Adversarial Rademacher Complexity of Deep Neural Networks

Jiancong Xiao JIANCONGXIAO@LINK.CUHK.EDU.CN
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China

Yanbo Fan FANYANBO0124@GMAIL.COM
Tencent AI Lab
Shenzhen, China

Ruoyu Sun* SUNRUOYU@CUHK.EDU.CN
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China

Zhi-Quan Luo LUOZQ@CUHK.EDU.CN
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China

Abstract

Deep neural networks (DNNs) are highly vulnerable to adversarial attacks. Ideally, a robust model should perform well on both perturbed training data and unseen perturbed test data. While fitting perturbed training data is relatively easy, generalizing to perturbed test data remains a significant challenge. This motivates the study of generalization guarantees from a learning theory perspective. This paper focuses on adversarial Rademacher complexity (ARC), first introduced by Yin et al. (2018) and Khim and Loh (2018). Their work primarily addressed linear functions and highlighted the open question of how to bound ARC for neural networks. Since then, several attempts have been made, with the latest results applying ARC only to two-layer neural networks. The main challenge arises from the dynamic nature and unknown closed-form solution of adversarial examples. In this paper, we resolve this issue and provide the first bound on ARC for deep neural networks. Our bound is qualitatively comparable to Rademacher complexity bounds in similar settings. The key ingredient is a new concept we introduce, termed intermediate adversarial examples, along with a framework for calculating the covering number that is compatible with them. Finally, we present experiments to analyze poor robust generalization, demonstrating that the weight norm is a crucial factor influencing the robust generalization gap.

Keywords: Rademacher Complexity, Adversarial Robustness, Generalization Bounds, Neural Networks

1. Introduction

Deep neural networks (DNNs) (Krizhevsky et al., 2012; Hochreiter and Schmidhuber, 1997) have achieved remarkable success in various machine learning tasks, including computer vision (CV) and natural language processing (NLP). However, they have been shown to be vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014). More specifically, a well-trained model can perform poorly on slightly perturbed data samples. Incorporating perturbed samples into the training dataset can improve robustness in practice, but it does not always lead to satisfactory performance. One major issue arises from generalization: while training a model to fit perturbed training samples

*. Corresponding Author.

is relatively easy, such a model often fails to generalize well to adversarial examples in the test set. For instance, when applying ResNet to CIFAR-10, adversarial training can achieve nearly 100% robust accuracy on the training set, yet only 47% robust accuracy on the test set (Madry et al., 2017). Recent works (Gowal et al., 2020; Rebuffi et al., 2021) have mitigated the overfitting issue, but it still has a 20% robust generalization gap between robust test accuracy (approximately 60%) and robust training accuracy (around 80%). Therefore, it is interesting to provide a theoretical understanding of adversarially robust generalization. This paper focuses on Rademacher complexity.

In classical learning theory, the generalization gap can be bounded in terms of Rademacher complexity with high probability. Rademacher complexity is defined as

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}) = \mathbb{E}_{\sigma} \frac{1}{n} \left[\sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sigma_i h(x_i, y_i) \right], \tag{1}$$

where $\mathcal{S} = \{x_i, y_i\}_{i=1,\cdots,n}$ is the sample dataset with n samples, $\mathcal{H}$ is the hypothesis function class, and σ_i are i.i.d. Rademacher random variables, *i.e.,* σ_i takes values 1 or -1 with equal probability. Techniques for deriving upper bounds on the Rademacher complexity of deep neural networks have been extensively studied, including layer-peeling (Neyshabur et al., 2015; Golowich et al., 2018) and covering number arguments (Bartlett et al., 2017). For more details, see Section 2.

Khim and Loh (2018) and Yin et al. (2018) concurrently extended Rademacher complexity to adversarial settings. They demonstrated that the robust generalization gap can be bounded by the Rademacher complexity of the adversarial loss, defined as $\tilde{h}(x_i, y_i) = \max_{x_i' \in \mathcal{B}(x_i)} h(x_i', y_i)$, where $\mathcal{B}(x)$ is a norm ball around sample x, and $\tilde{\mathcal{H}}$ represents the hypothesis class of adversarial losses. This specific form of Rademacher complexity is referred to as adversarial Rademacher complexity. Their primary contribution was establishing bounds for linear function classes.

For neural networks, it may seem straightforward to extend methods for standard losses to adversarial losses. However, Khim and Loh (2018); Yin et al. (2018) both pointed out that providing upper bounds in adversarial settings is significantly more challenging due to the presence of the $\max$ operator in adversarial loss. As a result, they relied on surrogate losses, leaving the following question for future work:

How can the adversarial Rademacher complexity of deep neural networks be bounded?

Since 2018, several attempts have been made to tackle this problem. Awasthi et al. (2020) attempted to extend the bounds on adversarial Rademacher complexity from linear functions to two-layer neural networks. Gao and Wang (2021) proposed a more meaningful surrogate loss: the FGSM loss. Broadly, existing attempts to tackle this problem can be categorized into two main approaches.

Type 1: Adversarial Loss in Shallow Networks. The first approach focuses on obtaining closed-form solutions for the optimal adversarial examples x_i^* and analyzing the adversarial loss, given by $\max_{x'} h(x', y) = h(x_i^*, y)$. Khim and Loh (2018); Yin et al. (2018) introduced adversarial Rademacher complexity and provided bounds for linear functions using this method. Awasthi et al. (2020) extended this analysis to two-layer neural networks, deriving bounds in this setting. However, in deeper networks, obtaining closed-form solutions becomes intractable, making it unclear how to extend this approach to multi-layer architectures and derive generalization bounds in deep neural networks.

Table 1: Comparison of our work with the two types of attempts on bounding Adversarial Rademacher complexity: Type 1: Adversarial Loss in Shallow Networks (Yin et al., 2018; Khim and Loh, 2018; Awasthi et al., 2020). Type 2: Surrogate Loss in Deep Networks (Yin et al., 2018; Khim and Loh, 2018; Gao and Wang, 2021). Our work distinguishes itself by providing the first bound for the Adversarial Rademacher Complexity of DNNs.

	Loss	Networks	Techniques	Limitation
Type 1	**Adversarial Loss**	$\leq$ Two-Layer	Optimal Attack	Cannot be applied to DNNs
Type 2	Surrogate Loss	**Multi-Layer**	Change Definition	Cannot bound the robust generalization gap
Ours	**Adversarial Loss**	**Multi-Layer**	Lemma 1 & 2	-

Type 2: Surrogate Loss in Deep Networks. This approach uses a surrogate loss $\hat{h}(x, y) \approx \tilde{h}(x, y) = \max_{x'} h(x', y)$ to bypass the main difficulty posed by the $\max$ operator, where the surrogate loss does not explicitly contain a $\max$ term. Examples of $\hat{h}(x, y)$ include tree-transformation loss (Khim and Loh, 2018), SDP relaxation loss (Yin et al., 2018), and FGSM loss (Gao and Wang, 2021). However, this approach provides upper bounds for the Rademacher complexity of the surrogate loss rather than the adversarial loss, and thus cannot bound the robust generalization gap. For a more detailed discussion, see Appendix B.2.

In summary, these two types of attempts aim to eliminate the $\max$ operator using different approaches. The methods and their limitations are summarized in Table 1. To our knowledge, the problem of bounding the adversarial Rademacher complexity of deep neural networks has remained unsolved since it was first raised in 2018. In this paper, we resolve this problem and provide the first bound for the adversarial Rademacher complexity of deep neural networks. Our approach is based on the covering number, which serves as an upper bound for Rademacher complexity. In adversarial settings, this problem becomes:

How to calculate the covering number of the adversarial hypothesis class?

The first challenge for this problem is that the closed-form expression of optimal adversarial examples is not known. To address this, we introduce a concept called intermediate adversarial examples, which allow us to bound the covering number of the linear function class without requiring access to the closed-form solution of the optimal adversarial example. Using this approach, we reproduce the bound in the linear setting. The formal definition of intermediate adversarial examples is provided later in Lemma 1.

The second challenge arises from the conflict between existing methods for calculating the covering number of DNNs and the dynamic nature of intermediate adversarial examples. Current techniques for computing the covering number of DNNs assume a static training set, whereas adversarial examples are model-dependent and evolve dynamically. To resolve this issue, we introduce a lemma called Layer-wise Induction for Adversarial Hypothesis Class, which is designed to be compatible with our intermediate adversarial examples. The formal statement of this lemma is provided later in Lemma 2. By combining these two techniques, we establish the first bound on the adversarial Rademacher complexity of DNNs.

Main Result. For depth-l, width-h fully-connected neural networks, assume that the weight matrices $W_1, W_2, \ldots, W_l$ in each of the l layers have Frobenius norms bounded by $M_1, \ldots, M_l$, and

all n samples are bounded by B. Then, with high probability,

$$\text{Adversarial Rademacher Complexity} \leq \mathcal{O}\bigg(\frac{(B+\epsilon)h\sqrt{l\log l}\prod_{j=1}^{l}M_j}{\sqrt{n}}\bigg).$$

We provide a comparison with existing bounds in similar settings. We show that our bound is comparable to (1) the upper bound for standard Rademacher complexity and (2) the upper bound for adversarial Rademacher complexity of two-layer neural networks (Awasthi et al., 2020). Additionally, we provide a lower bound for adversarial Rademacher complexity and extend the results to multi-class classification settings. Finally, we study some empirical implications of our bounds. Our experiments indicate that the weight norm is positively correlated with the robust generalization gap. These findings contribute to a deeper theoretical understanding of adversarial robustness in deep learning models.

2. Related Work

Adversarial Attacks and Defense. Since 2013, it has been well established that deep neural networks trained using standard gradient descent are highly vulnerable to small perturbations in input data (Szegedy et al., 2013; Goodfellow et al., 2014; Chen et al., 2017; Carlini and Wagner, 2017; Madry et al., 2017). Research on improving the robustness of neural networks has followed two main directions. One line of work focuses on developing defense mechanisms to enhance model robustness against adversarial attacks (Wu et al., 2020; Gowal et al., 2020). Another line aims to design stronger adversarial attacks to evaluate and challenge existing defenses (Athalye et al., 2018; Tramer et al., 2020; Chen et al., 2017; Xiao et al., 2022c).

Robust Generalization. Prior research has demonstrated that increasing the amount of training data can improve robust generalization (Schmidt et al., 2018; Raghunathan et al., 2019; Zhai et al., 2019). Several works have analyzed generalization in adversarial settings through the lens of VC-dimension (Attias et al., 2021; Montasser et al., 2019). Neyshabur et al. (2017b) applied a PAC-Bayesian framework to derive generalization bounds for neural networks, which was later extended to adversarial settings by Farnia et al. (2018); Xiao et al. (2023). Sinha et al. (2017) examined robust generalization in the context of distributional robustness, while Allen-Zhu and Li (2020) explored it from the perspective of feature purification. Additionally, Javanmard et al. (2020) studied generalization properties in the setting of linear regression.

Rademacher Complexity. Neyshabur et al. (2015) applied a layer-peeling technique to derive a generalization bound for depth-l neural networks. Specifically, assuming that the Frobenius norms of the weight matrices $W_1, W_2, \ldots, W_l$ are bounded by $M_1, \ldots, M_l$, and that all n input instances have ℓ_2-norm bounded by B, they showed that the generalization gap between the population risk and the empirical risk is bounded with high probability by $\mathcal{O}(B2^l\prod_{j=1}^{l}M_j/\sqrt{n})$. Additionally, Bartlett et al. (2017) provided a spectral norm-based bound on the Rademacher complexity by controlling the covering number of the function class of deep neural networks. The relevant work on Adversarial Rademacher Complexity is discussed in the Introduction, with further details provided in Appendix B. For a more detailed discussion, see Appendix B.3.

3. Preliminaries

3.1 Generalization Gap and Rademacher Complexity

Generalization Gap. In the classical machine learning framework, we consider a function class $\mathcal{F}$ (e.g., linear functions, neural networks). The learning objective is to find a function $f \in \mathcal{F}$ that minimizes the population risk:

$$R(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f(x), y)],$$

where $\mathcal{D}$ denotes the underlying data distribution and $\ell(\cdot)$ is the loss function. Since $\mathcal{D}$ is typically unknown, we minimize the empirical risk in practice. Given n independent and identically distributed (i.i.d.) samples $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the empirical risk is defined as:

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i).$$

The generalization gap is then defined as the difference between the population risk and the empirical risk:

$$\text{Generalization Gap} := R(f) - R_n(f).$$

Let the hypothesis class be defined as $\mathcal{H} = \{h \mid h(x, y) = \ell(f(x), y), f \in \mathcal{F}\}$, which connects the loss function to the function class. The Rademacher complexity framework leads to the following generalization bound.

Proposition 1 (Bartlett and Mendelson (2002))**.** *Let the loss function $\ell(f(x), y)$ be bounded with range $[0, C]$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following inequality holds for all $f \in \mathcal{F}$:*

$$R(f) \le R_n(f) + 2C\mathcal{R}_{\mathcal{S}}(\mathcal{H}) + 3C\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

3.2 Robust Generalization Gap and Adversarial Rademacher Complexity

Robust Generalization Gap. In the context of adversarial robustness, we define the robust population risk and robust empirical risk as follows:

$$\tilde{R}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\|x'-x\|_p\le\epsilon} \ell(f(x'), y) \quad \text{and} \quad \tilde{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \max_{\|x_i'-x_i\|_p\le\epsilon} \ell(f(x_i'), y_i).$$

Throughout this paper, we focus on general ℓ_p attacks where $p \ge 1$. We denote by $\mathcal{B}(x)$ the general perturbation set around point x. For ℓ_p attacks, this set is defined as $\mathcal{B}(x) = \{x' \mid \|x' - x\|_p \le \epsilon\}$. The robust generalization gap is then defined as:

$$\text{Robust Generalization Gap} := \tilde{R}(f) - \tilde{R}_n(f).$$

Let the adversarial loss be defined as $\tilde{\ell}(f(x), y) := \max_{x'\in\mathcal{B}(x)} \ell(f(x'), y)$. We then define the adversarial hypothesis class as:

$$\tilde{\mathcal{H}} = \left\{\tilde{h} : \tilde{h}(x, y) = \tilde{\ell}(f(x), y), f \in \mathcal{F}\right\}. \tag{2}$$

Then, according to Proposition 1, the robust generalization gap can be bounded by the Rademacher complexity of $\tilde{\mathcal{H}}$. We have the following robust generalization bound.

