Journal of Machine Learning Research 27 (2026) 1-46 Submitted 4/23; Revised 5/26; Published 6/26

# 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** YANBOFAN@NJU.EDU.CN
*Nanjing University*
*Suzhou, 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*

**Editor:** Po-Ling Loh

## 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 DNNs can fit perturbed training data, 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 Khim and Loh (2018) and Yin et al. (2019). 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., 2014; Goodfellow et al., 2015). More specifically, a

*. Corresponding Author.

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 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., 2018). 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 $\sigma_i$ are i.i.d. Rademacher random variables, *i.e.,* $\sigma_i$ 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. (2019) 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. (2019) 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. (2020a) 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 $x^*$ and analyzing the

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., 2019; Khim and Loh, 2018; Awasthi et al., 2020a). Type 2: Surrogate Loss in Deep Networks (Yin et al., 2019; 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** | Lemmas 5 & 6 | - |

adversarial loss, given by $\max_{x'} h(x', y) = h(x^*, y)$. Khim and Loh (2018); Yin et al. (2019) introduced adversarial Rademacher complexity and provided bounds for linear functions using this method. Awasthi et al. (2020a) 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.

**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., 2019), 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 5.

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 6. 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,

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

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., 2020a). 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., 2014; Goodfellow et al., 2015; Chen et al., 2017; Carlini and Wagner, 2017; Madry et al., 2018). 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., 2025).

**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., 2022; Montasser et al., 2019). Neyshabur et al. (2018) 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. (2018) examined robust generalization in the context of distributional robustness, while Allen-Zhu and Li (2022) 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 $\ell_2$-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 $f \in \mathcal{F}$:*

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

### 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 $\ell_p$ attacks where $p \geq 1$. We denote by $\mathcal{B}(x)$ the general perturbation set around point $x$. For $\ell_p$ attacks, this set is defined as $\mathcal{B}(x) = \{x' \mid \|x' - x\|_p \leq \epsilon\}$. 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:

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

and assume $0 \in \tilde{\mathcal{H}}$. 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. (2019))** *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 $f \in \mathcal{F}$:*

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

**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 ($\zeta$-cover)** *Let $\zeta > 0$ 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 $\zeta$-cover of $\mathcal{V}$ if for any $v \in \mathcal{V}$, there exists $v' \in \mathcal{C}$ such that $d(v, v') \leq \zeta$. The $\zeta$-covering number of $\mathcal{V}$, denoted as $\mathcal{N}(\mathcal{V}, d(\cdot, \cdot), \zeta)$, is defined as the minimum cardinality $|\mathcal{C}|$ over all possible $\zeta$-covers.*

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 $\zeta$-covering number of $\mathcal{H}$ is then defined as $\mathcal{N}(\mathcal{H}, \|\cdot\|_{\mathcal{S}}, \zeta)$.

**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 $L_\rho$-Lipschitz activation function and $\rho(0) = 0$, $W_j$ 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. We use the superscript binary to indicate the binary setting. In this case, functions $f \in \mathcal{F}^{\text{binary}}$ have scalar outputs, i.e., $h_l = 1$. Without this superscript, the notation refers to the general multiclass setting, where $f$ outputs a vector. 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 the data matrix whose $i$-th row is $x_i^\top$.

## 4. Main Challenges in Bounding ARC

In this section, we discuss the fundamental challenges encountered in bounding ARC and present our approach to addressing these challenges.

The main difficulty in bounding the ARC of DNNs comes from the fact that adversarial perturbations destroy the recursive structure that underlies the existing analyses of Rademacher complexity for DNNs.

To explain this point, recall that the existing approaches for bounding the Rademacher complexity of deep neural networks mainly follow two related routes. The first route directly peels the Rademacher complexity layer by layer. Let $\mathcal{F}_l$ denote the $l$-layer network class, this approach establishes a recursive relationship of the form

$$\mathcal{R}_{\mathcal{S}}(\mathcal{H}_l) \quad \text{in terms of} \quad \mathcal{R}_{\mathcal{S}}(\mathcal{H}_{l-1}),$$

where the last layer is peeled off and the remaining part is treated as an $(l-1)$-layer network class. The second route bounds the Rademacher complexity through covering numbers. In this case, the key step is again recursive: one constructs the covering number of the $l$-layer class, denoted as $\mathcal{N}_l$, from the covering numbers of the previous layers, schematically,

$$\mathcal{N}_l \quad \text{in terms of} \quad \mathcal{N}_1, \ldots, \mathcal{N}_{l-1}.$$

Although these two routes look different, both of them rely on the same implicit fixed-sample structure. At each recursive step, the input feature set to the current layer is treated as fixed once the previous layers have been handled.

More concretely, consider an $l$-layer neural network and write

$$h_0(x) = x, \qquad h_j(x) = \sigma(W_j h_{j-1}(x)), \quad j = 1, \ldots, l-1.$$

In the standard, non-adversarial setting, the empirical process is evaluated on the fixed samples $\{x_i\}_{i=1}^n$. Thus, at the $j$-th layer, the relevant feature set is

$$\{h_{j-1}(x_i) : i = 1, \ldots, n\}.$$

This feature set is generated from the fixed training samples and the previous layers. Hence, after the previous layers have been peeled or covered, the analysis of the next layer can be performed conditionally on this empirical feature set. This is the fixed-sample structure implicitly used by both the direct Rademacher peeling method and the covering number method.

The adversarial setting breaks this structure. In ARC, the relevant input for $x_i$ is selected by an inner maximization, for example

$$x_i^*(f) \in \arg \max_{\|x - x_i\| \le \epsilon} \ell(f(x), y_i).$$

Thus the effective sample is no longer the fixed sample $\{x_i\}_{i=1}^n$, but the network-dependent sample

$$\{x_i^*(f) : i = 1, \ldots, n\}.$$

This network dependence prevents the standard recursive arguments from going through.

For the direct Rademacher peeling route, the obstruction is the following. A recursion from an $(l-1)$-layer ARC to an $l$-layer ARC would require comparing the adversarial process of the $(l-1)$-layer subnetwork with that of the $l$-layer network. However, the adversarial examples associated with these two objects are generally different:

$$x_i^*(W_{l-1}, \cdots, W_1) \neq x_i^*(W_l, W_{l-1}, \cdots, W_1).$$

The former is selected according to an $(l-1)$-layer network, while the latter is selected according to the full $l$-layer network. Therefore, the recursive step

$$\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}_{l-1}) \to \mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}_l)$$

does not have the same fixed samples on both sides. The sample on which the $(l-1)$-layer quantity is evaluated is not the sample that appears inside the $l$-layer adversarial process. This mismatch is precisely what prevents a direct layer-by-layer peeling argument.

