Adversarial Rademacher Complexity of Deep Neural Networks

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

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

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

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

Abstract

Deep neural networks are vulnerable to adversarial attacks. Ideally, a robust model shall perform well on both the perturbed training data and the unseen perturbed test data. It is found empirically that fitting perturbed training data is not hard, but generalizing to perturbed test data is quite difficult. To better understand adversarial generalization, it is of great interest to study the adversarial Rademacher complexity (ARC) of deep neural networks. However, how to bound ARC in multi-layers cases is largely unclear due to the difficulty of analyzing adversarial loss in the definition of ARC. There have been two types of attempts of ARC. One is to provide the upper bound of ARC in linear and one-hidden layer cases. However, these approaches seem hard to extend to multi-layer cases. Another is to modify the adversarial loss and provide upper bounds of Rademacher complexity on such surrogate loss in multi-layer cases. However, such variants of Rademacher complexity are not guaranteed to be bounds for meaningful robust generalization gaps (RGG). In this paper, we provide a solution to this unsolved problem. Specifically, we provide the first bound of adversarial Rademacher complexity of deep neural networks. Our approach is based on covering numbers. We provide a method to handle the robustified function classes of DNNs such that we can calculate the covering numbers. Finally, we provide experiments to study the empirical implication of our bounds and provide an analysis of poor adversarial generalization.

Keywords: Rademacher Complexity, Adversarial Robustness, Generalization Bounds

1. Introduction

Deep neural networks (DNNs) (Krizhevsky et al., 2012; Hochreiter and Schmidhuber, 1997) have become successful in many machine learning tasks such as computer vision (CV) and natural language processing (NLP). But they are shown to be vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014). More specifically, a well-trained model performs badly on slightly perturbed data samples. Adding perturbed samples into the training dataset can improve robustness in practice, but it still cannot lead to satisfactory performance. One of the major challenges comes from generalization. It is found that training a model to fit perturbed training samples is

* Corresponding Author.
relatively easy, but such a model does not perform well on the adversarial examples of the test set. For example, when applying ResNet to CIFAR-10, adversarial training can achieve nearly 100% robust accuracy on the training set, but it only gets 47% robust accuracy on the test set (Madry et al., 2017). Recent works (Gowal et al., 2020; Rebuffi et al., 2021) mitigated the overfitting issue, but it still has a 20% robust generalization gap between robust test accuracy (about 60%) and robust training accuracy (about 80%). Therefore, it is interesting to provide a theoretical understanding of adversarial generalization.

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

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

(1)

where $S = \{x_i, y_i\}_{i=1,\ldots,n}$ is the sample dataset, $H$ is the hypothesis function class, and $\sigma_i$ are i.i.d. Rademacher random variable, i.e., $\sigma_i$ equals to $1$ or $-1$ with equality probability. Rademacher complexity of deep neural nets has been well studied. Neyshabur et al. (2015) used a layer-peeling technique to provide a bound: for depth-$l$ neural nets, 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 $n$ input instances have $\ell_2$-norm bounded by $B$, the generalization gap between population risk and empirical risk is bounded by $O(B^2 \prod_{j=1}^l M_j / \sqrt{n})$ with high probability. Bartlett et al. (2017) provided a spectral norm bound for Rademacher complexity by bounding the covering number of function class of DNNs. For more details, see Section 3.

The work of (Khim and Loh, 2018; Yin et al., 2019) concurrently extended Rademacher complexity to adversarial settings to study the causes of poor adversarial generalization. They showed that robust generalization gap (in misclassification error) is bounded by adversarial Rademacher complexity (ARC), which is defined as

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

(2)

with high probability, where $B(x, \epsilon)$ is a norm ball around sample $x$ with radius $\epsilon$.

However, The work of (Khim and Loh, 2018; Yin et al., 2019) both pointed out that it is difficult to provide a bound for ARC in deep neural networks cases due to the $\max$ operation in Eq. (2). Existing methods, e.g., layer peeling (Neyshabur et al., 2015) and covering number argument (Bartlett et al., 2017), cannot be directly applied. There have been two types of attempts to study adversarial Rademacher complexity of neural networks.

**Type 1: Optimal attacks in Linear and one-hidden layer cases.** The work of (Khim and Loh, 2018; Yin et al., 2019) both provided a bound for ARC in linear cases when they introduced ARC. The work of (Awasthi et al. 2020) analyzed the max problem in Eq. (2) in one-hidden layer cases and provided a bound for ARC in this case. These approaches seem hard to be extended to multi-layer cases.

**Type 2: Surrogate loss in multi-layer cases.** Standard method cannot be applied in multi-layer cases. Three work used Lipschitz surrogate loss $\hat{h}(x, y) \approx \max_{x'} h(x', y)$ to bypass the difficulty and applied the layer peeling technique (Neyshabur et al. 2015) or covering number argument
Table 1: Comparison of our work with the two types of attempts on adversarial Rademacher complexity. Type 1: Thm. 2 in (Yin et al., 2019), Lemma 1 & Corollary 1 in (Khim and Loh, 2018), and the work of (Awasthi et al., 2020). Type 2: Thm. 8 in (Yin et al., 2019), Lemma 2 & Corollary 2 in (Khim and Loh, 2018), and the work of (Gao and Wang, 2021). We provide the first bound for adversarial Rademacher complexity of DNNs, resulting the first bound of robust generalization gap of DNNs.

(Bartlett et al., 2017). Surrogate loss \( \hat{h}(x, y) \) includes tree-transformation loss (Khim and Loh, 2018), SDP relaxation loss (Yin et al., 2019), and FGSM loss (Gao and Wang, 2021). However, these approaches change the definition of ARC. It is not guaranteed to be a meaningful bound for robust generalization gap.

To the best of our knowledge, how to bound the adversarial Rademacher complexity of deep neural networks has been an unsolved problem since it was raised in 2018. In this paper, we provide the first bound for adversarial Rademacher complexity of deep neural networks. Our approach is still based on covering number. Then, the main difficulty becomes how to calculate the covering number of robustified function classes. We provide a novel robustified weight perturbation bound such that we can calculate the covering number of robustified function classes.

Specifically, for depth-\( l \), width-\( h \) fully-connected neural nets, with high probability,

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

We provide a comparison with the 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 ARC in one-hidden layer cases. We also provide a lower bound for ARC correspondingly.

Finally, we trained 88 models to study some empirical implications of our bounds. We find that the weight norm positively relates to the adversarial generalization gap in our experiments.

Our contributions are summarized as follows:

1. How to bound ARC is an unsolved problem raised in 2018 with several attempts in recent years. We give the first upper bound. The bound provides a further understanding of the robust generalization of DNNs.

2. Technical contribution: Our approach is based on covering numbers. We provide an robustified weight perturbation bound such that we can calculate the covering number of robustified function classes. This technique might be helpful in another problem.

3. We conduct experiments on CIFAR-10 and CIFAR-100 to study the relationship between the factors in our bounds and the robust generalization gap.
2. Preliminaries

2.1 Generalization Gap and Rademacher Complexity

**Generalization Gap.** We start from the classical machine learning framework. Let $\mathcal{F}$ be the hypothesis class (e.g., linear functions, neural networks). The goal of the learning problem is to find $f \in \mathcal{F}$ to minimize the population risk $R(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(f(x), y)]$, where $\mathcal{D}$ is the true data distribution, and $\ell(\cdot)$ is the loss function. Since $\mathcal{D}$ is unknown, we minimize the empirical risk in practice. Given $n$ i.i.d samples $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, the empirical risk is $R_n(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$. The generalization gap is defined as follow:

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

**Rademacher Complexity.** A classical measure of the generalization gap is Rademacher complexity (Bartlett and Mendelson, 2002). Given the hypothesis class $\mathcal{H}$, the (empirical) Rademacher complexity is defined as

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

where $\sigma_i$ are i.i.d Rademacher random variables, i.e. $\sigma_i$ equals to 1 or $-1$ with equal probability. Define the function class $\mathcal{L} = \{\ell(f(x), y) | f \in \mathcal{F}\}$, then we have the following generalization bound.

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

$$R(f) \leq R_n(f) + 2\mathcal{R}(\mathcal{L}) + 3C\sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$

2.2 Robust Generalization Gap and Adversarial Rademacher Complexity

**Robust Generalization Gap (RGG).** Let $\tilde{\ell}(f(x), y) := \max_{\|x' - x\|_p \leq \epsilon} \ell(f(x'), y)$ be the adversarial loss. The robust population risk and the robust empirical risk are

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

respectively. In this paper we consider general $\ell_p$ attacks for $p \geq 1$. The robust generalization gap is defined as follow:

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

Let the adversarial hypothesis class be $\tilde{\mathcal{L}} = \{\tilde{\ell}(f(x), y) | f \in \mathcal{F}\}$, according to Proposition 1, the robust generalization gap can be bounded by the Rademacher complexity of $\tilde{\mathcal{L}}$. We have the following adversarial generalization bound.