Proposition 2 (Yin et al. (2018)). *Let the loss function $\ell(f(x), y)$ be bounded with range $[0, C]$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following inequality holds for all $f \in \mathcal{F}$:*

$$\tilde{R}(f) \leq \tilde{R}_n(f) + 2C\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) + 3C\sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$

Definition 3 (Adversarial Rademacher Complexity). *Following Proposition 2, we define the Adversarial Rademacher Complexity (ARC) as the Rademacher complexity of the adversarial hypothesis class $\tilde{\mathcal{H}}$:*

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) = \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{\tilde{h}\in\tilde{\mathcal{H}}} \sum_{i=1}^{n} \sigma_i \tilde{h}(x_i', y_i)\right] = \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h\in\mathcal{H}} \sum_{i=1}^{n} \sigma_i \max_{x_i'\in\mathcal{B}(x_i)} h(x_i', y_i)\right],$$

The Rademacher complexity can be further upper bounded using the covering number, which we define as follows.

Definition 4 (ε-cover). *Let $\varepsilon > 0$ and $(\mathcal{V}, d(\cdot,\cdot))$ be a metric space, where $d(\cdot,\cdot)$ is a (pseudo)-metric. A subset $\mathcal{C} \subset \mathcal{V}$ is called a ε-cover of $\mathcal{V}$ if for any $v \in \mathcal{V}$, there exists $v' \in \mathcal{C}$ such that $d(v, v') \leq \varepsilon$. The ε-covering number of $\mathcal{V}$, denoted as $\mathcal{N}(\mathcal{V}, d(\cdot,\cdot), \varepsilon)$, is defined as the minimum cardinality $|\mathcal{C}|$ over all possible ε-covers*[1].

We now specialize this concept to hypothesis classes. Given a sample dataset $\mathcal{S}$, we define a pseudometric on $\mathcal{H}$ as: $\|h\|_{\mathcal{S}}^2 = \frac{1}{n}\sum_{i=1}^{n} h(x_i, y_i)^2$. The ε-covering number of $\mathcal{H}$ is then defined as $\mathcal{N}(\mathcal{H}, \|\cdot\|_{\mathcal{S}}, \varepsilon)$. We define the diameter of $\mathcal{H}$ as: $D \triangleq 2\max_{h\in\mathcal{H}} \|h\|_{\mathcal{S}}$.

Function Class. We consider depth-l, width-h fully-connected neural networks,

$$\mathcal{F} = \{x \mapsto W_l \rho\left(W_{l-1}\rho\left(\cdots \rho\left(W_1 x\right)\cdots\right)\right) \mid \|W_j\| \leq M_j,\ j = 1, \ldots, l\}. \tag{3}$$

where $\rho(\cdot)$ is an element-wise L_ρ-Lipschitz activation function and $\rho(\cdot) = 0$, W_j are $h_j \times h_{j-1}$ matrices, for $j = 1, \cdots, l$. h_0 equals to the input dimension d. Let $h = \max\{h_0, \cdots, h_l\}$ be the width of the neural networks. Denote the (a, b)-group norm $\|W\|_{a,b}$ as the a-norm of the b-norm of the rows of W. We consider two cases in Equation (3): the Frobenius norm and the $\|\cdot\|_{1,\infty}$-norm. The corresponding function classes are denoted as $\mathcal{F}_2$ and $\mathcal{F}_{1,\infty}$, respectively. Additionally, let the training data be $x_1, \cdots, x_n \in \mathbb{R}^d$. We assume that $\|X\|_{p,\infty} = B$, where X is a matrix of the all the samples, the i^{th} rows of x is x_i^T.

4. Main Challenges in Bounding ARC

In this section, we discuss the fundamental challenges encountered when calculating the covering number of an adversarial hypothesis class and present our approach to addressing these challenges.

4.1 Challenge 1: Distance Between Two Adversarial Functions

For simplicity, we consider the binary classification case and use a 1-dimensional function as an example. Following (Yin et al., 2018; Awasthi et al., 2020), let the loss function be $\ell(f(x), y) =$

1. We use two different Greek alphabet: ε for adversarial attacks and ϵ for covering number.

$\phi(yf(x))$, where ϕ is a non-increasing function. Then

$$\max_{x'} \ell\left(f\left(x'\right), y\right) = \phi\left(\min_{x'} yf\left(x'\right)\right).$$

Assume that the function ϕ is L_ϕ-Lipschitz, by Talagrand's Lemma (Ledoux and Talagrand, 2013), we have $\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) \leq L_\phi \mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{F}})$, where we define the adversarial hypothesis class as

$$\tilde{\mathcal{F}} = \left\{ \tilde{f} : (x, y) \mapsto \inf_{\|x - x'\|_p \leq \epsilon} yf\left(x'\right) \mid f \in \mathcal{F} \right\}. \tag{4}$$

Let $f_w(x) : \mathbb{R} \to \mathbb{R}$ with $|w| \leq M$ and its adversarial function $g_w(x) \triangleq \min_{x' \in [x-\epsilon, x+\epsilon]} f_w(x')$. Suppose we have only one sample x with $|x| = B$. Suppose $f_w(x)$ is L-Lipschitz w.r.t. w in x, e.g., $wx, \sin(w)x^5$. Note that $g_w(x)$ may not be Lipschitz continuous w.r.t. w. The problem is as follows:

Problem 1. *Bound the size of an ε-cover $\mathcal{N}(\tilde{\mathcal{F}}, \|\cdot\|_{\mathcal{S}}, \varepsilon)$ of the adversarial hypothesis class $\tilde{\mathcal{F}} = \left\{ g(x) \triangleq \min x' \in [x-\epsilon, x+\epsilon] f_w(\cdot) : |w| \leq M \right\}$ given sample x. Here, the ε-cover is a set of functions whose distance to any function in $\tilde{\mathcal{F}}$ is no more than ε.*

To help better understand Problem 1, we consider a simpler problem of a standard function class $\mathcal{F} = \{f_w(\cdot) : \mathbb{R} \to \mathbb{R} \mid |w| \leq M\}$.

Problem 2. *Bound the size of an ε-cover $\mathcal{N}(\mathcal{F}, \|\cdot\|_{\mathcal{S}}, \varepsilon)$ of $\mathcal{F} = \{f_w(\cdot) : \mathbb{R} \to \mathbb{R} \mid |w| \leq M\}$ given sample x.*

The idea is to build a relation between a cover of the function class and a cover of the parameter region, which is a subset of Euclidean space. For this purpose, we only need to relate the distance in the parameter space to the distance in the function space. More specifically, suppose $|w_1 - w_2| \leq \epsilon_w$, then $|f_{w_1}(x) - f_{w_2}(x)| \leq L|w_1 - w_2| \leq \epsilon_w L$. As a result, $|w_1 - w_2| \leq \epsilon_w = \varepsilon/L \Rightarrow |f_{w_1}(x) - f_{w_2}(x)| \leq \varepsilon$. This implies that an ϵ_w-cover of $[-M, M]$ leads to a ε-cover of $\mathcal{F}$. In the rest of the paper, we refer to $|f_{w_1}(x) - f_{w_2}(x)|$ as weight perturbation. The next step is to bound the size of the ϵ_w-cover of $[-M, M]$. In fact, $\{M, M - 2\epsilon_w, M - 4\epsilon_w, \dots\}$ is one such cover, with size no more than $2M/(2\epsilon_w) = M/\epsilon_w$. Therefore, for general L-Lipschitz function, the ε-cover of $\mathcal{F}$ is no more than $M/\epsilon_w = ML/\varepsilon$.

Linear function. For particular functions, the Lipschitz constant L needs to be estimated. We use the linear function as an example. In this case, $|f_{w_1}(x) - f_{w_2}(x)| = |w_1 x - w_2 x| = |w_1 - w_2||x| = |w_1 - w_2|B$. This implies that the Lipschitz constant $L = B$. Therefore, the ε-cover of $\mathcal{F}$ is no more than $M/\epsilon_w = MB/\varepsilon$.

Now we return to Problem 1. It is evident that the problem would be resolved if we knew the Lipschitz constant of the adversarial function. However, the adversarial function $g_w(x)$ might not be Lipschitz continuous. A small change in the input x might cause a large change in the function value. The following naive method illustrates the challenge and demonstrates why applying the standard approach fails to provide a bound.

Assume $f_w(x) = wx$. Let $x_i^* = \inf_{\|x - x'\| \leq \epsilon} f_{w_i}(x'), i = 1, 2$. A naive method is to use triangle inequality to get $|f_{w_1}(x_1^*) - f_{w_2}(x_2^*)| = |w_1 x_1^* - w_2 x_2^*| \leq |(w_1 - w_2)x_1^*| + |w_2(x_1^* - x_2^*)| \leq \epsilon_w B + 2M\epsilon \overset{\text{want}}{=} \varepsilon$. Thus an ε-cover of the function space can be built from an $((\varepsilon - 2M\epsilon)/B)$-cover

in w-space. This is only possible when $\varepsilon > M\epsilon$, thus we cannot build an ε-cover for arbitrarily small $\varepsilon > 0$. As a result, as $n \to \infty$, the corresponding ARC bound would have a non-vanishing term, which is undesirable.

The key issue seems to be controlling the extra term $|w_2(x_1^* - x_2^*)|$. For a linear function f_w, this can be resolved by obtaining the closed-form solutions of x_1^* and x_2^*. This is essentially the type 1 attempts. However, for general f_w (e.g., DNNs), the relation of the worst-perturbed points x_1^* and x_2^* is unclear.

Our solution to Problem 1 is provided as followed. Let

$$x_1^* = \arg \inf_{\|x-x'\| \le \epsilon} w_1 x', \quad \text{and} \quad x_2^* = \arg \inf_{\|x-x'\| \le \epsilon} w_2 x',$$

and

$$\bar{x} = \begin{cases} x_2^* & if \quad w_1 x_1^* \ge w_2 x_2^* \\ x_1^* & if \quad w_1 x_1^* < w_2 x_2^*. \end{cases}.$$

If $w_1 x_1^* \ge w_2 x_2^*$, we have $w_1 x_1^* - w_2 x_2^* \le w_1 x_2^* - w_2 x_2^* = w_1 \bar{x} - w_2 \bar{x}$. If $w_1 x_1^* < w_2 x_2^*$, we have $w_2 x_2^* - w_1 x_1^* \le w_2 x_1^* - w_1 x_1^* = w_2 \bar{x} - w_1 \bar{x}$. The choice of $\bar{x}$ is illustrated in Figure 1. Combine these two inequalities, we have

$$|w_1 x_1^* - w_2 x_2^*| \le |w_1 \bar{x} - w_2 \bar{x}| \le |w_1 - w_2| \, |\bar{x}| \le \epsilon_w \, (B + \epsilon) \, . \tag{5}$$

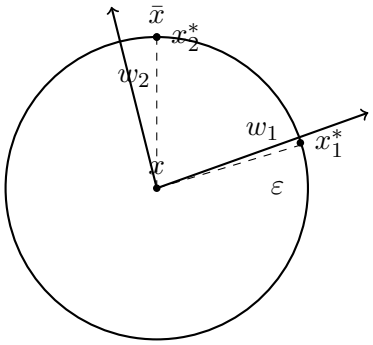


Figure 1: A schematic illustration of points x_1^* and x_2^* in the ball $\|x' - x\| \le \varepsilon$ that minimize $w_1 x'$ and $w_2 x'$, respectively. The point $\bar{x}$ is chosen depending on which inner product is smaller. The inequality $|w_1 x_1^* - w_2 x_2^*| \le |w_1 \bar{x} - w_2 \bar{x}| \le |w_1 - w_2| \|\bar{x}\| \le \varepsilon_w (B + \varepsilon)$ follows from comparing x_1^*, x_2^*, and $\bar{x}$.

Therefore, the ε-cover of $\tilde{\mathcal{F}}$ is no more than $M/\epsilon_w = M(B + \epsilon)/\varepsilon$.

Remark 1. This approach recovers the upper bound of ARC for linear functions, as established in Khim and Loh (2018); Yin et al. (2018). The main advantage of this approach is that it provides a bridge for computing the distance between two adversarial functions without requiring the closed-form solution of adversarial examples. Consequently, this method shows potential for extension to multi-layer neural networks. However, we will demonstrate an additional challenge prevents the direct application of this approach to multi-layer neural networks in the following subsection. Nevertheless, our proof reveals that the definition of $\bar{x}$ is a crucial step. We refer to this as the *intermediate adversarial example* and present the corresponding lemma for general functions below.

Lemma 1 (Intermediate Adversarial Example)**.** *Given (x, y) and perturbation set $\mathcal{B}(x)$. For all $\tilde{h}_1, \tilde{h}_2 \in \tilde{\mathcal{H}}$ with their standard counterparts $h_1, h_2 \in \mathcal{H}$, there exists an adversarial example $x'(\tilde{h}_1, \tilde{h}_2) \in \mathcal{B}(x)$, s.t.*

$$\left|\tilde{h}_1(x, y) - \tilde{h}_2(x, y)\right| \leq \left|h_1\left(x'\left(\tilde{h}_1, \tilde{h}_2\right), y\right) - h_2\left(x'\left(\tilde{h}_1, \tilde{h}_2\right), y\right)\right|.$$

We refer to this adversarial example $x'(\tilde{h}_1, \tilde{h}_2) \in \mathcal{B}(x)$ as intermediate adversarial example.

4.2 Challenge 2: Weight-Dependent Nature of Adversarial Examples

While our approach successfully bounds the ARC for linear functions using intermediate adversarial examples, extending this method to DNNs presents inherent challenges. The primary difficulty arises from a fundamental conflict between the intermediate adversarial example approach and established methods for bounding DNN Rademacher complexity, such as layer peeling and covering number techniques. The conflict stems from the dynamic nature of intermediate adversarial examples: the point $\bar{x}$ varies as we move from $(l-1)$-layer to l-layer networks. This variability contradicts a key requirement of traditional methods, which rely on fixed inputs at each layer. We use the covering number approach by Bartlett et al. (2017) to illustrate this challenge. Let X_i denote the fixed output of the i-th layer. The covering number of the hypothesis class $\mathcal{H}$ are derived through induction, expressing the bound in Eq. (6) as a sum of covering numbers over matrix space of $W_j x_{j-1}$.

$$\ln \mathcal{N}\left(\mathcal{H}, \|\cdot\|_{\mathcal{S}}, \varepsilon\right) \leq \sum_{j=1}^{l} \sup_{(W_1, \ldots, W_{j-1})} \ln \mathcal{N}\left(\left\{W_j x_{j-1} : \left\|W_j^\top\right\|_{2,1} \leq a_j\right\}, \|\cdot\|, \delta_j\right). \quad (6)$$

for some ε. In the adversarial setting, however, x_{j-1} is not fixed; it depends on both the weights of the preceding layers $(W_1, \ldots, W_{j-1})$ and the weights of subsequent layers $(W_j, \ldots, W_l)$. This interdependence between layers prevents the direct application of traditional covering number bounds to the adversarial setting. The same issue arises in the layer peeling approach, as detailed in Appendix B. To address this challenge, we propose an alternative decomposition based on Lemma 1, which bounds the covering number of the adversarial hypothesis class using the covering number of the weight space.