For the covering number route, the obstruction appears in two places: one is the comparison between two nearby networks, and the other is again the recursive step. Let $f$ and $f'$ be two networks.

First, in the standard setting, one controls their empirical discrepancy on the same fixed samples:

$$\big|f(x_i) - f'(x_i)\big|.$$

In the adversarial setting, however, the two adversarial examples

$$x_i^*(f) = x_i^*(W_l, \cdots, W_1) \text{ and } x_i^*(f') = x_i^*(W_l', \cdots, W_1')$$

are different in general. Therefore, the relevant comparison is no longer between two networks evaluated on the same input, but between two networks evaluated on different network-dependent adversarial examples.

Second, in the standard setting, this common sample set allows the covering number to be calculated recursively across layers. In the adversarial setting, however, the covering numbers of different layers are evaluated on different adversarial examples, since the adversarial examples depend on the network being covered. Consequently, similar to the direct Rademacher-complexity route, the usual layerwise covering recursion

$$\mathcal{N}_1, \ldots, \mathcal{N}_{l-1} \longrightarrow \mathcal{N}_l$$

does not directly go through.

This motivates us to construct an adversarial example that can play the role of a common reference point in the recursive analysis. The goal is to define layerwise adversarial examples that are not fixed in the same sense as the original samples in the standard setting, but are sufficiently stable and structured to support the desired recursions. To this end, we introduce intermediate adversarial examples. They serve as replacements for the fixed intermediate feature sets in the non-adversarial theory. They allow us to track adversarially selected inputs through the network layer by layer and thereby recover a recursive structure for the adversarial Rademacher complexity analysis.

### 4.1 Intermediate Adversarial Examples

In this section, we use binary classification with a one-dimensional linear function as a simple example to illustrate how intermediate adversarial examples are defined and used to bound the ARC. Following Yin et al. (2019) and Awasthi et al. (2020a), we define the loss function as $\ell(f(x), y) = \phi(yf(x))$, where $\phi$ 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 $\phi$ is $L_\phi$-Lipschitz, by Talagrand's Lemma (Ledoux and Talagrand, 2013), we have $\mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{H}}) \leq L_\phi \mathcal{R}_{\mathcal{S}}(\tilde{\mathcal{F}}^{\text{binary}})$, where we define the adversarial hypothesis class as

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

Let $f_w(x) : \mathcal{X} \to \mathbb{R}$ with $|w| \leq M$ and $\mathcal{X} \subset \mathbb{R}$, and define its adversarial function as

$$g_w(x) \triangleq \min_{x' \in [x-\epsilon,\, x+\epsilon]} f_w(x').$$

For simplicity, suppose we have only one sample $x$ with $|x| = B$. The problem is as follows:

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

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

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

The idea is to relate a cover of the function class to a cover of the parameter region. Consider the one-dimensional linear function $f_w(x) = wx, w \in [-M, M]$. For two parameters $w_1, w_2 \in [-M, M]$, we have

$$|f_{w_1}(x) - f_{w_2}(x)| = |(w_1 - w_2)x| \leq |w_1 - w_2|\,|x|.$$

Therefore, if $|x| \leq B$ and $|w_1 - w_2| \leq \epsilon_w := \frac{\zeta}{B}$, then $|f_{w_1}(x) - f_{w_2}(x)| \leq \zeta$. Thus, an $\epsilon_w$-cover of the parameter interval $[-M, M]$ induces a $\zeta$-cover of the function class $\mathcal{F}^{\text{binary}}$ on the domain $\{x : |x| \leq B\}$. It remains to bound the size of such a parameter cover. For example,

$$\{M, M - 2\epsilon_w, M - 4\epsilon_w, \ldots\}$$

is a $2\epsilon_w$-spaced grid over $[-M, M]$, whose size is at most $\frac{2M}{2\epsilon_w} = \frac{M}{\epsilon_w} = \frac{MB}{\zeta}$. Hence, for the one-dimensional linear class, the covering number satisfies

$$\mathcal{N}(\mathcal{F}^{\text{binary}}, \zeta) \leq \frac{MB}{\zeta}.$$

Now we return to Problem 1 and show how to define and use intermediate adversarial examples to solve this problem. Given an original example $x$, let

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

The intended idea is that $x_1^*$ and $x_2^*$ are two different adversarial examples derived from the same original sample $x$, constructed against $w_1$ and $w_2$, respectively. Let

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

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 $w_2 x_2^* - w_1 x_1^* \le w_2 x_1^* - w_1 x_1^* = w_2 \bar{x} - w_1 \bar{x}$. Combine these two inequalities, we have

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

The choice of $\bar{x}$ in the two-dimensional case is illustrated in Figure 1. Consider the original sample $x = (0, 0)$. Given $w_1$ and $w_2$, the points $x_1^*$ and $x_2^*$ lie on the boundary of the $\epsilon$-ball in the directions opposite to $w_1$ and $w_2$, respectively. If we assume that the magnitude of $w_1$ is larger than that of $w_2$, then $w_1^\intercal x_1^* < w_2^\intercal x_2^*$ (In the two-dimensional case, $wx$ becomes the inner product between $w$ and $x$). Accordingly, we set $\bar{x} = x_1^*$.

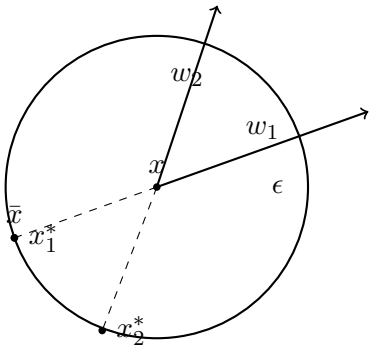



Figure 1: A schematic illustration of the choice of $\bar{x}$. The points $x_1^*$ and $x_2^*$ within the ball $\|x' - x\| \le \epsilon$ are the minimizers of $w_1^\intercal x'$ and $w_2^\intercal x'$, respectively. The point $\bar{x}$ is chosen as either $x_1^*$ or $x_2^*$, depending on which of $w_1^\intercal x'$ and $w_2^\intercal x'$ is smaller.