**Proposition 2** (Yin et al., 2019). Suppose that the range of the loss function $\tilde{\ell}(f(x), y)$ is $[0, C]$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the following holds for all $f \in \mathcal{F}$,

$$\tilde{R}(f) \leq \tilde{R}_n(f) + 2\mathcal{R}(\tilde{\mathcal{L}}) + 3C\sqrt{\frac{\log \frac{2}{\delta}}{2n}}.$$
**Binary Classification.** We first discuss the binary classification case, then we discuss the extension to the multi-class classification case in Section 5. Following (Yin et al., 2019; Awasthi et al., 2020), we assume that the loss function can be written as $\ell(f(x), y) = \phi(yf(x))$ where $\phi$ is a non-increasing function. Then

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

Assume that the function $\phi$ is $L_\phi$-Lipschitz, by Talagrand’s Lemma (Ledoux and Talagrand, 2013), we have $R_S^{\tilde{\phi}} \leq L_\phi R_S(\tilde{F})$, where we define the robustified function class as

$$\tilde{F} = \{ \tilde{f} : (x, y) \to \inf_{\|x-x'\|_p \leq \epsilon} yf(x') | f \in F \}. \tag{3}$$

Despite of the loss function $\ell()$, ARC can also be defined in the following form.

**Definition 3 (Adversarial Rademacher Complexity).** We define the Rademacher complexity of the robustified function class $\tilde{F}$ in Eq. (3), i.e.

$$R_S(\tilde{F}) = \mathbb{E}_\sigma \frac{1}{n} \left[ \sup_{f \in F} \sum_{i=1}^n \sigma_i \inf_{\|x-x'\|_p \leq \epsilon} yf(x') \right],$$

as adversarial Rademacher complexity.

Our goal is to give upper bounds for adversarial Rademacher complexity. Then, it induces the guarantee of the robust generalization gap.

**Hypothesis Class.** We consider depth-$l$, width-$h$ fully-connected neural networks,

$$F = \{ x \to W_l \rho(W_{l-1} \rho(\cdots \rho(W_1 x) \cdots)), \|W_j\| \leq M_j, j = 1, \cdots, l \}, \tag{4}$$

where $\rho(\cdot)$ is an element-wise $L_\rho$-Lipschitz activation function, $W_j$ are $h_j \times h_{j-1}$ matrices, for $j = 1, \cdots, l$. We have $h_l = 1$ and $h_0 = d$ is the input dimension. 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, Frobenius norm and $\|\cdot\|_1,\infty$-norm in equation (4). Additionally, we assume that $\|X\|_{p,\infty} = B$, where $X$ is a matrix of the all the samples, the $i^{th}$ rows of $X$ is $x_i^T$.

### 3. Related Work

**Adversarial Attacks and Defense.** Since 2013, it has now been well known that deep neural networks trained by standard gradient descent are highly susceptible to small corruptions to the input data (Szegedy et al., 2013; Goodfellow et al., 2014; Chen et al., 2017; Carlini and Wagner, 2017; Madry et al., 2017). A lines of work aimed at increasing the robustness of neural networks (Wu et al., 2020; Gowal et al., 2020). Another works aimed at finding more powerful attacks (Athalye et al., 2018; Tramer et al. 2020; Chen et al., 2017; Xiao et al., 2022c).
Adversarial Generalization. The work of (Schmidt et al. 2018; Raghunathan et al., 2019; Zhai et al. 2019) has shown that more data can help achieve better adversarial generalization. The work of (Attias et al., 2021. Montasser et al. 2019) explains generalization in adversarial settings using VC-dimension. (Neyshabur et al., 2017b) uses a PAC-Bayesian approach to provide a generalization bound for neural networks. (Sinha et al., 2017) study the adversarial generalization in terms of distributional robustness. The work of (Allen-Zhu and Li. 2020) explains adversarial generalization through the lens of feature purification. The work of (Javanmard et al. 2020) studied the generalization properties in the setting of linear regression. The work of (Xing et al., 2021; Xiao et al., 2022a) studied adversarial generalization using the notion of uniform algorithmic stability.

Adversarial Rademacher Complexity. The related work of ARC are discussed in the Introduction. In Appendix B, we provide the details of the above bounds.

4. Bounds of Adversarial Rademacher Complexity

4.1 Challenge of Bounding Adversarial Rademacher Complexity

4.1.1 Layer Peeling

To use the layer peeling techniques, Rademacher complexity should be in the structure of composition functions. However, the max operation in the definition of ARC destroys this structure. For more detail, see Appendix B.3.

Recent Attempts. Khim and Loh (2018) introduced a surrogate loss, tree-transformation loss, to get rid of the max operation. Then, the layer peeling bound can be applied to adversarial settings.

4.1.2 Covering Number

In this part, we discuss the difficulty of calculating the covering number of a robustified function class, and how we resolve it. For simplicity, We use the 1-dimensional function as an example.

We first state the problem for scalar functions. Consider \( f_w(x) : \mathbb{R} \rightarrow [-B, B] \) for scalars \(|w| \leq M\), and its robustified function \( g_w(x) \triangleq \min_{x' \in [x-\epsilon,x+\epsilon]} f_w(x') \). Here \( f_w(x) \) is \( L \)-Lipschitz w.r.t. \( w \) in \( x \in [-B, B] \), e.g., \( w, w^2, x^5, \sin(w) x^5 \). Note that \( g_w(x) \) may not be Lipschitz continuous w.r.t. \( w \). The problem is the following.

**Math Problem 1:** Bound the size of an \( \zeta \)-cover of the adversarial hypothesis function space \( \tilde{F} = \{ g(x) \triangleq \min_{x' \in [x-\epsilon,x+\epsilon]} f_w(\cdot) : |w| \leq M \} \). Here, the \( \zeta \)-cover is a set of functions whose distance to any function in \( F \) is no more than \( \zeta \).

**Review:** Calculating Covering Number of Function Space  To help readers better understand the problem, we consider a simpler problem of standard function class \( F = \{ f_w(\cdot) : \mathbb{R} \rightarrow [-B, B] : |w| \leq M \} \).

**Math Problem 2:** Bound the size of an \( \zeta \)-cover of the function space \( F = \{ f_w(\cdot) : \mathbb{R} \rightarrow [-B, B] : |w| \leq M \} \).

The idea is to build a relation between a cover of function class and a cover of the parameter region (a subset of Euclidean space). For this purpose, we only need to relate the distance in the parameter space to the distance in the function space. More specifically, suppose \(|w_1 - w_2| \leq \epsilon_w\), then \(|f_{w_1}(x) - f_{w_2}(x)| = |w_1 x - w_2 x| = |w_1 - w_2||x| \leq \epsilon_w B\). As a result, \(|w_1 - w_2| \leq \epsilon_w = \zeta/B \Rightarrow |f_{w_1}(x) - f_{w_1}(x)| \leq \zeta \). This implies an \( \epsilon \)-cover of \([-B, B]\) leads to an \( \zeta \)-cover of \( F \).
The proof sketch is the following.

**Step 1:** Relate the distance in the parameter space to the distance in the function space. More specifically, \(|w_1 - w_2| \leq \epsilon_w\), then \(|f_{w_1}(x) - f_{w_2}(x)| = |w_1 x - w_2 x| = |w_1 - w_2| |x| \leq \epsilon_w B\).

**Step 2:** Relate a cover of function class and a cover of the parameter region (a subset of Euclidean space). As a result, \(|w_1 - w_2| \leq \epsilon_w \Rightarrow f_{w_1}(x) - f_{w_2}(x)| \leq \zeta. This implies an \(\epsilon/B\)-cover of \([-M, M]\) (region in w-space) leads to an \(\zeta\)-cover of \(F\).

**Step 3:** Bound the size of the \(\epsilon_w\)-cover of \([-M, M]\). In fact, \(\{M, M - 2\epsilon_w, M - 4\epsilon_w, \ldots \}\) is one such cover, with size no more than \(2M/(2\epsilon_w) = M/\epsilon_w\).

**Conclusion:** The \(\zeta\)-cover of \(F\) is no more than \(M/\epsilon_w = MB/\zeta\).

**Discussing Problem 1 on Robustified Function Space.** Now we come back to Problem 1. We can adopt the same proof procedure for Problem 2: Step 2 and Step 3 would be the same, but Step 1 needs to significantly modified.

**Assumption 1:** If \(|w_1 - w_2| \leq \epsilon_w\), then \(|g_{w_1}(x) - g_{w_2}(x)| \leq \epsilon_w/\psi(B)\).

**Claim:** Under Assumption 1, the size of \(\zeta\)-cover of \(\tilde{F}\) is no more than \(M/\psi(B)/\epsilon_w\).

The remaining problem is to prove Assumption 1 holds, namely, solve the following math problem,

**Math Problem 3:** Find some function \(\psi(B) : \mathbb{R} \to \mathbb{R}\), such that if \(|w_1 - w_2| \leq \epsilon_w\), then \(|g_{w_1}(x) - g_{w_2}(x)| \leq \epsilon_w/\psi(B)\), where \(g_{w_1}(x) \triangleq \min_{x\in[x-\epsilon,x+\epsilon]} f_{w_1}(x)\).

In other words, we need to find \(\epsilon_w\) s.t. \(\inf_{\|x'-x\|\leq \epsilon} f_{w_1}(x') -\inf_{\|x-x\|\leq \epsilon} f_{w_2}(x') \leq \zeta\), for any \(|w_1 - w_2| \leq \epsilon_w, \|x_0\| \leq B\). Let \(x^*_1 = \inf_{\|x-x\|\leq \epsilon} f_{w_1}(x'), \hat{i} = 1, 2\), then we want to show

\[
|f_{w_1}(x^*_1) - f_{w_2}(x^*_2)| \leq \zeta.
\]

(5)

In the rest of our paper, we call Eq. (5) as robustified weight perturbation bound. Correspondingly, we call \(|f_{w_1}(x) - f_{w_2}(x)| \leq \zeta\) as weight perturbation bound in standard settings. It is not hard to obtain a weight perturbation bound in standard settings, see (Bartlett et al., 2017). However, it is unclear how to obtain an robustified weight perturbation bound in the literature. A small change in input will cause a large change in function value.