Lemma 2 (Layer-wise Induction for Adversarial Hypothesis Class)**.** *Let $(\delta_1, \ldots, \delta_l)$ be given, along with Lipschitz activation function ρ (where ρ is L_ρ-Lipschitz and $\rho(0) = 0$). Let the loss function $\ell(f(x), y)$ be L_ϕ-Lipschitz with respect to the first argument. Let the function class be $\mathcal{F}_2$ or $\mathcal{F}_{1,\infty}$ and the adversarial hypothesis class be defined in* (2)*. Define*

$$\varepsilon = L_\phi \sum_{j=1}^{l} L_\rho^{l-1} \frac{\prod_{k=1}^{l} M_k}{M_j} \max\left\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\right\} \left(\|x\|_{p,\infty} + \epsilon\right) \delta_j, \quad (7)$$

where $r = 2$ for Frobenius norm and $r = 1$ for $(1, \infty)$-norm. Then:

$$\ln\left(\mathcal{N}\left(\tilde{\mathcal{H}}, \|\cdot\|_{\mathcal{S}}, \varepsilon\right)\right) \leq \sum_{j=1}^{l} \ln\left(\mathcal{N}\left(\left\{W_j \mid \|W_j\| \leq M_j\right\}, \|\cdot\|, \delta_j\right)\right).$$

Since the upper bound is expressed as a sum of covering numbers over the weight spaces W_j rather than the weight-input products $W_j x_{j-1}$, it effectively resolves the issue of input dependency on subsequent layer weights.

5. Bounds for Adversarial Rademacher Complexity

5.1 Binary Classification

We begin by establishing bounds for the ARC in the binary classification setting.

Theorem 5 (Frobenius Norm Bound)**.** *Consider the function class $\mathcal{F}_2$, and its corresponding adversarial function class $\tilde{\mathcal{H}}$ in Eq.* (2)*. The ARC of deep neural networks, $\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}})$, satisfies*

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) \leq \frac{24L_\phi}{\sqrt{n}} \max\left\{1, q^{\frac{1}{2}-\frac{1}{p}}\right\} \left(\|X\|_{p,\infty} + \epsilon\right) L_\rho^{l-1} \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l)} \prod_{j=1}^{l} M_j.$$

Furthermore, under the assumptions that $L_\phi = 1$, $L_\rho = 1$, $p \leq 2$, $\|X\|_{p,\infty} = B$, and $h = \max\{h_0, \cdots, h_l\}$, we have

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) \leq \mathcal{O}\left(\frac{(B+\epsilon)h\sqrt{l\log(l)}\prod_{j=1}^{l} M_j}{\sqrt{n}}\right). \tag{8}$$

The proof is based on bounding the Rademacher complexity using the covering number, which is known as Dudley's integral. Specifically, we proceed as follows:

- We first establish an upper bound on the distance between two adversarial functions using Lemma 1 (Intermediate Adversarial Example).
- Next, we apply Lemma 2 to relate the covering number of the adversarial hypothesis class to the covering number of the weight norm.
- This reduces the problem to bounding the covering number of a norm ball, a well-established result in mathematical analysis.

The complete proof is provided in Appendix A.

Theorem 6 ($(1,\infty)$-norm Bound)**.** *Consider the function class $\mathcal{F}_{1,\infty}$, and its corresponding adversarial function class $\tilde{\mathcal{H}}$ in Eq.* (2)*. The ARC of deep neural networks, $\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}})$, satisfies*

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) \leq \frac{24}{\sqrt{n}} \left(\|X\|_{p,\infty} + \epsilon\right) L_\rho^{l-1} \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l)} \prod_{j=1}^{l} M_j.$$

In the case of the $(1,\infty)$-norm, the bound is similar to that of the Frobenius norm, except for the additional term $\max\left\{1, d^{1/2-1/p}\right\}$. Therefore, for all $p \geq 1$, the $(1,\infty)$-norm bound maintains the same order as in Eq. (8).

We compare our bound to the bounds in similar settings. Specifically, we compare our bound with the covering number bounds for (standard) Rademacher complexity (Bartlett et al., 2017) and the bound of ARC in two-layer cases.

Covering Number Bound for Standard Rademacher complexity. The work of Bartlett et al. (2017) used a covering number argument to show that the generalization gap is bounded by

$$\tilde{\mathcal{O}}\left(\frac{B\prod_{j=1}^{l}\|W_j\|}{\sqrt{n}}\left(\sum_{j=1}^{l}\frac{\|W_j\|_{2,1}^{2/3}}{\|W_j\|^{2/3}}\right)^{3/2}\right).$$

Our bound differs in two key aspects. First, it includes an additional dependence on ϵ, which is unavoidable in adversarial settings. Second, as discussed in Neyshabur et al. (2017b) and Golowich et al. (2018), the term $\left(\sum_{j=1}^{l} \frac{\|W_j\|_{2,1}^{2/3}}{\|W_j\|^{2/3}}\right)$ admits the following bounds:

$$l^{\frac{3}{2}} \leq \left(\sum_{j=1}^{l} \frac{\|W_j\|_{2,1}^{2/3}}{\|W_j\|^{2/3}}\right)^{3/2} \leq l^{\frac{3}{2}} h.$$

On the other hand, our bound exhibits a dependence on network size of $\mathcal{O}(\sqrt{l\log(l)}h)$. The main difference arises from the need for two distinct approaches to perform layer-wise induction in standard and adversarial settings. Compared to the upper bound on network size dependence in existing standard results, our bound maintains a comparable dependence on depth l and width h.

Bound for ARC in Two-Layer Cases. The work of Awasthi et al. (2020) established that the ARC is bounded by

$$\mathcal{O}\left(\frac{(B+\epsilon)\sqrt{h_1 d}\sqrt{\log n}M_1 M_2}{\sqrt{n}}\right)$$

in the two-layer setting. Applying our bound to the two-layer case, we obtain

$$\mathcal{O}\left(\frac{(B+\epsilon)\sqrt{h_1 d}M_1 M_2}{\sqrt{n}}\right),$$

which is strictly tighter. Notably, under the same conditions for other factors, our bound exhibits a lower dependence on the sample size n.

Theorem 7 (Lower Bound)**.** *Consider the function class $\mathcal{F}_2$. Let $\tilde{\mathcal{F}}_2$ denote its corresponding adversarial function class as defined in Eq.* (4)*. There exists an activation function and a dataset S such that the ARC of deep neural networks satisfies*

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{F}}_2) \geq \Omega\left(\frac{(B+\epsilon)\prod_{j=1}^{l} M_j}{\sqrt{n}}\right).$$

The proof is provided in Appendix A. For function classes $\mathcal{F}_{1,\infty}$ with the $(1,\infty)$-norm, ARC admits the same lower bound. From this bound, we observe a gap in depth l and width h between the upper and lower bounds, while the other terms remain unavoidable. In the next section, we extend the ARC analysis to multi-class classification.

5.2 Multi-Class Classification

The setting for multi-class classification follows (Bartlett and Mendelson, 2002). In a K-class classification problem, let $\mathcal{Y} = \{1, 2, \cdots, K\}$. The functions in the hypothesis class $\mathcal{F}$ map $\mathcal{X}$ to $\mathbb{R}^K$, the k-th output of f is the score of $f(x)$ assigned to the k-th class.

Define the margin operator $M(f(x), y) = [f(x)]_y - \max_{y' \neq y}[f(x)]_{y'}$. The function makes a correct prediction if and only if $M(f(x), y) > 0$. We consider a particular loss function $\ell(f(x), y) = \phi_\gamma(M(f(x), y))$, where $\gamma > 0$ and $\phi_\gamma : \mathbb{R} \to [0, 1]$ is the ramp loss:

$$\phi_\gamma(t) = \begin{cases} 1 & t \leq 0 \\ 1 - \frac{t}{\gamma} & 0 < t < \gamma \\ 0 & t \geq \gamma. \end{cases}$$

$\phi_\gamma(t) \in [0,1]$ and $\phi_\gamma(\cdot)$ is $1/\gamma$-Lipschitz. The loss function $\ell(f(x), y)$ satisfies:

$$\mathbb{1}\left(y \neq \arg\max_{y' \in [K]} [f(x)]_{y'}\right) \leq \ell\left(f(x), y\right) \leq \mathbb{1}\left([f(x)]_y \leq \gamma + \max_{y' \neq y} [f(x)]_{y'}\right).$$

Define the function class $\ell_{\mathcal{F}} := \{(x,y) \mapsto \phi_\gamma(M(f(x),y)) : f \in \mathcal{F}\}$. In adversarial training, the adversarial hypothesis class is defined as

$$\tilde{\ell}_{\mathcal{F}} := \left\{(x,y) \mapsto \max_{x' \in \mathcal{B}(x)} \phi_\gamma\left(M\left(f(x'), y\right)\right) : f \in \mathcal{F}\right\}. \tag{9}$$

Then, the following generalization bound holds.

Corollary 8 ((Yin et al., 2018)). *Consider the above adversarial multi-class classification setting. For any fixed $\gamma > 0$, we have with probability at least $1-\delta$, for all $f \in \mathcal{F}$,*

$$\begin{aligned}
&\mathbb{P}_{(x,y)\sim\mathcal{D}}\left\{\exists x' \in \mathbb{B}_x^p(\epsilon) \text{ s.t. } y \neq \arg\max_{y' \in [K]} \left[f(x')\right]_{y'}\right\} \\
\leq &\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left(\exists x_i' \in \mathbb{B}_{x_i}^p(\epsilon) \text{ s.t. } \left[f(x_i')\right]_{y_i} \leq \gamma + \max_{y' \neq y} \left[f(x_i')\right]_{y'}\right) \\
&+ 2\mathcal{R}_S(\tilde{\ell}_{\mathcal{F}}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.
\end{aligned}$$

Using the same idea as in binary settings, we can calculate the covering number of the adversarial hypothesis class via intermediate adversarial examples. Then, we have the following bound for ARC.

Theorem 9. *Given the function class $\mathcal{F}_2$, and the corresponding adversarial hypothesis class $\tilde{\ell}_{\mathcal{F}}$ in Eq.* (9)*, the ARC of deep neural networks $\mathcal{R}_S(\tilde{\ell}_{\mathcal{F}})$ satisfies*

$$\mathcal{R}_S(\tilde{\ell}_{\mathcal{F}}) \leq \frac{48}{\gamma\sqrt{n}} \max\left\{1, q^{\frac{1}{2}-\frac{1}{p}}\right\} \left(\|X\|_{p,\infty} + \epsilon\right) L_\rho^{l-1} \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l)} \prod_{j=1}^{l} M_j. \tag{10}$$

The proof is based on the fact that $\phi_\gamma(M(\cdot, y))$ is $2/\gamma$-Lipshitz, which is provided in Appendix A. Notably, our bound is independent of the number of classes K, improving from $\mathcal{O}(K)$ (Yin et al., 2018) to $\mathcal{O}(1)$. Comparing Theorems 9 and 5, we observe that the bound for the multiclass setting is as tight as that for the binary case. Therefore, from the perspective of ARC, increasing the number of classes does not impact adversarially robust generalization.

6. Experiments

6.1 Comparing Standard and Adversarial Rademacher Complexity

We now examine the relationship between the bounds for standard and adversarial Rademacher complexity. We begin by recalling the upper bound for standard Rademacher complexity from Golowich et al. (2018):

$$\mathcal{R}_S(\mathcal{H}) \leq \mathcal{O}\left(\frac{B\sqrt{l}\prod_{j=1}^{l} M_j}{\gamma\sqrt{n}}\right). \tag{11}$$

Next, we categorize the factors in the bounds into two groups.

Algorithm-Independent Factors. The bounds include five algorithm-independent factors: the number of samples n, depth l, width h, sample size B, and perturbation intensity ϵ. For notational convenience, we define $C_{std} = B\sqrt{l}/\sqrt{n}$ and $C_{adv} = (B+\epsilon)h\sqrt{l\log l}/\sqrt{n}$ as the constants for standard and adversarial Rademacher complexity, respectively. By definition, $C_{adv} > C_{std}$.

Algorithm-Dependent Factors. There are two remaining terms: the product of upper bounds on matrix norms, $\prod_{j=1}^{l} M_j$, and the margin, γ. By definition, these terms are independent of the algorithm. However, they are implicitly algorithm-dependent. The bound is universal and holds for all neural networks with weights satisfying $\|W_j\| \leq M_j$ for $j = 1, \ldots, l$. Conversely, once a neural network is trained with specific weight norms $\|W_j\|$, we can set $M_j = \|W_j\|$ for all j, ensuring that the bound applies specifically to the trained network and is the tightest possible bound for it. Similarly, γ is set to the margin of the trained network. Thus, the two algorithm-dependent factors in the bound are the product of weight norms and the margin. We define the ratio $W_{std} := \prod_{j=1}^{l} \|W_j\|/\gamma$ for standard training and $W_{adv} := \prod_{j=1}^{l} \|W_j\|/\gamma$ for adversarial training. Our experimental results, presented in the next subsection and Appendix C, consistently show that $W_{adv} > W_{std}$.

Generalization Gap Analysis. Let $\mathcal{E}(\cdot)$ denote the standard generalization gap and $\tilde{\mathcal{E}}(\cdot)$ represent the robust generalization gap. We use f_{std} and f_{adv} to denote models trained using standard and adversarial training, respectively. Our analysis aims to understand why adversarially trained models exhibit substantially larger robust generalization gaps compared to the standard generalization gaps of normally trained models (i.e., why $\tilde{\mathcal{E}}(f_{adv}) > \mathcal{E}(f_{std})$). While prior work (Zhang et al., 2021) has established that Rademacher complexity is large in these settings, we can still analyze the relationship between robust generalization gaps and our identified factors through the standard and ARC bounds:

$$\tilde{\mathcal{E}}(f_{adv}) \propto C_{adv}W_{adv} \quad and \quad \mathcal{E}(f_{std}) \propto C_{std}W_{std}.$$

Notably, the bounds apply universally to any model. We can analyze the standard Rademacher complexity bound for adversarially trained models, i.e., $C_{std}W_{adv}$, and vice versa. To isolate the individual effects of factors C_{adv} and W_{adv}, we examine two additional generalization gaps: the robust generalization gap of standard-trained models ($\tilde{\mathcal{E}}(f_{std})$) and the standard generalization gap of adversarially-trained models ($\mathcal{E}(f_{adv})$). These gaps help decompose the contributions of each factor:

$$\tilde{\mathcal{E}}(f_{std}) \propto C_{adv}W_{std} \quad and \quad \mathcal{E}(f_{adv}) \propto C_{std}W_{adv}.$$

In the previous section, we identified the normalized product of weight norms $\prod_{j=1}^{l} \|W_j\|/\gamma$ as a key algorithm-dependent factor in the ARC bounds. To empirically validate our theoretical analysis, we conducted extensive experiments comparing these terms between standard and adversarial training settings. Since our bounds generalize to convolutional neural networks, we evaluated VGG architectures (Simonyan and Zisserman, 2014) on both CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009). Our analysis encompasses 88 trained models, with additional experimental results provided in Appendix C.