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

**Remark 1.** This approach recovers the upper bound of ARC for linear functions, as established in Khim and Loh (2018); Yin et al. (2019). 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 5 (Intermediate Adversarial Example)** *Given $(x, y)$ and perturbation set $\mathcal{B}(x)$. Suppose $\mathcal{B}(x)$ is compact. 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 Layer-wise Induction with Intermediate Adversarial Examples

We now return to the general multidimensional, multilayer, multiclass setting. 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 $\mathcal{H}$ is derived through induction, expressing the bound in Eq. (6) as a sum of covering numbers over matrix space of $W_j x_{j-1}$.

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

for some $\zeta$. 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 5, which bounds the covering number of the adversarial hypothesis class using the covering number of the weight space.

**Lemma 6 (Layer-wise Induction for Adversarial Hypothesis Class)** *Let $(\delta_1, \ldots, \delta_l)$ be given, along with Lipschitz activation function $\rho$ (where $\rho$ is $L_\rho$-Lipschitz and $\rho(0) = 0$). Let the loss function $\ell(f(x), y)$ be $L_\phi$-Lipschitz with respect to the first argument, i.e., $|\ell(u, y) - \ell(v, y)| \leq L_\phi \|u - v\|_2$. Let the function class be $\mathcal{F}_2$ or $\mathcal{F}_{1,\infty}$ and the adversarial hypothesis class be defined in (2). Define*

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

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

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

The proof is based on Lemma 5, which provides the bridge for the layer-wise induction given in Appendix 6. 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 7 (Frobenius Norm Bound)** *Consider the function class $\mathcal{F}_2^{\text{binary}}$, 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*

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

*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 5 (Intermediate Adversarial Example).
- Next, we apply Lemma 6 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 8 ($(1,\infty)$-norm Bound)** *Consider the function class $\mathcal{F}_{1,\infty}^{\text{binary}}$, 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 $\epsilon$, which is unavoidable in adversarial settings. Second, as discussed in Neyshabur et al. (2018) 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. (2020a) established that the ARC is bounded by

$$\mathcal{O}\left(\frac{(B+\epsilon)\sqrt{h_1 d}\sqrt{\log n}M_1M_2}{\sqrt{n}}\right)$$

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

$$\mathcal{O}\left(\frac{(B+\epsilon)\sqrt{h_1 d}M_1M_2}{\sqrt{n}}\right),$$

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

**Theorem 9 (Lower Bound)** *Consider the function class $\mathcal{F}_2^{\text{binary}}$. Let $\tilde{\mathcal{F}}_2^{\text{binary}}$ 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*

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

The proof is based on reducing the problem of lower bounding the ARC of DNNs to that of linear hypothesis classes, and it is provided in Appendix A. For function classes 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 $M(f(x), y) = [f(x)]_y - \max_{y' \neq y} [f(x)]_{y'}$, where $[f(x)]_y$ denotes the $y$-th output of $f(x)$. 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

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

and assume $0 \in \tilde{\mathcal{H}}$. Then, the following generalization bound holds.

**Corollary 10 ((Yin et al., 2019))** *Consider the above adversarial multi-class classification setting. For any fixed $\gamma > 0$, we have with probability at least $1 - \delta$, for all $f \in \mathcal{F}$,*

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

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

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

The proof is based on the fact that $\phi_\gamma(M(\cdot, y))$ is $2/\gamma$-Lipschitz, which is provided in Appendix A. Notably, our bound does not introduce an additional coordinate-wise multiplicative factor in the number of classes $K$. This should not be interpreted as saying that the bound is independent of $K$. In the multiclass setting, the output dimension is $h_l = K$, and the bound also depends on the last-layer norm constraint $M_l$. Thus, the number of classes can affect the bound through the architecture and the norm constraints. The point is that our covering-number argument controls the multiclass adversarial loss class directly, rather than decomposing it into $K$ scalar coordinate classes.

## 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):

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

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 $\epsilon$. 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, $\gamma$. By definition, these terms are independent of the algorithm. However, they are implicitly algorithm-dependent (Bartlett et al., 2017; Neyshabur et al., 2018, 2017), with more discussion provided in Appendix C.2.1. 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, $\gamma$ 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:

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

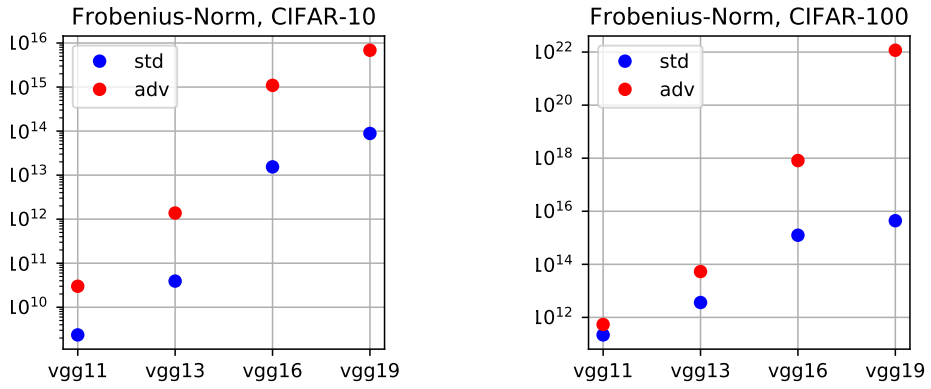
In the previous section, we identified the normalized product of weight norms $\prod_{j=1}^{l} \|W_j\|/\gamma$ as a key algorithm-dependent factor in the ARC bounds. To empirically validate our theoretical analysis, we conducted extensive experiments comparing these terms between standard and adversarial training settings. Since our bounds generalize to convolutional neural networks, we evaluated VGG architectures (Simonyan and Zisserman, 2015) on both CIFAR-10 and CIFAR-100 datasets (Krizhevsky, 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 $5 \times 10^{-4}$, which was empirically determined to be optimal for robust accuracy, though we explored other values in ablation studies. For adversarial training, we implemented $\ell_\infty$ PGD (Madry et al., 2018) 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. (2017), 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.

**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[1]. 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.

1. 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.




(a) (b)

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

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% |

**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% robust 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 $\epsilon$.

**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.

## Acknowledgments

We thank the action editor and the anonymous reviewers for their thoughtful comments and constructive suggestions. This work was supported by Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWS-KCCYB-2024016); Guangdong Provincial Key Laboratory of Mathematical Foundations for Artificial Intelligence (2023B1212010001).