**Naive Method for Linear \(f_w\) Fails.** Assume \(f_w(x) = wx\). A naive method is to use triangle inequality to get \(|f_{w_1}(x^*_1) - f_{w_2}(x^*_2)| = |w_1 x^*_1 - w_2 x^*_2| \leq |(w_1 - w_2)x^*_1| + |w_2(x^*_1 - x^*_2)| \leq \epsilon_w B + 2M\epsilon \Rightarrow \zeta. Thus an \(\epsilon\)-cover of the function space can be built from an \((\zeta - 2M\epsilon)/B\)-cover in w-space. This is only possible when \(\zeta > M\epsilon\), thus we cannot build an \(\zeta\)-cover for arbitrarily small \(\zeta > 0\). As a result, as \(n \to \infty\), the corresponding adversarial Rademacher complexity bound would have a non-vanishing term, which is undesirable.

**Challenge.** The key issue seems to be controlling the extra term \(|w_2(x^*_1 - x^*_2)|\). For a linear function \(f_w\), this can be resolved by noting \(x^*_1 = x^*_2\) when \(|w_1| > \epsilon_w\). However, for general \(f_w\) (possibility non-convex), the relation of the worst-perturbed points \(x^*_1\) and \(x^*_2\) is unclear.

**Recent Attempts.** Yin et al. (2019) introduced a SDP-relaxation loss. Then, ARC \(\approx \text{RC defined in SDP-relaxation loss. They directly applied the standard weight perturbation bound provided in the work of (Bartlett et al., 2017). Gao and Wang (2021) considered FGSM loss. Then, the weight perturbation can be bounded in terms of \(|f_{w_1}(x) - f_{w_2}(x)|\) and \(|\nabla f_{w_1}(x) - \nabla f_{w_2}(x)|\). Then, they provide a bound for RC defined in FGSM loss. However, these two approaches change the definition of ARC.


Our approach. To solve this problem, we provide an robustified weight perturbation bound (Step 2, next subsection). Then, we provide a bound for the covering number of robustified function classes and provide a bound for adversarial Rademacher complexity.

4.2 Main Result on Adversarial Rademacher Complexity and Proof Sketch

Our main result states an upper bound of adversarial Rademacher complexity in Frobenius norm.

**Theorem 4** (Frobenius Norm Bound). Given the function class \( \mathcal{F} \) in equation (4) under Frobenius Norm, and the corresponding robustified function class \( \tilde{\mathcal{F}} \) in equation (3). The adversarial Rademacher complexity of deep neural networks \( \mathcal{R}_S(\mathcal{F}) \) satisfies

\[
\mathcal{R}_S(\tilde{\mathcal{F}}) \leq \frac{24}{\sqrt{n}} \max\{1, q^{\frac{1}{2}} \} \left( \|X\|_{p,\infty} \right) L_{\rho}^{l-1} \sqrt{\sum_{j=1}^{l} h_{j} h_{j-1} \log(3l) \prod_{j=1}^{l} M_{j}}. \tag{6}
\]

By assuming that \( L_{\rho} = 1, p \leq 2, \|X\|_{p,\infty} = B, \) and \( h = \max\{h_{0}, \ldots, h_{l}\} \), we have

\[
\mathcal{R}_S(\tilde{\mathcal{F}}) \leq O \left( \frac{(B + \epsilon) h \sqrt{\log(l)} \prod_{j=1}^{l} M_{j}}{\sqrt{n}} \right). \tag{7}
\]

Our proof is based on calculating the covering number of \( \tilde{\mathcal{F}} \). Below we sketch the proof. The completed proof is provided in Appendix A.

**Step 1: Diameter of \( \tilde{\mathcal{F}} \).** We first calculate the diameter of \( \tilde{\mathcal{F}} \). We have

\[
2 \max_{f \in \tilde{\mathcal{F}}} \|f\|_{S} \leq 2 L_{\rho}^{l-1} \max\{1, q^{\frac{1}{2}} \} \left( \|X\|_{p,\infty} + \epsilon \right) \prod_{j=1}^{l} M_{j} \overset{\Delta}{=} D.
\]

**Step 2: Distance to \( \tilde{\mathcal{F}}^c \).** Let \( C_{j} \) be \( \delta_j \)-covers (Definition 1) of \( \{|W_{j}|_{F} \leq M_{j}\}, j = 1, 2, \ldots, l \). Let \( \mathcal{F}^{c} = \{f^{c}: \mathbf{x} \rightarrow W_{j_{1}}^{c} \rho(W_{j_{1}-1}^{c} \rho(\cdots \rho(W_{j_{1}}^{c} \mathbf{x}) \cdots)), W_{j}^{c} \in C_{j}, j = 1, 2, \ldots, l \} \) and \( \tilde{\mathcal{F}}^{c} = \{\tilde{f}(\mathbf{x}, y) \rightarrow \inf_{\|\mathbf{x} - \mathbf{x}'\|_{p} \leq \epsilon} y f(\mathbf{x}') | f \in \mathcal{F}^{c}\} \).

For all \( \tilde{f} \in \tilde{\mathcal{F}} \), we need to find the smallest distance to \( \tilde{\mathcal{F}}^{c} \), i.e. we need to calculate the

\[
\max_{f \in \tilde{\mathcal{F}}} \min_{f^{c} \in \mathcal{F}^{c}} \|\tilde{f} - f^{c}\|_{S}.
\]

Let

\[
x_{i}^{c} = \arg \inf_{\|\mathbf{x}_{i} - \mathbf{x}'_{i}\|_{p}} y_{i} f(\mathbf{x}'_{i}), \quad x_{i}^{c} = \arg \inf_{\|\mathbf{x}_{i} - \mathbf{x}'_{i}\|_{p}} y_{i} f^{c}(\mathbf{x}'_{i}),
\]

and

\[
\bar{x}_{i} = \begin{cases} x_{i}^{c} & \text{if } f(\mathbf{x}_{i}) \geq f^{c}(\mathbf{x}_{i}) \\ x_{i}^{c} & \text{if } f(\mathbf{x}_{i}) < f^{c}(\mathbf{x}_{i}). \end{cases}
\]

Define \( g_{b}^{0}(\cdot) \) as

\[
g_{b}^{0}(\bar{x}) = W_{b} \rho(W_{b-1} \rho(\cdots W_{a} \rho(W_{a}^{c} \mathbf{x}) \cdots))).
\]
In words, for the layers $b \geq j > a$ in $g_b^0(\cdot)$, the weight is $W^c_j$, for the layers $a \geq j \geq 1$ in $g_b^0(\cdot)$, the weight is $W^c_j$. Then we have $f(\tilde{x}_i) = g^0_b(\tilde{x}_i)$, $f(\tilde{x}_i) = g^1_b(\tilde{x}_i)$. We can decompose

$$|f(\tilde{x}_i) - f^c(\tilde{x}_i)| \leq |g^0_b(\tilde{x}_i) - g^1_b(\tilde{x}_i)| + \cdots + |g^{l-1}_b(\tilde{x}_i) - g^l_b(\tilde{x}_i)| \leq \sum_{j=1}^{l} \frac{D\delta_j}{2M_j}$$  \hspace{1cm} (8)

**Step 3: Covering Number of $\tilde{F}$.** Then, we can calculate the $\epsilon$-covering number $\mathcal{N}(\tilde{F}, \parallel \cdot S, \epsilon)$. Because $\tilde{F}^c$ is a $\epsilon$-cover of $\tilde{F}$. The cardinality of $\tilde{F}^c$ is

$$\mathcal{N}(\tilde{F}, \parallel \cdot S, \epsilon) = |\tilde{F}^c| \leq \left( \frac{3lD}{2\epsilon} \right) \sum_{j=1}^{l} h_j h_{j-1}.$$  \hspace{1cm} (9)

**Step 4: Integration.** By Dudley’s integral, we obtain the bound in Theorem 4.

### 4.3 Further Discussions of the Bounds

**Theorem 5** ($\| \cdot \|_{1,\infty}$-Norm Bound). 
Given the function class $\mathcal{F}$ in equation (4) under $\| \cdot \|_{1,\infty}$-norm, and the corresponding robustified function class $\tilde{\mathcal{F}}$ in equation (3). The adversarial Rademacher complexity of deep neural networks $\mathcal{R}_S(\tilde{\mathcal{F}})$ satisfies

$$\mathcal{R}_S(\tilde{\mathcal{F}}) \leq \frac{24}{\sqrt{n}} (\|X\|_{p,\infty} + \epsilon) L_p^{-1} \left[ \sum_{j=1}^{l} h_j h_{j-1} \log(3l) \right] \prod_{j=1}^{l} M_j.$$  \hspace{1cm} (10)

The proof is similar to the proof of Theorem 4 and is deferred to Appendix A. In the case of $\| \cdot \|_{1,\infty}$-norm, the bound is similar to the bound in the Frobenius norm case except the term $\max\{1, q^{1/2 - 1/p}\}$. Therefore, for all $p \geq 1$, the $\| \cdot \|_{1,\infty}$-norm bound have the same order in Eq. (7). Similarly, we can obtain a spectral norm bound in Eq. (7) if $M_j$ is the bound of the spectral norm of $W_j$ and $h_j$ is the rank in each of the layers $j$.