Training Protocol. We employed SGD optimization with a three-stage learning rate schedule: 0.1 for the first 100 epochs, 0.01 for the next 50 epochs, and 0.001 for the final 50 epochs. Weight decay was primarily set to 5×10^{-4}, which was empirically determined to be optimal for robust accuracy, though we explored other values in ablation studies. For adversarial training, we implemented ℓ_∞ PGD (Madry et al., 2017) with $\epsilon = 8/255$ perturbation intensity, using 20 steps during training and 40 steps during testing, with a step size of $2/255$ for inner maximization.

Margin Computation. Following Neyshabur et al. (2017a), we defined margins differently for standard and adversarial training. For standard training, the margin was calculated as the 5th percentile of $f(x_i)[y_i] - \max_{y \neq y_i} f(x)[y]$ across all training points. For adversarial training, we used the 5th percentile of margins computed on PGD-adversarial examples. Since both the standard-trained and adversarially-trained models achieved 100% training accuracy, the margins of all samples are positive. This ensures that the 5th percentile of margins is also positive. the Detailed ablation studies on percentile selection are presented in Appendix C.3.

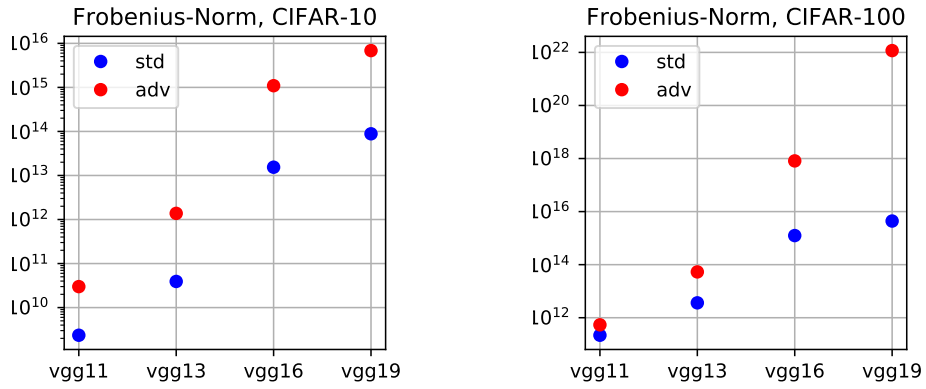


Figure 2: Comparison of Frobenius norms between standard and adversarial training models on CIFAR-10 and CIFAR-100 datasets.

Higher Weight Norms in Adversarially-Trained Models. Figure 2 compares weight norms between standard and adversarial training across VGG architectures on both CIFAR-10 and CIFAR-100 datasets[2]. Using a logarithmic scale for visualization, our results consistently demonstrate that adversarially-trained models have larger weight norms than their standard-trained counterparts ($W_{adv} \geq W_{std}$). Additional ablation studies in Appendix C further confirm this relationship across different experimental conditions.

Analysis of Standard and Robust Generalization Gaps. Table 2 presents standard and robust generalization gaps for both training approaches, using VGG-19 on CIFAR-10 as our primary example. Standard-trained models exhibit small standard generalization gaps ($\mathcal{E}(f_{std}) = 10.45\%$), while adversarially-trained models show larger standard generalization gaps ($\mathcal{E}(f_{adv}) = 26.34\%$). This increased gap aligns with the known phenomenon that adversarial training typically compromises standard generalization, possibly due to overfitting to adversarial examples. The robust generalization gap presents a striking contrast: standard-trained models show minimal robust generalization gaps ($\tilde{\mathcal{E}}(f_{std}) = 0$), but this is not indicative of good performance. Rather, it reflects uniformly poor robustness, with both training and test robust accuracy approaching 0%. Conversely, adversarially-trained models exhibit large robust generalization gaps ($\tilde{\mathcal{E}}(f_{adv}) = 58.90\%$). This substantial gap in robust generalization is a key phenomenon that we aim to analyze and explain.

Interpreting Zero Robust Generalization Gap in Standard Training. Table 2 reveals that $\tilde{\mathcal{E}}(f_{std}) = 0\%$ represents a degenerate case where standard-trained models achieve 100% ro-

2. While larger models naturally exhibit higher weight norms due to their increased parameter count, our focus is on the relative difference between adversarially-trained and standard-trained models.

Table 2: Comparison of four generalization gaps for VGG-19 trained on CIFAR-10: standard and robust gaps for both training methods. Note: $\tilde{\mathcal{E}}(f_{std}) = 0\%$ reflects complete model failure (100% training error), while other cases achieve near-zero training errors.

	Standard-trained models		Adversarially-trained models	
Types of Generalization Gaps	Standard	Robust	Standard	Robust
Training Errors	0%	100%	0%	0.02%
Test Errors	10.45%	100%	26.34%	58.92%
Generalization Gaps	$\mathcal{E}(f_{std})$=10.45%	$\tilde{\mathcal{E}}(f_{std})$=0%	$\mathcal{E}(f_{adv})$=26.34%	$\tilde{\mathcal{E}}(f_{adv})$=58.90%

bust training error, indicating complete failure to fit any adversarial examples in the training set. This renders both the generalization gap and its corresponding Rademacher complexity bound $\tilde{\mathcal{E}}(f_{std}) \leq \mathcal{O}(C_{adv}W_{std}/\sqrt{n})$ trivial. In contrast, the other three cases achieve near-zero training errors, providing meaningful generalization gaps. We focus our analysis on understanding why $\tilde{\mathcal{E}}(f_{adv}) > \mathcal{E}(f_{std})$ by examining the relationship $\tilde{\mathcal{E}}(f_{adv}) > \mathcal{E}(f_{adv}) > \mathcal{E}(f_{std})$.

Impact of C_{adv} on Generalization Gaps. Comparing generalization gaps for adversarially-trained models, we observe that the robust generalization gap significantly exceeds the standard generalization gap ($\tilde{\mathcal{E}}(f_{adv}) = 58.90\% > \mathcal{E}(f_{adv}) = 26.34\%$). Using Rademacher complexity bounds as approximations for these gaps, we can express this relationship as $\tilde{\mathcal{E}}(f_{adv}) \propto C_{adv}W_{adv}$ and $\mathcal{E}(f_{adv}) \propto C_{std}W_{adv}$. This suggests that C_{adv} directly contributes to the increased robust generalization gap, as it scales with the perturbation intensity ϵ.

Impact of W_{adv} on Standard Generalization. When comparing standard generalization gaps, we observe that adversarially-trained models exhibit poorer generalization compared to standard training ($\mathcal{E}(f_{adv}) = 26.34\% > \mathcal{E}(f_{std}) = 10.45\%$). This widely observed degradation in standard generalization can be understood through Rademacher complexity bounds. Using these bounds as approximations ($\mathcal{E}(f_{adv}) \propto C_{std}W_{adv}$ and $\mathcal{E}(f_{std}) \propto C_{std}W_{std}$), we can attribute the increased generalization gap to larger weight norms in adversarially-trained models (W_{adv}), demonstrating its positive correlation with generalization degradation.

The relationship between generalization gaps ($\tilde{\mathcal{E}}(f_{adv}) > \mathcal{E}(f_{adv}) > \mathcal{E}(f_{std})$) can be characterized by the corresponding complexity terms: $C_{adv}W_{adv} > C_{std}W_{adv} > C_{std}W_{std}$. This analysis reveals that robust generalization challenges stem from two distinct sources: (1) an algorithm-independent component C_{adv}, which is inherent to the minimax nature of adversarial training and thus unavoidable, and (2) an algorithm-dependent component W_{adv}, reflecting increased weight norms in adversarially-trained models, which might be addressable through improved training techniques.

Role of Weight Decay. Our analysis suggests that controlling weight norms could improve robust generalization. In Appendix C.4, we investigate the effects of incrementally increasing weight decay. While larger weight decay values reduce weight norms and improve generalization, they also degrade training performance, revealing a fundamental trade-off between training accuracy and generalization. At weight decay of 10^{-2}, training fails completely, though notably, even in this regime, adversarially-trained models maintain larger weight norms than standard-trained models.

Neural Network Representation Capacity. We hypothesize that the increased weight norms in adversarial training stem from fundamental representation requirements: neural networks with

small weight norms appear insufficient to fit adversarial examples in the training set, forcing the optimization to converge to solutions with larger weight norms. However, as our analysis shows, these large-norm solutions typically exhibit poor generalization properties, leading to robust overfitting. While this hypothesis aligns with our observations, comprehensive validation would require large-scale experimentation beyond our current scope.

7. Conclusion

Limitation. The main limitation is that norm-based bounds tend to be excessively large in practical scenarios. As shown in Figure 2, the bounds for VGG networks exceed 10^9 in experiments on the CIFAR-10 dataset. The key challenge is how to obtain tighter norm-based bounds in real-world settings, not only for adversarial robustness but also in standard scenarios. This remains an open problem.

We present the first bounds on adversarial Rademacher complexity for deep neural networks, providing new theoretical insights into robust generalization. Our analysis reveals that robust generalization challenges arise from two distinct sources: an algorithm-independent factor inherent to the adversarial setting, and an algorithm-dependent factor related to neural network weight norms. Through extensive empirical validation, we establish clear correlations between these factors and robust generalization performance. These findings open new directions for both theoretical research in understanding adversarial training and practical improvements in robust generalization methods.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190*, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. 2021.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. In *International Conference on Machine Learning*, pages 431–441. PMLR, 2020.

Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

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

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of evasion adversaries. *arXiv preprint arXiv:1806.01471*, 2018.

Chen Dan, Yuting Wei, and Pradeep Ravikumar. Sharp statistical guaratees for adversarially robust gaussian classification. In *International Conference on Machine Learning*, pages 2345–2355. PMLR, 2020.

Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2018.

Qingyi Gao and Xiao Wang. Theoretical investigation of generalization bounds for adversarial learning of deep neural networks. *Journal of Statistical Theory and Practice*, 15(2):1–28, 2021.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pages 297–299. PMLR, 2018.

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

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

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.

Justin Khim and Po-Ling Loh. Adversarial risk bounds via function transformation. *arXiv preprint arXiv:1810.09519*, 2018.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*, 2018.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

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. *arXiv preprint arXiv:1801.02613*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017a.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017b.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2, 2017.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

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.

Hossein Taheri, Ramtin Pedarsani, and Christos Thrampoulidis. Asymptotic behavior of adversarial training in binary classification. *arXiv preprint arXiv:2010.13275*, 2020.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *arXiv preprint arXiv:2004.05884*, 2020.

Jiancong Xiao, Yanbo Fan, Ruoyu Sun, Jue Wang, and Zhi-Quan Luo. Stability analysis and generalization bounds of adversarial training. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022a. URL https://openreview.net/forum?id=78aj7sPX4s-.

Jiancong Xiao, Zeyu Qin, Yanbo Fan, Baoyuan Wu, Jue Wang, and Zhi-Quan Luo. Adaptive smoothness-weighted adversarial training for multiple perturbations with its stability analysis. *arXiv preprint arXiv:2210.00557*, 2022b.

Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding adversarial robustness against on-manifold adversarial examples. *arXiv preprint arXiv:2210.00430*, 2022c.

Jiancong Xiao, Jiawei Zhang, Zhi-Quan Luo, and Asuman E. Ozdaglar. Smoothed-SGDmax: A stability-inspired algorithm to improve adversarial generalization. In *NeurIPS ML Safety Workshop*, 2022d. URL https://openreview.net/forum?id=4rksWKdGovR.

Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. PAC-bayesian adversarially robust generalization bounds for deep neural networks. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023. URL https://openreview.net/forum?id=CG0oM1LmbP.

Jiancong Xiao, Jiawei Zhang, Zhi-Quan Luo, and Asuman E. Ozdaglar. Uniformly stable algorithms for adversarial training and beyond. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 54319–54340. PMLR, 21–27 Jul 2024.

Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=xz80iPFIjvG.

Dong Yin, Kannan Ramchandran, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. *arXiv preprint arXiv:1810.11914*, 2018.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pages 7085–7094. PMLR, 2019.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, 2021.

A. Proofs of Technical Results

A.1 Proof of Lemma 1

Proof Let

$$x(\tilde{h}_1) = \arg\max_{x' \in \mathcal{B}(x)} h_1(x', y), \quad x(\tilde{h}_2) = \arg\max_{x' \in \mathcal{B}(x)} h_2(x', y).$$

Then,

$$\left|\tilde{h}_1(x,y) - \tilde{h}_2(x,y)\right| \leq \max\left\{\left|h_1(x(\tilde{h}_1),y) - h_2(x(\tilde{h}_1),y)\right|, \left|h_1(x(\tilde{h}_2),y) - h_2(x(\tilde{h}_2),y)\right|\right\}.$$

It is because

$$h_1(x(\tilde{h}_1),y) - h_2(x(\tilde{h}_2),y) \leq h_1(x(\tilde{h}_1),y) - h_2(x(\tilde{h}_1),y)$$

and

$$h_2(x(\tilde{h}_2),y) - h_1(x(\tilde{h}_1),y) \leq h_2(x(\tilde{h}_2),y) - h_1(x(\tilde{h}_2),y).$$

Let

$$\bar{x}(\tilde{h}_1,\tilde{h}_2) = \begin{cases} x(\tilde{h}_1), \text{ if } h_1(x(\tilde{h}_1),y) \geq h_2(x(\tilde{h}_2),y) \\ x(\tilde{h}_2), \text{ if } h_1(x(\tilde{h}_1),y) < h_2(x(\tilde{h}_2),y). \end{cases} \tag{12}$$

We have

$$\left|\tilde{h}_1(x,y) - \tilde{h}_2(x,y)\right| \leq \left|h_1(\bar{x}(\tilde{h}_1,\tilde{h}_2),y) - h_2(\bar{x}(\tilde{h}_1,\tilde{h}_2),y)\right|.$$

The expression in Eq. (12) is the intermediate adversarial examples. ■

A.2 Proof of Lemma 2

The proof of Lemma 2 requires the following Lemma.