## References

Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd annual symposium on foundations of computer science (FOCS)*, pages 977–988. IEEE, 2022.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for adversarially robust learning. *Journal of Machine Learning Research*, 23(175):1–31, 2022.

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, 2020a.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. On the rademacher complexity of linear hypothesis sets. *arXiv preprint arXiv:2007.11045*, 2020b.

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.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

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 adversaries. *Advances in Neural Information Processing Systems*, 31, 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.

Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *Machine Learning with Applications*, 3:100017, 2021.

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. *International Conference on Learning Representations*, 2015.

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.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 514–529, 2018.

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. Learning multiple layers of features from tiny images. *Master's thesis, Deptartment of Computer Science, University of Toronto*, 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. *International Conference on Learning Representations*, 2018.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

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 Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *International Conference on Learning Representations*, 2018.

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

Hamed Rahimian and Sanjay Mehrotra. Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85, 2022.

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.

Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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. *International Conference on Learning Representations*, 2015.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

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

Hossein Taheri, Ramtin Pedarsani, and Christos Thrampoulidis. Asymptotic behavior of adversarial training in binary linear classification. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 127–132. IEEE, 2022.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 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. *Advances in neural information processing systems*, 33:2958–2969, 2020.

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

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, Jiawei Zhang, Zhi-Quan Luo, and Asuman E. Ozdaglar. Smoothed-SGDmax: A stability-inspired algorithm to improve adversarial generalization. In *NeurIPS ML Safety Workshop*, 2022c.

Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. Pac-bayesian spectrally-normalized bounds for adversarially robust generalization. *Advances in Neural Information Processing Systems*, 36:36305–36323, 2023.

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.

Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding adversarial robustness against on-manifold adversarial examples. *Pattern Recognition*, 159: 111071, 2025.

Yue Xing, Qifan Song, and Guang Cheng. On the generalization properties of adversarial training. In *International Conference on Artificial Intelligence and Statistics*, pages 505–513. PMLR, 2021a.

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

Yue Xing, Ruizhi Zhang, and Guang Cheng. Adversarially robust estimate and risk analysis in linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 514–522. PMLR, 2021c.

Yue Xing, Qifan Song, and Guang Cheng. Phase transition from clean training to adversarial training. *Advances in Neural Information Processing Systems*, 35:9330–9343, 2022a.

Yue Xing, Qifan Song, and Guang Cheng. Unlabeled data help: Minimax analysis and adversarial robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 136–168. PMLR, 2022b.

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

Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*, 2018.

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.

## Appendix A. Proofs of Technical Results

### A.1 Proof of Lemma 5

**Proof**

Since $\mathcal{B}(x)$ is compact and for all $h \in \mathcal{H}$, $h(\cdot, y)$ is continuous, $\arg\max_{x' \in \mathcal{B}(x)} h(x', y)$ is attainable. Let

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

By the property of $\arg\max$, we have

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

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

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

**Case 1:** If $h_1(x(\tilde{h}_1), y) \geq h_2(x(\tilde{h}_2), y)$, then

$$0 \leq h_1(x(\tilde{h}_1), y) - h_2(x(\tilde{h}_2), y) \leq h_1(x(\tilde{h}_1), y) - h_2(x(\tilde{h}_1), y).$$

Since both $h_1(x(\tilde{h}_1), y) - h_2(x(\tilde{h}_2), y)$ and $h_1(x(\tilde{h}_1), y) - h_2(x(\tilde{h}_1), y)$ are nonnegative, it follows that

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

In this case, $\bar{x}(\tilde{h}_1, \tilde{h}_2) = x(\tilde{h}_1)$, thus

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

**Case 2:** If $h_1(x(\tilde{h}_1), y) < h_2(x(\tilde{h}_2), y)$, then

$$0 \leq h_2(x(\tilde{h}_2), y) - h_1(x(\tilde{h}_1), y) \leq h_2(x(\tilde{h}_2), y) - h_1(x(\tilde{h}_2), y).$$

Since both $h_2(x(\tilde{h}_2), y) - h_1(x(\tilde{h}_1), y)$ and $h_2(x(\tilde{h}_2), y) - h_1(x(\tilde{h}_2), y)$ are nonnegative, it follows that

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

In this case, $\bar{x}(\tilde{h}_1, \tilde{h}_2) = x(\tilde{h}_2)$, thus

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

Therefore, in either case 1 or case 2, we conclude that

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

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

### A.2 Proof of Lemma 6

The proof of Lemma 6 requires the following Lemma.

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

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

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

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

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

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

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

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

■

**Lemma 2** *Let $A$ be an $m \times k$ matrix and $b$ be an $n$-dimensional vector. Then,*

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

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

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

■

**Lemma 3** *Let $A$ be an $m \times k$ matrix and $b$ be an $n$-dimensional vector. Then,*

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

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

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

■

Now, we move the the proof of Lemma 6.

**Proof** In Frobenius norm case, let $r = 2$ and $\mathcal{C}_j$ denote $\delta_j$-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 $\delta_j$-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

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

Notably, the definition of $\mathcal{C}_j$ and $\mathcal{C}_j^m$ are different, particularly in dimension. 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 5, 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 $L_\phi$-Lipschitz with respect to the first argument,

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

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 $f(\bar{x}_i) = g_l^0(\bar{x}_i)$, $f^c(\bar{x}_i) = g_l^l(\bar{x}_i)$. We can decompose

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

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 2, (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 2, inequality (ii) is due to Lemma 2 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

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

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

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

where

$$D = 2L_\rho^{l-1} \prod_{k=1}^{l} M_k \max\left\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\right\} (\|x\|_{p,\infty} + \epsilon).$$

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

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

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

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

Therefore,

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

■

### A.3 Proof of Theorem 7 and 8

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

**Proposition 4 (Dudley's integral)** *The Rademacher complexity $\mathcal{R}_S(\mathcal{F}^{\text{binary}})$ satisfies*

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

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 $\mathcal{F}^{\text{binary}}$ from its covering number bounds.

**Lemma 5 (Covering number of norm-balls (Lemma 6.27 in Mohri et al. (2018))** *Let $\mathcal{B}$ be a $\ell_p$ norm ball with radius $W$. Let $d(x_1, x_2) = \|x_1 - x_2\|_p$. Define the $\zeta$-covering number of $B$ as $\mathcal{N}(\mathcal{B}, d(\cdot,\cdot), \zeta)$, we have*

$$\mathcal{N}\left(\mathcal{B}, d(\cdot,\cdot), \zeta\right) \leq \left(\frac{3W}{\zeta}\right)^d.$$

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

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

Now we move to the proof of Theorem 7 and 8.