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) RC.** The work of (Bartlett et al. 2017) used a covering numbers argument to show that the generalization gap is bounded by

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

As discussed in the work of (Neyshabur et al., 2017b) and (Golowich et al. 2018), this bound is no smaller than $\mathcal{O}(B \sqrt{lMh} \prod_{j=1}^{l} M_j / \sqrt{n})$. Our bound has an additional dependence on $\epsilon$, which is unavoidable in adversarial settings, and has a different dependence on the network size $O(\sqrt{l \log(l)}h)$. We have a lower dependence on depth-$l$ and a higher dependence on width-$h$.

**Bound for ARC in Two-layers Cases.** The work of (Awasthi et al., 2020) showed that the ARC is bounded by

$$O\left( \frac{(B + \epsilon) \sqrt{h_1 d \sqrt{\log n} M_1 M_2}}{\sqrt{n}} \right)$$  \hspace{1cm} (11)
in two-layers cases. If we apply our bound to two-layers cases, our bound becomes $O((B + \epsilon)\sqrt{\log M_1 M_2/\sqrt{n}})$, which is strictly better than the bound in Eq. (11). If other factors remain the same, our bound has a lower dependence on the sample size $n$.

**Theorem 6** (Lower Bound). *Given the function class $\mathcal{F}$ of DNNs in equation (4) with Relu activation functions, and the corresponding robustified function class $\tilde{\mathcal{F}}$ in equation (3). There exists a dataset $S$, s.t. the adversarial Rademacher complexity of deep neural networks $\mathcal{R}_S(\tilde{\mathcal{F}})$ satisfies*

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

The proof is provided in Appendix A. The gap between the upper bound and the lower bound is $O(h \sqrt{\log t})$. In the next section, we extend the adversarial Rademacher complexity to the Multi-class classification cases.

**5. Margin Bounds for Multi-Class Classification**

**5.1 Setting for Multi-Class Classification**

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

Define the margin operator $M(f(x), y) = [f(x)]_y - \max_{y' \neq y} [f(x)]_{y'}$. The function makes a correct prediction if and only if $M(f(x), y) > 0$. We consider a particular loss function $\ell(f(x), y) = \phi_{\gamma}(M(f(x), y))$, where $\gamma > 0$ and $\phi_{\gamma} : \mathbb{R} \rightarrow [0, 1]$ 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 $\phi'_{\gamma}(t)$ is $1/\gamma$-Lipschitz. The loss function $\ell(f(x), y)$ satisfies:

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

Define the function class $\ell_{\mathcal{F}} := \{(x, y) \mapsto \phi_{\gamma}(M(f(x), y)) : f \in \mathcal{F}\}$.

In adversarial training, let $\mathbb{B}_x^p(\epsilon) = \{x' : \|x' - x\|_p \leq \epsilon\}$ and we define the robustified function class $\tilde{\ell}_{\mathcal{F}} := \{(x, y) \mapsto \max_{x' \in \mathbb{B}_x^p(\epsilon)} \ell(f(x'), y) : f \in \mathcal{F}\}$. We have

**Corollary 7** (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 $f \in \mathcal{F},$

$$\mathbb{P}_{(x,y) \sim \mathcal{D}} \left\{ \exists x' \in \mathbb{B}_x^p(\epsilon) \text{ s.t. } y \neq \arg\max_{y' \in [K]} [f(x')]_{y'} \right\} \leq \frac{1}{n} \sum_{i=1}^{n} 1(\exists x'_i \in \mathbb{B}_x^p(\epsilon) \text{ s.t. } [f(x'_i)]_{y_i} \leq \gamma + \max_{y' \neq y} [f(x'_i)]_{y'}) + 2\mathcal{R}_S(\tilde{\ell}_{\mathcal{F}}) + \frac{3}{2n} \log \frac{2}{\delta}.\]
5.2 Adversarial Rademacher Complexity in Multi-Class Cases

Under the multi-class setting, we have the following bound for adversarial Rademacher complexity.

**Theorem 8.** Given the function class $\mathcal{F}$ in equation (4) under Frobenius Norm, and the corresponding robustified function class $\tilde{\mathcal{F}}$ in equation (3). The adversarial Rademacher complexity of deep neural networks $\mathcal{R}_S(\tilde{\ell}_{\tilde{F}})$ satisfies

$$\mathcal{R}_S(\tilde{\ell}_{\tilde{F}}) \leq \frac{48K}{\gamma \sqrt{n}} \max\{1, q^{\frac{1}{2} - \frac{1}{p}}\}(\|X\|_{p,\infty} + \epsilon)L_{\rho}^{l-1} \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l) \prod_{j=1}^{l} M_j}.$$  \hspace{1cm} (14)

The $\|\cdot\|_{1,\infty}$-norm bound is similar, except the term $\max\{1, q^{\frac{1}{2} - \frac{1}{p}}\}$. Below we sketch the proof.

The completed proof is provided in Appendix A.

Step 1: Let $\tilde{\mathcal{F}}^k = \{(\mathbf{x}, y) \rightarrow \inf_{\mathbf{x}' \in B} (|f(\mathbf{x}')|_y - |f(\mathbf{x})|_y), f \in \mathcal{F}\}$, then $\mathcal{R}_S(\tilde{\ell}_{\mathcal{F}}) \leq K\mathcal{R}_S(\tilde{\mathcal{F}}^k)$. Step 2: By the Lipschitz property of $\phi_y(\cdot)$, $\mathcal{R}_S(\tilde{\ell}_{\tilde{F}^k}) \leq \frac{1}{\gamma} \mathcal{R}_S(\mathcal{F}^k)$. Step 3: The calculation of $\mathcal{R}_S(\tilde{\mathcal{F}}^k)$ follows the binary case.

5.3 Comparison of Standard and Adversarial Rademacher Complexity

Now, we compare the difference between the bounds for (standard) Rademacher complexity and adversarial Rademacher complexity. We have shown that

$$\mathcal{R}_S(\ell_{\mathcal{F}}) \leq \mathcal{O}\left(\frac{B \sqrt{l} \prod_{j=1}^{l} M_j}{\gamma \sqrt{n}}\right) \quad \text{and} \quad \mathcal{R}_S(\tilde{\ell}_{\mathcal{F}}) \leq \mathcal{O}\left(\frac{(B + \epsilon)h \sqrt{l} \log l \prod_{j=1}^{l} M_j}{\gamma \sqrt{n}}\right),$$  \hspace{1cm} (15)

where we use the upper bound of $\mathcal{R}_S(\mathcal{F})$ in (Golowich et al., 2018).

**Algorithm-Independent Factors.** In the two bounds, the algorithm-independent factors include sample size $B$, perturbation intensity $\epsilon$, depth-$l$, and width-$h$. To simplify the notations, we let $C_{std} = B \sqrt{l}$ and $C_{adv} = (B + \epsilon)h \sqrt{l} \log l$ be the constants in standard and adversarial Rademacher complexity, respectively. We simply have $C_{adv} > C_{std}$.

**Algorithm-Dependent Factors.** In the two bounds, the margins $\gamma$ and the product of the matrix norms $\prod_{j=1}^{l} \|W_j\|$ depend on the training algorithms. To simplify the notations, we define $W_{std} := \prod_{j=1}^{l} \|W_j\|/\gamma$ if the training algorithm is standard training. Correspondingly, let $W_{adv} := \prod_{j=1}^{l} \|W_j\|/\gamma$ if the training algorithm is adversarial training. As shown in the next section and Appendix C, $W_{adv} > W_{std}$ holds in our experiments.

**Notation of generalization gaps.** In the next section, we use $\mathcal{E}(\cdot)$ and $\tilde{\mathcal{E}}(\cdot)$ to denote the standard and robust generalization gap. We use $f_{std}$ and $f_{adv}$ to denote the standard- and adversarially-trained model. Our goal is to analyze the correlation between these factors and generalization gap to study why the robust generalization gap of an adversarial training model is large, which is quite different from the standard generalization gap of a standard-trained model, i.e., we want to analyze why $\tilde{\mathcal{E}}(f_{adv}) > \mathcal{E}(f_{std})$. Based on the standard and adversarial Rademacher complexity bounds, it is suggested that

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

To analyze the individual effect of factors $C_{adv}$ and $W_{adv}$, we further introduce two kinds of generalization gaps, the robust generalization gap of a standard-trained model ($\tilde{\mathcal{E}}(f_{std})$) and the
standard generalization gap of an adversarially-trained model ($\mathcal{E}(f_{\text{adv}})$). The standard and adversarial Rademacher complexity suggest that
\[
\tilde{\mathcal{E}}(f_{\text{std}}) \propto C_{\text{adv}}W_{\text{std}} \quad \text{and} \quad \mathcal{E}(f_{\text{adv}}) \propto C_{\text{std}}W_{\text{adv}}.
\]

6. Experiment

As we discuss in the previous section, the product of weight norm $\prod_{j=1}^{l} \|W_j\|/\gamma$ are algorithm-dependent factors in the ARC bounds. We provide experiments comparing the difference between these terms in standard and adversarial settings. Since the bounds also hold for convolution neural networks, we consider the experiments of training VGG networks (Simonyan and Zisserman, 2014) on CIFAR-10 (Krizhevsky et al., 2009) and CIFAR-100. We trained 88 models to study the weight norm and generalization gap. Additional experiments are provided in Appendix C.1.