Lemma 3 (Awasthi et al. (2020), cf. Lemma 1)**.** *If* $x_i^* \in \{x_i' \mid \|x_i - x_i'\|_p \leq \epsilon\}$*, then*

$$\|x_i^*\|_{r^*} \leq \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}(\|X\|_{p,\infty} + \epsilon).$$

Proof If $p \geq r^*$, applying Hölder's inequality with $1/r^* = 1/p + 1/s$, we obtain

$$\|x_i^*\|_{r^*} \leq \sup \|\mathbf{1}\|_s \|x_i^*\|_p = \|\mathbf{1}\|_s \|x_i^*\|_p = d^{\frac{1}{s}} \|x_i^*\|_p = d^{1-\frac{1}{r}-\frac{1}{p}} \|x_i^*\|_p.$$

Equality holds when all entries are equal. If $p < r^*$, we have

$$\|x_i^*\|_{r^*} \leq \|x_i^*\|_p.$$

Equality holds when one entry equals one, and all others are zero. Thus,

$$\begin{aligned} \|x_i^*\|_{r^*} &\leq \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}\|x_i^*\|_p \\ &\leq \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}(\|x_i\|_p + \|x_i - x_i^*\|_p) \\ &\leq \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}(\|X\|_{p,\infty} + \epsilon). \end{aligned}$$

■

Lemma 4. *Let A be an $m \times k$ matrix and b be an n-dimensional vector. Then,*

$$\|Ab\|_2 \leq \|A\|_F \|b\|_2.$$

Proof Let A_i denote the rows of A for $i = 1, \ldots, m$. Then,

$$\|Ab\|_2 = \sqrt{\sum_{i=1}^{m} (A_i b)^2} \leq \sqrt{\sum_{i=1}^{m} \|A_i\|_2^2 \|b\|_2^2} = \sqrt{\sum_{i=1}^{m} \|A_i\|_2^2} \cdot \sqrt{\|b\|_2^2} = \|A\|_F \|b\|_2.$$

■

Lemma 5. *Let A be an $m \times k$ matrix and b be an n-dimensional vector. Then,*

$$\|Ab\|_\infty \leq \|A\|_{1,\infty} \|b\|_\infty.$$

Proof Let A_i denote the rows of A for $j = 1, \ldots, m$. Then,

$$\|Ab\|_\infty = \max |A_i b| \leq \max \|A_i\|_1 \|b\|_\infty = \|A\|_{1,\infty} \|b\|_\infty.$$

■

Now, we move the the proof of Lemma 2.

Proof In Frobenius norm case, let $r = 2$ and $\mathcal{C}_j$ denote δ_j-covers of the set $\{\|W_j\|_F \leq M_j\}$, for $j = 1, 2, \cdots, l$. Define

$$\mathcal{F}^c = \left\{ f^c : x \mapsto W_l^c \rho \left(W_{l-1}^c \rho \left(\cdots \rho \left(W_1^c x \right) \cdots \right) \right), W_j^c \in \mathcal{C}_j,\ j = 1, 2, \ldots, l \right\};$$

In $(1, \infty)$-norm case, let $r = 1$ and $\mathcal{C}_j^m$ be δ_j-covers of $\{\|W_j^m\|_1 \leq M_j\}$, $j = 1, 2, \cdots, l$, $m = 1, \cdots, h_j$, where W_j^m is the m^{th} row of W_j^m. Let

$$\mathcal{F}^c = \left\{ f^c : x \mapsto W_l^c \rho \left(W_{l-1}^c \rho \left(\cdots \rho \left(W_1^c x \right) \cdots \right) \right), W_j^{cm} \in \mathcal{C}_j^m,\ m = 1, \ldots, h_j,\ j = 1, 2, \ldots, l \right\}.$$

Then, the following discussion holds for both $\mathcal{F}_2$ and $\mathcal{F}_{1,\infty}$. Define the adversarial hypothesis class as

$$\tilde{\mathcal{H}}^c = \left\{ \tilde{h} : \tilde{h}(x, y) = \tilde{\ell} \left(f(x), y \right), f \in \mathcal{F}^c \right\}.$$

For any $\tilde{h} \in \tilde{\mathcal{H}}$, we aim to determine the smallest distance to $\tilde{\mathcal{H}}^c$, , which involves computing

$$\max_{\tilde{h} \in \tilde{\mathcal{H}}} \min_{\tilde{h}^c \in \tilde{\mathcal{H}}^c} \left\| \tilde{h} - \tilde{h}^c \right\|_S.$$

$\forall (x_i, y_i), i = 1, \cdots, n$, given $\tilde{h}$ and $\tilde{h}^c$, by Lemma 1, there exist an intermediate adversarial example $\bar{x}_i$, such that,

$$\left| \tilde{h}(x_i, y_i) - \tilde{h}^c(x_i, y_i) \right| \leq \left| h(\bar{x}_i, y_i) - h^c(\bar{x}_i, y_i) \right|.$$

Since the loss function $\ell(f(x), y)$ is L_ϕ-Lipschitz with respect to the first argument,

$$\left| \tilde{h}(x_i, y_i) - \tilde{h}^c(x_i, y_i) \right| \leq L_\phi \left| f(\bar{x}_i) - f^c(\bar{x}_i) \right|.$$

Define $g_b^a(\cdot)$ as

$$g_b^a(\bar{x}) = W_b\rho\left(W_{b-1}\rho\left(\cdots W_{a+1}\rho\left(W_a^c\cdots\rho\left(W_1^c\bar{x}\right)\cdots\right)\right)\right).$$

In words, for the layers $b \geq j > a$ in $g_b^a(\cdot)$, the weight is W_j, for the layers $a \geq j \geq 1$ in $g_b^a(\cdot)$, the weight is W_j^c. Then we have $f(\bar{x}_i) = g_l^0(\bar{x}_i)$, $f(\bar{x}_i) = g_l^l(\bar{x}_i)$. We can decompose

$$\begin{aligned}
|f(\bar{x}_i) - f^c(\bar{x}_i)| &= \left|g_l^0(\bar{x}_i) - g_l^l(\bar{x}_i)\right| \\
&= \left|g_l^0(\bar{x}_i) - g_l^1(\bar{x}_i) + \cdots + g_l^{l-1}(\bar{x}_i) - g_l^l(\bar{x}_i)\right| \\
&\leq \left|g_l^0(\bar{x}_i) - g_l^1(\bar{x}_i)\right| + \cdots + \left|g_l^{l-1}(\bar{x}_i) - g_l^l(\bar{x}_i)\right|. \qquad (13)
\end{aligned}$$

To bound the gap $|f(\bar{x}_i) - f^c(\bar{x}_i)|$, we first calculate $|g_l^{j-1}(\bar{x}_i) - g_l^j(\bar{x}_i)|$ for $j = 1, \cdots, l$.

$$\begin{aligned}
\left|g_l^{j-1}(\bar{x}_i) - g_l^j(\bar{x}_i)\right| &= \left|W_l\rho\left(g_{l-1}^{j-1}(\bar{x}_i)\right) - W_l\rho\left(g_{l-1}^j(\bar{x}_i)\right)\right| \\
&\overset{(i)}{\leq} \|W_l\|\left\|\rho\left(g_{l-1}^{j-1}(\bar{x}_i)\right) - \rho\left(g_{l-1}^j(\bar{x}_i)\right)\right\|_{r^*} \\
&\overset{(ii)}{\leq} L_\rho M_l\left\|g_{l-1}^{j-1}(\bar{x}_i) - g_{l-1}^j(\bar{x}_i)\right\|_{r^*} \\
&\overset{(iii)}{=} L_\rho M_l\left\|W_{l-1}\rho\left(g_{l-2}^{j-1}(\bar{x}_i)\right) - W_{l-1}\rho\left(g_{l-2}^j(\bar{x}_i)\right)\right\|_{r^*} \\
&\leq \cdots \\
&\leq L_\rho^{l-j}\prod_{k=j+1}^{l} M_k\left\|W_j\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right) - W_j^c\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*}.
\end{aligned}$$

where (i) is due to Lemma 4, (ii) is due to the bound of $\|W_j\|$ and the Lipschitz of $\rho(\cdot)$, (iii) is because of the definition of $g_b^a(\bar{x})$. Then

$$\begin{aligned}
\left|g_l^{j-1}(\bar{x}_i) - g_l^j(\bar{x}_i)\right| &\leq L_\rho^{l-j}\prod_{k=j+1}^{l} M_k\left\|W_j\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right) - W_j^c\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*} \\
&= L_\rho^{l-j}\prod_{k=j+1}^{l} M_k\left\|\left(W_j - W_j^c\right)\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*} \\
&\overset{(i)}{\leq} L_\rho^{l-j}\prod_{k=j+1}^{l} M_k\left\|W_j - W_j^c\right\|\left\|\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*} \\
&\overset{(ii)}{\leq} L_\rho^{l-j}\prod_{k=j+1}^{l} M_k\delta_j\left\|\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*}. \qquad (14)
\end{aligned}$$

where inequality (i) is due to Lemma 4, inequality (ii) is due to Lemma 4 and inequality (iii) is due to the assumption that $\|W_j - W_j^c\| \leq \delta_j$. It is lefted to bound $\|\rho(g_{j-1}^{j-1}(\bar{x}_i)))\|_{r^*}$, we have

$$\begin{aligned}
\left\|\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right)\right\|_{r^*} &= \left\|\rho\left(g_{j-1}^{j-1}(\bar{x}_i)\right) - \rho(0)\right\|_{r^*} \\
&\leq L_\rho \left\|g_{j-1}^{j-1}(\bar{x}_i)\right\|_{r^*} \\
&= L_\rho \left\|W_{j-1}^c \rho\left(g_{j-2}^{j-2}(\bar{x}_i)\right)\right\|_{r^*} \\
&\leq L_\rho \left\|W_{j-1}^c\right\| \left\|\rho\left(g_{j-2}^{j-2}(\bar{x}_i)\right)\right\|_{r^*} \\
&\leq L_\rho M_{j-1} \left\|\rho\left(g_{j-2}^{j-2}(\bar{x}_i)\right)\right\|_{r^*} \\
&\leq \cdots \\
&\leq L_\rho^{j-1} \prod_{k=1}^{j-1} M_k \max\left\{1, d^{\frac{1}{2}-\frac{1}{r}-\frac{1}{p}}\right\} \left(\|x\|_{p,\infty} + \epsilon\right). \qquad (15)
\end{aligned}$$

combining Eq. (14) and (15), we have

$$\begin{aligned}
\left|g_l^{j-1}(\bar{x}_i) - g_l^j(\bar{x}_i)\right| &\leq L_\rho^{l-1} \frac{\prod_{k=1}^l M_k}{M_j} \delta_j \max\left\{1, d^{\frac{1}{2}-\frac{1}{r}-\frac{1}{p}}\right\} \left(\|x\|_{p,\infty} + \epsilon\right) \\
&= \frac{D\delta_j}{2M_j}. \qquad (16)
\end{aligned}$$

Therefore, combining Eq. (13) and (16), we have

$$\begin{aligned}
|f(\bar{x}_i) - f^c(\bar{x}_i)| &\leq \left|g_l^0(\bar{x}_i) - g_l^1(\bar{x}_i)\right| + \cdots + \left|g_l^{l-1}(\bar{x}_i) - g_l^l(\bar{x}_i)\right| \\
&\leq \sum_{j=1}^{l} \frac{D\delta_j}{2M_j}.
\end{aligned}$$

Then

$$\max_{\tilde{f} \in \tilde{\mathcal{F}}} \min_{\tilde{f}^c \in \tilde{\mathcal{F}}^c} \left\|\tilde{f} - \tilde{f}^c\right\|_S \leq \sum_{j=1}^{l} \frac{D\delta_j}{2M_j}.$$

Let $\delta_j = 2M_j\varepsilon / L_\phi l D$, $j = 1, \cdots, l$, we have

$$\max_{\tilde{h} \in \tilde{\mathcal{H}}} \min_{\tilde{h}^c \in \tilde{\mathcal{H}}^c} \left\|\tilde{h} - \tilde{h}^c\right\|_S \leq L_\phi \sum_{j=1}^{l} \frac{D\delta_j}{2M_j} \leq \varepsilon.$$

We then calculate the ε-covering number $\mathcal{N}(\tilde{\mathcal{H}}, \|\cdot\|_S, \varepsilon)$. Because $\tilde{\mathcal{H}}^c$ is a ε-cover of $\tilde{\mathcal{H}}$. The cardinality of $\tilde{\mathcal{H}}^c$ is

$$\begin{aligned}
\mathcal{N}\left(\tilde{\mathcal{H}}, \|\cdot\|_S, \varepsilon\right) &= \left|\tilde{\mathcal{H}}^c\right| \\
&= \begin{cases} \prod_{j=1}^l |\mathcal{C}_j| & \text{if } r = 2; \\ \prod_{j=1}^l \prod_{m=1}^{h_j} \left|\mathcal{C}_j^m\right| & \text{if } r = 1. \end{cases} \\
&= \prod_{j=1}^{l} \mathcal{N}\left(\{W_j \mid \|W_j\| \leq M_j\}, \|\cdot\|, \delta_j\right).
\end{aligned}$$

Therefore,

$$\ln\left(\mathcal{N}\left(\tilde{\mathcal{H}}, \|\cdot\|_{\mathcal{S}}, \varepsilon\right)\right) \leq \sum_{j=1}^{l} \ln\left(\mathcal{N}\left(\{W_j \mid \|W_j\| \leq M_j\}, \|\cdot\|, \delta_j\right)\right).$$

■

A.3 Proof of Theorem 5 and 6

Before we provide the proof, we first introduce the Dudley's integral.

Proposition 1 (Dudley's integral)**.** *The Rademacher complexity $\mathcal{R}_S(\mathcal{F})$ satisfies*

$$\mathcal{R}_S(\mathcal{F}) \leq \inf_{\delta \geq 0}\left[8\delta + \frac{12}{\sqrt{n}}\int_{\delta}^{D/2}\sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_S, \varepsilon)}d\varepsilon\right].$$

This proposition is a well-established result in statistical learning theory, with detailed proofs available in standard references such as Wainwright (2019). Using this relationship between covering numbers and Rademacher complexity, we can derive upper bounds on the Rademacher complexity of function class $\mathcal{F}$ from its covering number bounds.

Lemma 6 (Covering number of norm-balls)**.** *Let $\mathcal{B}$ be a ℓ_p norm ball with radius W. Let $d(x_1, x_2) = \|x_1 - x_2\|_p$. Define the ε-covering number of B as $\mathcal{N}(\mathcal{B}, d(\cdot,\cdot), \varepsilon)$, we have*

$$\mathcal{N}\left(\mathcal{B}, d(\cdot,\cdot), \varepsilon\right) \leq \left(1 + \frac{2W}{\varepsilon}\right)^{d}.$$

In the case of Frobenius norm ball of $m \times k$ matrices, we have the dimension $d = m \times k$ and

$$\mathcal{N}\left(\mathcal{B}, \|\cdot\|_F, \varepsilon\right) \leq \left(1 + \frac{2W}{\varepsilon}\right)^{m\times k} \leq \left(\frac{3W}{\varepsilon}\right)^{m\times k}.$$

Now we move to the proof of Theorem 5.

