in S and S' (where larger training sample size n tends to reduce the probability of using that single different example to update model parameters in AT), and gets "magnified" by the perturbation J along the AT training trajectory. The expansiveness parameter of J in this paper effectively captures this "magnification" factor. Thus, the eventual difference between w and w' depends on not only the sample size n but also the expansiveness parameter of J. Then the exploitation of the expansiveness of J brings sample size n into the bounds.

5.6 Revisit of PGD-based AT

The upper bounds in the last section suggest that the expansiveness parameter of π affect the UAS parameter of A_π and therefore the generalization gap w.r.t A_π. We now analyze the expansiveness parameter for the PGD perturbation π^{PGD} and then derive generalization bound for the PGD-based AT (i.e., A_π with $\pi = \pi^{\mathrm{PGD}}$).

To begin, we assume that the gradient $\nabla_x f$ is Lipschitz, namely, that there exist positive constants η and $\Gamma_{\mathcal{W}}$ such that for any $y \in \mathcal{Y}$, any $x, x' \in \mathcal{X}$ and any $w, w' \in \mathcal{W}$

$$\|\nabla_{x'} f(w', x', y) - \nabla_x f(w, x, y)\| \leq \eta \|x - x'\| + \Gamma_{\mathcal{W}} \|w - w'\| \tag{5.37}$$

Lemma 5.3 (Expansiveness of PGD)**.** *Consider the mapping π^{PGD} defined in (5.7). Suppose that f satisfies the condition (5.37) and the mapping G is $\alpha-$ Lipschitz.*

$$q_c(\pi^{\mathrm{PGD}}) \leq \min\left(\sum_{k=0}^{K-1} \mu^i \nu, \frac{2\sqrt{d}\epsilon}{c}\right) \tag{5.38}$$

where $\nu = \lambda\alpha\Gamma_{\mathcal{W}}$ and $\mu = 1 + \lambda\alpha\eta$.

We defer the proof to Appendix C.1.

For all J-losses for which $q_c(J)$ is uniformly bounded by q^*, plugging this bound to (5.27) immediately gives a generalization bound that vanishes as $\mathcal{O}(1/n)$. However, one of the most important J-loss, the one defined using sign-PGD attack, fails to satisfy this boundedness condition and the bound (5.27) does not apply.

To carefully study such a setting, let $J^{\mathrm{sign-PGD}} := \pi^{\mathrm{sign-PGD}}$, where $\pi^{\mathrm{sign-PGD}}$ is π^{PGD} with function G taken as the sign function. We have the following results.

Corollary 5.2. *Let $J = J^{\mathrm{sign-PGD}}$. Suppose that for any S and S' with $S \simeq S'$, $\|A(S) - A(S')\| < B$ with probability 1. Under the condition of Theorem 5.11, for any $\rho > 0$, there exists some N (depending on ρ), such that when $n > N$,*

$$\mathrm{GG}_n(J, A_\pi) < (1 - \delta_n(A_\pi)/B)\, L_{\mathcal{X}} \cdot 2\epsilon\sqrt{d} + \rho. \tag{5.39}$$

The proof is left in Appendix C.1.

Remark 5.16. Note that for $J^{\mathrm{sign-PGD}}$-loss and without additional information on $\|A(S) - A(S')\|$, it appears difficult to arrive at a generalization bound that vanishes with n and the bound given here converges to a constant. Although this may not mean that AT with $J^{\mathrm{sign-PGD}}$-loss does not have a vanishing generalization error, it nonetheless reveals certain intrinsic difficulty of generalization for this setting. Specifically, for large n, the perturbation radius (that defines the J-loss) and the input dimension appear to fight against the UAS parameter $\delta_n(A_\pi)$; when UAS parameter decreases – which pushes towards better generalization, $\epsilon\sqrt{d}$ is amplified more significantly – causing poorer generalization.

Convergence analysis of PGD Lemma 5.3 suggests that PGD attacks with fewer steps K tends to have smaller expansiveness parameter, leading to improved generalization performance in the corresponding PGD-AT. However, as K decreases, the perturbations generated by PGD may stay further from the optimal perturbation J^*, suggesting a trade-off between generalization and robustness.

To explore this, we now present a convergence analysis for the PGD attacks defined in (5.7). Specifically, we consider PGD attacks with the mapping G that satisfies the following condition:

$$\nabla_x f(w,x,y)^T G(\nabla_x f(w,x,y)) > 0 \tag{5.40}$$

for any $(x,y) \in \mathcal{X} \times \mathcal{Y}$ and any $w \in \mathcal{W}$. Note that this condition simply requires the direction of the modified gradient $G(\nabla_x f(w,x,y))$ to align near the direction of the original gradient, within 90 degree angle.

Lemma 5.4 (Convergence of PGD)**.** *Suppose that $f(w,x,y)$ satisfies the condition (5.37). Let $x^* = J^*(x;y,w)$ and suppose that $\nabla_x f(w,x^*,y) = 0$. Suppose $\|G(\nabla_x f(w,x,y))\|^2 \le C$ for any (w,x,y). For any mapping G that satisfies the condition (5.40), performing the $K-$step PGD (5.7) with step size $\lambda = \frac{1}{\sqrt{K}}$ results in*

$$f(w,x^*,y) - \frac{1}{K}\sum_{k=1}^{K} f(w,x^k,y) \le \frac{(2C+d^*)}{2K} + \frac{d^*(\eta^2+\eta+1)}{2} \tag{5.41}$$

where $d^* = \max_{k\in\{1,\cdots,K\}} \|x^k - x^*\|^2$ and $x^k := T^k_{x,y}(x;w)$ denotes the perturbed instance generated by the $k-$step PGD with $k \le K$.

We defer the proof of this lemma to Appendix C Section C.1. The lemma provides an upper bound for the difference between the maximal loss $f(w,x^*,y)$ and the average of the losses achieved by K-step PGD (averaged over the K steps). If the achieved loss $f(w,x^k,y)$ increases over the K steps, the bound becomes

$$f(w,x^*,y) - f(w,x^K,y) \le \frac{(2C+d^*)}{2K} + \frac{d^*(\eta^2+\eta+1)}{2}$$

Notably this upper bound decays with K, but converges to a positive constant. This should come as no surprise since without stronger conditions or knowledge on f (e.g., concavity), it is hopeless to have PGD attacks to reach the true maximal loss value $f(w,x^*,y)$.

If we further assume loss functions $f(w,x,y)$ to be concave in x and consider the "raw-gradient (RG)"-PGD where the mapping G is taken as the identity map, we have the following convergence upper bound for PGD by directly adapting the Theorem 3.7 in [12]:

Lemma 5.5 (Convergence of RG-PGD with concave functions)**.** *Suppose that $f(w,x,y)$ satisfies the condition (5.37) and is concave in x. Let the mapping G in (5.6) be the identity map. Then the $K-$ step PGD (5.7) with step size $\lambda = \frac{1}{\eta}$ satisfies*

$$f(w, x^*, y) - f(w, x^K, y) \leq \frac{3\eta\|x - x^*\|^2 + f(w, x^*, y) - f(w, x, y)}{K} \tag{5.42}$$

where $x^* = J^*(x; y, w)$ *and* $x^K := T^K_{x,y}(x; w)$.

The bound obviously vanishes with K.

Trade-off between robustness and generalization We here rewrite the notation in (5.7) as

$$\pi^{\mathrm{PGD}}_K(x; y, w) := T^K_{x,y}(x; w) \tag{5.43}$$

to emphasize its dependency on K in the PGD attack.

We define the *(expected) robustness gap (on training set)* as

$$\mathrm{RG}(J^*, \pi) := \mathbb{E}_{S,A}\left[R_S[A_\pi(S), J^*] - R_S[A_\pi(S), \pi]\right] \tag{5.44}$$

This term characterizes the robustness of a model on the training set against J^* when it is trained by AT using some other adversarial perturbation π.

For shorter notation, let $w = A_\pi(S)$ and consider $\mathrm{RG}(J^*, \pi^{\mathrm{PGD}}_K)$. We have

$$\begin{aligned}
&\mathrm{RG}(J^*, \pi^{\mathrm{PGD}}_K) \\
=&\mathbb{E}_{S,A}\left[\frac{1}{n}\sum_{i=1}^n f(w, J^*(x_i; y_i, w), y_i) - f(w, \pi^{\mathrm{PGD}}_K(x_i; y_i, w), y_i)\right] && (5.45)\\
\leq& \sup_{(x,y,w)}\left[f(w, J^*(x; y, w), y) - f(w, \pi^{\mathrm{PGD}}_K(x; y, w), y)\right] && (5.46)\\
=& \sup_{(x,y,w)}\left[f(w, x^*, y) - f(w, x^K, y)\right] && (5.47)
\end{aligned}$$

where $x^* = J^*(x; y, w)$ and $x^K := \pi^{\mathrm{PGD}}_K(x; y, w)$. Since the results in Lemma 5.4 and 5.5 apply for arbitrary choice of (w, x, y), they suggest that a smaller robustness gap $\mathrm{RG}(J^*, \pi)$ can be achieved for π^{PGD}_K with larger K.

However, Lemma 5.3 on the other hand suggests that π^{PGD}_K with smaller K tends to achieve a smaller expansiveness parameter $q_c(\pi^{\mathrm{PGD}}_K)$ and therefore the corresponding generalization gap $\mathrm{GG}_n(J^*, A_\pi)$ with $\pi = \pi^{\mathrm{PGD}}_K$ tends to be smaller for smaller K. This suggests a potential trade-off between generalization and the robustness (measured by $\mathrm{RG}(J^*, \pi^{\mathrm{PGD}}_K)$).

5.6.1 Experiments

To investigate how the expansiveness property affects generalization, we consider a smooth approximation of the sign function by a tanh function, i.e., $\mathrm{sgn}(x) \approx \tanh_\gamma(x) := \tanh(\gamma x)$. Notably, the approximation error here vanishes with increase γ. By replacing $\mathrm{sgn}(x)$ in PGD AT with $\tanh_\gamma(x)$, we may control the expansiveness of π^{PGD}.

We conduct experiment for PGD-AT when G is chosen as $\tanh_\gamma$ as well as the identity map. Specially, for π^{PGD} with different choice of G, we refer to it as "sign-PGD" when $G(x) = \mathrm{sgn}(x)$, as "$\tanh_\gamma$-PGD" when $G(x) = \tanh_\gamma(x)$ and as "raw gradient (RG)-PGD" when $G(x) = x$. In all the experiments, we primarily consider the J-loss defined in (5.1) as our evaluation metric, with the loss function in f taken as the 0-1 loss and refer to this metric as J-(0-1) loss. We mainly use J from $\{\tanh_\gamma\text{-PGD}, \text{sign-PGD}, J^{\mathrm{id}}\}$. The experiments are conducted on CIFAR-10, CIFAR-100 [47] and SVHN [66]. Our experimental setting is elaborated in Appendix C.2, which follows from the setting in [78].

Figure 5.2 (a) presents the results of experiments conducted on CIFAR10, where the models are trained using $\tanh_\gamma$-PGD AT (i.e., A_π with $\pi = \tanh_\gamma$-PGD) with various γ values. Each model is trained for 200 epochs and is evaluated using the J-(0-1) loss for $J \in \{\tanh_\gamma\text{-PGD}, \text{sign-PGD}, J^{\mathrm{id}}\}$ (distinguished by colors), where γ matches the corresponding value in π. We use star-shaped dots and circle-shaped dots to respectively denote the J-(0-1) loss measured on the training set and the testing set. The gaps between each pairs of curves in the same color category then represents the generalization gap of the trained models evaluated by different J-(0-1) loss.

Smaller expansiveness results in reduced generalization gap By decreasing γ in π, the generalization gaps reduce, as shown by the narrowing gaps across all pairs of the curves in the same color. The observed experimental results demonstrate that AT with less expansive π tends to achieve a smaller generalization gap, consistent with the generalization bound of (5.26). Similar trends are also observed on SVHN and CIFAR100 (see Appendix C.4 Figure C.1).

Sign-PGD appears to be a stronger perturbation Due to the mismatch between π and J, the model trained by the algorithm A_π may still have a large empirical risk $\mathbb{E}[R_S[A_\pi(S), J]]$, which in turn results in a high population risk $\mathbb{E}[R_{\mathcal{D}}[A_\pi(S), J]]$ even if the generalization gap $\mathrm{GG}_n(J, A_\pi)$ is small. This is illustrated in Figure 5.2 (a) as the blue star-shaped curve consistently stays higher than the green star-shaped curve with a notably large gap. As γ increases, the $\tanh_\gamma$ function gradually approaches the sign

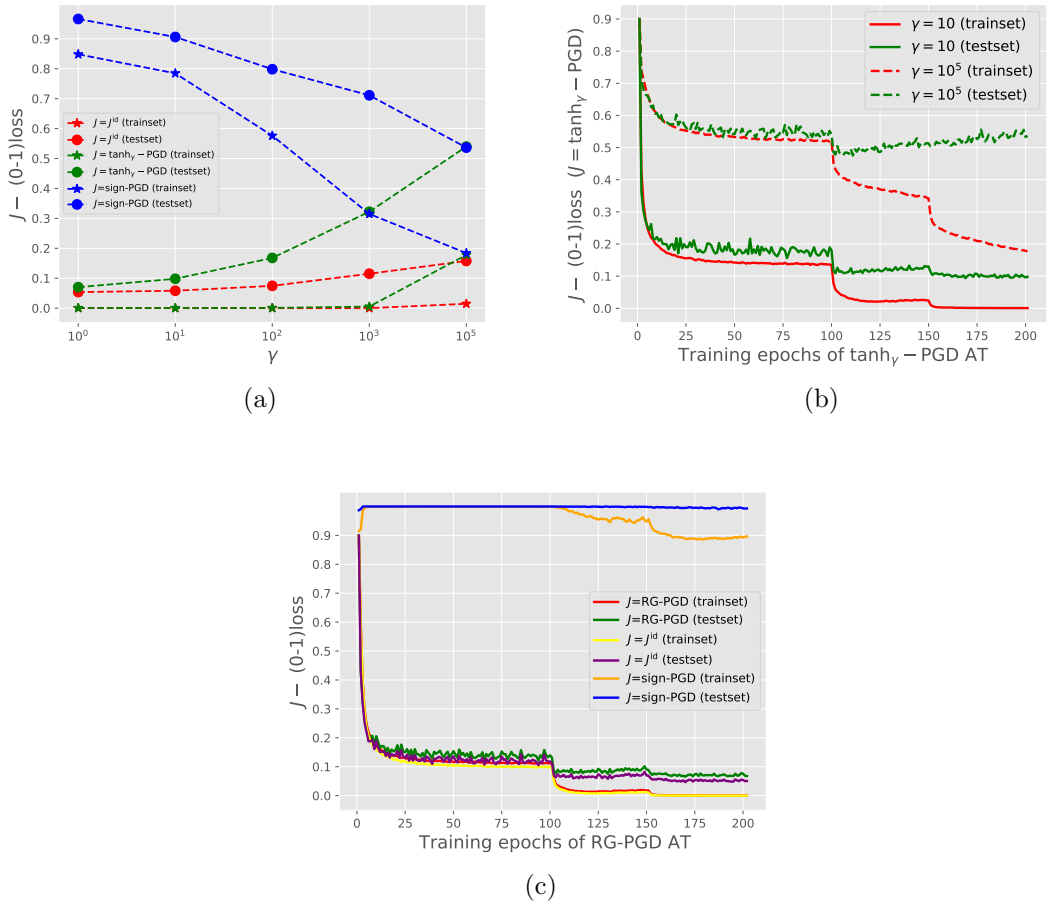


Figure 5.2: Experiments on CIFAR-10. (a) Models trained with $\tanh_\gamma$-PGD AT with different γ and evaluated by J-(0-1) loss on the training and testing set. (b) J-(0-1) loss with $J = \tanh_\gamma$-PGD measured along the training trajectories of two sets of $\tanh_\gamma$-PGD AT. (c) J-(0-1) loss measured along the trajectory of the RG-PGD AT with different choice of J.

function, leading to an intersection of the green and the blue curves. This indicates that sign-PGD is a stronger perturbation compare to the $\tanh_\gamma$-PGD, as the model trained with $\tanh_\gamma$-PGD AT can still be vulnerable to the sign-PGD attack on the training set.