Training Settings. For both standard and adversarial training, we use the stochastic gradient descent (SGD) optimizer, along with a learning rate schedule, which is 0.1 over the first 100 epochs, down to 0.01 over the following 50 epochs, and finally be 0.001 in the last 50 epochs. Weight decay is set to be $5 \times 10^{-4}$ in most of the models, which is shown to be optimal for robust accuracy. Weight decay is also set to be other values for ablation studies. For adversarial settings, we adopt the $\ell_\infty$ PGD adversarial training (Madry et al., 2017). The perturbation intensity is set to be $8/255$. We set the number of steps as 20 and further increase it to 40 in the testing phase. For the stepsize in the inner maximization, we set it as $2/255$.

Calculation of Margins. We adopt the setting in (Neyshabur et al., 2017a). In standard training, we set the margin over training set to be 5th-percentile of the margins of the data points in $S$. i.e. $\text{Prc}_{5}\{f(x_i)[y_i] - \max_{y \neq y_i} f(x)[y] | (x_i, y_i) \in S\}$. In adversarial settings, we set the margin over training set to be 5th-percentile of the margins of the PGD-adversarial examples of $S$. The choice of 5th-percentile is because the training accuracy is 100% in all the experiments. We provide ablation studies about the percentile in the Appendix C.3.

Adversarially-trained Models Have Larger Weight Norms. In Figure 1 we show the experiments of standard and adversarial training VGG on CIFAR-10 and CIFAR-100\(^2\). The y-axis is in the logarithm scale. In these eight cases, the weight norms of adv-trained models are larger than that of std-trained models, i.e., $W_{\text{adv}} \geq W_{\text{std}}$. Ablation studies are provided in Appendix C. $W_{\text{adv}} \geq W_{\text{std}}$ can also be observed in these experiments.

Standard and Robust Generalization Gap. In Table 2, we show the standard and robust generalization gap of both standard-trained and adversarially-trained models. We use the results of

---

1. https://github.com/JiancongXiao/Adversarial-Rademacher-Complexity
2. Larger models have more parameters and yield larger weight norm. These experiments aim to show the difference between adv-trained and std-trained models.
training VGG-19 on CIFAR-10 to discuss the experiments. Firstly, we can see that $\tilde{E}(f_{std})$ is small (=10.45%). On the other hand, an adversarial-trained model has a larger standard generalization gap ($E(f_{adv})=26.34\%$). It is a widely observed phenomenon that adversarial training hurts standard generalization. One possible reason is that adversarial training overfits the adversarial examples and performs worse on the original examples. Secondly, the robust generalization gap of a standard-trained model is very small ($\tilde{E}(f_{std}) = 0$). It is because the standard-trained model can easily be attacked on both the training set and test set. Then, the robust training accuracy and test accuracy are closed to 0%. Therefore, the robust generalization gap is also 0. On the contrary, $\tilde{E}(f_{adv})=58.90\%$, i.e. the adversarial generalization is bad. This is also observed in the previous studies, and we aim to discuss this phenomenon.

$\tilde{E}(f_{std})=0\%$ is a degenerated case. In Table 2, we can see that the robust training error for a standard-trained model is equal to 100%. Since the model does not fit any adversarial examples in the training set, there is nothing to generalize to the adversarial examples in the test set. The generalization gap becomes meaningless. And the Rademacher complexity bound $\tilde{E}(f_{std}) \leq O(C_{adv}W_{std}/\sqrt{n})$ becomes a trivial bound. In the other three cases, the training errors are all $\approx 0\%$. The generalization gaps are meaningful. We aim to analyze $\tilde{E}(f_{adv}) > \tilde{E}(f_{std})$ by analyzing $\tilde{E}(f_{adv}) > E(f_{adv}) > \tilde{E}(f_{std})$.

**The effect of $C_{adv}$.** We first compare the difference between $\tilde{E}(f_{adv})$ and $E(f_{adv})$. We can see that $\tilde{E}(f_{adv}) = 58.90\% > E(f_{adv}) = 26.34\%$. For an adversarially-trained model, the robust generalization gap is larger than the standard generalization gap. If we use the bounds of adversarial and standard Rademacher complexity as approximations of the robust and standard generalization gap, i.e., $\tilde{E}(f_{adv}) \propto C_{adv}W_{adv}$ and $E(f_{adv}) \propto C_{std}W_{adv}$, $C_{adv}$ is positivley related to the robust generalization gap.

**The effect of $W_{adv}$**. Similarly, we compare the difference between $E(f_{adv})$ and $\tilde{E}(f_{std})$. We can see that $\tilde{E}(f_{adv}) = 26.34\% > E(f_{std}) = 10.45\%$. This is a widely observed phenomenon that adversarial training hurts standard generalization. It can also be explained by the Rademacher bounds. If we use the bounds of standard Rademacher complexity as approximations, i.e., $E(f_{adv}) \propto C_{std}W_{adv}$ and $\tilde{E}(f_{std}) \propto C_{std}W_{std}$, $W_{adv}$ is positive related to the robust generalization gap.

In summary, we can use a simple formula, $C_{adv}W_{adv} > C_{std}W_{adv} > C_{std}W_{std}$, to describe the effect of these factors on generalization gap $\tilde{E}(f_{adv}) > E(f_{adv}) > \tilde{E}(f_{std})$. The difficulty of adversarial generalization might come from two parts, the constant $C_{adv}$ and the weight norms $W_{adv}$. The first part $C_{adv}$ is independent of the algorithms. It comes from the minimax problem of adversarial training itself, and it cannot be avoided. The second part $W_{adv}$ depends on the algorithms.

**Weight Decay.** The above analysis suggests controlling the weight norms to improve adversarial generalization. In Appendix C.4, we provide an experiment to slightly increase the weight decay.
Larger weight decay yields smaller weight norms and smaller generalization. But it also hurts training errors. It seems to be a trade-off between training error and generalization error. When the weight decay is increased to $10^{-2}$, the training fails. In this case, the weight norm of the adv-trained model is still larger than that of the std-trained model.

**Discussion on Representation Power of DNNs.** In our opinion, the reason why adversarial training yields a larger weight norm might be due to the representation power of DNNs. DNNs with small weight norms cannot fit adversarial examples on the training set. Then, the algorithms will find DNNs with large weight norms. However, DNNs with large weight norms cannot generalize well, as it is suggested by the above analysis. Therefore, robust overfitting appears. To study this hypothesis requires large-scale experiments, beyond the scope of our work.

7. Conclusion
In this paper, we provide a method and give the first bounds of adversarial Rademacher complexity of deep neural networks. Based on the analysis, the difficulty of adversarial generalization may come from two parts. One is algorithm-independent, the other one is algorithm-dependent and related to the weight norm of the neural network. We empirically study the correlation between these factors and adversarial generalization. We think our results will motivate more theoretical research to understand adversarial training and empirical research to improve the generalization of adversarial training.
References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

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

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

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

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

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

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

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

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

Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

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

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. 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 Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017a.
Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017b.

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

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

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

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

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

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

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

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

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

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

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

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

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

Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding adversarial robustness against on-manifold adversarial examples. *arXiv preprint arXiv:2210.00430*, 2022c.
Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. In Thirty-Fifth Conference on Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=xz80iPFIjuG

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

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

A.1 Proof of Theorem 4

Before we provide the proof, we first introduce the following definition and lemma.

**Covering Number.** Our method for analyzing adversarial Rademacher complexity is based on the covering number. We first provide the definition of covering number.

**Definition 1** ($\varepsilon$-cover). Let $\varepsilon > 0$ and $(V,d(\cdot, \cdot))$ be a metric space, where $d(\cdot, \cdot)$ is a (pseudo)-metric. We say $C \subset V$ is an $\varepsilon$-cover of $V$, if for any $v \in V$, there exists $v' \in C$ s.t. $d(v,v') \leq \varepsilon$. Define the smallest $|C|$ as the $\varepsilon$-covering number of $V$ and denote as $\mathcal{N}(V,d(\cdot, \cdot),\varepsilon)$.

Next, we define the $\varepsilon$-covering number of a function class $\mathcal{F}$. Given the sample dataset $S = \{ (x_1, y_1), \cdots, (x_n, y_n) \}$ with $x_i \in \mathbb{R}^t$, let $\|f\|_S^2 = \frac{1}{n} \sum_{i=1}^n f(x_i, y_i)^2$ be a pseudometric of $\mathcal{F}$.

Define the $\varepsilon$-covering number of $\mathcal{F}$ to be $\mathcal{N}(\mathcal{F}, \| \cdot \|_S, \varepsilon)$. Let $D$ be the diameter of $\mathcal{F}$ with $D \triangleq 2 \max_{f \in \mathcal{F}} \|f\|_S$.

**Proposition 2** (Dudley’s integral). The Rademacher complexity $\mathcal{R}_S(\mathcal{F})$ satisfies

$$
\mathcal{R}_S(\mathcal{F}) \leq \inf_{\delta > 0} \left[ 8\delta + \frac{12}{\sqrt{n}} \int_0^D \sqrt{\log \mathcal{N}(\mathcal{F}, \| \cdot \|_S, \varepsilon)} \, d\varepsilon \right].
$$

The proof of this proposition can be found in statistic textbooks (e.g. (Wainwright, 2019)). Based on this, we can use the bound of the covering number of the function class $\mathcal{F}$ to give an upper bound of the Rademacher complexity.