Proof We first consider the Lipschitz constant of the loss function $\ell(f(x), y) = \phi(yf(x))$ in binary settings. Since

$$\left|\phi\left(yf(x_1)\right) - \phi\left(yf(x_2)\right)\right| \leq L_\phi \left|yf(x_1) - yf(x_2)\right| = L_\phi \left|f(x_1) - f(x_2)\right|,$$

the loss function $\ell(f(x), y) = \phi(yf(x))$ is L_ϕ-Lipschitz with respect to the first argument. Based on Lemma 2, define

$$\varepsilon = L_\phi \sum_{j=1}^{l} L_\rho^{l-1} \frac{\prod_{k=1}^{l} M_k}{M_j} \max\left\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\right\}\left(\|x\|_{p,\infty} + \epsilon\right)\delta_j.$$

where $r = 2$ for Frobenius norm and $r = 1$ for $(1, \infty)$-norm. Then:

$$\begin{aligned}
\ln\left(\mathcal{N}\left(\tilde{\mathcal{H}}, \|\cdot\|_S, \varepsilon\right)\right) &\leq \sum_{j=1}^{l} \ln\left(\mathcal{N}\left(\{W_j \mid \|W_j\| \leq M_j\}, \|\cdot\|, \delta_j\right)\right) \\
&= \sum_{j=1}^{l} \ln|\mathcal{C}_j| \\
&\overset{(i)}{\leq} \sum_{j=1}^{l} \ln\left(\frac{3M_j}{\delta_j}\right)^{h_j h_{j-1}} \\
&= \ln\left(\frac{3L_\phi l D}{2\varepsilon}\right) \sum_{j=1}^{l} h_j h_{j-1}.
\end{aligned}$$

where inequality (i)) is due to Lemma 6. By Dudley's integral, we have

$$\begin{aligned}
\mathcal{R}_S(\tilde{\mathcal{H}}) &\leq \inf_{\delta \geq 0}\left[8\delta + \frac{12}{\sqrt{n}} \int_{\delta}^{L_\phi D/2} \sqrt{\log \mathcal{N}(\mathcal{H}, \|\cdot\|_S, \varepsilon)} d\varepsilon\right] \\
&\leq \inf_{\delta \geq 0}\left[8\delta + \frac{12}{\sqrt{n}} \int_{\delta}^{L_\phi D/2} \sqrt{\left(\sum_{j=1}^{l} h_j h_{j-1}\right) \log(3L_\phi l D/2\varepsilon)} d\varepsilon\right] \\
&= \inf_{\delta \geq 0}\left[8\delta + \frac{12 L_\phi D \sqrt{\sum_{j=1}^{l} h_j h_{j-1}}}{\sqrt{n}} \int_{\delta/L_\phi D}^{1/2} \sqrt{\log(3l/2\varepsilon)} d\varepsilon\right]. \qquad (17)
\end{aligned}$$

Let $\delta \to 0$[3]. Then, we evaluate the integration

$$\int_0^{1/2} \sqrt{\log\left(\frac{3l}{2\varepsilon}\right)} d\varepsilon.$$

Let $u = \frac{3l}{2\varepsilon}$, so $\varepsilon = \frac{3l}{2u}$ and $d\varepsilon = -\frac{3l}{2u^2} du$. When $\varepsilon = 0$, $u \to \infty$. When $\varepsilon = \frac{1}{2}$, $u = \frac{3l}{2 \cdot \frac{1}{2}} = 3l$. The integral becomes:

$$\int_{\infty}^{3l} \sqrt{\log u}\left(-\frac{3l}{2u^2}\right) du = \frac{3l}{2} \int_{3l}^{\infty} \frac{\sqrt{\log u}}{u^2} du.$$

Let $t = \log u$, so $u = e^t$, $du = e^t dt$. When $u = 3l$, $t = \log(3l)$. The integral transforms to:

$$\frac{3l}{2} \int_{\log(3l)}^{\infty} \sqrt{t}\, e^{-t} dt.$$

Let $v = \sqrt{t}$, $dw = e^{-t} dt$. Then $dv = \frac{1}{2\sqrt{t}} dt$, $w = -e^{-t}$, integration by part:

$$\int \sqrt{t} e^{-t} dt = -\sqrt{t} e^{-t} + \frac{1}{2} \int \frac{e^{-t}}{\sqrt{t}} dt.$$

3. Simply let $\delta = L_\phi D/\sqrt{n}$, we can obtain a bound in $\mathcal{O}(\sqrt{\log n/n})$. To get rid of the $\log n$ term, we can let $\delta \to 0$.

Essentially, the integral $\int_a^\infty t^{1/2}e^{-t}dt$ is the upper incomplete gamma function $\Gamma\left(\frac{3}{2}, a\right)$. Using properties of Γ:

$$\Gamma\left(\frac{3}{2}, \log(3l)\right) = \sqrt{\pi}\,\text{erfc}\left(\sqrt{\log(3l)}\right) + \sqrt{\log(3l)}\,e^{-\log(3l)}.$$

Here $\text{erfc}(x)$ is the complementary error function defined as:

$$\text{erfc}(x) = 1 - \text{erf}(x),$$

where the error function $\text{erf}(x)$ is given by:

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

Therefore, the integration is

$$\frac{3l}{2}\left(\frac{\sqrt{\log(3l)}}{3l} + \frac{\sqrt{\pi}}{2}\text{erfc}\left(\sqrt{\log(3l)}\right)\right) = \frac{1}{2}\left(\sqrt{\log(3l)} + \frac{3l}{2}\sqrt{\pi}\,\text{erfc}\left(\sqrt{\log(3l)}\right)\right).$$

Finally, we provide the upper bound of the integration.

$$\begin{aligned}
\int_0^{1/2}\sqrt{\log(3l/2\varepsilon)}d\varepsilon &= \frac{1}{2}\left(\frac{3l}{2}\sqrt{\pi}\text{erfc}(\sqrt{\log 3l}) + \sqrt{\log 3l}\right) \\
&\leq \frac{1}{2}\left(\frac{3l}{2}\sqrt{\pi}\exp(-\sqrt{\log 3l}^2) + \sqrt{\log 3l}\right) \\
&= \frac{1}{2}\left(\frac{\sqrt{\pi}}{2} + \sqrt{\log 3l}\right) \\
&\leq \frac{1}{2}\left(2\sqrt{\log 3l}\right) \\
&= \sqrt{\log 3l}. \qquad (18)
\end{aligned}$$

Plugging Eq. (18) to Eq. (17), we have

$$\mathcal{R}_S(\tilde{\mathcal{H}}) \leq \frac{24}{\sqrt{n}}\max\left\{1, d^{\frac{1}{2}-\frac{1}{r}-\frac{1}{p}}\right\}(\|x\|_{p,\infty} + \epsilon)\,L_\rho^{l-1}\sqrt{\sum_{j=1}^{l} h_j h_{j-1}\log(3l)}\prod_{j=1}^{l} M_j.$$

■

A.4 Proof of Theorem 7

Let $\rho(\cdot)$ be the identity activation function. The following discussion hods for both $\mathcal{F} = \mathcal{F}_2$ and $\mathcal{F}_{1,\infty}$. The proof of the theorem is based on constructing a linear network. By the definition of Rademacher complexity, if $\mathcal{H}'$ is a subset of $\mathcal{H}$, then

$$\begin{aligned}\mathcal{R}_{\mathcal{S}}(\mathcal{H}') &= \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h\in\mathcal{H}'}\sum_{i=1}^{n}\sigma_i h(x_i,y_i)\right]\\ &\le \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{n}\sigma_i h(x_i,y_i)\right]\\ &= \mathcal{R}_{\mathcal{S}}(\mathcal{H}).\end{aligned}$$

This inequality follows directly from the fact that restricting the hypothesis class cannot increase the supremum in the definition of Rademacher complexity.

Therefore, it suffices to lower bound the complexity of $\tilde{\mathcal{F}}'$ under a specific distribution $\mathcal{D}$, where $\tilde{\mathcal{F}}'$ is a subset of $\tilde{\mathcal{F}}$. We define

$$\tilde{\mathcal{F}}' = \left\{ x \mapsto \inf_{\|x'-x\|_p\le\epsilon} yM_l \cdot M_2 w^T x \mid w\in\mathbb{R}^q, \|w\|_2 \le M_1 \right\}.$$

This formulation constrains the function class while maintaining a meaningful lower bound on its complexity.

We first prove that $\tilde{\mathcal{F}}'$ is a subset of $\tilde{\mathcal{F}}$. In $\tilde{\mathcal{F}}$, we set the activation function $\rho(\cdot)$ to be the identity mapping. Define

$$W_1 = \begin{bmatrix} w \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{h_1\times h_0}, \quad W_j = \begin{bmatrix} M_j & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{h_j\times h_{j-1}}, \quad j=2,\ldots,l. \tag{19}$$

Since $\|W_j\| \le M_j$, imposing the additional constraint in Eq. (19) on $\tilde{\mathcal{F}}$ reduces it to $\tilde{\mathcal{F}}'$, confirming that $\tilde{\mathcal{F}}'$ is a subset of $\tilde{\mathcal{F}}$.

To proceed, we need to establish a lower bound for the adversarial Rademacher complexity of linear hypothesis classes. This result follows from the work of Yin et al. (2019); Awasthi et al. (2020), which we state below.

Proposition 4. *Given the function class*

$$\mathcal{G} = \{x \to yw^T x | w\in\mathbb{R}^q, \|w\|_r \le W\}$$

and

$$\tilde{\mathcal{G}} = \{x \to \inf_{\|x'-x\|_r\le\epsilon} yw^T x | w\in\mathbb{R}^q, \|w\|_r \le W\},$$

the adversarial Rademacher complexity $\mathcal{R}_S(\tilde{\mathcal{G}})$ satisfies

$$\mathcal{R}_S(\tilde{\mathcal{G}}) \ge \max\left\{\mathcal{R}_S(\mathcal{G}), \frac{\epsilon \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}W}{2\sqrt{n}}\right\}.$$

Since the standard Rademacher complexity

$$\mathcal{R}_S(\mathcal{G}) = \frac{W}{m}\mathbb{E}_\sigma \| \sum_{i=1}^{n} \sigma_i x_i \|_{r*},$$

let $\|x_i\| = B$ with equal entries for $i = 1, \cdots, n$, by Lemma 3, we have

$$\mathcal{R}_S(\mathcal{G}) = \frac{W}{m}\mathbb{E}_\sigma | \sum_{i=1}^{n} \sigma_i | \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\} B.$$

By Khintchine's inequality, we know that there exists a universal constant $c > 0$ such that

$$\mathbb{E}_\sigma | \sum_{i=1}^{n} \sigma_i | \geq c\sqrt{n}.$$

Then, we have

$$\mathcal{R}_S(\mathcal{G}) = \frac{cW}{\sqrt{n}} \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\} B.$$

Therefore,

$$\begin{aligned}
\mathcal{R}_S(\tilde{\mathcal{G}}) &\geq \max\left\{ \mathcal{R}_S(\mathcal{G}), \frac{\epsilon \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\} W}{2\sqrt{n}} \right\} \\
&\geq \frac{1}{1+2c}\mathcal{R}_S(\mathcal{G}) + \frac{2c}{1+2c} \times \frac{\epsilon \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\} W}{2\sqrt{n}} \\
&\geq \frac{c}{1+2c}\left(\frac{(B+\epsilon) \max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\} W}{\sqrt{n}} \right).
\end{aligned}$$

Let $W = \prod_{j=1}^{l} M_j$, we have

$$\mathcal{R}_S(\tilde{\mathcal{F}}) \geq \Omega\left(\frac{\max\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\}(B+\epsilon) \prod_{j=1}^{l} M_j}{\sqrt{n}} \right),$$

where $r = 2$ for frobenius norm bound and $r = 1$ for $\|\cdot\|_{1,\infty}$-norm bound.

A.5 Proof of Theorem 9

Proof We consider the Lipschitz constant of the loss function $\ell(f(x), y) = \phi_\gamma(M(f(x), y))$, where $\gamma > 0$. First, by the definition of the ramp loss $\phi_\gamma : \mathbb{R} \to [0, 1]$, ϕ_γ is $1/\gamma$-Lipschitz. Second, the following lemma provides the Lipchitz constant of the margin operator.

Lemma 7 (Bartlett et al. (2017), cf. Lemma A.3)**.** *For every y and every $p \geq 1$, $M(\cdot, y)$ is 2-Lipschitz with respect to $\|\cdot\|_p$.*

Therefore, the loss function $\ell(f(x), y) = \phi_\gamma(M(f(x), y))$ is $2/\gamma$ Lipshitz. The upper bound is obtained by letting $L_\phi = 2/\gamma$ in Theorem 5. ■

B. Discussion on Existing Methods for Rademacher Complexity

In this section, we review existing approaches for calculating Rademacher complexity, examine related work in the field, and highlight the challenges in analyzing adversarial Rademacher complexity.

B.1 Existing Bounds for Standard Rademacher Complexity

Layer Peeling Technique. The Rademacher complexity of multi-layer neural networks is primarily calculated using the 'layer peeling' technique (Neyshabur et al., 2015). For a function class $\mathcal{F}$ and function g, we define the composition $g \circ \mathcal{F}$ as $\{g \circ f | f \in \mathcal{F}\}$. Talagrand's Lemma establishes that $\mathcal{R}_{\mathcal{S}}(g \circ f) \leq L_g \mathcal{R}_{\mathcal{S}}(\mathcal{F})$. Applying this result to neural networks, we can show that $\mathcal{R}_{\mathcal{S}}(\mathcal{F}_l) \leq 2L_\rho M_j \mathcal{R}_{\mathcal{S}}(\mathcal{F}_{l-1})$, where $\mathcal{F}_l$ represents the function class of l-layer neural networks. Since the Rademacher complexity of a linear function class is bounded by $\mathcal{O}(BM_1/\sqrt{n})$, induction yields an upper bound of $\mathcal{O}(B2^l L_\rho^{l-1} \prod_{j=1}^{l} M_j/\sqrt{n})$. For activation functions like ReLU where $L_\rho = 1$, we can simplify this bound by eliminating the L_ρ term.