**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 $L_\phi$-Lipschitz with respect to the first argument. Based on Lemma 6, define

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

where $r = 2$ for Frobenius norm, which corresponds to Theorem 7, and $r = 1$ for $(1,\infty)$-norm, and which corresponds to Theorem 8. Then:

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

where inequality (i)) is due to Lemma 5.

Next, we consider the diameter of $\tilde{\mathcal{H}}$. Since $0 \in \tilde{\mathcal{H}}$, for all $x, y$, there exists a function $f' \in \mathcal{F}$, such that $\tilde{h}'(x,y) = \max_{x' \in \mathcal{B}(x)} \ell(f'(x'), y) = 0$. $\forall (x_i, y_i), i = 1, \cdots, n$, given $\tilde{h}$ and $\tilde{h}'$, by Lemma 5, there exist an intermediate adversarial example $\bar{x}_i$, such that

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

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

$$\begin{aligned}
\left|\tilde{h}(x_i, y_i) - \tilde{h}'(x_i, y_i)\right| \leq & L_\phi \left|f(\bar{x}_i) - f'(\bar{x}_i)\right| \\
\leq & L_\phi \left(|f(\bar{x}_i)| + \left|f'(\bar{x}_i)\right|\right) \\
\leq & 2L_\rho^l \prod_{k=1}^{l} M_k \max\left\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\right\} (\|x\|_{p,\infty} + \epsilon)
\end{aligned}$$

where the last inequality is due to Equation (15). Therefore, the diameter of $\tilde{\mathcal{H}}$ is no larger than

$$4L_\phi L_\rho^l \prod_{k=1}^{l} M_k \max\left\{1, d^{1-\frac{1}{r}-\frac{1}{p}}\right\} (\|x\|_{p,\infty} + \epsilon),$$

which we denoted as $D_{\tilde{\mathcal{H}}}$. By Dudley's integral, we have

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

Let $\zeta' = 2\zeta/D_{\tilde{\mathcal{H}}}$. By definition $D_{\tilde{\mathcal{H}}} = 2L_\phi D$, then $\zeta' = 2\zeta/D_{\tilde{\mathcal{H}}} = \zeta/(L_\phi D)$. Therefore,

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

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

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

By Cauchy–Schwarz,

$$\int_0^1 \sqrt{\log\left(\frac{3l}{2\zeta'}\right)}\, d\zeta' \;\le\; \sqrt{\int_0^1 \log\left(\frac{3l}{2\zeta'}\right)\, d\zeta'}.$$

A direct computation gives

$$\int_0^1 \log\left(\frac{3l}{2\zeta'}\right)\, d\zeta' = \log\frac{3l}{2}\int_0^1 d\zeta' - \int_0^1 \log\zeta'\, d\zeta' = \log\frac{3l}{2} + 1,$$

since $\int_0^1 \log\zeta'\, d\zeta' = -1$. Therefore

$$\int_0^1 \sqrt{\log\left(\frac{3l}{2\zeta'}\right)}\, d\zeta' \;\le\; \sqrt{\log\frac{3l}{2} + 1}.$$

Finally, because $l \ge 2$ implies $\log(3l) \ge \log 6$, we have

$$\sqrt{\log\frac{3l}{2} + 1} = \sqrt{\log(3l) + (1 - \log 2)} \;\le\; 2\sqrt{\log(3l)}.$$

Plugging it to Eq. (17), we have

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

If $r = 1$, the bound reduces to the Frobenius norm bound in Theorem 7. If $r = 1$, the bound reduces to the $(1,\infty)$-norm bound in Theorem 8. ■

### A.4 Proof of Theorem 9

Let $\rho(\cdot)$ be the identity activation function. The following discussion hods for both $\ell_2$-norm and $\ell_{1,\infty}$ norm cases. 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 $\tilde{\mathcal{F}}^{\text{binary}'}$ under a specific distribution $\mathcal{D}$, where $\tilde{\mathcal{F}}^{\text{binary}'}$ is a subset of $\tilde{\mathcal{F}}^{\text{binary}}$. We define

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

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

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

$$W_1 = \begin{bmatrix} w \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{h_1 \times h_0}, \quad W_j = \begin{bmatrix} M_j & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{h_j \times h_{j-1}}, \quad j = 2, \ldots, l. \tag{18}$$

Since $\|W_j\| \leq M_j$, imposing the additional constraint in Eq. (18) on $\tilde{\mathcal{F}}^{\text{binary}}$ reduces it to $\tilde{\mathcal{F}}^{\text{binary}'}$, confirming that $\tilde{\mathcal{F}}^{\text{binary}'}$ is a subset of $\tilde{\mathcal{F}}^{\text{binary}}$.

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. (2020a), which we state below.

**Proposition 4** *Given the function class*

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

*and*

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

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

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

The standard Rademacher complexity of linear hypothesis classes can be expressed as

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

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

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

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

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

As discussed in Awasthi et al. (2020b), this lower bound is in tight in terms of dependence on sample size $n$ and dimension $d$. 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

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

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

### A.5 Proof of Theorem 11

**Proof** For any $a, b \in \mathbb{R}^K$, we have

$$\begin{aligned}
|M(a,y) - M(b,y)| &= \left|a_y - b_y - \left(\max_{y' \neq y} a_{y'} - \max_{y' \neq y} b_{y'}\right)\right| \\
&\leq |a_y - b_y| + \left|\max_{y' \neq y} a_{y'} - \max_{y' \neq y} b_{y'}\right| \\
&\leq 2\|a - b\|_2.
\end{aligned}$$

Since $\phi_\gamma$ is $1/\gamma$-Lipschitz, the loss function $\ell(f(x), y) = \phi_\gamma(M(f(x), y))$ is $2/\gamma$-Lipschitz with respect to the first argument.