Impact of AT on standard generalization The seminal work by [94] found that AT can negatively impact standard generalization. They constructed specific data models to demonstrate that achieving robustness and standard generalization can be inherently conflicting, suggesting an unavoidable trade-off between these two goals. This phenomenon has been extensively studied in subsequent research [42, 71, 77, 107, 115, 117]. Our experimental results offer further insights into this phenomenon from the perspective of algorithmic stability. Specifically, we find that the decline in standard generalization performance caused by AT can be attributed to the poor expansiveness condition of the sign-PGD method employed in AT. As shown by the trend of the red circle-shaped curve in Figure 5.2 (a), AT does not always harm standard generalization; a reduction in the J^{id}-(0-1) loss is observed as γ decreases. This suggests that the trade-off identified by [94] might be a side effect of the sign-PGD AT and is not necessarily unavoidable.

Generalization performance along training trajectories Figure 5.2 (b) plots the J-(0-1) loss with $J = \pi$ evaluated along the trajectory of the $\tanh_\gamma$-PGD AT with $\gamma = 10$ (the solid curves) and $\gamma = 10^5$ (the dashed curves). The dashed curves exhibit a phenomenon similar to robust overfitting observed in [78]: after the first learning rate decay (the 100^{th} epoch), as the training loss continuously decreases, the testing loss starts to *elevate*. This phenomenon does not appear in the AT with $\gamma = 10$, as shown in the trend of the solid curves. We conduct additional experiments for RG-PGD AT. As shown in Figure 5.2 (c), the generalization gap remains small across all groups of J-(0-1) loss throughout the training. Similar to the previous results, the model trained by this AT variant exhibits notable vulnerability to the sign-PGD perturbation, as indicated by the consistently high values of the orange and blue curves.

These findings demonstrate that removing or altering the sign function in PGD leads to a non-negligible influence on both robust generalization and resistance to perturbations on the training set. This highlights the crucial role of the sign function in PGD-AT, which deserves a more careful and further in-depth investigation.

5.7 Revisit of sign function in PGD

For simplicity, we write $f(w, x, y)$ as $f(x)$ in this section. The sign-PGD perturbation can be treated as an iterative optimization algorithm for solving the constrained optimization problem $\max_{\hat{x}\in\mathbb{B}_\infty(x,\epsilon)} f(\hat{x})$. It is related to the sign gradient methods, which has been used for different purposes, such as for training neural networks (e.g., [79]) and for gradient compression (e.g., [8]).

We now show that the sign gradient method can be viewed as a Steepest Descend (or ascend in our context) Method (SDM) w.r.t a $\infty-$norm ball (e.g., see Chapter 9.4 in [10]). Specifically, for the loss $f(x^k)$ at the k^{th} iteration in SDM, it updates x^k by finding a steepest ascend direction v within a small neighborhood of x^k such that the loss $f(x^{k+1})$ with $x^{k+1} = x^k + v$ is locally maximized. Such a neighborhood can be chosen as a $p-$norm ball around x^k (i.e., $\mathbb{B}_p(x^k, \lambda_p)$) with a small radius λ_p. Finding v introduces a new optimization problem: $\max_{v\in\mathbb{B}_p(x^k,\lambda_p)} f(x^k+v)$, which is then approximately solved by replacing $f(x^k+v)$ with its linear approximation around x^k, namely, solving $\max_{v\in\mathbb{B}_p(x^k,\lambda_p)} f(x^k) + \nabla f(x^k)^T v$ which is equivalent to solving $\max_{v\in\mathbb{B}_p(x^k,\lambda_p)} \nabla f(x^k)^T v$ whose closed form solution is

$$v^* = \lambda_p G_p(\nabla f(x^k)), \quad \text{where} \quad G_p(\nabla f(x^k)) := \frac{\text{sgn}(\nabla f(x^k)) \odot |\nabla f(x^k)|^{q-1}}{\|\nabla f(x^k)\|_q^{q-1}} \tag{5.48}$$

where we require $1/p + 1/q = 1$. The operator $\odot$ denotes the element-wise product. The closed form (5.48) then gives the following updating rule of SDM as

$$x^{k+1} = x^k + \lambda G_p(\nabla f(x^k)) \tag{5.49}$$

As a special case, when $p = 1$ with $q = \infty$, SMD turns into the coordinate gradient method with $G_1(\nabla f(x^k)) = \text{sgn}(\max_i \nabla f(x^k)[i])e_i$ and $i = \arg\max_j |\nabla f(x^k)[j]|$, where e_i denotes the standard basis vector. When $p = q = 2$, we have $G_2(\nabla f(x^k)) = \nabla f(x^k)/\|\nabla f(x^k)\|_2$

When $p = \infty$ with $q = 1$, the mapping G_∞ reduces to the sign function, indicating that the sign-PGD is indeed a (projected) SDM w.r.t $\mathbb{B}_\infty(x^k, \lambda_\infty)$. It is then curious to investigate *the generalization performance of the model trained by AT using the G_p-PGD with $p \neq \infty$.*[1].

We conduct experiments for the G_p-PGD-based AT following the same experimental setting as in the previous section, except that λ_p is adjusted to maintain the same volume of the balls $\mathbb{B}_p(0, \lambda_p)$ across different p values (details in Appendix C.3).

[1]Note that in the G_p-PGD we still consider projecting onto $\mathbb{B}_\infty(x, \epsilon)$ when p is taken other than ∞.

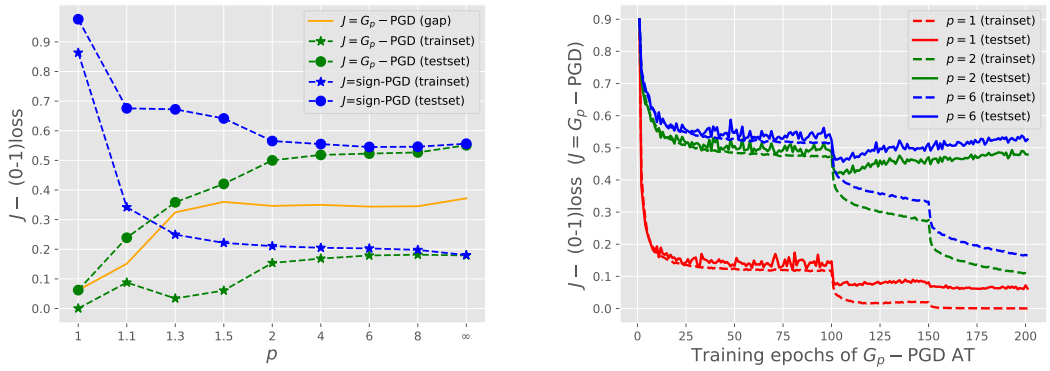


Figure 5.3: Experiments for G_p-PGD AT: (a) Model trained with various p values and evaluated by J-(0-1) loss with $J = \pi$ and $J =$ sign-PGD. (b) Training curves of the AT with various p values. (c) Standard generalization performance of the models trained by the AT, where the green curves are copied from (a) for a clearer presentation.

Larger p results in larger generalization gap Figure 5.3 (a) presents the experimental results on CIFAR10 (results on the other datasets are in Appendix C.4 Figure C.2 and C.3). The models are trained by A_π with $\pi = G_p$-PGD for various p and are evaluated by the J-(0-1) loss with $J = \pi$ (green curves) as well as $J =$sign-PGD (blue curves). The yellow curve represents the generalization gap for models trained with G_p-PGD. As shown, a larger p tends to result in larger generalization gaps. Indeed, nearly all G_p-PGD with $p \geq 1.3$ cause notably overfitting in AT with generalization gaps exceeding 30%.

The consistently higher position of blue star-shaped curves over the green star-shaped curve also suggests that sign-PGD is the strongest perturbations among the G_p-PGD.

Generalization performance along training trajectory Figure 5.3 (b) further exhibits the overfitting in G_p-PGD AT by plotting training curves for $p = \{1, 2, 6\}$, where continued training causes a rise of the testing errors (the blue and green curves), in contrast with the red curves, which demonstrate a good generalization.

Impact of p on standard generalization Figure 5.3 (c) shows how the G_p-PGD AT affect standard generalization where the red curves deontes the J-(0-1) loss with $J = J^{\text{id}}$ and the green curves are copied from Figure 5.3 (a) for a clearer comparison. An enlarging standard generalization gap is also observed in G_p-PGD AT with larger p.

The observed overfitting caused by the G_p-PGD family is potentially attributed to that nearly all the members in $\{G_p : p \in [1, \infty]\}$ have a poor Lipschitzness, as shown in the following lemma, which leads to a bad expansiveness of G_p-PGD.

Lemma 5.6. *Consider the mapping $G_p : \mathbb{R}^d \to \mathbb{R}^d$ specified in (5.48) with $p \in [1, \infty]$. Let $\mathcal{I} := \{1, \cdots, d\}$. If G_p is α_p-Lipschitz over the set $H(r) \subseteq \mathbb{R}^d$ with $H(r) := \{x \in \mathbb{R}^d : \min_{i \in \mathcal{I}} |x[i]| \geq r\}$ for some $r > 0$, then we have*

$$\alpha_p \geq \frac{1}{r d^{\frac{1}{p}}} \tag{5.50}$$

We defer the proof in Appendix C.1. This lower bound also implies that α_p is unbounded in $\mathbb{R}^d$, noting that the lower bound approaches infinity as $r \to 0$. Except for this extreme case, it is reasonable to assume that the gradients $\nabla_x f(x)$ lies in a set $H(r)$ with sufficiently small r where all the members in $\{G_p : p \in [1, \infty]\}$ have a bounded but large Lipschitz constant. Noteworthy, the lower bound increases, as p ranges from 1 to infinity, suggesting that the increased generalization gap in Figure 5.3 (a) is attributed to the increasing expansiveness of G_p-PGD caused by the rise in α_p.

5.8 Concluding Remarks

In this chapter, we carry out a stability analysis and present novel generalization bounds for models trained using AT with an arbitrary adversarial perturbation π and evaluated on a loss induced by an arbitrary perturbation J. At the heart of our analysis is the introduction of a notion of "expansiveness" for the perturbation maps (J and π), which governs the behavior of the derived bounds. Specifically, we show that whenever the expansiveness parameter of J is strictly bounded, our generalization bounds vanish with sample size n as $\mathcal{O}(1/n)$ and a small expansiveness parameter of π further helps generalization. On the other hand, when the J-loss (i.e., the loss induced by perturbation J) is defined with J taken as the sign-PGD perturbation, the expansiveness parameter of J is no longer bounded. In this case, our bound reveals an intrinsic tension between the stability parameter, and the perturbation radius, and the ambient data dimension, in their respective roles on generalization – specifically, the bound converges to a constant. When considering $\pi = J$, this helps to explain the robust overfitting phenomenon of sign-PGD AT as shown in Figure 5.1. Additional advantages of our bounds include the following. Our generic bound (Theorem 5.12) is applicable to AT algorithms based on any form of adversarial perturbations. Our bounds do not rely on any assumption on the adversarial loss directly, since we only make assumptions on the standard loss and all properties of the adversarial loss are induced via perturbation map J. Finally, varying the form of J potentially enables this framework to be applicable to settings where generalization on other performance metrics is of interest.

We zoom into models trained with multi-step PGD, and further demonstrate that the sign function used in the perturbation is an important cause of robust overfitting for such AT methods. We experimentally replace the sign function in PGD with a smooth approximation $\tanh_\gamma$, where $\tanh_\gamma(x) = \tanh(\gamma x)$ and the parameter γ controls the smoothness of the function and hence the expansiveness of the PGD perturbation (decreasing γ decreases the expansiveness). Our experiments show that reducing γ results in smaller generalization gaps. These results validate our bound and its implication on generalization. Interestingly our experiments also reveal that sign-PGD appears as a stronger attack than $\tanh_\gamma$-PGD and the raw gradient (RG)-PGD attack, even on the training set. Performing AT with $\tanh_\gamma$-PGD and RG-PGD may be inadequate for defending against the sign-PGD attacks on the training set. Our observations suggest that sign-function, a building block of PGD-based AT, appears to play a peculiar role: comparing with the $\tanh_\gamma$ counter-part, the sign function helps to better solve the inner maximization problem but at the same time cause the perturbation π to suffer from bad expansiveness and results in poor generalization. This aspect of sign-PGD has been largely overlooked to date, since most theoretical

analysis of PGD removes the sign function in their consideration (i.e., studying RG-PGD instead).

In this work, we also recognize sign-PGD as an iterative method for solving the inner maximization problem where each step is principled by a locally linear approximation of the loss function. Based on this principle, we extend sign-PGD to a wider family of perturbations. We show theoretically that every member in this family suffers from poor expansiveness. This result seems to point to certain intrinsic difficulty in training models adversarially.