Golowich et al. (2018) achieves an improved bound by reducing the depth dependence from 2^l to $\sqrt{l}$. Their key insight involves reformulating the Rademacher complexity expression $\mathbb{E}_\sigma[\cdot]$ as $\mathbb{E}_\sigma \exp\ln[\cdot]$. This transformation allows for layer peeling to be performed within the $\ln(\cdot)$ function, effectively containing the exponential 2^l term inside the logarithm and yielding the improved $\sqrt{l}$ dependence.

Covering Number. Bartlett et al. (2017) established a generalization gap bound using covering numbers:

$$\tilde{\mathcal{O}}\Big(\frac{B\prod_{j=1}^{l}\|W_j\|}{\sqrt{n}}\Big(\sum_{j=1}^{l}\frac{\|W_j\|_{2,1}^{2/3}}{\|W_j\|^{2/3}}\Big)^{3/2}\Big),$$

where $\|\cdot\|$ denotes the spectral norm. Their proof employs layer-wise induction: for each layer j, let W_j represent the layer's weight matrix and X_j denote the network's output after processing through layers 1 to $j-1$. The bound is derived by inductively computing the matrix covering number $\mathcal{N}(\{W_j X_j\}, \|\cdot\|_2, \epsilon)$ for each layer.

B.2 Existing Bounds for Rademacher Complexity on Surrogate Loss

For linear models, ARC bounds can be derived directly from its definition (Khim and Loh, 2018; Yin et al., 2019). However, extending these analyses to multi-layer networks presents additional challenges. We begin by examining approaches that utilize surrogate loss functions.

Tree Transformation Loss. Khim and Loh (2018) introduced a tree transformation T and demonstrated that $\max_{\|x-x'\|\leq\epsilon} \ell(f(x), y) \leq \ell(Tf(x), y)$. This leads to an upper bound on the adversarial population risk. Specifically, for any $\delta \in (0, 1)$:

$$\tilde{R}(f) \leq R(Tf) \leq R_n(Tf) + 2L\mathcal{R}_S(T \circ \mathcal{F}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2n}},$$

where $R_n(Tf)$ represents the empirical risk of the transformed function, $\mathcal{R}_S(T \circ \mathcal{F})$ is the Rademacher complexity of the transformed function class, and L is the Lipschitz constant. This result bounds the robust population risk in terms of empirical risk and the standard Rademacher

complexity of the transformed function class $T \circ \mathcal{F}$. While $\mathcal{R}_S(T \circ \mathcal{F})$ serves as an approximation of adversarial Rademacher complexity, this bound has two key limitations: the empirical risk term $R_n(Tf)$ differs from the objective used in practical adversarial training, and the bound does not directly characterize the robust generalization gap between training and test performance.

SDP Relaxation Surrogate Loss. In (Yin et al., 2019), the authors introduced an SDP surrogate loss to approximate the adversarial loss for two-layer neural networks. This surrogate loss is defined as:

$$\hat{\ell}(f(x), y) = \phi_\gamma\bigg(M(f(x), y) - \frac{\epsilon}{2} \max_{k\in[K], z=\pm 1} \max_{P\succeq 0, diag(P)\leq 1} \langle zQ(w_{2,k}, W_1), P\rangle\bigg).$$

Using this formulation, the adversarial Rademacher complexity can be approximated by computing the Rademacher complexity of this surrogate loss function. However, this approach shares the same limitations as the previously discussed method, as it still does not directly characterize the robust generalization gap between training and test performance.

FGSM Attack Loss. Gao and Wang (2021) analyzed Rademacher complexity in the adversarial setting by focusing on Fast Gradient Sign Method (FGSM) adversarial examples to handle the $\max$ operation in the adversarial loss. Under specific gradient assumptions, they derived an upper bound for the adversarial Rademacher complexity using the loss $\ell(f(x_{\text{FGSM}}), y)$. Their analysis requires that $|\nabla \ell(f(x), y)| \geq \kappa$ holds for all x in the domain, where κ appears in the denominator of their final bound. This assumption proves problematic for two reasons: first, it is a strong condition that may not hold in practice, and second, the bound becomes unbounded as κ approaches zero. Moreover, since their approach modifies the original loss function, it cannot provide guarantees on the robust generalization gap.

B.3 Layer Peeling Technique for ARC

We first briefly introduce the layer peeling technique in standard settings.

$$\begin{aligned}
\mathcal{R}_{\mathcal{S}}(\mathcal{H}) &= \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h\in\mathcal{H}} \sum_{i=1}^n \sigma_i h(x_i)\right] \\
&= \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}, \|W_l\|\leq M_l} \sum_{i=1}^n \sigma_i W_l \rho(h'(x_i))\right] \\
&\leq M_l \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}} \left\|\sum_{i=1}^n \sigma_i \rho(h'(x_i))\right\|\right] \\
&\leq 2M_l L_\rho \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}} \sum_{i=1}^n \sigma_i h'(x_i)\right] \\
&= 2M_l L_\rho \mathcal{R}_{\mathcal{S}}(\mathcal{H}_{l-1}).
\end{aligned}$$

In adversarial settings, if we directly apply the layer peeling technique, we have

$$\begin{aligned}
\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) &= \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h\in\mathcal{H}} \sum_{i=1}^{n} \sigma_i \max_{\|x_i - x_i'\|\le\epsilon} h(x_i')\right] \\
&= \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}, \|W_l\|\le M_l} \sum_{i=1}^{n} \sigma_i W_l \rho(h'(x_i^*(h)))\right] \\
&\le M_l \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}} \left\|\sum_{i=1}^{n} \sigma_i \rho(h'(x_i^*(h)))\right\|\right] \\
&\le 2M_l L_\rho \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}} \sum_{i=1}^{n} \sigma_i h'(x_i^*(h))\right] \\
&\neq 2M_l L_\rho \mathbb{E}_\sigma \frac{1}{n}\left[\sup_{h'\in\mathcal{H}_{l-1}} \sum_{i=1}^{n} \sigma_i h'(x_i^*(h'))\right] \\
&= 2M_l L_\rho \mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}_{l-1}).
\end{aligned}$$

where $x_i^*(h)$ and $x_i^*(h')$ denote the optimal adversarial examples for l-layer and $(l-1)$-layer neural networks, respectively. Since $x_i^*(h) \neq x_i^*(h')$ in general, the optimal adversarial examples differ between architectures of different depths, which prevents the direct extension of layer peeling techniques to the adversarial setting.

B.4 Comparison of Adversarial Generalization Bounds

VC-Dimension Bounds. The VC dimension is a fundamental tool in statistical learning theory for bounding generalization gaps. Several works, including Cullina et al. (2018), Montasser et al. (2019), and Attias et al. (2021), have extended this framework to the adversarial setting. However, these approaches fail to provide computable bounds on the adversarial generalization gap, as we explain below. Let $\mathcal{H}$ denote the hypothesis class (for example, the set of neural networks with a fixed architecture).

Cullina et al. (2018) introduced the concept of adversarial VC dimension (AVC) and established bounds on the adversarial generalization gap in terms of $\text{AVC}(\mathcal{H})$. However, they did not provide methods to compute the AVC for neural networks, thus leaving their bounds non-computable in practice.

Montasser et al. (2019) took a different approach by defining an adversarial function class $\mathcal{L}_{\mathcal{H}}^{\mathcal{U}}$, where $\mathcal{L}$ represents the loss function and $\mathcal{U}$ denotes the uncertainty set. While their bound on the adversarial generalization gap using $\mathcal{L}_{\mathcal{H}}^{\mathcal{U}}$ differs from the $\text{AVC}(\mathcal{H})$ approach of Cullina et al. (2018), they similarly did not provide a method to compute their bound, rendering it non-computable in practice.

Attias et al. (2021) analyze the case where the perturbation set $U(x)$ is finite, containing exactly k possible adversarial examples for each sample x. Under this assumption, they bound the adversarial generalization gap by:

$$\mathcal{O}\left(\frac{1}{\varepsilon^2}\left(\sqrt{k \cdot VC(\mathcal{H})}\log\left(\frac{3}{2}+a\right) k \cdot VC(\mathcal{H})\right) + \log\frac{1}{\delta}\right).$$

Since VC($\mathcal{H}$) can be upper-bounded by the number of network parameters, this result provides a computable bound, improving upon previous approaches. However, this computability comes with a significant limitation: the bound depends on k, the number of allowed perturbed samples, which deviates from the standard notion of adversarial generalization where $U(x)$ is an infinite set. In contrast, our bound applies to the original adversarial generalization framework where $U(x)$ is infinite ($k = +\infty$).

Adversarial Generalization in Alternative Settings. Several studies have explored adversarial generalization in specific model architectures. Javanmard et al. (2020) analyzed generalization properties in linear regression, while multiple researchers (Taheri et al., 2020; Javanmard et al., 2020; Dan et al., 2020) investigated adversarial generalization using Gaussian mixture models. Studies on uniform stability in adversarial training (Xing et al., 2021; Xiao et al., 2022a,b,d, 2024) suggest that poor generalization may result from the non-smooth nature of adversarial loss functions.

Certified Robustness. Research on certified robustness focuses on guaranteeing model performance within norm-constrained neighborhoods of training data. Cohen et al. (2019) developed certification methods through random smoothing, while Lecuyer et al. (2019) approached certification through differential privacy frameworks.

Geometric Perspectives on Adversarial Examples. Several works have examined the geometric properties of adversarial examples (Gilmer et al., 2018; Khoury and Hadfield-Menell, 2018). The off-manifold hypothesis, first proposed by Szegedy et al. (2013), suggests that adversarial examples deviate from the underlying data manifold. Supporting this view, Song et al. (2017) demonstrated using generative models that adversarial examples typically occupy low-probability regions of the data distribution. Similarly, Ma et al. (2018) employed Local Intrinsic Dimensionality (LID) to show that adversarial subspaces are both low-probability and distinct from the data submanifold.

C. Additional Experiments

In this section, we present additional experimental results to further validate our theoretical findings and explore their implications.

C.1 Experiments on VGG Architectures

Figure 3 presents experimental results for VGG-11 and VGG-13 architectures, while Figure 4 demonstrates findings for VGG-16 and VGG-19. Our analysis reveals a substantial difference in the product of Frobenius norms between standard and adversarial training methods, which corresponds to poor generalization performance in the adversarial setting.

$\|\cdot\|_{1,\infty}$-Norm Bounds. The $\|\cdot\|_{1,\infty}$-norm bounds are shown in Figure 5. Similar the the Frobenius norm bounds,The gap of $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$ between adversarial training and standard training are large. But the magnitude of $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$ is larger than the magnitude of $\prod_{j=1}^{l} \|W_j\|_F$.

C.2 Ablation Study on Margin Distribution

Figure 6 illustrates the margin distributions at the 1st, 3rd, and 5th percentiles of the training dataset. As the robust training accuracy reaches 100%, the choice of percentile does not significantly impact our analysis. Across all percentiles, standard training consistently achieves larger margins compared

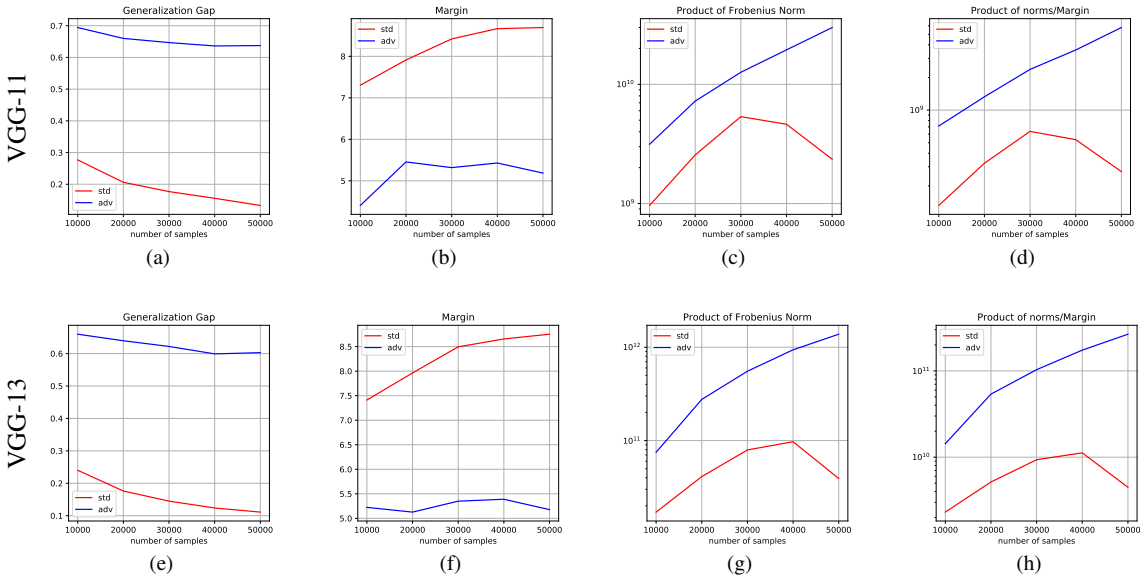


Figure 3: Comparison of Frobenius norm products across VGG architectures. Results from standard training (red lines) and adversarial training (blue lines) for VGG-11 (top row) and VGG-13 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin γ, (c,g) Product of layer-wise Frobenius norms $\prod_{j=1}^{l} \|W_j\|_F$, and (d,h) Ratio of Frobenius norm product to margin $\prod_{j=1}^{l} \|W_j\|_F / \gamma$.