**Lemma 3** (Covering number of norm-balls). Let $B$ be a $\ell_p$ norm ball with radius $W$. Let $d(x_1, x_2) = \|x_1 - x_2\|_p$. Define the $\varepsilon$-covering number of $B$ as $\mathcal{N}(B, d(\cdot, \cdot), \varepsilon)$, we have

$$
\mathcal{N}(B, \| \cdot \|_p, \varepsilon) \leq (1 + 2W/\varepsilon)^d.
$$

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

$$
\mathcal{N}(B, \| \cdot \|_{F}, \varepsilon) \leq (1 + 2W/\varepsilon)^{m \times k} \leq (3W/\varepsilon)^{m \times k}.
$$

**Lemma 4.** If $x_i^* \in \{ x_i' \| \| x_i - x_i' \|_p \leq \varepsilon \}$, we have

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

Proof: If $p \geq r^*$, by Holder’s inequality with $1/r^* = 1/p + 1/s$,

$$
\|x_i^*\|_r \leq \sup \|1\|_s \|x_i^*\|_p = \|1\|_s \|x_i^*\|_p = d^\frac{1}{2} \|x_i^*\|_p = d^{1 - \frac{1}{r^*}} \|x_i^*\|_p.
$$

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

$$
\|x_i^*\|_r \leq \|x_i^*\|_p.
$$

Equality holds when one of the entries of $\theta$ equals to one and the others equal to zero. Then

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

□
Lemma 5. Let $A$ be an $m \times k$ matrix and $b$ be a $n$-dimension vector, we have

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

Proof: let $A_i$ be the rows of $A$, $i = 1 \cdots m$, we have

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

$\square$

Step 1: Diameter of $\hat{F}$. We first calculate the diameter of $\hat{F}$. $\forall f \in F$, given $(x_i, y_i)$, let $x_i^* \in \arg \inf_{\|x_i - x_i^*\|_p \leq \epsilon_p} yf(x_i^*)$, and we let $x_i^*$ be the output of $x_i^*$ pass through the first to the $l - 1$ layer, we have

$$|\hat{f}(x_i, y_i)| = \|x_i - x_i^*\|_p \leq \epsilon_p yf(x_i^*)
= |W_1 \rho(W_{l-1} x_i^{l-1})|
\leq (i) \|W_1\|_F \cdot \|\rho(W_{l-1} x_i^{l-1})\|_2
= \|W_1\|_F \cdot \|\rho(W_{l-1} x_i^{l-1}) - \rho(0)\|_2
\leq (ii) L_\rho M_j \|W_{l-1}(x_i^{l-1})\|_2
\leq \cdots
\leq L_\rho^{l-1} \prod_{j=2}^{l} M_j \|W_1 x_i^*\|_2
\leq L_\rho^{l-1} \prod_{j=1}^{l} M_j \cdot \|x_i^*\|_2
\leq (iii) L_\rho^{l-1} \prod_{j=1}^{l} M_j \max\{1, d^{\frac{1}{2} - \frac{1}{p}}\} (\|X\|_{p,\infty} + \epsilon),$$

where inequality (i) is because of Lemma 5, inequality (ii) is because of the Lipschitz property of activation function $\rho(\cdot)$, inequality (iii) is because of Lemma 4. Therefore, we have

$$2 \max_{f \in \mathcal{F}} \|\hat{f}\|_S = 2 \left(\frac{1}{n} \sum_{i=1}^{n} |\hat{f}(x_i, y_i)|^2\right)^{\frac{1}{2}} \leq 2 L_\rho^{l-1} \max\{1, d^{\frac{1}{2} - \frac{1}{p}}\} (\|X\|_{p,\infty} + \epsilon) \prod_{j=1}^{l} M_j \overset{\Delta}{=} D.$$ 

Step 2: Distance to $\tilde{\mathcal{F}}^c$. Let $C_j$ be $\delta_j$-covers of $\{\|W_j\|_F \leq M_j\}$, $j = 1, 2, \cdots, l$. Let

$$\mathcal{F}^c = \{f^c : x \rightarrow W_1^c \rho(W_{l-1} \rho(\cdots \rho(W_1^c x) \cdots)) \mid W_j^c \in C_j, j = 1, 2 \cdots, l\}$$

and $\tilde{\mathcal{F}}^c = \{\hat{f} : (x, y) \rightarrow \inf_{\|x - x\|_p \leq \epsilon} yf(x') \mid f \in \mathcal{F}^c\}.$
For all $\tilde{f} \in \mathcal{F}$, we need to find the smallest distance to $\mathcal{F}^c$, i.e. we need to calculate the

$$
\max_{f \in \mathcal{F}} \min_{f' \in \mathcal{F}^c} \| \tilde{f} - f' \|_S.
$$

$\forall (x_i, y_i), i = 1, \cdots, n$, given $\tilde{f}$ and $\tilde{f}'$ with $\|W_j - W'_j\|_F \leq \delta_j, j = 1, \cdots, l$, consider

$$
|\tilde{f}(x_i, y_i) - \tilde{f}'(x_i, y_i)|
= |\inf_{\|x_i - x'_i\|_p} y_i f(x'_i) - \inf_{\|x_i - x'_i\|_p} y_i f'(x'_i)|
$$

Let

$$
x'_i = \arg \inf_{\|x_i - x'_i\|_p} y_i f(x'_i), \quad \text{and} \quad x_i^c = \arg \inf_{\|x_i - x'_i\|_p} y_i f'(x'_i),
$$

we have

$$
|\tilde{f}(x_i, y_i) - \tilde{f}'(x_i, y_i)|
= |y_i f(x'_i) - y_i f'(x'_i)|
= |f(x'_i) - f'(x'_i)|.
$$

Let

$$
\bar{x}_i = \begin{cases} 
  x'_i & \text{if } f(x'_i) \geq f'(x'_i) \\
  x_i^c & \text{if } f(x'_i) < f'(x'_i)
\end{cases}
$$

Then,

$$
|\tilde{f}(x_i, y_i) - \tilde{f}'(x_i, y_i)|
\leq |f(\bar{x}_i) - f'(\bar{x}_i)|
$$

Define $g^c_0(\cdot)$ as

$$
g^c_0(\bar{x}) = W_0 \rho(W_{b-1} \cdots W_{a+1} \rho(W_a^c \cdots \rho(W_1^c \bar{x}) \cdots))).
$$

In words, for the layers $b \geq j > a$ in $g^c_0(\cdot)$, the weight is $W_j$, for the layers $a \geq j \geq 1$ in $g^c_0(\cdot)$, the weight is $W_j^c$. Then we have $f(\bar{x}_i) = g^c_0(\bar{x}_i)$, $f(\bar{x}_i) = g^c_0(\bar{x}_i)$. We can decompose

$$
|f(\bar{x}_i) - f^c(\bar{x}_i)|
= |g^c_0(\bar{x}_i) - g^c_0(\bar{x}_i)|
= |g^c_0(\bar{x}_i) - g^c_1(\bar{x}_i) + \cdots + g^c_{l-1}(\bar{x}_i) - g^c_l(\bar{x}_i)|
\leq |g^c_0(\bar{x}_i) - g^c_1(\bar{x}_i)| + \cdots + |g^c_{l-1}(\bar{x}_i) - g^c_l(\bar{x}_i)|.
$$

(16)
To bound the gap $|f(\bar{x}_i) - f^c(\bar{x}_i)|$, we first calculate $|g^{j-1}_i(\bar{x}_i) - g^j_i(\bar{x}_i)|$ for $j = 1, \ldots, l$.

\begin{align*}
|g^{j-1}_i(\bar{x}_i) - g^j_i(\bar{x}_i)| &= |W_i(\rho(g^{j-1}_{i-1}(\bar{x}_i)) - W_i(\rho(g^j_{i-1}(\bar{x}_i)))| \\
&\leq \|W_i\|_F \|\rho(g^{j-1}_{i-1}(\bar{x}_i)) - \rho(g^j_{i-1}(\bar{x}_i))\|_2 \\
&\leq L_\rho M_i \|g^{j-1}_{i-1}(\bar{x}_i) - g^j_{i-1}(\bar{x}_i)\|_2 \\
&\leq L_\rho M_i \|W_{i-1}(\rho(g^{j-1}_{i-2}(\bar{x}_i)) - W_{i-1}(\rho(g^j_{i-2}(\bar{x}_i)))\|_2 \\
&\leq \cdots \\
&\leq L_\rho^{l-j} \prod_{k=j+1}^{L} M_k \|W_j(\rho(g^{j-1}_{j-1}(\bar{x}_i)) - W_j(\rho(g^j_{j-1}(\bar{x}_i)))\|_2,
\end{align*}

where (i) is due to Lemma 5, (ii) is due to the bound of $\|W_j\|$ and the Lipschitz of $\rho(\cdot)$, (iii) is because of the definition of $g^0_i(\bar{x})$. Then

\begin{align*}
|g^{j-1}_i(\bar{x}_i) - g^j_i(\bar{x}_i)| &\leq L_\rho^{l-j} \prod_{k=j+1}^{L} M_k \|W_j(\rho(g^{j-1}_{j-1}(\bar{x}_i)) - W_j(\rho(g^j_{j-1}(\bar{x}_i)))\|_2 \\
&= L_\rho^{l-j} \prod_{k=j+1}^{L} M_k \|W_j - W_j^c\|_F \|\rho(g^{j-1}_{j-1}(\bar{x}_i))\|_2 \\
&\leq L_\rho^{l-j} \prod_{k=j+1}^{L} M_k \|W_j - W_j^c\|_F \|\rho(g^{j-1}_{j-1}(\bar{x}_i))\|_2 \\
&\leq L_\rho^{l-j} \prod_{k=j+1}^{L} M_k \|\rho(g^{j-1}_{j-1}(\bar{x}_i))\|_2,
\end{align*}