Now we apply Lemma 6 to the adversarial hypothesis class $\tilde{\mathcal{H}}$ in Eq. (9) with $L_\phi = 2/\gamma$ and $r = 2$. For any $(\delta_1, \ldots, \delta_l)$, define

$$\zeta = \frac{2}{\gamma}\sum_{j=1}^{l} L_\rho^{l-1} \frac{\prod_{k=1}^{l} M_k}{M_j} \max\left\{1, d^{\frac{1}{2}-\frac{1}{p}}\right\} (\|X\|_{p,\infty} + \epsilon)\, \delta_j.$$

Then

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

By the standard covering number bound for Euclidean balls,

$$\ln \mathcal{N}\left(\{W_j : \|W_j\|_F \leq M_j\}, \|\cdot\|_F, \delta_j\right) \leq h_j h_{j-1} \log\left(\frac{3M_j}{\delta_j}\right).$$

Following the same choice of $\delta_j$ and the same Dudley integral calculation as in the proof of Theorem 7, we obtain

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

This completes the proof. ■

## Appendix 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 $L_\rho$ 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 $\kappa$ 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 $\kappa$ 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.

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

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. (2022), 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 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 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. (2022) 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:

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

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., 2022; 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., 2021b; Xiao et al., 2022a,b,c, 2024) suggest that poor generalization may result from the non-smooth nature of adversarial loss functions.

**Related Work on Regularization and Robustness.** The connection between robustness and regularization has been widely studied in the literature. A line of work has explored Jacobian regularization, where robustness is promoted by directly penalizing the sensitivity of model outputs to input perturbations (Jakubovitz and Giryes, 2018; Yu et al., 2018; Ross and Doshi-Velez, 2018; Finlay and Oberman, 2021). Another line has investigated weight regularization, where robustness emerges from penalizing the norm of the model parameters (Blanchet et al., 2019; Sinha et al., 2018; Rahimian and Mehrotra, 2022). Since weight decay is equivalent to $\ell_2$ regularization, our results naturally relate to this literature, suggesting that weight decay not only controls model capacity but may also contribute to adversarial robustness. We highlight these connections to situate our contribution within the broader context of robustness through regularization. Related work by Xing and collaborators investigates adversarial robustness through robust statistics, generalization theory, training dynamics, and minimax analysis, covering topics such as robust linear regression, the generalization behavior of adversarial training, phase transitions from clean to adversarial training, and the benefit of unlabeled data for robustness (Xing et al., 2021c,a, 2022a,b).

**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. (2014), suggests that adversarial examples deviate from the underlying data manifold. Supporting this view, Song et al. (2018) 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.

## Appendix 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

#### C.2.1 Discussion of optimizing over Gamma

By definition, these terms are independent of the algorithm. However, they are implicitly algorithm-dependent (Bartlett et al., 2017; Neyshabur et al., 2018). The optimization over the margin parameter $\gamma$ is standard in empirical evaluations of margin-based generalization bounds. For example, Bartlett et al. (2017) derive a spectrally-normalized margin bound based on Rademacher-complexity arguments and empirically evaluate the bound as a function of the normalized margin. The experiment is reported in Figure 1 in their paper. Similarly, Neyshabur et al. (2017) empirically compute and compare several generalization bounds, including Rademacher-complexity-based norm bounds and spectrally-normalized margin bounds. Their public implementation[2] reports the following bounds:

2. `https://github.com/bneyshabur/generalization-bounds.`

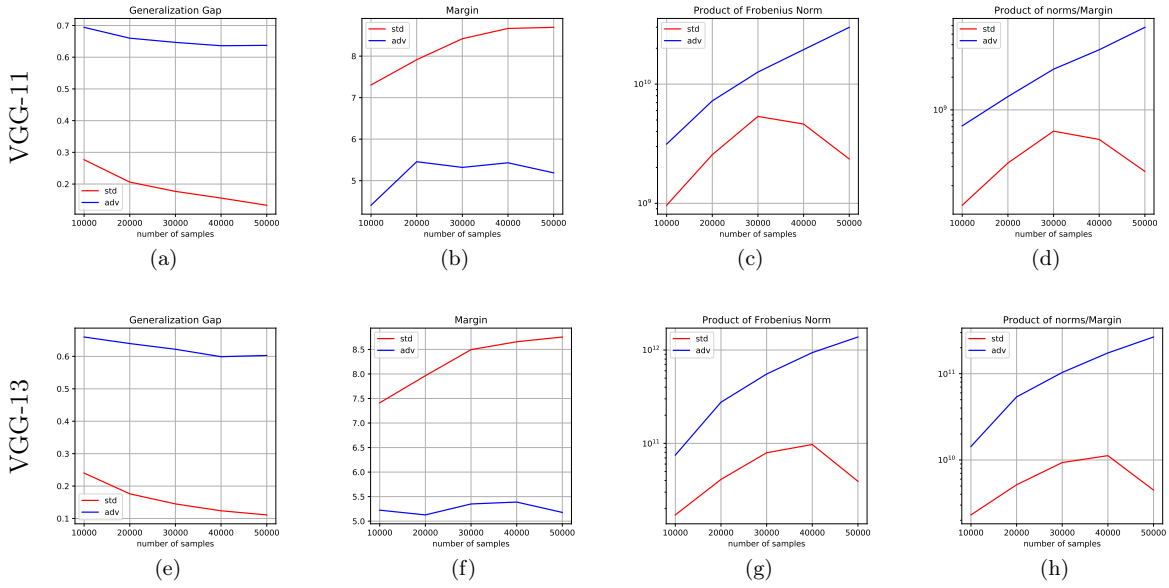



Figure 3: Comparison of Frobenius norm products across VGG architectures. Results from standard training (red lines) and adversarial training (blue lines) for VGG-11 (top row) and VGG-13 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin $\gamma$, (c,g) Product of layer-wise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