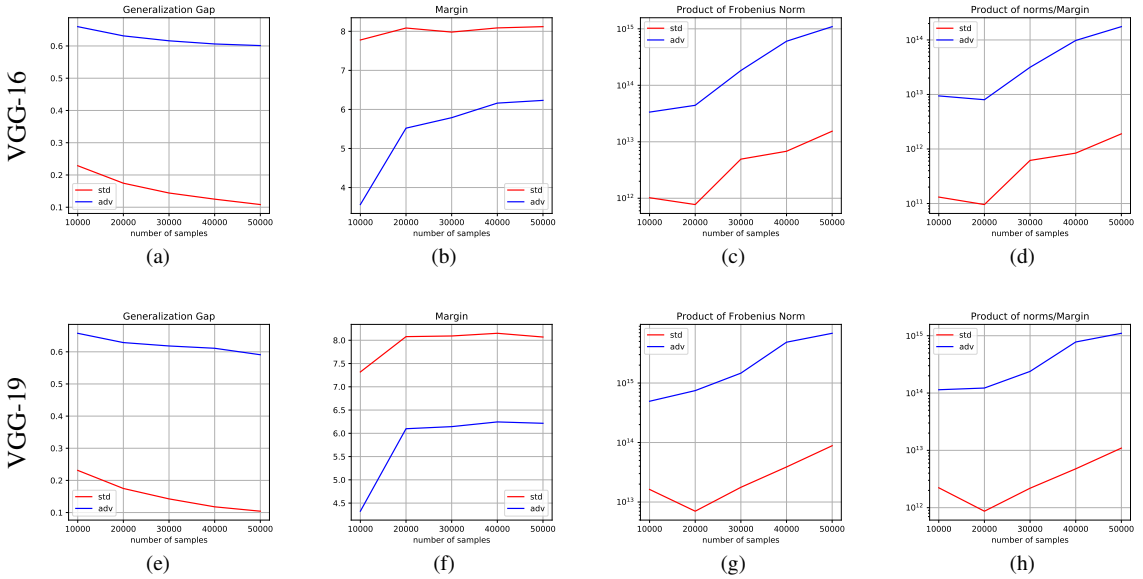
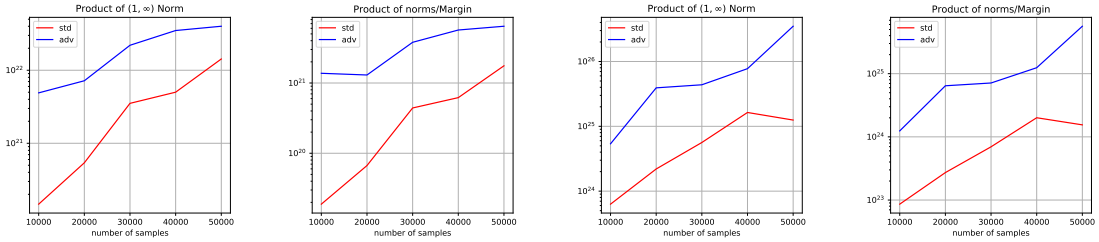


Figure 4: CIFAR-10 experimental results comparing standard training (red lines) and adversarial training (blue lines) for VGG-16 (top row) and VGG-19 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin, (c,g) Product of layer-wise Frobenius norms $\prod_{j=1}^{l} \|W_j\|_F$, and (d,h) Ratio of Frobenius norm product to margin $\prod_{j=1}^{l} \|W_j\|_F / \gamma$.



(a) (b) (c) (d)

Figure 5: Comparison of $\|\cdot\|_{1,\infty}$-norm products on CIFAR-10 between standard training (red lines) and adversarial training (blue lines) for: VGG-16 (a,b) and VGG-19 (c,d). The plots show: (a,c) Product of layer-wise $\|\cdot\|_{1,\infty}$-norms $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$, and (b,d) Ratio of norm product to margin $\prod_{j=1}^{l} \|W_j\|_{1,\infty}/\gamma$.

to adversarial training. Since margins appear in the denominator of the Rademacher complexity upper bound, these smaller margins in adversarial training contribute, albeit modestly, to its poorer generalization performance.

C.3 Experiments on CIFAR-100

Performance Analysis. Table 3 presents comparative results between standard and adversarial training on CIFAR-100 using VGG-16 and VGG-19 architectures. Our experiments reveal that the standard CIFAR-100 training set size of 50,000 samples is insufficient to effectively train VGG networks to acceptable performance levels. This limitation makes it challenging to analyze weight norm trends through CIFAR-100 experiments. Nevertheless, we compare the product of weight norms between standard and adversarial training methods to gain insights into their relative behaviors.

Product of Weight Norms. Figure 7 presents the training results for VGG-16 and VGG-19 architectures on CIFAR-100. Consistent with our CIFAR-10 experiments, adversarially trained models exhibit significantly larger weight norms compared to their standard-trained counterparts.

Table 3: Performance comparison between standard and adversarial training on CIFAR-100 using VGG-16 and VGG-19 architectures. Results show clean accuracy for standard training and robust accuracy (against PGD attacks) for adversarial training.

No. of Samples	10000	20000	30000	40000	50000
VGG-16-STD	0.26	0.44	0.54	0.60	0.63
VGG-16-ADV	0.12	0.15	0.17	0.18	0.19
VGG-19-STD	0.32	0.47	0.53	0.58	0.62
VGG-19-ADV	0.12	0.16	0.17	0.19	0.21

C.4 Weight Decay

Theoretical upper bounds on adversarial Rademacher complexity suggest that adding weight regularization (weight decay) can improve generalization performance. We experimentally validate this theoretical insight in Figure 8, comparing adversarial training with and without weight decay.

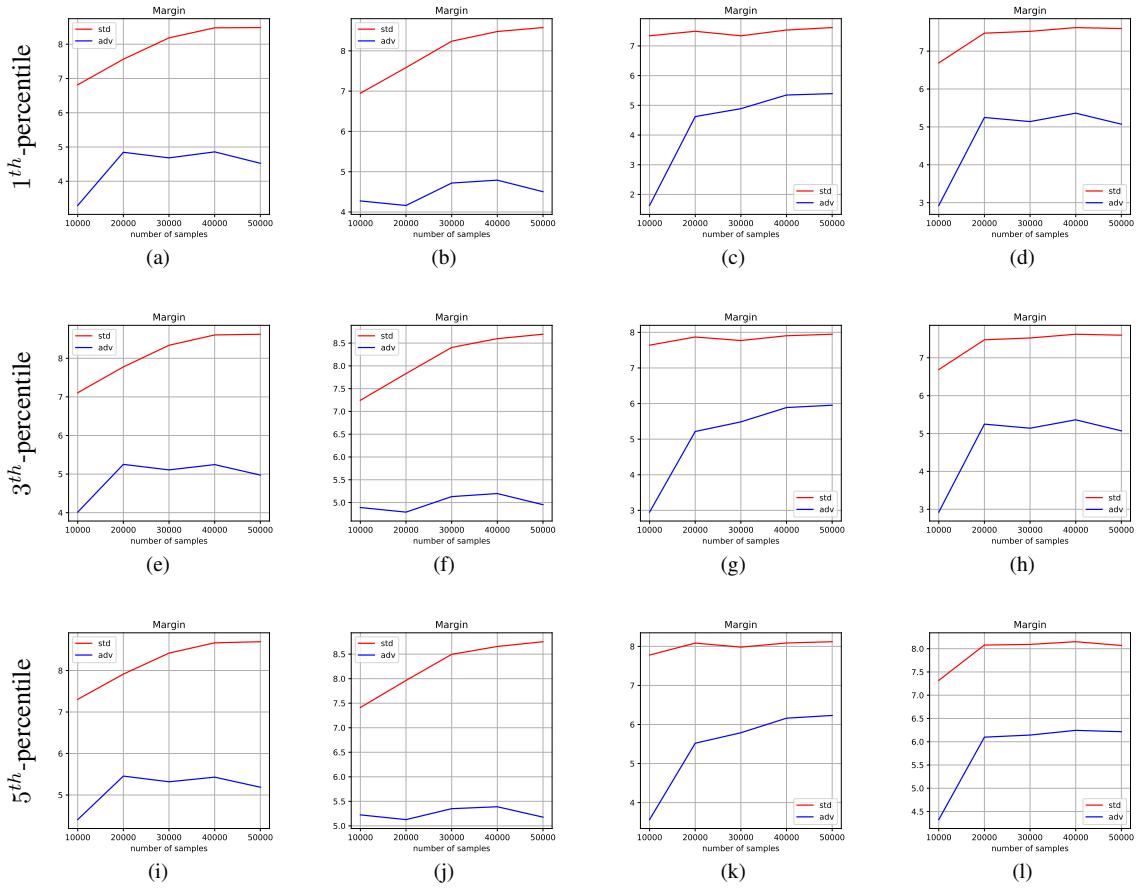


Figure 6: Margin analysis across VGG architectures: Results from VGG-11 (first column), VGG-13 (second column), VGG-16 (third column), and VGG-19 (fourth column).

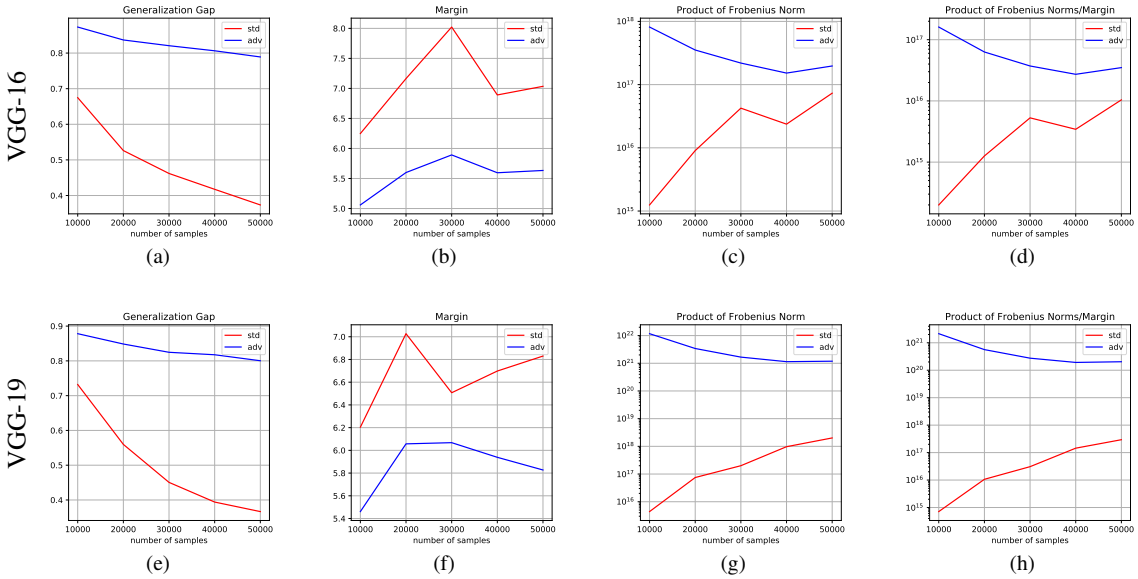


Figure 7: CIFAR-100 experimental results comparing standard training (red lines) and adversarial training (blue lines) for VGG-16 (top row) and VGG-19 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin γ, (c,g) Product of layer-wise Frobenius norms $\prod_{j=1}^{l} \|W_j\|_F$, and (d,h) Ratio of Frobenius norm product to margin $\prod_{j=1}^{l} \|W_j\|_F / \gamma$.

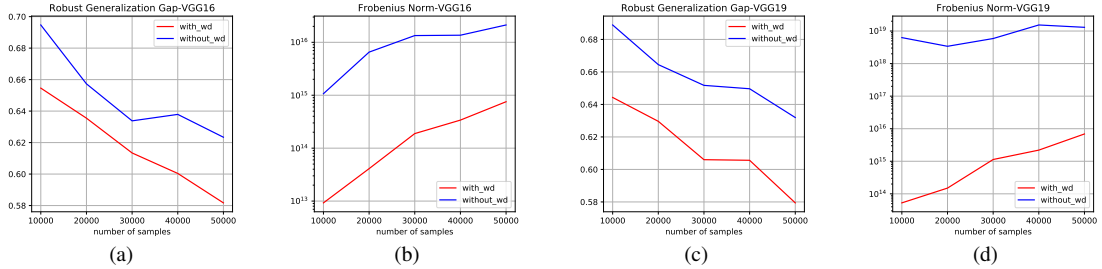


Figure 8: Impact of weight decay on VGG architectures: Results for VGG-16 (a,b) and VGG-19 (c,d), comparing models trained with and without weight decay. The plots show: (a,c) Robust generalization gap and (b,d) Product of layer-wise Frobenius norms.

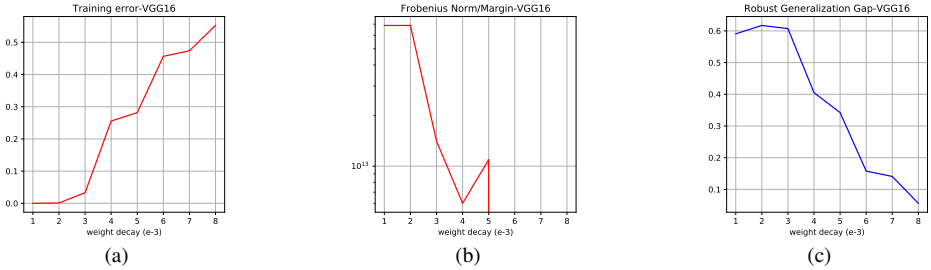


Figure 9: Weight decay effects on model behavior for values ranging from 1×10^{-3} to 9×10^{-3}: (a) Training error performance, (b) Frobenius norm of model weights, and (c) Robust generalization gap between training and test performance.

The results in Figures 8(a) and (c) demonstrate that incorporating weight decay reduces the robust generalization gap. Additionally, Figures 8(b) and (d) show that adversarial training with weight decay yields smaller weight norm products. These experimental findings establish a clear empirical connection between the robust generalization gap and weight norm products, supporting our theoretical analysis.

In Figure 9, we examine the effect of weight decay values ranging from 1×10^{-3} to 9×10^{-3}. The training error increases with higher weight decay values. At low weight decay (1×10^{-3}), the model achieves minimal training error, indicating a high capacity to fit the training data. However, as weight decay increases, the model's flexibility is constrained, leading to higher training errors. This behavior aligns with the regularization effect of weight decay, which penalizes large weights and reduces the model's ability to overfit. At very high weight decay values (9×10^{-3}), the model risks underfitting, as excessive regularization prevents it from capturing meaningful patterns in the data.

The Frobenius norm of the model weights decreases monotonically with increasing weight decay. This trend reflects the direct impact of weight decay on the optimization process, where larger decay values impose stricter penalties on weight magnitudes. The reduction in the Frobenius norm indicates a simplification of the model, which is a key objective of regularization. However, excessively small weight magnitudes can lead to underfitting, highlighting the need for careful tuning of the weight decay parameter.

The generalization gap, defined as the difference between training and test performance, demonstrates a non-linear relationship with weight decay. At low weight decay values, the gap is large, indicating poor generalization due to overfitting. As weight decay increases, the gap narrows, reaching a minimum at intermediate values (e.g., 3×10^{-3} to 7×10^{-3}). This reduction in the gap signifies improved generalization, as the model achieves a better balance between fitting the training data and maintaining performance on unseen data. At very high weight decay values, the gap may stabilize or slightly increase, as both training and test performance degrade due to underfitting.

In conclusion, the results highlight the trade-offs associated with weight decay. Model performance deteriorates significantly within this range. At weight decay values of 6×10^{-3} and higher, the margin becomes negative, indicating high training error. Subsequently, the weight norm to margin ratio also becomes negative. At 9×10^{-3}, the model fails to learn entirely, with both training and test errors reaching 90%. Our analysis suggests the optimal weight decay range for minimizing weight norm lies between 1×10^{-3} and 5×10^{-3}. The smallest weight norm observed was 6.01×10^{12} (with weight decay = 4×10^{-3}). However, this value remains larger than the weight norm achieved through standard training (1.90×10^{12}). These findings emphasize the importance of selecting an appropriate weight decay value to achieve robust model performance.