Chapter 6

Summary and Future Works

6.1 Summary

In this thesis, we investigate the adversarial robustness of deep learning models, with a primary focus on understanding the robust generalization behavior of deep neural networks (DNNs) trained via adversarial training (AT). Initially, in Chapter 3, we analyze the robust generalization of linear classifiers as a foundational step, connecting their performance with classical notions such as Rademacher complexity.

We then extend our analysis to more complex deep learning models in Chapter 4, exploring their training dynamics under AT. Our findings highlight the critical role played by the perturbation operator in AT. Specifically, we characterize the relationship between the perturbation operator and the learning difficulty on the induced distributions by deriving a novel generalization bound and by conducting extensive experiments, demonstrating that the local dispersion of the perturbation operator significantly impacts the generalization performance of models learned on the induced distribution. Additionally, our experiments reveal a clear correlation between the evolution of the induced distribution along AT and the robust generalization performance resulted by AT.

In Chapter 5, we further deepen our understanding of AT dynamics through algorithmic stability analysis. Here, we propose a novel framework that considers generalization with respect to perturbation-induced losses, explicitly distinguishing between the perturbations used during training and those used during evaluation. Within this framework, we derive improved generalization bounds and conduct comprehensive experiments that underscore the importance of the expansiveness property of perturbation operators, showing that it substantially influences the generalization of models trained by AT.

6.2 Limitations

The main limitation of this work is that in most theoretical analysis, we have only developed an upper bound for the generalization of AT algorithms. Like all theoretical results based on upper bounds, they are adequate for understanding performance guarantees but may be inadequate to explain poor generalization. Nonetheless, our experimental results have suggested that our upper bound may well explain robust overfitting.

Furthermore, in Chapter 4, while we establish a connection between local dispersion and the learning difficulty of the induced data distribution, the theoretical framework does not fully explain the underlying causes of increased local dispersion during AT. Nor does it provide improved AT algorithms based on the theoretical insights.

6.3 Future Directions

Several promising directions remain open for future exploration:

- **Exploring Connections between Local Dispersion and Expansiveness:** While local dispersion and expansiveness measure the sensitivity of the perturbation operators from different perspectives —local dispersion focusing on sensitivity with respect to data inputs and expansiveness concerning model parameters —they share conceptual similarities. Future work could aim to thoroughly investigate and clarify the mathematical and conceptual relationships between these two notions, potentially leading to a unified analytical framework.

- **Designing Improved Perturbation Operators for AT:** A meaningful direction for future research involves developing perturbation operators that not only efficiently solve the "inner maximization" problem of AT but also exhibit low expansiveness parameters. Integrating such perturbation operators into AT could achieve improved robust generalization performance.

- **Investigating "Forgetting" in AT:** Each iteration of AT generates new adversarial examples based on the current model parameters. A natural question arises as to whether models trained via AT progressively "forget" adversarial examples generated during the early training phases. Understanding this phenomenon could lead to new strategies for stabilizing and improving adversarial training methods.

- **Developing New AT Schemes:** An intriguing future direction is exploring alternative training paradigms, such as initially training models with standard (non-adversarial) objectives and subsequently fine-tuning with adversarial perturbations. Inspired by the prevalent foundation model fine-tuning paradigm, this strategy could be enriched further by integrating the low-rank adaptation (LoRA) [40] techniques, exploring the potential existence of robust models in the low rank subspaces around the standard-trained models.

References

[1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

[2] Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. *CoRR*, abs/1810.02180, 2018.

[3] Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. *CoRR*, abs/2004.13617, 2020.

[4] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness, 2021.

[5] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[6] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33:4381–4391, 2020.

[7] Amine Bennouna, Ryan Lucas, and Bart Van Parys. Certified robust neural networks: Generalization and corruption resistance. *arXiv preprint arXiv:2303.02251*, 2023.

[8] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[9] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[10] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[12] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[13] Tuan Anh Bui, Trung Le, Quan Tran, He Zhao, and Dinh Phung. A unified wasserstein distributional robustness framework for adversarial training. *arXiv preprint arXiv:2202.13437*, 2022.

[14] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

[15] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2024.

[16] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021.

[17] Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

[18] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide deep learning for recommender systems, 2016.

[19] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[20] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[21] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of evasion adversaries, 2018.

[22] Zhun Deng, Hangfeng He, Jiaoyang Huang, and Weijie Su. Towards understanding the dynamics of the first-order adversaries. In *International Conference on Machine Learning*, pages 2484–2493. PMLR, 2020.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[24] Dimitrios I. Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Lower bounds for adversarially robust PAC learning. *CoRR*, abs/1906.05815, 2019.

[25] Chengyu Dong, Liyuan Liu, and Jingbo Shang. Double descent in adversarial training: An implicit label noise perspective. *CoRR*, abs/2110.03135, 2021.

[26] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. *CoRR*, abs/2106.01606, 2021.

[27] Yinpeng Dong, Ke Xu, Xiao Yang, Tianyu Pang, Zhijie Deng, Hang Su, and Jun Zhu. Exploring memorization in adversarial training. *arXiv preprint arXiv:2106.01606*, 2021.

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

[29] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2018.

[30] Farzan Farnia and Asuman Ozdaglar. Train simultaneously, generalize better: Stability of gradient-based minimax learners. In *International Conference on Machine Learning*, pages 3174–3185. PMLR, 2021.

[31] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier, 2018.

[32] Shaopeng Fu and Di Wang. Theoretical analysis of robust overfitting for wide dnns: An ntk approach. *arXiv preprint arXiv:2310.06112*, 2023.

[33] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[34] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[35] Muhammad Zaid Hameed and Beat Buesser. Boundary adversarial examples against adversarial overfitting, 2022.

[36] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.

[39] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering, 2017.

[40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[41] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019.

[42] Adel Javanmard, Mahdi Soltanolkotabi, and Hamed Hassani. Precise tradeoffs in adversarial training for linear regression. In *Conference on Learning Theory*, pages 2034–2078. PMLR, 2020.

[43] Sekitoshi Kanai, Masanori Yamada, Hiroshi Takahashi, Yuki Yamanaka, and Yasutoshi Ida. Relationship between nonsmoothness in adversarial training, constraints of attacks, and flatness in the input space. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. accepted.

[44] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[45] Justin Khim and Po-Ling Loh. Adversarial risk bounds via function transformation, 2019.

[46] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

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

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[49] Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. Wasserstein distributionally robust optimization: Theory and applications in machine learning. In *Operations research & management science in the age of analytics*, pages 130–166. Informs, 2019.

[50] Soichiro Kumano, Hiroshi Kera, and Toshihiko Yamasaki. Theoretical understanding of learning from adversarial perturbations, 2024.

[51] Soichiro Kumano, Hiroshi Kera, and Toshihiko Yamasaki. Wide two-layer networks can learn from adversarial perturbations, 2025.

[52] Ilja Kuzborskij and Christoph Lampert. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 2815–2824. PMLR, 2018.

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

[54] Yunwen Lei, Zhenhuan Yang, Tianbao Yang, and Yiming Ying. Stability and generalization of stochastic gradient methods for minimax problems. In *International Conference on Machine Learning*, pages 6175–6186. PMLR, 2021.

[55] Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pages 5809–5819. PMLR, 2020.

[56] Binghui Li, Jikai Jin, Han Zhong, John Hopcroft, and Liwei Wang. Why robust generalization in deep learning is difficult: Perspective of expressive power. *Advances in Neural Information Processing Systems*, 35:4370–4384, 2022.

[57] Yan Li, Ethan X. Fang, Huan Xu, and Tuo Zhao. Inductive bias of gradient descent based adversarial training on separable data. *CoRR*, abs/1906.02931, 2019.

[58] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[59] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

[60] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality, 2018.

[61] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[62] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, MA, 2 edition, 2018.

[63] Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. *CoRR*, abs/1902.04217, 2019.

[64] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[65] Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, pages 605–638. PMLR, 2018.

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

[67] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[68] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[69] Anh Nguyen, Jason Yosinski, and Jeff 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*, pages 427–436, 2015.

[70] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning,

Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[71] Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, and Shuicheng Yan. Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, pages 17258–17277. PMLR, 2022.

[72] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, 2016.

[73] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings, 2015.

[74] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[75] Zhuang Qian, Kaizhu Huang, Qiu-Feng Wang, and Xu-Yao Zhang. A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies, 2022.

[76] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in neural information processing systems*, 32, 2019.

[77] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. *arXiv preprint arXiv:2002.10716*, 2020.

[78] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. *CoRR*, abs/2002.11569, 2020.

[79] Martin Riedmiller and Heinrich Braun. Rprop: a fast adaptive learning algorithm. In *Proc. of the Int. Symposium on Computer and Information Science VII*, 1992.

[80] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[81] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *CoRR*, abs/1804.11285, 2018.

[82] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019.

[83] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

[84] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[85] Vasu Singla, Sahil Singla, David Jacobs, and Soheil Feizi. Low curvature activations reduce overfitting in adversarial training. *CoRR*, abs/2102.07861, 2021.

[86] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training, 2020.

[87] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[88] Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*, volume 3, page 4, 2017.

[89] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. *CoRR*, abs/2104.04448, 2021.

[90] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[91] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[92] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[93] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples, 2017.

[94] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[95] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019.

[96] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[97] Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, Cambridge, UK, 2019.

[98] Yihan Wang, Shuang Liu, and Xiao-Shan Gao. Data-dependent stability analysis of adversarial training. *arXiv preprint arXiv:2401.03156*, 2024.

[99] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. *arXiv preprint arXiv:2112.08304*, 2021.

[100] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail?, 2023.

[101] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[102] Dongxian Wu, Yisen Wang, and Shutao Xia. Revisiting loss landscape for adversarial robustness. *CoRR*, abs/2004.05884, 2020.

[103] Jiancong Xiao, Yanbo Fan, Ruoyu Sun, and Zhi-Quan Luo. Adversarial rademacher complexity of deep neural networks, 2022.

[104] Jiancong Xiao, Yanbo Fan, Ruoyu Sun, Jue Wang, and Zhi-Quan Luo. Stability analysis and generalization bounds of adversarial training. *Advances in Neural Information Processing Systems*, 35:15446–15459, 2022.

[105] Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[106] Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. *Advances in neural information processing systems*, 34:26523–26535, 2021.

[107] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. Adversarial robustness through local lipschitzness. *CoRR*, abs/2003.02460, 2020.

[108] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness, 2020.

[109] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey, 2024.

[110] Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher complexity for adversarially robust generalization. *CoRR*, abs/1810.11914, 2018.

[111] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond, 2022.

[112] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 2019.

[113] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[114] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

[115] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[116] Huishuai Zhang, Da Yu, Yiping Lu, and Di He. Adversarial noises are linearly separable for (nearly) random neural networks, 2022.

[117] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. *CoRR*, abs/2002.11242, 2020.

[118] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.

APPENDICES

Appendix A

Omitted proofs in Chapter 3

Proof of Proposition 3.3.

Proof. We have that

$$
\begin{aligned}
&\max_{\hat{x}\in\mathbb{B}_p(x,\epsilon)} \mathbb{I}(yh_w(\hat{x}) < 0) \\
=&\mathbb{I}\left(\min_{\|\delta\|_p\leq\epsilon} yh_w(x+\delta) < 0\right) && \text{(A.1)}\\
\leq&\tilde{\phi}\left(\min_{\|\delta\|_p\leq\epsilon} yh_w(x+\delta)\right) && \text{(A.2)}\\
=&\tilde{\phi}(yw^Tx - \epsilon\|w\|_q) && \text{(A.3)}\\
\leq&\tilde{\phi}(yw^Tx) + \epsilon\|w\|_q && \text{(A.4)}\\
=&\tilde{\phi}(yh_w(x)) + \epsilon\|w\|_q && \text{(A.5)}
\end{aligned}
$$

Let $f(x,y;w) := \tilde{\phi}(yh_w(x)) + \epsilon\|w\|_q$ and denote $\tilde{\mathcal{H}} := \{f(x,y;w) : h_w \in \mathcal{H}\}$ for $\mathcal{H} := \{h_w : \|w\|_q \leq B\}$. Since each $f \in \tilde{\mathcal{H}}$ is bounded within $[0, \epsilon B + 1]$, utilizing (A.5)

and adopting Theorem 3.2, we have that with probability $1-\eta$ over sampling S from $\mathcal{D}^n$,

$$
\begin{aligned}
R_{\mathcal{D}}^{\mathrm{adv}}(w) &:= \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\hat{x}\in\mathbb{B}_p(x,\epsilon)} \mathbb{I}(yh_w(\hat{x}) < 0) \\
&\leq \mathbb{E}_{(x,y)\sim\mathcal{D}} \tilde{\phi}(yh_w(x)) + \epsilon\|w\|_q && \text{(A.6)} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \tilde{\phi}(y_i h_w(x_i)) + \epsilon\|w\|_q + 2(\epsilon B+1)\mathrm{Rad}_n(\hat{\mathcal{H}}; S) + 3(\epsilon B+1)\sqrt{\frac{\log(2/\eta)}{2n}} && \text{(A.7)} \\
&\leq \frac{1}{n}\sum_{i=1}^{n} \phi(y_i h_w(x_i)) + \epsilon\|w\|_q + 2(\epsilon B+1)\mathrm{Rad}_n(\hat{\mathcal{H}}; S) + 3(\epsilon B+1)\sqrt{\frac{\log(2/\eta)}{2n}} && \text{(A.8)}
\end{aligned}
$$

where inequality (A.8) is due to that $\tilde{\phi}(z) \leq \phi(z)$ for any z. This completes the proof.

□

Proof of Theorem 3.4.

Proof. We have that

$$
\begin{aligned}
&\mathrm{Rad}_n(\tilde{\mathcal{H}}; S) && \text{(A.9)} \\
=&\mathbb{E}_\Sigma\left[\sup_{\|w\|_q\leq B} \frac{1}{n}\sum_{i=1}^{n} \sigma_i(\tilde{\phi}(y_i h_w(x_i)) + \epsilon\|w\|_q)\right] && \text{(A.10)} \\
\leq&\mathbb{E}_\Sigma\left[\sup_{\|w\|_q\leq B} \frac{1}{n}\sum_{i=1}^{n} \sigma_i\tilde{\phi}(y_i h_w(x_i))\right] + \mathbb{E}_\Sigma\left[\sup_{\|w\|_q\leq B} \frac{1}{n}\sum_{i=1}^{n} \sigma_i\epsilon\|w\|_q\right] && \text{(A.11)}
\end{aligned}
$$

where (A.10) follows by the definition of Rademacher complexity and (A.11) is derived by the fact that supremum of summations is smaller than summations of supremum.