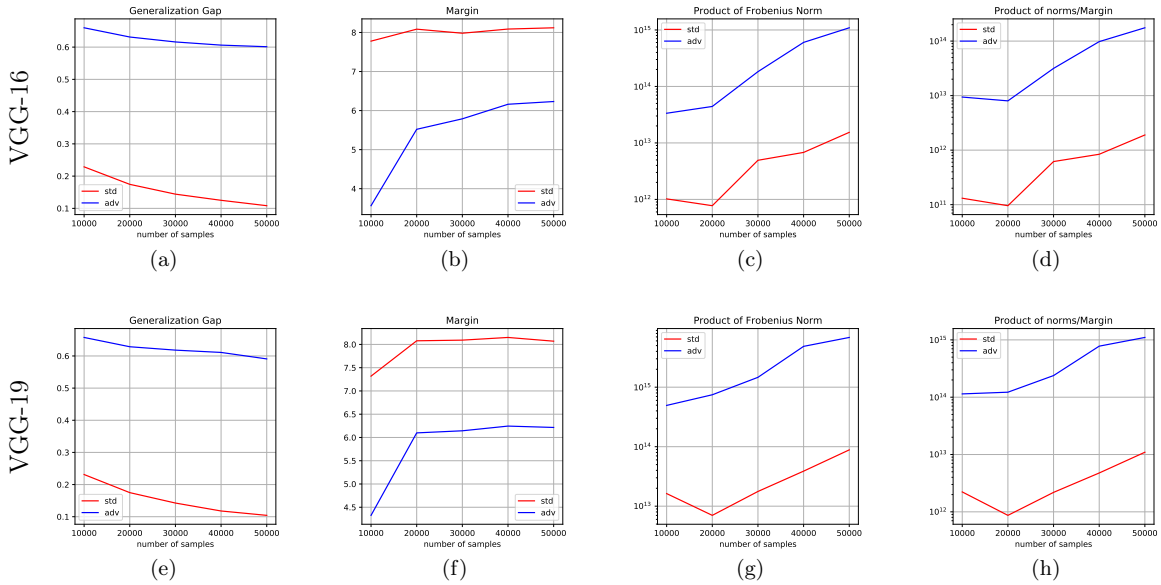



Figure 4: CIFAR-10 experimental results comparing standard training (red lines) and adversarial training (blue lines) for VGG-16 (top row) and VGG-19 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin, (c,g) Product of layer-wise Frobenius norms $\prod_{j=1}^{l} \|W_j\|_F$,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

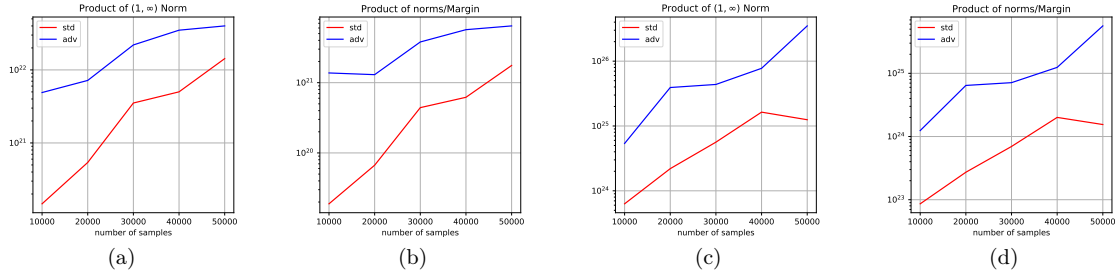



Figure 5: Comparison of $\|\cdot\|_{1,\infty}$-norm products on CIFAR-10 between standard training (red lines) and adversarial training (blue lines) for: VGG-16 (a,b) and VGG-19 (c,d). The plots show: (a,c) Product of layer-wise $\|\cdot\|_{1,\infty}$-norms $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$, and (b,d)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

Table 3: Generalization bounds computed in Neyshabur et al. (2017).

| **Bound** | **Reference** | **Type** |
|---|---|---|
| $L_{1,\infty}$ bound | Bartlett and Mendelson (2002), with depth dependence from Golowich et al. (2018) | Rademacher |
| $L_{3,1.5}$ bound | Neyshabur et al. (2015), with depth dependence from Golowich et al. (2018) | Rademacher |
| Frobenius bound | Neyshabur et al. (2015), with depth dependence from Golowich et al. (2018) | Rademacher |
| $\text{Spec}_{L_1}$ bound | Bartlett et al. (2017) | Rademacher margin |
| $\text{Spec}_{\text{Fro}}$ bound | Neyshabur et al. (2018) | PAC-Bayesian margin |

#### C.2.2 Results

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 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 4 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 4: 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 |

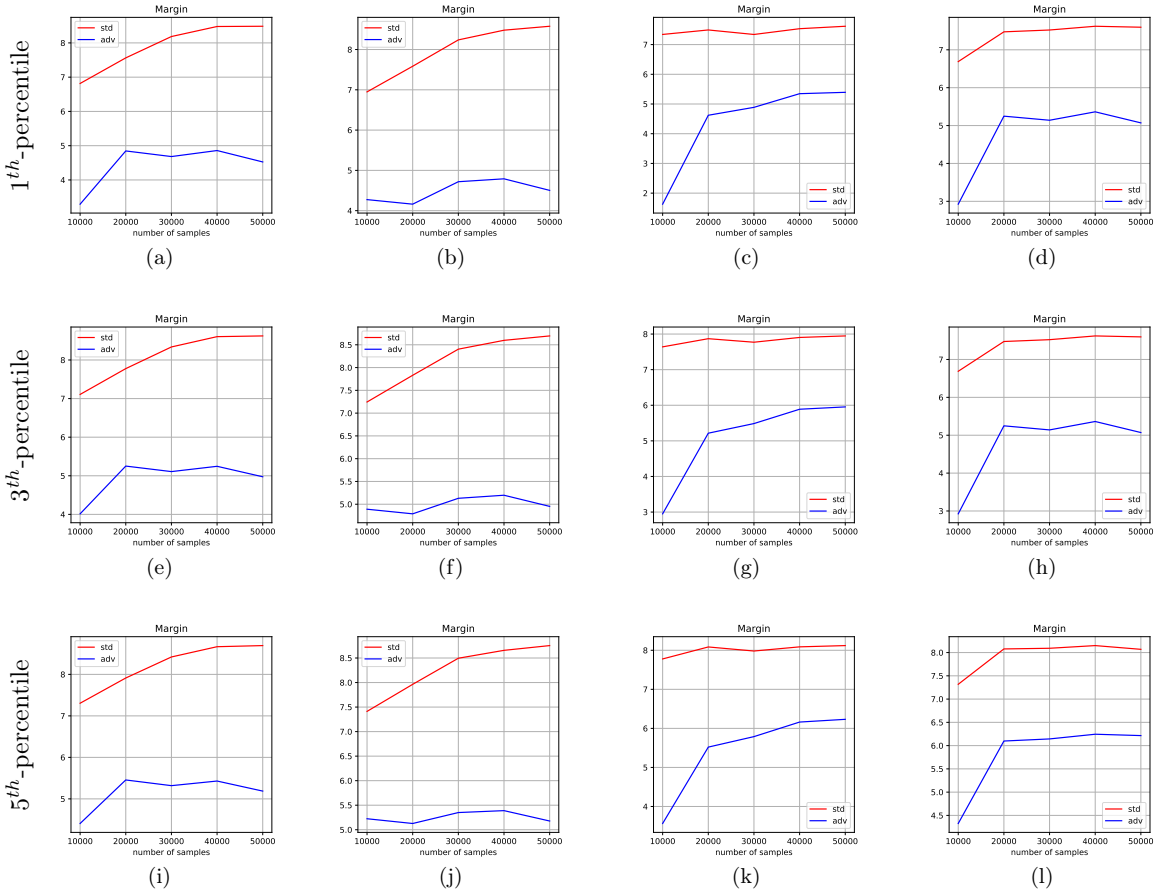



Figure 6: Margin analysis across VGG architectures: Results from VGG-11 (first column), VGG-13 (second column), VGG-16 (third column), and VGG-19 (fourth column).