(17)

where inequality (i) is due to Lemma 5, inequality (ii) is due to Lemma 5 the assumption that $\|W_j - W_j^c\|_F \leq \delta_j$. It is lefted to bound $\|\rho(g^{j-1}_{j-1}(\bar{x}_i))\|_2$, we have

\begin{align*}
\|\rho(g^{j-1}_{j-1}(\bar{x}_i))\|_2 &= \rho(g^{j-1}_{j-1}(\bar{x}_i)) - \rho(0)\|_2 \\
&\leq L_\rho \|g^{j-1}_{j-1}(\bar{x}_i))\|_2 \\
&= L_\rho \|W_j^c - \rho(g^{j-2}_{j-2}(\bar{x}_i))\|_2 \\
&\leq L_\rho \|W_j^c - \rho(g^{j-2}_{j-2}(\bar{x}_i))\|_2 \\
&\leq L_\rho M_j \|\rho(g^{j-2}_{j-2}(\bar{x}_i))\|_2 \\
&\leq \cdots \\
&\leq L_\rho^{j-1} \prod_{k=1}^{j-1} M_k \max\{1, d^{\frac{1}{2}-\frac{1}{p}}\}(\|X\|_{p,\infty} + \epsilon).
\end{align*}

(18)
combining Eq. (17) and (18), we have
\[ |g_j^{l-1}(\bar{x}_i) - g_j^l(\bar{x}_i)| \leq \frac{L_{j-1}}{M_j} \prod_{k=1}^l M_k \delta_j \max\{1, d^{\frac{i-1}{p}}\} (\|X\|_{p,\infty} + \epsilon) \]
\[ = \frac{D\delta_j}{2M_j}. \quad (19) \]

Therefore, combining Eq. (16) and (19), we have
\[ |f(\bar{x}_i) - f^c(\bar{x}_i)| \leq |g_0^l(\bar{x}_i) - g_1^l(\bar{x}_i)| + \cdots + |g_l^{l-1}(\bar{x}_i) - g_l^l(\bar{x}_i)| \]
\[ \leq \sum_{j=1}^l \frac{D\delta_j}{2M_j}. \quad (20) \]

Then
\[ \max_{\tilde{f} \in \tilde{F}} \min_{\tilde{f}^c \in \tilde{F}^c} \|\tilde{f} - \tilde{f}^c\|_S \leq \sum_{j=1}^l \frac{D\delta_j}{2M_j}. \]

Let \( \delta_j = \frac{2M_j \epsilon}{dD} \), \( j = 1, \cdots, l \), we have
\[ \max_{\tilde{f} \in \tilde{F}} \min_{\tilde{f}^c \in \tilde{F}^c} \|\tilde{f} - \tilde{f}^c\|_S \leq \sum_{j=1}^l \frac{D\delta_j}{2M_j} \leq \epsilon. \]

**Step 3: Covering Number of \( \tilde{F} \).** We then calculate the \( \epsilon \)-covering number \( \mathcal{N}(\tilde{F}, \|\cdot\|_S, \epsilon) \). Because \( \tilde{F}^c \) is a \( \epsilon \)-cover of \( \tilde{F} \). The cardinality of \( \tilde{F}^c \) is
\[ \mathcal{N}(\tilde{F}, \|\cdot\|_S, \epsilon) = |\tilde{F}^c| = \prod_{j=1}^l |C_j| \leq \prod_{j=1}^l \left( \frac{3M_j}{\delta_j} \right)^{h_j h_{l-1}} = \left( \frac{3l D}{2\epsilon} \right)^{\sum_{j=1}^l h_j h_{l-1}}, \]

where inequality (i) is due to Lemma 3.

**Step 4, Integration.** By Dudley’s integral, we have
\[
\mathcal{R}_S(\tilde{F}) \\
\leq \inf_{\delta \geq 0} \left[ 8\delta + \frac{12}{\sqrt{n}} \int_{\delta}^{D/2} \sqrt{\log N(\mathcal{F}, \| \cdot \|_S, \varepsilon)} d\varepsilon \right] \\
\leq \inf_{\delta \geq 0} \left[ 8\delta + \frac{12}{\sqrt{n}} \int_{\delta}^{D/2} \left( \sum_{j=1}^{l} h_j h_{j-1} \right) \log(3D/2\varepsilon) d\varepsilon \right] \\
= \inf_{\delta \geq 0} \left[ 8\delta + \frac{12D}{\sqrt{n}} \frac{\sum_{j=1}^{l} h_j h_{j-1}}{\sqrt{n}} \int_{\delta/D}^{\sqrt{\log(3D/2\varepsilon)}} d\varepsilon \right]. \quad (21)
\]

Let \( \delta \to 0 \). Integration by part, we have

\[
\int_0^{1/2} \sqrt{\log(3l/2\varepsilon)} d\varepsilon \\
= \frac{1}{2} \left( \frac{3l}{2} \sqrt{\pi \text{erfc}(\sqrt{\log 3l})} + \sqrt{\log 3l} \right) \\
\leq \frac{1}{2} \left( \frac{3l}{2} \sqrt{\pi \exp(-\sqrt{\log 3l^2})} + \sqrt{\log 3l} \right) \\
= \frac{1}{2} \left( \sqrt{\frac{\pi}{2}} + \sqrt{\log 3l} \right) \\
\leq \frac{1}{2} \left( 2\sqrt{\log 3l} \right) \\
= \sqrt{\log 3l}. \quad (22)
\]

Plugging Eq. (22) to Eq. (21), we have

\[
\mathcal{R}_S(\tilde{F}) \leq \frac{24}{\sqrt{n}} \max\{1, d^{2-\frac{1}{p}}\} \left( \|X\|_{p,\infty} + \varepsilon \right) L^{l-1}_\rho \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l)} \prod_{j=1}^{l} M_j.
\]

**A.2 Proof of Theorem 5**

The proof is similar to the proof of the Frobenius norm bound with two difference. The first one is inequality in Lemma 5 is replaced by Lemma 6 below. The second one is that the \( C_j \) covering sets are for each neurons in this case. While \( C_j \) cover each matrices in the previous case. Below we provide the complete proof. We first introduce the following inequality.

**Lemma 6.** Let \( A \) be a \( m \times k \) matrix and \( b \) be a \( n \)-dimension vector, we have

\[
\|A \cdot b\|_{\infty} \leq \|A\|_{1,\infty} \|b\|_{\infty}.
\]

**Proof:** let \( A_i \) be the rows of \( A \), \( j = 1 \cdots m \), we have

\[
\|A \cdot b\|_{\infty} = \max \|A_i b\| \leq \max \|A_i\|_1 \|b\|_{\infty} = \|A\|_{1,\infty} \|b\|_{\infty}.
\]

3. Simply let \( \delta = D/\sqrt{n} \), we can obtain a bound in \( \mathcal{O}(\sqrt{\log n/n}) \). To get rid of the \( \log n \) term, we can let \( \delta \to 0 \).
Step 1: Diameter of $\tilde{F}$. We first calculate the diameter of $\tilde{F}$. \(\forall f \in \mathcal{F},\) given \((x_i, y_i)\), let \(x_i^* \in \arg \inf_{\|x_i - x_i'\|_p \leq \epsilon_p} yf(x_i')\), and we let \(x_i^l\) be the output of \(x_i^*\) pass through the first to the \(l-1\) layer, we have

\[
|\tilde{f}(x_i, y_i)| = \left| \inf_{\|x_i - x_i'\|_p \leq \epsilon_p} yf(x_i') \right| = |W_{l}\rho(W_{l-1}x_i^{l-1})| \\
\leq \|W_{l}\|_{1,\infty} \cdot \|\rho(W_{l-1}x_i^{l-1})\|_{\infty} = \|W_{l}\|_{1,\infty} \cdot \|\rho(W_{l-1}x_i^{l-1}) - \rho(0)\|_{\infty} \\
\leq L_{\rho}M_{l}\|W_{l-1}(x_i^{l-1})\|_{\infty} \leq \cdots \leq L_{\rho}^{l-1}\prod_{j=1}^{l} M_{j} \cdot \|x_i^*\|_{\infty} \\
\overset{(iii)}{\leq} L_{\rho}^{l-1}\prod_{j=1}^{l} M_{j}(\|X\|_{p,\infty} + \epsilon),
\]

where inequality (i) is because of Lemma 6, inequality (ii) is because of the Lipschitz property of activation function \(\rho(\cdot)\), inequality (iii) is because of Lemma 4. Therefore, we have

\[
2\max_{f \in \mathcal{F}} \|\tilde{f}\|_{S} = 2\left(\frac{1}{n} \sum_{i=1}^{n} |\tilde{f}(x_i, y_i)|^2\right)^{\frac{1}{2}} \leq 2L_{\rho}^{l-1}(\|X\|_{p,\infty} + \epsilon)\prod_{j=1}^{l} M_{j} \overset{\Delta}{=} D.
\]