For the first term in (A.11), we have that

$$
\begin{aligned}
&\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\phi}(y_i h_w(x_i))\right] \\
\leq&\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B}\frac{1}{n}\sum_{i=1}^{n}\sigma_i y_i h_w(x_i)\right] && \text{(A.12)}\\
=&\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h_w(x_i)\right] && \text{(A.13)}\\
=&\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B}\frac{1}{n}\sum_{i=1}^{n}\sigma_i w^T x_i\right] && \text{(A.14)}\\
=&\frac{1}{n}\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B} w^T\left(\sum_{i=1}^{n}\sigma_i x_i\right)\right] && \text{(A.15)}\\
\leq&\frac{1}{n}\mathbb{E}_{\Sigma}\left[\sup_{\|w\|_q\leq B}\|w\|_q\left\|\sum_{i=1}^{n}\sigma_i x_i\right\|_p\right] && \text{(A.16)}\\
=&\frac{B}{n}\mathbb{E}_{\Sigma}\left\|\sum_{i=1}^{n}\sigma_i x_i\right\|_p && \text{(A.17)}\\
\leq&\max\left(1,d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{B}{n}\mathbb{E}_{\Sigma}\left\|\sum_{i=1}^{n}\sigma_i x_i\right\|_2 && \text{(A.18)}\\
\leq&\max\left(1,d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{B}{n}\left(\mathbb{E}_{\Sigma}\left\|\sum_{i=1}^{n}\sigma_i x_i\right\|_2^2\right)^{\frac{1}{2}} && \text{(A.19)}\\
=&\max\left(1,d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{B}{n}\left(\mathbb{E}_{\Sigma}\left[\sum_{i,j=1}^{n}\sigma_i\sigma_j x_i^T x_j\right]\right)^{\frac{1}{2}} && \text{(A.20)}\\
=&\max\left(1,d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{B}{n}\left(\left[\sum_{i=1}^{n}\|x_i\|_2^2\right]\right)^{\frac{1}{2}} && \text{(A.21)}\\
\leq&\max\left(1,d^{\frac{1}{p}-\frac{1}{2}}\right)\frac{BR}{\sqrt{n}} && \text{(A.22)}
\end{aligned}
$$

where (A.12) is derived by utilizing the Talagrand's lemma (see Lemma 5.7 in [62])

and by the fact that $\tilde{\phi}$ is a 1-Lipschitz function. Equality (A.13) holds since each y_i takes value in $\{-1, +1\}$. (A.16) follows from the Hölder's inequality and (A.18) follows from the inequality relation between 2-norm and other $p-$norms. (A.19) is dervied by the Jensen's inequality and by the fact that square root is a concave function. (A.21) is due to that $\mathbb{E}[\sigma_i \sigma_j] = \mathbb{E}[\sigma_i]\mathbb{E}[\sigma_j] = 0$ and (A.22) is derived by the condition that $\|x_i\|_2 \leq R$.

For the second term in (A.11), we have that

$$
\begin{aligned}
&\mathbb{E}_\Sigma \left[\sup_{\|w\|_q \leq B} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \epsilon \|w\|_q \right] \\
=&\frac{\epsilon}{n} \mathbb{E}_\Sigma \left[\sup_{\|w\|_q \leq B} \|w\|_q \left(\sum_{i=1}^{n} \sigma_i \right) \right] && \text{(A.23)} \\
=&\frac{\epsilon}{n} \mathbb{E}_\Sigma \left[\frac{B}{2} \left(1 + \operatorname{sgn} \left(\sum_{i=1}^{n} \sigma_i \right) \right) \left(\sum_{i=1}^{n} \sigma_i \right) \right] && \text{(A.24)} \\
=&\frac{\epsilon B}{2n} \mathbb{E}_\Sigma \left[\left(\sum_{i=1}^{n} \sigma_i \right) \operatorname{sgn} \left(\sum_{i=1}^{n} \sigma_i \right) \right] && \text{(A.25)} \\
=&\frac{\epsilon B}{2n} \mathbb{E}_\Sigma \left[\left| \sum_{i=1}^{n} \sigma_i \right| \right] && \text{(A.26)} \\
\leq&\frac{\epsilon B}{2n} \sqrt{n} && \text{(A.27)} \\
=&\frac{\epsilon B}{2\sqrt{n}} && \text{(A.28)}
\end{aligned}
$$

where (A.25) is due to that $\mathbb{E}_\Sigma \left[\sum_{i=1}^{n} \sigma_i\right] = \sum_{i=1}^{n} \mathbb{E}[\sigma_i] = 0$ and (A.27) is derived by the Khintchine inequality. □

Appendix B

Omitted proofs and results in Chapter 4

B.1 Detailed Experimental setup

Our Reduced ImageNet is made by aggregating several semantically similar subsets of the original ImageNet, resulting in a total of 66594 images. This dataset is then partitioned into a training set containing 5,000 images per class and a testing set containing approximately 1,000 images per class. Compared to the restricted ImageNet in [80], our dataset has a more balanced sample size across each classes. Table B.1 illustrates the specific classes from the original ImageNet that have been aggregated in our dataset.

For adversarial training (AT), the settings on different datasets are summarized in Table B.2. Data augmentation is performed on these datasets except for MNIST during the training. For CIFAR-10 and CIFAR-100 we follow the data augmentation setting in [78]. For our reduced ImageNet, we adopt the same data augmentation scheme that is used on the restricted ImageNet in [107].

For the induced distribution experiments (IDEs) on each datasets, the settings are outlined in Table B.3. It is important to note that for each of the individual IDEs that is conducted on the same dataset, we maintain consistent training settings. This includes using the same model architecture with identical model size and the same level of regularization. This ensures a fair comparison of the IDE results obtained from the same dataset. Furthermore, the model is trained to achieve zero training error in all the IDEs, excluding the situation that the degeneration in model performance could be attributed to inadequate training procedures.

Classes in the reduced ImageNet	Classes in ImageNet
"dog"	86 to 90
"cat"	(8,10,55,95,174)
"truck"	279 to 283
"car"	272 to 276
"beetles"	623 to 627
"turtle"	458 to 462
"crab"	612 to 616
"fish"	450 to 454
"snake"	477 to 481
"spider"	604 to 608

Table B.1: The left column presents the classes within our reduced ImageNet dataset, with each class being an aggregation of the corresponding classes from the full-scale ImageNet dataset, as depicted in the right column.

	MNIST	CIFAR-10	CIFAR-100	Reduced ImageNet
model	small CNN	PRN18	WRN-34	PRN-50
optimizer	Adam	SGD	SGD	SGD
weight deacy	None	5×10^{-4}	5×10^{-4}	None
batch size	128	128	128	128
ϵ	0.3	8/255	8/255	4/255
PGD step size	0.01	2/255	2/255	0.9/255
number of PGD	40	10	10	5

Table B.2: Settings in PGD and AT across different datasets

	MNIST	CIFAR-10	CIFAR-100	Reduced ImageNet
model	small CNN	PRN-18	WRN-34	PRN-50
optimizer	Adam	SGD	SGD	SGD
weight deacy	None	5×10^{-4}	5×10^{-4}	5×10^{-4}
batch size	128	128	128	128

Table B.3: Settings in the IDE across different datasets

B.2 Proofs

B.2.1 Proof of (4.9)

We have that

$$\begin{aligned}
\tilde{\gamma}_\phi(x,y) &:= \mathbb{E}_{\rho,\rho'}\|\mathcal{Q}_\phi(x+\rho,y)-\mathcal{Q}_\phi(x+\rho',y)\|_2^2 \\
&= \mathbb{E}_{\rho,\rho'}\left(\mathcal{Q}_\phi(x+\rho,y)-\mathcal{Q}_\phi(x+\rho',y)\right)^T\left(\mathcal{Q}_\phi(x+\rho,y)-\mathcal{Q}_\phi(x+\rho',y)\right) && \text{(B.1)}\\
&= \mathbb{E}_{\rho,\rho'}\left[\|\mathcal{Q}_\phi(x+\rho,y)\|_2^2+\|\mathcal{Q}_\phi(x+\rho',y)\|_2^2-2\mathcal{Q}_\phi(x+\rho',y)^T\mathcal{Q}_\phi(x+\rho,y)\right] && \text{(B.2)}\\
&= 2\mathbb{E}_\rho\|\mathcal{Q}_\phi(x+\rho,y)\|_2^2-2\|\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)\|_2^2 && \text{(B.3)}
\end{aligned}$$

On the other hand, we have that

$$\begin{aligned}
\tilde{\gamma}_\phi(x,y) &:= \mathbb{E}_{\rho,\rho'}\|\mathcal{Q}_\phi(x+\rho,y)-\mathcal{Q}_\phi(x+\rho',y)\|_2^2 \\
&= \mathbb{E}_{\rho,\rho'}\|\mathcal{Q}_\phi(x+\rho,y)-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)-\left(\mathcal{Q}_\phi(x+\rho',y)-\mathbb{E}_{\rho'}\mathcal{Q}_\phi(x+\rho',y)\right)\|_2^2 && \text{(B.4)}\\
&= 2\mathbb{E}_\rho\|\mathcal{Q}_\phi(x+\rho,y)-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)\|_2^2-2\|\mathbb{E}_\rho\left[\mathcal{Q}_\phi(x+\rho,y)-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)\right]\|_2^2 && \text{(B.5)}\\
&= 2\mathbb{E}_\rho\|\mathcal{Q}_\phi(x+\rho,y)-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)\|_2^2 && \text{(B.6)}\\
&= 2\mathbb{E}_\rho\left[\sum_{i=1}^{d}\left(\mathcal{Q}_\phi(x+\rho,y)[i]-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)[i]\right)^2\right] && \text{(B.7)}\\
&= 2\sum_{i=1}^{d}\mathbb{E}_\rho\left(\mathcal{Q}_\phi(x+\rho,y)[i]-\mathbb{E}_\rho\mathcal{Q}_\phi(x+\rho,y)[i]\right)^2 && \text{(B.8)}\\
&= 2\text{Trace}\left(\text{COV}(\mathcal{Q}_\phi(x+\rho,y))\right) && \text{(B.9)}
\end{aligned}$$

where equality (B.5) is derived by applying the results of (B.3). We use $\mathcal{Q}_\phi(x+\rho,y)[i]$ to denote the i^{th} coordinate of the vector $\mathcal{Q}_\phi(x+\rho,y)$.

□

B.2.2 Proof of Lemma 4.1

With a little abuse of notation, let (t,y) denote an instance drawn from $\mathcal{D}_*$ and let (v,y) denote an instance drawn from the induced distribution $\tilde{\mathcal{D}}_\phi$ associate with a perturbation

$\mathcal{Q}_\phi$. For shorter notations, we will denote $z := (t, y)$, $u := (v, y)$ and $f(u) := f(v, y)$ and simply write $\mathcal{Q}_\phi$ as $\mathcal{Q}$.

Denote by $g(u_1 \cdots u_m) := \sup_{\theta\in\Theta} \mathrm{GG}(\theta; \tilde{S}_\phi, \tilde{\mathcal{D}}_\phi) = \sup_{\theta\in\Theta} \left| \frac{1}{m} \sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E} f_\theta(u) \right|$. We have for any $1 \le j \le m$

$$\sup_{u_1,\cdots,u_m,u_j'} \left| g(u_1, \cdots, u_m) - g(u_1, \cdots, u_j', u_{j+1}, \cdots u_m) \right| \tag{B.10}$$

$$= \sup_{u_1,\cdots,u_m,u_j'} \left| \sup_{\theta\in\Theta} \left| \frac{1}{m} \sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E} f_\theta(u) \right| - \sup_{\theta\in\Theta} \left| \frac{1}{m} \left(\sum_{i=1, i\neq j}^{m} f_\theta(u_i) + f_\theta(u_j') \right) - \mathbb{E}_u f_\theta(u) \right| \right| \tag{B.11}$$

$$\le \sup_{u_1,\cdots,u_m,u_j'} \sup_{\theta\in\Theta} \left| \left| \frac{1}{m} \sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E} f_\theta(u) \right| - \left| \frac{1}{m} \left(\sum_{i=1, i\neq j}^{m} f_\theta(u_i) + f_\theta(u_j') \right) - \mathbb{E}_u f_\theta(u) \right| \right| \tag{B.12}$$

$$\le \sup_{u_1,\cdots,u_m,u_j'} \sup_{\theta\in\Theta} \left| \frac{1}{m} \sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E}_u f_\theta(u) - \frac{1}{m} \left(\sum_{i=1, i\neq j}^{m} f_\theta(u_i) + f_\theta(u_j') \right) + \mathbb{E}_u f_\theta(u) \right| \tag{B.13}$$

$$= \sup_{\theta\in\Theta} \sup_{u_j, u_j'} \frac{1}{m} \left| f_\theta(u_j) - f_\theta(u_j') \right| \tag{B.14}$$

$$\le \frac{1}{m} \sup_{\theta\in\Theta} \sup_{u_j} |f_\theta(u_j)| + \frac{1}{m} \sup_{\theta\in\Theta} \sup_{u_j'} \left| f_\theta(u_j') \right| \tag{B.15}$$

$$\le \frac{2B}{m} \tag{B.16}$$

where the inequality (B.13) follows from the inverse triangle inequality. The inequality (B.15) and (B.16) make use of the triangle inequality and the boundedness condition of f.

With the result derived above, by McDiarmid inequality, we have for all $\mu > 0$

$$\Pr\left[g(u_1 \cdots u_m) - \mathbb{E}_U g(u_1 \cdots u_m) \ge \mu \right] \le \exp\left(\frac{-m\mu^2}{B} \right)$$

where we use $U := (u_1, \cdots, u_m)$. This is equivalent to saying that with probability $1 - \tau$, we have

$$g(u_1 \cdots u_m) \le \mathbb{E}_U g(u_1 \cdots u_m) + 2B \sqrt{\frac{\log \frac{1}{\tau}}{2m}} \tag{B.17}$$

□

B.2.3 Proof of Theorem 4.6

Following the notations in the proof of Lemma 4.1, we now derive an upper bound for the term $\mathbb{E}_U g(u_1 \cdots u_m)$.