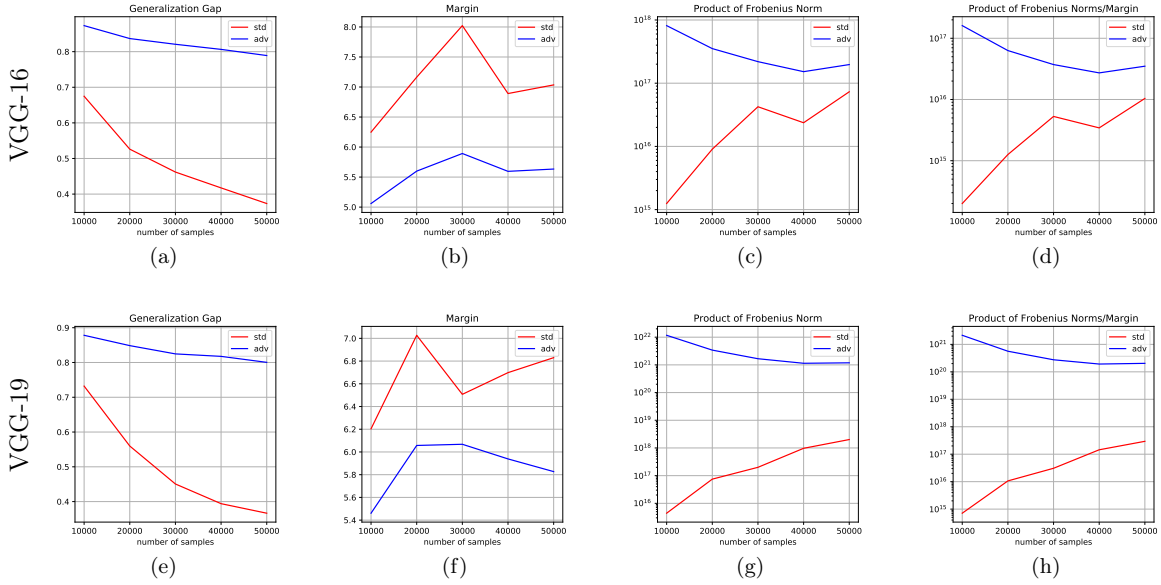



Figure 7: CIFAR-100 experimental results comparing standard training (red lines) and adversarial training (blue lines) for VGG-16 (top row) and VGG-19 (bottom row). The plots show: (a,e) Generalization gap, (b,f) Training set margin $\gamma$, (c,g) Product of layer-wise Frobenius norms $\prod_{j=1}^{l}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

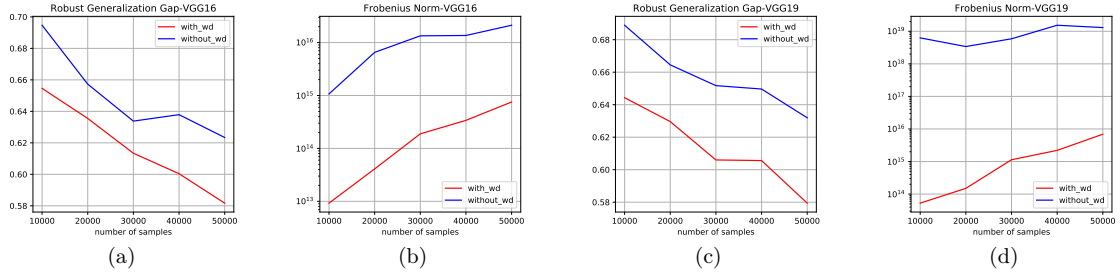
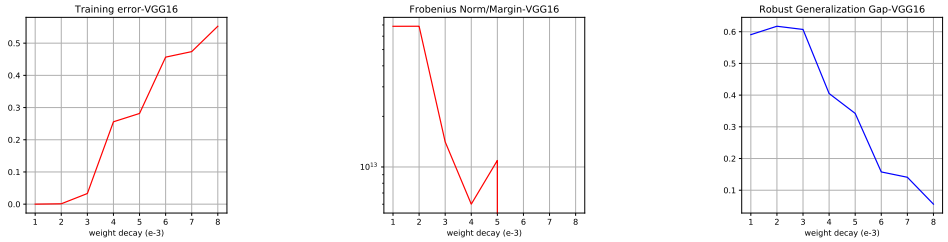



Figure 8: Impact of weight decay on VGG architectures: Results for VGG-16 (a,b) and VGG-19 (c,d), comparing models trained with and without weight decay. The plots show: (a,c) Robust generalization gap and (b,d) Product of layer-wise Frobenius norms.




(a) (b) (c)

Figure 9: Weight decay effects on model behavior for values ranging from $1\times10^{-3}$ to $9\times10^{-3}$: (a) Training error performance, (b) Frobenius norm of model weights, and (c) Robust generalization gap between training and test performance.

### 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. 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 $1 \times 10^{-3}$ to $9 \times 10^{-3}$. The training error increases with higher weight decay values. At low weight decay ($1 \times 10^{-3}$), 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 ($9 \times 10^{-3}$), 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., $3 \times 10^{-3}$ to $7 \times 10^{-3}$). 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 $6 \times 10^{-3}$ and higher, the margin becomes negative, indicating high training error. Subsequently, the weight norm to margin ratio also becomes negative. At $9 \times 10^{-3}$, 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 $1 \times 10^{-3}$ and $5 \times 10^{-3}$. The smallest weight norm observed was $6.01 \times 10^{12}$ (with weight decay $= 4 \times 10^{-3}$). However, this value remains larger than the weight norm achieved through standard training ($1.90 \times 10^{12}$). These findings emphasize the importance of selecting an appropriate weight decay value to achieve robust model performance.