For shorter notations, let $Z := (z_1, \cdots, z_m)$, $\Gamma := (\rho_1, \cdots, \rho_m)$ and $F_\theta(Z, \Gamma) := \frac{1}{m}\sum_{i=1}^{m} f_\theta(\mathcal{Q}(x_i + \rho_i, y_i), y_i)$. We have

$$\mathbb{E}_U g(u_1 \cdots u_m) \tag{B.18}$$

$$= \mathbb{E}_U \sup_{\theta\in\Theta} \left| \frac{1}{m}\sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E} f_\theta(u) \right| \tag{B.19}$$

$$= \mathbb{E}_U \sup_{\theta\in\Theta} \left| \frac{1}{m}\sum_{i=1}^{m} f_\theta(u_i) - \mathbb{E}_{\hat{U}} \left[\frac{1}{m}\sum_{i=1}^{m} f_\theta(\hat{u}_i) \right] \right| \tag{B.20}$$

$$\leq \mathbb{E}_U \sup_{\theta\in\Theta} \left[\mathbb{E}_{\hat{U}} \left| \frac{1}{m}\sum_{i=1}^{m} f_\theta(u_i) - \frac{1}{m}\sum_{i=1}^{m} f_\theta(\hat{u}_i) \right| \right] \tag{B.21}$$

$$\leq \mathbb{E}_U \mathbb{E}_{\hat{U}} \sup_{\theta\in\Theta} \left| \frac{1}{m}\sum_{i=1}^{m} f_\theta(u_i) - \frac{1}{m}\sum_{i=1}^{m} f_\theta(\hat{u}_i) \right| \tag{B.22}$$

$$= \mathbb{E}_Z \mathbb{E}_\Gamma \mathbb{E}_{\hat{Z}} \mathbb{E}_{\hat{\Gamma}} \sup_{\theta\in\Theta} \left| \frac{1}{m}\sum_{i=1}^{m} f_\theta(\mathcal{Q}(x_i + \rho_i, y_i), y_i) - \frac{1}{m}\sum_{i=1}^{m} f_\theta(\mathcal{Q}(\hat{x}_i + \hat{\rho}_i, \hat{y}_i), \hat{y}_i) \right| \tag{B.23}$$

$$= \mathbb{E}_Z \mathbb{E}_\Gamma \mathbb{E}_{\hat{Z}} \mathbb{E}_{\hat{\Gamma}} \sup_{\theta\in\Theta} \left| F_\theta(Z, \Gamma) - \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) + \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) - F_\theta(\hat{Z}, \hat{\Gamma}) + \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) - \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) \right| \tag{B.24}$$

$$\leq \mathbb{E}_Z \mathbb{E}_\Gamma \sup_{\theta\in\Theta} \left| F(Z, \Gamma) - \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) \right| + \mathbb{E}_{\hat{Z}} \mathbb{E}_{\hat{\Gamma}} \sup_{\theta\in\Theta} \left| F_\theta(\hat{Z}, \hat{\Gamma}) - \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) \right|$$

$$+ \mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta\in\Theta} \left| \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) - \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) \right| \tag{B.25}$$

$$= \underbrace{2\mathbb{E}_Z \mathbb{E}_\Gamma \sup_{\theta\in\Theta} \left| F_\theta(Z, \Gamma) - \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) \right|}_{①} + \underbrace{\mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta\in\Theta} \left| \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) - \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) \right|}_{②} \tag{B.26}$$

where (B.21) follows from Jensen's inequality and (B.22) is due to that the supremum of expectation is less than equal to expectation of the supremum. The inequality (B.25) is

derived by the triangle inequality and the fact that supremum of sum is less than equal to sum of supremum. We now individually construct upper bounds for the term ① and ②.

For the term ①, we have

$$
\begin{aligned}
&2\mathbb{E}_Z\mathbb{E}_\Gamma \sup_{\theta\in\Theta}\left|F_\theta(Z,\Gamma)-\mathbb{E}_{\bar\Gamma}F_\theta(Z,\bar\Gamma)\right| \\
\le&2\mathbb{E}_Z\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma} \sup_{\theta\in\Theta}\left|F_\theta(Z,\Gamma)-F_\theta(Z,\bar\Gamma)\right| && \text{(B.27)}\\
=&2\mathbb{E}_Z\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma} \sup_{\theta\in\Theta}\left|\frac{1}{m}\sum_{i=1}^m f_\theta(\mathcal{Q}(x_i+\rho_i,y_i),y_i)-\frac{1}{m}\sum_{i=1}^m f_\theta(\mathcal{Q}(x_i+\bar\rho_i,y_i),y_i)\right| && \text{(B.28)}\\
=&\frac{2}{m}\mathbb{E}_Z\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma} \sup_{\theta\in\Theta}\mathbb{E}_\Sigma\left|\sum_{i=1}^m \sigma_i\left(f_\theta(\mathcal{Q}(x_i+\rho_i,y_i),y_i)-f_\theta(\mathcal{Q}(x_i+\bar\rho_i,y_i),y_i)\right)\right| && \text{(B.29)}\\
\le&\frac{2}{m}\mathbb{E}_Z\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma} \sup_{\theta\in\Theta}\sqrt{\sum_{i=1}^m\left|f_\theta(\mathcal{Q}(x_i+\rho_i,y_i),y_i)-f_\theta(\mathcal{Q}(x_i+\bar\rho_i,y_i),y_i)\right|^2} && \text{(B.30)}\\
\le&\frac{2}{m}\mathbb{E}_Z\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma}\sqrt{\sum_{i=1}^m\beta^2\|\mathcal{Q}(x_i+\rho_i,y_i)-\mathcal{Q}(x_i+\bar\rho_i,y_i)\|^2} && \text{(B.31)}\\
\le&\frac{2\beta}{m}\mathbb{E}_Z\sqrt{\mathbb{E}_\Gamma\mathbb{E}_{\bar\Gamma}\left[\sum_{i=1}^m\|\mathcal{Q}(x_i+\rho_i,y_i)-\mathcal{Q}(x_i+\bar\rho_i,y_i)\|^2\right]} && \text{(B.32)}\\
=&\frac{2\beta}{m}\mathbb{E}_Z\sqrt{\sum_{i=1}^m\mathbb{E}_\rho\mathbb{E}_{\bar\rho}\|\mathcal{Q}(x_i+\rho_i,y_i)-\mathcal{Q}(x_i+\bar\rho_i,y_i)\|^2} && \text{(B.33)}\\
=&\frac{2\beta}{m}\mathbb{E}_Z\sqrt{\sum_{i=1}^m\gamma(x_i,y_i)} && \text{(B.34)}\\
\le&\frac{2\beta}{m}\sqrt{\mathbb{E}_Z\left[\sum_{i=1}^m\gamma(x_i,y_i)\right]} && \text{(B.35)}\\
=&\frac{2\beta}{m}\sqrt{\sum_{i=1}^m\mathbb{E}_{z_i}\gamma(x_i,y_i)} && \text{(B.36)}\\
=&\frac{2\beta}{\sqrt{m}}\sqrt{\mathbb{E}_z\gamma(x,y)} && \text{(B.37)}
\end{aligned}
$$

The inequality (B.27) is derived similarly to inequality (B.21) and (B.22). In (B.29), we introduce Rademacher variables $\Sigma := (\sigma_1, \cdots, \sigma_m)$ (i.e., each random variable σ_i takes values in $\{-1, +1\}$ independently with equal probability 0.5). The Rademacher variables introduces a random exchange of the corresponding difference term. Since Γ and $\hat{\Gamma}$ are independently sampled from the same distribution, such a swap gives an equally likely configuration. Therefore, the equality (B.29) holds. The inequality (B.30) is given by the Khintchine's inequality. The inequality (B.31) makes use of the Lipschitz condition of f. (B.32) is derived from Jensen's inequality and due to that square root is a concave function. (B.34) is by the definition of the local dispersion of $\mathcal{Q}$. Again, we apply Jensen's inequality to obtain (B.35). Equation (B.36) and (B.37) follow from the settings that each $z_i = (x_i, y_i)$ is i.i.d.

For the term ②, we have

$$
\begin{aligned}
&\mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta \in \Theta} \left| \mathbb{E}_{\bar{\Gamma}} F_\theta(Z, \bar{\Gamma}) - \mathbb{E}_{\tilde{\Gamma}} F_\theta(\hat{Z}, \tilde{\Gamma}) \right| \\
=&\mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta \in \Theta} \left| \mathbb{E}_{\bar{\Gamma}} \left[\frac{1}{m} \sum_{i=1}^{m} f_\theta(\mathcal{Q}(x_i + \bar{\rho}_i, y_i), y_i) \right] - \mathbb{E}_{\tilde{\Gamma}} \left[\frac{1}{m} \sum_{i=1}^{m} f_\theta(\mathcal{Q}(\hat{x}_i + \tilde{\rho}_i, \hat{y}_i), \hat{y}_i) \right] \right| && \text{(B.38)} \\
=&\mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta \in \Theta} \left| \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\bar{\rho}_i} \left[f_\theta(\mathcal{Q}(x_i + \bar{\rho}_i, y_i), y_i) \right] - \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\tilde{\rho}_i} \left[f_\theta(\mathcal{Q}(\hat{x}_i + \tilde{\rho}_i, \hat{y}_i), \hat{y}_i) \right] \right| && \text{(B.39)} \\
=&\mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta \in \Theta} \left| \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right] - \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right] \right| && \text{(B.40)} \\
=&\frac{1}{m} \mathbb{E}_Z \mathbb{E}_{\hat{Z}} \mathbb{E}_\Sigma \sup_{\theta \in \Theta} \left| \sum_{i=1}^{m} \sigma_i \left(\mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right] - \mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right] \right) \right| && \text{(B.41)} \\
\leq&\frac{1}{m} \mathbb{E}_Z \mathbb{E}_{\hat{Z}} \sup_{\theta \in \Theta} \sqrt{\sum_{i=1}^{m} \left| \left(\mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right] - \mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right] \right) \right|^2} && \text{(B.42)}
\end{aligned}
$$

where equation (B.39) and (B.40) are due to each $\hat{\rho}_i$ and $\tilde{\rho}_i$ is i.i.d. Again, we introduce Rademacher variables at (B.41) and apply Khintchine's inequality to get (B.42). For the term $\left| \left(\mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right] - \mathbb{E}_{\rho} \left[f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right] \right) \right|^2$, we have

$$
\begin{aligned}
&\left| \mathbb{E}_\rho f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) - \mathbb{E}_\rho f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right|^2 && \text{(B.43)} \\
\leq&\left(\left| \mathbb{E}_\rho f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right| + \left| \mathbb{E}_\rho f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right| \right)^2 && \text{(B.44)} \\
\leq&2 \left| \mathbb{E}_\rho f_\theta(\mathcal{Q}(x_i + \rho, y_i), y_i) \right|^2 + 2 \left| \mathbb{E}_\rho f_\theta(\mathcal{Q}(\hat{x}_i + \rho, \hat{y}_i), \hat{y}_i) \right|^2 && \text{(B.45)}
\end{aligned}
$$

where inequality (B.45) is derived by the inequality $(a+b)^2 \leq 2(a^2+b^2)$. We also have that

$$
\begin{aligned}
&|\mathbb{E}_\rho f_\theta(\mathcal{Q}(x+\rho,y),y)|^2 \\
\leq&(\mathbb{E}_\rho\,|f_\theta(\mathcal{Q}(x+\rho,y),y)-f_\theta(x+\rho,y)|+|f_\theta(x+\rho,y)|)^2 && \text{(B.46)}\\
\leq&(\mathbb{E}_\rho\,|f_\theta(\mathcal{Q}(x+\rho,y),y)-f_\theta(x+\rho,y)|+B)^2 && \text{(B.47)}\\
\leq&(\mathbb{E}_\rho\beta\|\mathcal{Q}(x+\rho,y)-(x+\rho)\|_2+B)^2 && \text{(B.48)}
\end{aligned}
$$

The inequalities (B.46)-(B.48) respectively make use of the triangle inequality, Jensen's inequality, and the boundedness and lipschitz condition of f.

Returning to (B.42), we then have

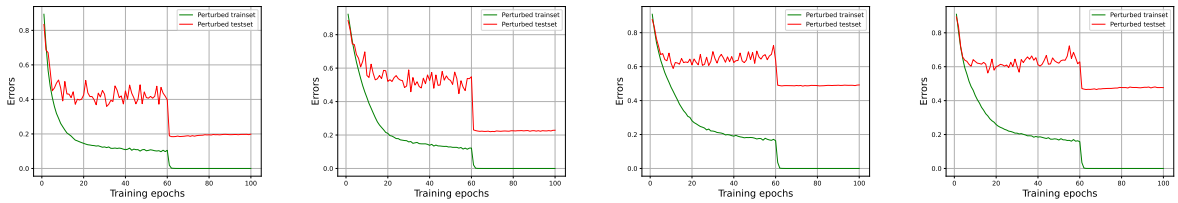
$$
\begin{aligned}
&\frac{1}{m}\mathbb{E}_Z\mathbb{E}_{\hat{Z}}\sup_{\theta\in\Theta}\sqrt{\sum_{i=1}^m|(\mathbb{E}_\rho\left[f_\theta(\mathcal{Q}(x_i+\rho,y_i),y_i)\right]-\mathbb{E}_\rho\left[f_\theta(\mathcal{Q}(\hat{x}_i+\rho,\hat{y}_i),\hat{y}_i)\right])|^2}\\
\leq&\frac{1}{m}\mathbb{E}_Z\mathbb{E}_{\hat{Z}}\sup_{\theta\in\Theta}\sqrt{\sum_{i=1}^m 2\,|\mathbb{E}_\rho f_\theta(\mathcal{Q}(x_i+\rho,y_i),y_i)|^2+\sum_{i=1}^m 2\,|\mathbb{E}_\rho f_\theta(\mathcal{Q}(\hat{x}_i+\rho,\hat{y}_i),\hat{y}_i)|^2} && \text{(B.49)}\\
\leq&\frac{1}{m}\mathbb{E}_Z\mathbb{E}_{\hat{Z}}\sqrt{\sum_{i=1}^m 2(\mathbb{E}_\rho\beta\|\mathcal{Q}(x_i+\rho,y_i)-(x_i+\rho)\|_2+B)^2+\sum_{i=1}^m 2(\mathbb{E}_\rho\beta\|\mathcal{Q}(\hat{x}_i+\rho,\hat{y}_i)-(\hat{x}_i+\rho)\|_2+B)^2} && \text{(B.50)}\\
\leq&\frac{1}{m}\sqrt{\mathbb{E}_Z\mathbb{E}_{\hat{Z}}\left[\sum_{i=1}^m 2(\mathbb{E}_\rho\beta\|\mathcal{Q}(x_i+\rho,y_i)-(x_i+\rho)\|_2+B)^2+\sum_{i=1}^m 2(\mathbb{E}_\rho\beta\|\mathcal{Q}(\hat{x}_i+\rho,\hat{y}_i)-(\hat{x}_i+\rho)\|_2+B)^2\right]} && \text{(B.51)}\\
\leq&\frac{2}{\sqrt{m}}\sqrt{\mathbb{E}_z(\mathbb{E}_\rho\beta\|\mathcal{Q}(x+\rho,y)-(x+\rho)\|_2+B)^2} && \text{(B.52)}\\
\leq&\frac{2(\beta\sqrt{d}\epsilon+B)}{\sqrt{m}} && \text{(B.53)}
\end{aligned}
$$

The final line is due to that with $\|\mathcal{Q}(x+\rho)-(x+\rho)\|_\infty \leq \epsilon$ we have $\|\mathcal{Q}(x+\rho)-(x+\rho)\|_2 \leq \sqrt{d}\epsilon$. This gives the final result

$$
\mathbb{E}_U g(u_1\cdots u_m) \leq \frac{2\beta}{\sqrt{m}}\sqrt{\mathbb{E}_z\gamma(x,y)}+\frac{2(\beta\sqrt{d}\epsilon+B)}{\sqrt{m}}
$$

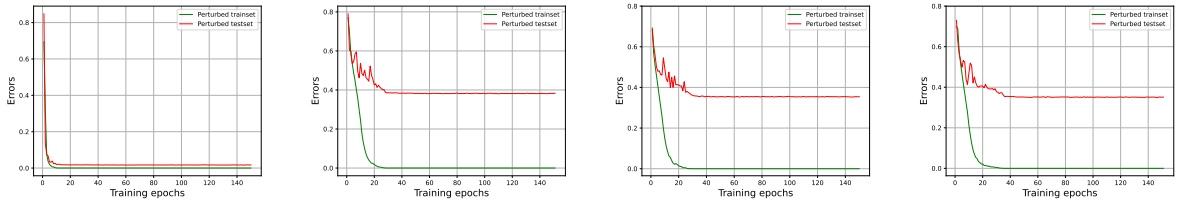
□

B.3 Omitted Figures



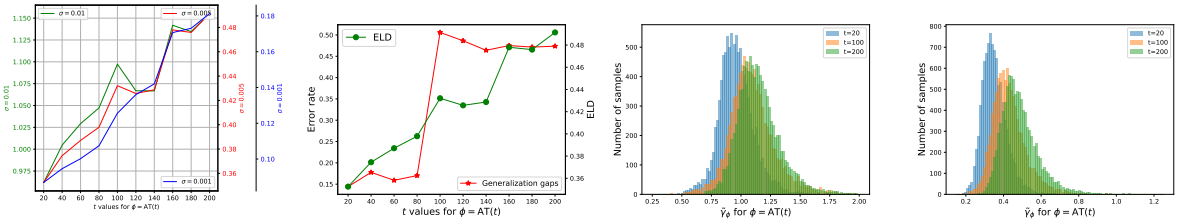
(a) ϕ=AT(0) (b) ϕ=AT(80) (c) ϕ=AT(120) (d) ϕ=AT(200)

Figure B.1: Experiments in Figure 4.1 reproduced on CIFAR-100.



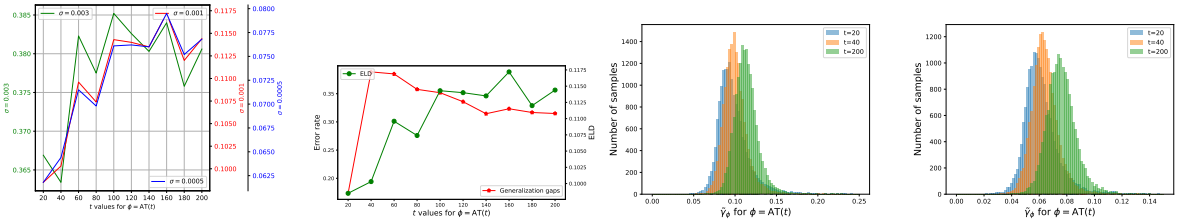
(a) ϕ=AT(0) (b) ϕ=AT(80) (c) ϕ=AT(120) (d) ϕ=AT(200)

Figure B.2: Experiments in Figure 4.1 reproduced on Reduced ImageNet.



(a) ELDs for different σ (b) $\sigma = 0.005$ (c) $\sigma = 0.01$ (d) $\sigma = 0.005$

Figure B.3: Experiments in Figure 4.2 reproduced on CIFAR-100.



(a) ELDs for different σ (b) $\sigma = 0.001$ (c) $\sigma = 0.001$ (d) $\sigma = 0.0005$

Figure B.4: Experiments in Figure 4.2 reproduced on Reduced ImageNet.

Appendix C

Omitted proofs and results in Chapter 5

C.1 Proofs

Proof of Lemma 5.2

$$\Delta_n(A, f_J) = \sup_{S \simeq S'} \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathbb{E}_A[f\left(A(S), J(x; y, A(S)), y\right) - f\left(A(S'); J(x; y, A(S')), y\right)] \tag{C.1}$$

$$= \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathbb{E}_A[f\left(A(S_*), J(x; y, A(S_*)), y\right) - f\left(A(S'_*); J(x; y, A(S'_*)), y\right)] \tag{C.2}$$

$$\leq \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathbb{E}_A[L_{\mathcal{X}}\|J(x; y, A(S_*)) - J(x; y, A(S'_*))\| + L_{\mathcal{W}}\|A(S_*) - A(S'_*)\|] \tag{C.3}$$

$$= \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathbb{E}_A L_{\mathcal{X}}\|J(x; y, A(S_*)) - J(x; y, A(S'_*))\| + L_{\mathcal{W}}\mathbb{E}_A\|A(S_*) - A(S'_*)\| \tag{C.4}$$

$$\leq \sup_{(x,y)\in\mathcal{X}\times\mathcal{Y}} \mathbb{E}_A L_{\mathcal{X}}\|J(x; y, A(S_*)) - J(x; y, A(S'_*))\| + L_{\mathcal{W}} \sup_{S\simeq S'} \mathbb{E}_A\|A(S) - A(S')\| \tag{C.5}$$

The inequality (C.3) is derived based on the condition (5.20). We now deal with the first term in (C.5).

For shorter notation, let $D(S_*, S'_*) := \|A(S_*) - A(S'_*)\|$. For any number $c^* \geq 0$, let $Q(S_*, S'_*; c^*) := \Pr(D(S_*, S'_*) < c^*)$. For any x, y we have

$$\begin{aligned}
&\mathbb{E}_A[L_{\mathcal{X}}\|J(x;y,A(S_*)) - J(x;y,A(S'_*))\|] && \text{(C.6)}\\
=&(1 - Q(S_*, S'_*; c^*))\mathbb{E}_A\left[L_{\mathcal{X}}\|J(x;y,A(S_*)) - J(x;y,A(S'_*))\| \mid D(S_*, S'_*) \geq c^*\right] && \text{(C.7)}\\
+&Q(S_*, S'_*; c^*)\mathbb{E}_A[L_{\mathcal{X}}\|J(x;y,A(S_*)) - J(x;y,A(S'_*))\| \mid D(S_*, S'_*) < c^*] && \text{(C.8)}\\
\leq&(1 - Q(S_*, S'_*; c^*))\mathbb{E}_A[q_{c^*}(J)L_{\mathcal{X}}D(S_*, S'_*) \mid D(S_*, S'_*) \geq c^*] + Q(S_*, S'_*; c^*)L_{\mathcal{X}}2\epsilon\sqrt{d} && \text{(C.9)}\\
\leq&q_{c^*}(J)L_{\mathcal{X}}\mathbb{E}_A D(S_*, S'_*) + Q(S_*, S'_*; c^*)L_{\mathcal{X}}2\epsilon\sqrt{d} && \text{(C.10)}\\
\leq&q_{c^*}(J)L_{\mathcal{X}}\sup_{S\simeq S'}\mathbb{E}_A D(S, S') + Q(S_*, S'_*; c^*)L_{\mathcal{X}}2\epsilon\sqrt{d} && \text{(C.11)}
\end{aligned}$$

The derivation above start by splitting the expectation into two conditional expectations conditioned on two complementary events (see the terms (C.7) and (C.8)) and then utilize the $c-$expansiveness property of J as well as the condition that $J(x, y, w) \in \mathbb{B}_\infty(x, \epsilon)$ to individually derive the first and second terms in (C.9). Plug the final expression above back in (C.5), the lemma is proved. □

Proof of the Theorem 5.11 Consider the AT algorithm specified in (5.4) and (5.5). For two datasets S and S' differing in only one sample and respectively containing n samples, let $\{w_t\}_{t=1}^T$ and $\{w'_t\}_{t=1}^T$ respectively denote the sequences of model parameters generated by running AT on S and S' for T iterations. Let c denote the smallest non-zero value of $\|w_t - w'_t\|$ across t and across the randomness of A when running AT algorithm A on S and S'. (Note that such a choice of c may be overly pessimistic, but it suffices to obtain the desired rate of vanishing of the generalization bound in this theorem). For arbitrary iteration $t \in \{1, \cdots, T-1\}$, we have

$$\begin{aligned}
&\mathbb{E}\|w_{t+1} - w'_{t+1}\|\\
\leq&\mathbb{E}\|w_t - \tau_t\nabla_{w_t}f\left(w_t, \pi(x;y,w_t), y\right) + \tau_t\nabla_{w'_t}f\left(w'_t, \pi(x;y,w_t), y\right) - w'_t\|\\
&+ \mathbb{E}\|\tau_t\nabla_{w'_t}f\left(w'_t, \pi(x';y',w'_t), y'\right) - \tau_t\nabla_{w'_t}f\left(w'_t, \pi(x;y,w_t), y\right)\| && \text{(C.12)}
\end{aligned}$$

Here the expectation is taken over all the randomness in w_t and w'_t. We use (x, y) and (x', y') respectively to denote the samples selected by the AT algorithm from S and S' at the iteration t. Inequality (C.12) is derived by adding and subtracting the term $\tau_t\nabla_{w'_t}f\left(w'_t, \pi(x;y,w_t), y\right)$ and then applying the triangle inequality. For the first term in (C.12), we have that

$$\begin{aligned}
&\mathbb{E}\|w_t - \tau_t\nabla_{w_t}f\left(w_t, \pi(x;y,w_t), y\right) + \tau_t\nabla_{w'_t}f\left(w'_t, \pi(x;y,w_t), y\right) - w'_t\|\\
&\leq \mathbb{E}\|w_t - w'_t\| + \tau_t\beta\mathbb{E}\|w_t - w'_t\| && \text{(C.13)}
\end{aligned}$$

by utilizing the triangle inequality and the condition (5.21). To deal with the second term in (C.12), we consider that at each iteration, with probability $1-1/n$ the samples selected by AT respectively from S and S' are the same. We have

$$\begin{aligned}
&\mathbb{E}\|\tau_t \nabla_{w'_t} f\left(w'_t, \pi(x'; y', w'_t), y'\right) - \tau_t \nabla_{w'_t} f\left(w'_t, \pi(x; y, w_t), y\right)\| \\
&\leq \left(1 - \frac{1}{n}\right) \tau_t \Gamma_{\mathcal{X}} \mathbb{E}\|\pi(x; y, w'_t) - \pi(x; y, w_t)\| + \frac{2\tau_t L_{\mathcal{W}}}{n} \qquad \text{(C.14)} \\
&\leq \left(1 - \frac{1}{n}\right) \tau_t \Gamma_{\mathcal{X}} q_c(\pi) \mathbb{E}\|w_t - w'_t\| + \frac{2\tau_t L_{\mathcal{W}}}{n} \qquad \text{(C.15)}
\end{aligned}$$

The first term in (C.14) and (C.16) make use of the condition (5.21) and then the expansiveness condition of π. Since f is $L_{\mathcal{W}}-$ Lipschitz w.r.t $\mathcal{W}$, we have $\|\nabla_w f(w; x, y)\| \leq L_{\mathcal{W}}$ for $\forall x, y, w$. The second term in (C.14) then follows.

Putting together and considering the step sizes $\tau_t \leq \frac{1}{\beta}$, we have

$$\begin{aligned}
&\mathbb{E}\|w_{t+1} - w'_{t+1}\| \\
&\leq \left(1 + \beta\tau_t + (1 - 1/n)\Gamma_{\mathcal{X}} q_c(\pi)\tau_t\right) \mathbb{E}\|w_t - w'_t\| + \frac{2\tau_t L_{\mathcal{W}}}{n} \qquad \text{(C.16)} \\
&\leq \left(1 + \beta\tau_t + \Gamma_{\mathcal{X}} q_c(\pi)\tau_t\right) \mathbb{E}\|w_t - w'_t\| + \frac{2\tau_t L_{\mathcal{W}}}{n} \qquad \text{(C.17)} \\
&\leq \left(2 + \Gamma_{\mathcal{X}} q_c(\pi)/\beta\right) \mathbb{E}\|w_t - w'_t\| + \frac{2L_{\mathcal{W}}}{n\beta} \qquad \text{(C.18)}
\end{aligned}$$

Unravelling the recursion, we have

$$\mathbb{E}\|w_T - w'_T\| \leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} \zeta^t \qquad \text{(C.19)}$$

where we take $\zeta = 2 + \Gamma_{\mathcal{X}} q_c(\pi)/\beta$. □

Proof of (5.28) Let $a > 2$ be a constant. For shorter notation let $Z = \|A_\pi(S_*) - A_\pi(S'_*)\|$. We will show that if the second moment $\mathbb{E}Z^2 = \mathcal{O}(\frac{1}{n^a})$, we can take $c^* = \mathbb{E}Z - t$ with $t = \Omega(\frac{1}{n^b})$ and $b \in (1, a/2)$, such that the probability $Q(c^*)$ decay at the rate of $\frac{1}{n^{a-2b}}$.

This is due to that

$$
\begin{aligned}
Q(c^*) &= \Pr\left[Z \leq c^*\right] && \text{(C.20)}\\
&= \Pr\left[Z \leq \mathbb{E}Z - t\right] && \text{(C.21)}\\
&\leq \Pr\left[t \leq |Z - \mathbb{E}Z|\right] && \text{(C.22)}\\
&\leq \frac{\mathrm{Var}(Z)}{t^2} && \text{(C.23)}\\
&\leq \frac{\mathbb{E}Z^2}{t^2} && \text{(C.24)}\\
&\leq \mathcal{O}\left(\frac{1/n^a}{1/n^{2b}}\right) = \mathcal{O}\left(\frac{1}{n^{a-2b}}\right) && \text{(C.25)}
\end{aligned}
$$

where the inequality (C.23) is based on the Chebyshev's inequality. Note that such a choice of t will guarantee that $c^* > 0$ such that the derivation above is nontrivial. This is because Theorem 5.11 implies that $\mathbb{E}Z \leq \delta_n(A_\pi) = \mathcal{O}(\frac{1}{n})$ and therefore $c^* = \mathcal{O}(\frac{1}{n} - \frac{1}{n^b})$. Taking $b > 1$ guarantees that $c^* > 0$.

Proof of the Theorem 5.14 and Corollary 5.1 The proof is based on a slight modification of the proof in Theorem 5.11. We start from the inequality (C.12). For the first term in (C.12), since that the loss function f is convex and $\tau_t \leq 1/\beta < 2/\beta$, according to Lemma 3.7.2 in [36], we have

$$
\begin{aligned}
&\mathbb{E}\|w_t - \tau_t \nabla_{w_t} f\left(w_t, \pi(x; y, w_t), y\right) + \tau_t \nabla_{w_t'} f\left(w_t', \pi(x; y, w_t), y\right) - w_t'\| \\
&\leq \mathbb{E}\|w_t - w_t'\| && \text{(C.26)}
\end{aligned}
$$

When f is further assumed to be $\mu-$ strongly convex, we have that $\mu \leq \beta$ since f is also $\beta-$smooth, implying that $\tau_t \leq \frac{1}{\beta} \leq \frac{2}{\beta+\mu}$. According to Lemma 3.7.3 in [36], we have inequality (C.27) as

$$
\begin{aligned}
&\mathbb{E}\|w_t - \tau_t \nabla_{w_t} f\left(w_t, \pi(x; y, w_t), y\right) + \tau_t \nabla_{w_t'} f\left(w_t', \pi(x; y, w_t), y\right) - w_t'\| \\
&\leq \left(1 - \frac{\beta\mu\tau_t}{\beta+\mu}\right) \mathbb{E}\|w_t - w_t'\| && \text{(C.27)}\\
&\leq \left(1 - \frac{1}{2}\tau_t\mu\right) \mathbb{E}\|w_t - w_t'\| && \text{(C.28)}
\end{aligned}
$$

In fact, since $\mu \leq \beta$, we also have $1 \leq \frac{2\beta}{\beta+\mu}$ and thus $\tau_t\mu \leq \frac{2\tau_t\mu\beta}{\beta+\mu}$ with $\tau_t\mu \leq 1$. The inequality (C.27) can be further simplified as (C.28).

The second term in (C.12) follows the same derivation as in the proof of Theorem 5.11. Putting together, when f is convex, we have

$$\begin{aligned}
&\mathbb{E}\|w_{t+1} - w'_{t+1}\| \\
&\leq (1 + \Gamma_{\mathcal{X}} q_c(\pi)\tau_t)\, \mathbb{E}\|w_t - w'_t\| + \frac{2\tau_t L_{\mathcal{W}}}{n} && \text{(C.29)} \\
&\leq (1 + \Gamma_{\mathcal{X}} q_c(\pi)/\beta)\, \mathbb{E}\|w_t - w'_t\| + \frac{2L_{\mathcal{W}}}{n\beta} && \text{(C.30)}
\end{aligned}$$

when f is $\mu-$ strongly convex, we have

$$\begin{aligned}
&\mathbb{E}\|w_{t+1} - w'_{t+1}\| \\
&\leq \left(1 - \frac{1}{2}\tau_t\mu + \Gamma_{\mathcal{X}} q_c(\pi)\tau_t\right) \mathbb{E}\|w_t - w'_t\| + \frac{2\tau_t L_{\mathcal{W}}}{n} && \text{(C.31)} \\
&\leq \left(1 - \frac{\mu}{2\beta} + \Gamma_{\mathcal{X}} q_c(\pi)/\beta\right) \mathbb{E}\|w_t - w'_t\| + \frac{2L_{\mathcal{W}}}{n\beta} && \text{(C.32)}
\end{aligned}$$

Unravelling the recursion, we have

$$\mathbb{E}\|w_T - w'_T\| \leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} \zeta^t \tag{C.33}$$

with $\zeta = 1 + \Gamma_{\mathcal{X}} q_c(\pi)/\beta$ when f is convex and $\zeta = 1 - \frac{\mu}{2\beta} + \Gamma_{\mathcal{X}} q_c(\pi)/\beta$ when f is $\mu-$ strongly convex. For the strongly convex case, if we let $q_c(\pi) < \frac{\mu}{2\Gamma_{\mathcal{X}}}$, we have $\zeta < 1$. In this case, the geometric series $\sum_{t=0}^{T} \zeta^t$ converges as $T \to \infty$ and entails a closed form. The bound in (C.33) can therefore be further simplified as

$$\begin{aligned}
\mathbb{E}\|w_T - w'_T\| &\leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{T} \zeta^t \\
&\leq \frac{2L_{\mathcal{W}}}{n\beta} \sum_{t=0}^{\infty} \zeta^t && \text{(C.34)} \\
&= \frac{2L_{\mathcal{W}}}{n\beta} \frac{1}{1-\zeta} && \text{(C.35)} \\
&= \frac{4L_{\mathcal{W}}}{n(\mu - 2q_c(\pi)\Gamma_{\mathcal{X}})} && \text{(C.36)}
\end{aligned}$$

This derives the bound in Corollary 5.1. □

Proof of Lemma 5.3 To establish the proof, we first discuss the expansive property of the one step PGD perturbation T For arbitrary $\hat{x} \in \mathcal{X}$, we have

$$\begin{aligned}
&\|T_{x,y}(\hat{x}; w) - T_{x,y}(\hat{x}; w')\| && \text{(C.37)}\\
=&\left\|\Pi_{\mathbb{B}_\infty(x,\epsilon)}\left[\hat{x} + \lambda G\left(\nabla_{\hat{x}} f(w, \hat{x}, y)\right)\right] - \Pi_{\mathbb{B}_\infty(x,\epsilon)}\left[\hat{x} + \lambda G\left(\nabla_{\hat{x}} f(w', \hat{x}, y)\right)\right]\right\| && \text{(C.38)}\\
\leq&\lambda \left\|G\left(\nabla_{\hat{x}} f(w, \hat{x}, y)\right) - G\left(\nabla_{\hat{x}} f(w', \hat{x}, y)\right)\right\| && \text{(C.39)}\\
\leq&\lambda\alpha \left\|\nabla_{\hat{x}} f(w, \hat{x}, y) - \nabla_{\hat{x}} f(w', \hat{x}, y)\right\| && \text{(C.40)}\\
\leq&\lambda\alpha\Gamma_{\mathcal{W}}\|w - w'\| && \text{(C.41)}
\end{aligned}$$

The inequality (C.39) is due to that the projection operation $\Pi_{\mathbb{B}_\infty(x,\epsilon)}$ is 1-expansive. The inequalities (C.40) and (C.41) are derived based on the Lipschitz condition of G and $\nabla_x f$.

For fixed $w \in \mathcal{W}$, we have for arbitrary $x', x'' \in \mathcal{X}$

$$\begin{aligned}
&\|T_{x,y}(x'; w) - T_{x,y}(x''; w)\| && \text{(C.42)}\\
=&\left\|\Pi_{\mathbb{B}_\infty(x,\epsilon)}\left[x' + \lambda G\left(\nabla_{x'} f(w, x', y)\right)\right] - \Pi_{\mathbb{B}_\infty(x,\epsilon)}\left[x'' + \lambda G\left(\nabla_{x''} f(w, x'', y)\right)\right]\right\| && \text{(C.43)}\\
\leq&\left\|x' + \lambda G\left(\nabla_{x'} f(w, x', y)\right) - x'' + \lambda G\left(\nabla_{x''} f(w, x'', y)\right)\right\| && \text{(C.44)}\\
\leq&\|x' - x''\| + \lambda\alpha \left\|\nabla_{x'} f(w, x', y) - \nabla_{x''} f(w, x'', y)\right\| && \text{(C.45)}\\
\leq&(1 + \lambda\alpha\eta)\|x' - x''\| && \text{(C.46)}
\end{aligned}$$

The derivation here follows the similar idea as above, utilizing the 1-expansiveness condition of $\Pi_{\mathbb{B}_\infty(x,\epsilon)}$ as well as the Lipschitz condition of G and the smoothness condition of f w.r.t $\mathcal{X}$.

We now derive the upper bound for the expansiveness of π^{PGD}. With a little abuse of notation, let $x_K = T^K_{x,y}(x; w)$ and similarly $x'_K = T^K_{x,y}(x; w')$. For shorter notation, let $\nu = \lambda\alpha\Gamma_{\mathcal{W}}$ and $\mu = 1 + \lambda\alpha\eta$

$$\begin{aligned}
&\|\pi^{\text{PGD}}(x; y, w) - \pi^{\text{PGD}}(x; y, w')\| && \text{(C.47)}\\
=&\|T^K_{x,y}(x; w) - T^K_{x,y}(x; w')\| && \text{(C.48)}\\
=&\|T_{x,y}(x_{K-1}; w) - T_{x,y}(x'_{K-1}; w')\| && \text{(C.49)}\\
\leq&\|T_{x,y}(x_{K-1}; w) - T_{x,y}(x_{K-1}; w')\| + \|T_{x,y}(x_{K-1}; w') - T_{x,y}(x'_{K-1}; w')\| && \text{(C.50)}\\
\leq&\pi\|w - w'\| + \mu\|x_{K-1} - x'_{K-1}\| && \text{(C.51)}\\
=&\pi\|w - w'\| + \mu\|T_{x,y}(x_{K-2}; w) - T_{x,y}(x'_{K-2}; w')\| && \text{(C.52)}\\
\leq&\sum_{k=0}^{K-1} \mu^i\nu\|w - w'\| && \text{(C.53)}
\end{aligned}$$

Note that the bound (C.53) holds for any choice of w, w'. On the other hand, using the condition that $T_{x,y}(\hat{x}; w) \in \mathbb{B}_\infty(x, \epsilon)$, we can derive that for any $w, w' \in \mathcal{W}$ with $\|w - w'\| > c$,

$$\|T_{x,y}(\hat{x}; w) - T_{x,y}(\hat{x}; w')\| \leq 2\sqrt{d}\epsilon = \frac{2\sqrt{d}\epsilon}{\|w - w'\|}\|w - w'\| \leq \frac{2\sqrt{d}\epsilon}{c}\|w - w'\| \tag{C.54}$$

Putting together, we have

$$q_c(\pi^{\text{PGD}}) \leq \min\left(\sum_{k=0}^{K-1} \mu^i \nu, \frac{2\sqrt{d}\epsilon}{c}\right) \tag{C.55}$$

This completes the proof. □

Proof of Corollary 5.2 We first establish the following result.

For any non-negative random variable Z bounded below B and any $c^* > 0$,

$$\Pr[Z \leq c^*] \leq \frac{B - \mathbb{E}(Z)}{B - c^*} \tag{C.56}$$

This result simply follows from $\Pr[Z \leq c^*] = \Pr[B - Z \geq B - c^*]$ and applying the Markov Inequality to random variable $B - Z$.

Now let $Z = A(S) - A(S')$ and $c^* = Bn^{-1/2}$ in Theorem 5.12. The second term in bound of Theorem 5.12 then reduces to $\left(1 - \frac{\sup_{S \simeq S'} \mathbb{E}\|A(S) - A(S')\|}{B(1-n^{-1/2})}\right) L_{\mathcal{X}} \cdot 2\epsilon\sqrt{d}$, which converges to $\left(1 - \frac{\sup_{S \simeq S'} \mathbb{E}\|A(S) - A(S')\|}{B}\right) L_{\mathcal{X}} \cdot 2\epsilon\sqrt{d}$ with n. It can be verified that the first term in the bound of Theorem 5.12 vanishes with n (as $n^{-1/2}$). The corollary then follows. □.

Proof of Lemma 5.6 The proof is established by noticing that all members in the set $\tilde{H}(r) := \{x \in \mathbb{R}^d : |x[i]| = r, \forall i \in \mathcal{I}\}$ achieves $1/(rd^{\frac{1}{p}})$−Lipschitz and thus the Lipschitz constant over $H(r)$ is greater than it. Specifically, for any $x, \hat{x} \in \tilde{H}(r)$ with $x \neq \hat{x}$, let

$\mathcal{I}_- := \{i \in \mathcal{I} : \mathrm{sgn}(x[i]) \neq \mathrm{sgn}(\hat{x}[i])\}$ and $\mathcal{I}_+ := \mathcal{I} - \mathcal{I}_-$. We have

$$\|G(x) - G(\hat{x})\|_2 \tag{C.57}$$

$$= \left\| \frac{\mathrm{sgn}(x) \odot |x|^{q-1}}{\|x\|_q^{q-1}} - \frac{\mathrm{sgn}(\hat{x}) \odot |\hat{x}|^{q-1}}{\|\hat{x}\|_q^{q-1}} \right\|_2 \tag{C.58}$$

$$= \left(\sum_{i=1}^{d} \left| \frac{\mathrm{sgn}(x[i])|x[i]|^{q-1}}{\|x\|_q^{q-1}} - \frac{\mathrm{sgn}(\hat{x}[i])|\hat{x}[i]|^{q-1}}{\|\hat{x}\|_q^{q-1}} \right|^2 \right)^{\frac{1}{2}} \tag{C.59}$$

$$= \left(\sum_{j \in \mathcal{I}_+} \left| \frac{\mathrm{sgn}(x[j])|x[j]|^{q-1}}{\|x\|_q^{q-1}} - \frac{\mathrm{sgn}(\hat{x}[j])|\hat{x}[j]|^{q-1}}{\|\hat{x}\|_q^{q-1}} \right|^2 + \sum_{k \in \mathcal{I}_-} \left| \frac{\mathrm{sgn}(x[k])|x[k]|^{q-1}}{\|x\|_q^{q-1}} - \frac{\mathrm{sgn}(\hat{x}[k])|\hat{x}[k]|^{q-1}}{\|\hat{x}\|_q^{q-1}} \right|^2 \right)^{\frac{1}{2}} \tag{C.60}$$

$$= \left(\sum_{k \in \mathcal{I}_-} \left| \frac{2r^{q-1}}{r^{q-1} d^{\frac{1}{p}}} \right|^2 \right)^{\frac{1}{2}} \tag{C.61}$$

$$= \sqrt{|\mathcal{I}_-|} \frac{2}{d^{\frac{1}{p}}} \tag{C.62}$$

where $|\mathcal{I}_-|$ denotes the cardinality of the set $\mathcal{I}_-$. The equality (C.61) is derived by noting that the first term in (C.60) is zero since $|x[j]| = |\hat{x}[j]|$ and $\mathrm{sgn}(|x[j]|) = \mathrm{sgn}(|\hat{x}[j]|)$ for each $j \in \mathcal{I}_+$ and noting that $\|x\|_q = r d^{\frac{1}{q}}$ for any $x \in \tilde{H}(r)$. The power term $\frac{q-1}{q}$ is replaced by $\frac{1}{p}$ since $1/q + 1/p = 1$. We also have

$$\|x - \hat{x}\|_2 \tag{C.63}$$

$$= \left(\sum_{i=1}^{d} |x[i] - \hat{x}[i]|^2 \right)^{\frac{1}{2}} \tag{C.64}$$

$$= \left(\sum_{j \in \mathcal{I}_+} |x[j] - \hat{x}[j]|^2 + \sum_{k \in \mathcal{I}_-} |x[k] - \hat{x}[k]|^2 \right)^{\frac{1}{2}} \tag{C.65}$$

$$= \left(\sum_{k \in \mathcal{I}_-} |2r|^2 \right)^{\frac{1}{2}} \tag{C.66}$$

$$= 2r \sqrt{|\mathcal{I}_-|} \tag{C.67}$$

Putting together, we have that for any $x, \hat{x} \in \tilde{H}(r)$ with $x \neq \hat{x}$,

$$\frac{\|G(x) - G(\hat{x})\|_2}{\|x - \hat{x}\|_2} = \frac{1}{rd^{\frac{1}{p}}} \leq \sup_{\substack{x', x'' \in Q(r) \\ x' \neq x''}} \frac{\|G(x') - G(x'')\|_2}{\|x' - x''\|_2} = \alpha_p \tag{C.68}$$

This completes the proof. □

Proof of Lemma 5.4 Since the following proof does not depend on the choice of w and y, for simplicity we will write $f(w, x, y)$ as $f(x)$ and $\nabla_x f(w, x, y)$ as $\nabla f(x)$ hereafter.

To establish the proof of Lemma 5.4, we first present and prove the following intermediate result.

Lemma C.1. *Suppose that the gradients of $f(x)$ satisfies the Lipschitz condition (5.37) and the mapping G in the PGD attack satisfies the condition (5.40). We have*

$$f(x') - f(x) - \lambda G(\nabla f(x))^T (x' - x) \leq \frac{\eta + 1}{2}\|x' - x\|^2 + \frac{1}{2}\|\nabla f(x)\|^2 + \frac{\lambda^2}{2}\|G(\nabla f(x))\|^2 \tag{C.69}$$

for any x' and x.

Proof. By the fundamental theorem of calculus, we have

$$f(x') - f(x) = \int_0^1 \frac{d}{dt} f(x + t(x' - x))dt \tag{C.70}$$

$$= \int_0^1 \nabla f(x + t(x' - x))^T (x' - x)dt \tag{C.71}$$

We therefore have

$$\begin{aligned} & f(x') - f(x) - \lambda G(\nabla f(x))^T (x' - x) \\ = & \int_0^1 \left[\nabla f(x + t(x' - x)) - \lambda G(\nabla f(x))\right]^T (x' - x)dt \qquad \text{(C.72)} \\ = & \int_0^1 \left[\nabla f(x + t(x' - x)) - \nabla f(x)\right]^T (x' - x)dt + \left[\nabla f(x) - \lambda G(\nabla f(x))\right]^T (x' - x) \qquad \text{(C.73)} \end{aligned}$$

For the first term in (C.73), we have

$$\begin{aligned}
&\int_0^1 \left[\nabla f(x+t(x'-x)) - \nabla f(x)\right]^T (x'-x)dt \\
\leq &\int_0^1 \|\nabla f(x+t(x'-x)) - \nabla f(x)\|\|x'-x\|dt && \text{(C.74)} \\
\leq &\int_0^1 \eta\|t(x'-x)\|\|x'-x\|dt && \text{(C.75)} \\
= &\frac{\eta}{2}\|x'-x\|^2 && \text{(C.76)}
\end{aligned}$$

where inequality (C.74) follows from the Cauchy−Schwarz inequality and inequality (C.75) is due to that the gradient of f is $\eta-$Lipschitz (i.e., condition (5.37)).

For the second term in (C.73), we have

$$\begin{aligned}
&\left[\nabla f(x) - \lambda G(\nabla f(x))\right]^T (x'-x) \\
\leq &\frac{1}{2}\|\nabla f(x) - \lambda G(\nabla f(x))\|^2 + \frac{1}{2}\|x'-x\|^2 && \text{(C.77)} \\
= &\frac{1}{2}\|\nabla f(x)\|^2 + \frac{1}{2}\|\lambda G(\nabla f(x))\|^2 - 2\lambda G(\nabla f(x))^T\nabla f(x) + \frac{1}{2}\|x'-x\|^2 && \text{(C.78)} \\
\leq &\frac{1}{2}\|\nabla f(x)\|^2 + \frac{1}{2}\|\lambda G(\nabla f(x))\|^2 + \frac{1}{2}\|x'-x\|^2 && \text{(C.79)}
\end{aligned}$$

Inequality (C.77) is due to that for any vector a and b we have $a^Tb \leq \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2$. Inequality (C.79) is derived based on the condition (5.40) that $G(\nabla f(x))^T\nabla f(x) > 0$.

The proof is completed by combining (C.76) and (C.79) together. □

We now present the proof for Lemma 5.4.

Proof. For simplicity, we write $f(w, x^*, y)$ as $f(x^*)$. Let $x^k := T^k_{x,y}(x; w)$. Additionally, let $\tilde{x}^{k+1} = x^k + \lambda G(\nabla f(x^k))$ and we therefore have $x^{k+1} = \Pi_{\mathbb{B}_\infty(x,\epsilon)}(\tilde{x}^{k+1})$.

According to Lemma C.1, we have that for any $k \leq K$

$$\begin{aligned}
&f(x^*) - f(x^k) \\
\leq &\lambda G(\nabla f(x^k))^T(x^* - x^k) + \frac{\eta+1}{2}\|x^* - x^k\|^2 + \frac{1}{2}\|\nabla f(x^k)\|^2 + \frac{\lambda^2}{2}\|G(\nabla f(x^k))\|^2 && \text{(C.80)}
\end{aligned}$$

For the first term in (C.80), we have that

$$\begin{aligned}
&\lambda G(\nabla f(x^k))^T(x^* - x^k) \\
=&(\tilde{x}^{k+1} - x^k)^T(x^* - x^k) && \text{(C.81)} \\
=&\frac{1}{2}\left(\|\tilde{x}^{k+1} - x^k\|^2 + \|x^* - x^k\|^2 - \|\tilde{x}^{k+1} - x^*\|^2\right) && \text{(C.82)} \\
\leq&\frac{1}{2}\left(\|\tilde{x}^{k+1} - x^k\|^2 + \|x^* - x^k\|^2 - \|x^{k+1} - x^*\|^2\right) && \text{(C.83)} \\
=&\frac{1}{2}\left(\|\lambda G(\nabla f(x^k))\|^2 + \|x^* - x^k\|^2 - \|x^{k+1} - x^*\|^2\right) && \text{(C.84)}
\end{aligned}$$

where equality (C.82) is due to that for any vector a and b, we have $2a^Tb = \|a\|^2 + \|b\|^2 - \|a-b\|^2$. Inequality (C.83) follows from the fact that since $x^* \in \mathbb{B}_\infty(x, \epsilon)$, we have $\|\tilde{x}^{k+1} - x^*\|^2 \leq \|\Pi_{\mathbb{B}_\infty(x,\epsilon)}(\tilde{x}^{k+1}) - x^*\|^2 = \|x^{k+1} - x^*\|^2$

For the other terms in (C.80), we have that

$$\begin{aligned}
&\frac{\eta+1}{2}\|x^* - x^k\|^2 + \frac{1}{2}\|\nabla f(x^k)\|^2 + \frac{\lambda^2}{2}\|G(\nabla f(x^k))\|^2 \\
=&\frac{\eta+1}{2}\|x^* - x^k\|^2 + \frac{1}{2}\|\nabla f(x^k) - \nabla f(x^*)\|^2 + \frac{\lambda^2}{2}\|G(\nabla f(x^k))\|^2 && \text{(C.85)} \\
\leq&\frac{\eta+1}{2}\|x^* - x^k\|^2 + \frac{\eta^2}{2}\|x^k - x^*\|^2 + \frac{\lambda^2 C}{2} && \text{(C.86)} \\
\leq&\frac{d^*(\eta^2+\eta+1)}{2} + \frac{\lambda^2 C}{2} && \text{(C.87)}
\end{aligned}$$

where equality (C.85) is derived based on the conditions that $\nabla f(x^*) = 0$. Inequality (C.86) is derived according to the Lipschitz condition of the gradients (5.37) and the condition that $\|G(\nabla f(x))\|^2 \leq C$ for any x.

Combining the results, we have that

$$
\begin{aligned}
& f(x^*) - \frac{1}{K}\sum_{k=1}^{K} f(x^k) \\
=& \frac{1}{K}\sum_{k=1}^{K} f(x^*) - f(x^k) && \text{(C.88)} \\
\leq& \frac{1}{2K}\sum_{k=1}^{K} \left(\|\lambda G(\nabla f(x^k))\|^2 + \|x^k - x^*\|^2 - \|x^{k+1} - x^*\|^2\right) + \frac{d^*(\eta^2+\eta+1)}{2} + \frac{\lambda^2 C}{2} && \text{(C.89)} \\
\leq& \frac{\lambda^2 C}{2} + \frac{\|x^1 - x^*\|^2 - \|x^{K+1} - x^*\|^2}{2K} + \frac{d^*(\eta^2+\eta+1)}{2} + \frac{\lambda^2 C}{2} && \text{(C.90)} \\
\leq& \lambda^2 C + \frac{\|x^1 - x^*\|^2}{2K} + \frac{d^*(\eta^2+\eta+1)}{2} && \text{(C.91)} \\
\leq& \lambda^2 C + \frac{d^*}{2K} + \frac{d^*(\eta^2+\eta+1)}{2} && \text{(C.92)}
\end{aligned}
$$

Taking $\lambda = \frac{1}{\sqrt{K}}$, we have

$$
f(x^*) - \frac{1}{K}\sum_{k=1}^{K} f(x^k) \leq \frac{(2C + d^*)}{2K} + \frac{d^*(\eta^2+\eta+1)}{2} \tag{C.93}
$$

This completes the proof. □

C.2 Hyper-parameter settings for the experiments

In our experiments, we follow the settings in [78]: The perturbation radius is set to be $\epsilon = 8/255$ w.r.t the $\infty-$norm for the three datasets. The pre-activation ResNet 18 (PRN-18) model [38] is used for CIFAR-10 and SVHN. The Wide ResNet 34 (WRN-34) model [113] is used for CIFAR-100. We set $K = 10$ for all the PGD variants with $\lambda = 2/255$ on CIFAR-10 and CIFAR-100, and set $\lambda = 1/255$ for SVHN. The initial learning rate of AT is set to be 0.1 for CIFAR-10 and CIFAR-100 and set to be 0.01 for SVHN. The learning rate is decayed by 0.1 at the 100^{th} and the 150^{th} epoch of the training. The batch size is set to be 128 and a weight decay of 5×10^{-4} is used for all the experiments. The experiments are conducted on our internal GPU clusters. Training PRN-18 on CIFAR-10 and SVHN for 200 epochs spends around 18 hours with two NVIDIA V100 GPUs, and training WRN-34 on CIFAR-100 requires around three days to complete with the same computing resources.

C.3 Computing λ_p

The volume of $\mathbb{B}_p(0, \lambda_p)$ is computed by

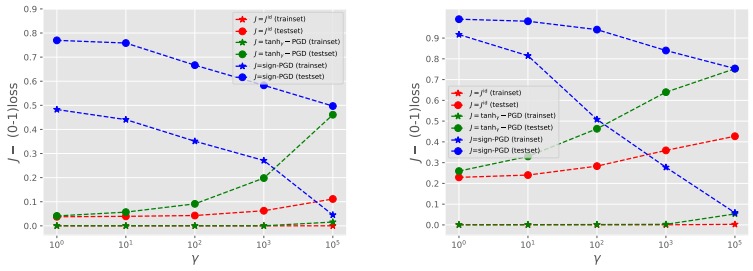
$$\text{vol}\left(\mathbb{B}_p(0, \lambda_p)\right) = \frac{\left(2\Gamma\left(\frac{1}{p}+1\right)\right)^d}{\Gamma\left(\frac{d}{p}+1\right)}\lambda_p^d \tag{C.94}$$

Here $\Gamma(\cdot)$ denotes the Euler's gamma function. For p other than ∞, to make $\text{vol}\left(\mathbb{B}_p(0, \lambda_p)\right) = \text{vol}\left(\mathbb{B}_\infty(0, \lambda_\infty)\right)$, we have

$$\lambda_p = \exp\left\{\frac{1}{d}\ln\Gamma(\frac{d}{p}+1) + \ln\frac{\lambda_\infty}{\Gamma(\frac{1}{p}+1)}\right\} \tag{C.95}$$

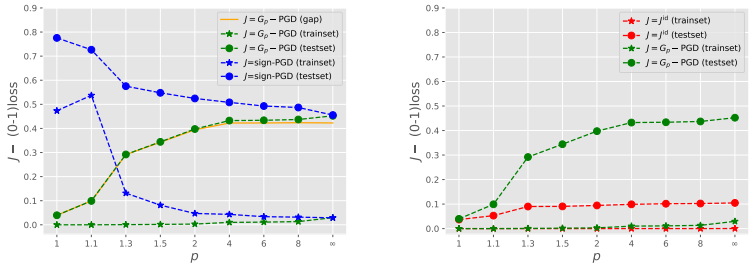
In the experiments, the value of λ_∞ (i.e., the step size for the sign-PGD) is set to be the same as in Section **??** and values for other λ_p is computed from (C.95).

C.4 Omitted figures



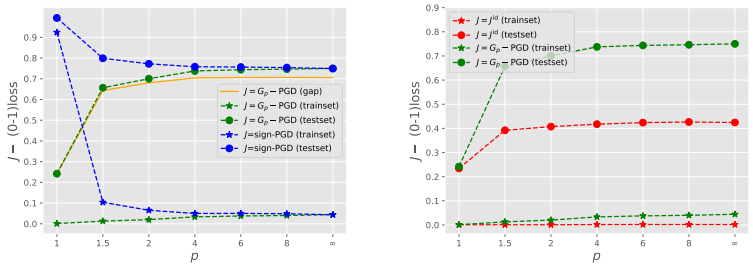
(a) (b)

Figure C.1: Experiments in Figure 5.2 reproduced on SVHN and CIFAR-100.



(a) (b)

Figure C.2: Experiments in Figure 5.3 reproduced on SVHN.



(a) (b)

Figure C.3: Experiments in Figure 5.3 reproduced on CIFAR-100.