Step 2: Distance to $\tilde{F}^c$. Let $C_{j}^{m}$ be \(\delta_j\)-covers of $\{|W_{j}^{m}|_{1} \leq M_{j}\}, j = 1, 2, \ldots, l, m = 1, \ldots, h_{j}$, where $W_{j}^{m}$ is the \(m\)th row of $W_{j}^{m}$. Let

\[
\mathcal{F}^c = \{f^c : x \rightarrow W_{l}^{c}\rho(W_{l-1}^{c}\rho(\cdots \rho(W_{1}^{c}x) \cdots))\}, W_{j}^{cm} \in C_{j}^{m}, m = 1, \ldots, h_{j}, j = 1, 2 \ldots, l\}
\]

and $\tilde{F}^c = \{\tilde{f} : (x, y) \rightarrow \inf_{\|x - x'\|_p \leq \epsilon} yf(x') | f \in \mathcal{F}^c\}.$

For all $\tilde{f} \in \tilde{F}$, we need to find the smallest distance to $\tilde{F}^c$, i.e. we need to calculate the

\[
\max_{f \in \mathcal{F}} \min_{\tilde{f} \in \tilde{F}^c} \|\tilde{f} - \tilde{f}^c\|_{S}.
\]

\(\forall (x_i, y_i), j = 1, \ldots, n,\) given $\tilde{f}$ and $\tilde{f}^c$ with $|W_{j}^{m} - W_{j}^{cm}| \leq \delta_j, m = 1, \ldots, h_{j}, j = 1, \ldots, l$, we have $\|W_{j} - W_{j}^c\|_{1,\infty} \leq \delta_j$. By the same argument as the step 2 of the proof of Theorem 4, we have
\[
\max_{f \in \mathcal{F}} \min_{\tilde{f} \in \mathcal{F}} \|\tilde{f} - f\|_S \leq \sum_{j=1}^{l} \frac{D\delta_j}{2M_j}.
\]

Let \(\delta_j = 2M_j\varepsilon/dD\), \(j = 1, \cdots, l\), we have
\[
\max_{f \in \mathcal{F}} \min_{\tilde{f} \in \mathcal{F}} \|\tilde{f} - f\|_S \leq \sum_{j=1}^{l} \frac{D\delta_j}{2M_j} \leq \varepsilon.
\]

**Step 3: Covering Number of \(\tilde{\mathcal{F}}\).** We then calculate the \(\varepsilon\)-covering number \(\mathcal{N}(\tilde{\mathcal{F}}, \| \cdot \|_S, \varepsilon)\). Because \(\tilde{\mathcal{F}}_c\) is a \(\varepsilon\)-cover of \(\tilde{\mathcal{F}}\). The cardinality of \(\tilde{\mathcal{F}}_c\) is
\[
\mathcal{N}(\tilde{\mathcal{F}}, \| \cdot \|_S, \varepsilon) = |\tilde{\mathcal{F}}_c| = l \prod_{j=1}^{l} h_j \left( \frac{h_j}{3} \right)^{h_j-1} \leq \left( \frac{3lD}{2\varepsilon} \right)^{\sum_{j=1}^{l} h_j} \left( \frac{3lD}{2\varepsilon} \right)^{h_j-1},
\]
where inequality (i) is due to Lemma 3.

**Step 4, Integration.** By the same argument as the step 4 of the proof of Theorem 4, integration by part, we have
\[
\mathcal{R}_S(\tilde{\mathcal{F}}) \leq \frac{24}{\sqrt{n}} \left( \|\mathbf{X}\|_{p,\infty} + \varepsilon \right) L_{\rho}^{l-1} \sqrt{\sum_{j=1}^{l} h_j h_{j-1} \log(3l) \prod_{j=1}^{l} M_j}.
\]

**A.3 Proof of Theorem 6**

The proof of the above Theorem is based on constructing a linear network. By the definition of Rademacher complexity, if \(\mathcal{H}'\) is a subset of \(\mathcal{H}\), we have
\[
\mathcal{R}_S(\mathcal{H}') = \mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{h \in \mathcal{H}'} \sum_{i=1}^{n} \sigma(h(x_i, y_i)) \right] \leq \mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{h \in \mathcal{H}} \sum_{i=1}^{n} \sigma(h(x_i, y_i)) \right] = \mathcal{R}_S(\mathcal{H}).
\]

Therefore, it is enough to lower bound the complexity of \(\tilde{\mathcal{F}}'\) in a particular distribution \(\mathcal{D}\), where \(\tilde{\mathcal{F}}'\) is a subset of \(\tilde{\mathcal{F}}\). Let
\[
\tilde{\mathcal{F}}' = \{ \mathbf{x} \rightarrow \inf_{\|\mathbf{x}' - \mathbf{x}\|_p \leq \varepsilon} y M_1 \cdot M_2 w^T \mathbf{x} | w \in \mathbb{R}^q, \|w\|_2 \leq M_1 \}.
\]

We first prove that \(\tilde{\mathcal{F}}'\) is a subset of \(\tilde{\mathcal{F}}\). In \(\tilde{\mathcal{F}}\), we let the activation function \(\rho(\cdot)\) be a identity mapping. Let
\[
W_1 = \begin{bmatrix} w \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{h_1 \times h_0}, \quad W_j = \begin{bmatrix} M_j & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{h_j \times h_{j-1}}, \quad j = 2, \cdots, l.
\]
Then, we have \( \|W_j\| \leq M_j \) and \( \tilde{F} \) with additional constraint in Eq. (23) reduce to \( \tilde{F}' \). In other words, \( \tilde{F}' \) is a subset of \( \tilde{F} \).

It turns out that we need to lower bound the adversarial Rademacher complexity of linear hypothesis. The results are given by the work of (Yin et al., 2019; Awasthi et al., 2020). Below we state the result.

**Proposition 4.** Given the function class

\[ G = \{ x \rightarrow y w^T x | w \in \mathbb{R}^q, \|w\|_r \leq W \} \]

and

\[ \tilde{G} = \{ x \rightarrow \inf_{\|x' - x\|_r \leq \epsilon} y w^T x | w \in \mathbb{R}^q, \|w\|_r \leq W \}, \]

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

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

Since the standard Rademacher complexity

\[ \mathcal{R}_S(G) = \frac{W}{m} \mathbb{E}_\sigma \left\| \sum_{i=1}^{n} \sigma_i x_i \right\|_{rs}, \]

let \( \|x_i\| = B \) with equal entries for \( i = 1, \cdots, n \), by Lemma 4, we have

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

By Khintchine’s inequality, we know that there exists a universal constant \( c > 0 \) such that

\[ \mathbb{E}_\sigma \left| \sum_{i=1}^{n} \sigma_i \right| \geq c \sqrt{n}. \]

Then, we have

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

Therefore,

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

\[ \geq \frac{1}{1 + 2c} \mathcal{R}_S(G) + \frac{2c}{1 + 2c} \times \frac{\epsilon \max \{1, d^{1 - \frac{1}{r} - \frac{1}{p}}\} W}{2\sqrt{n}} \]

\[ \geq \frac{c}{1 + 2c} \left( (B + \epsilon) \max \{1, d^{1 - \frac{1}{r} - \frac{1}{p}}\} W \right). \]

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

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

where \( r = 2 \) for frobenius norm bound and \( r = 1 \) for \( \| \cdot \|_{1,\infty} \)-norm bound.
A.4 Proof of Theorem 8

Proof: Firstly, we have

\[
\tilde{\ell}(f(x), y) = \max_{\|x - x'\| \leq \epsilon} \phi \gamma(M(f(x'), y)) = \phi \gamma(M(f(x'), y)) \\
= \phi \gamma(\inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x))) = \phi \gamma(\inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x))).
\]

Therefore, we have

\[
\max_{y' \neq y} \phi \gamma(\inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x))) = \max_k \phi \gamma(h_k(x, y)).
\]

If

\[
\inf_{y' \neq y} \inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x)) \leq \gamma,
\]

we have

\[
\inf_{y' \neq y} \inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x)) = \inf_k h_k(x, y).
\]

If

\[
\inf_{y' \neq y} \inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x)) > \gamma,
\]

we have

\[
\phi \gamma(\inf_{y' \neq y} \inf_{\|x - x'\| \leq \epsilon} (f(x)' - f(x))) = \phi \gamma(\inf_k h_k(x, y)) = 0.
\]

Therefore, we have

\[
\tilde{\ell}(f(x), y) = \phi \gamma(\inf_k h_k(x, y)) = \max_k \phi \gamma(h_k(x, y)).
\]

Define \(H^k = \{h^k(x, y) = \inf_{\|x - x'\| \leq \epsilon} ([f(x)]_y - f(x')]_y + \gamma \mathbb{1}(y = k) | f \in F\}\), we have

\[
\mathcal{R}_S(\tilde{\ell}_F) \leq K \mathcal{R}_S(\phi \gamma \circ H^k) \leq \frac{K}{\gamma} \mathcal{R}_S(H^k),
\]

(24)
where inequality (i) is the Lemma 9.1 of (Mohri et al. 2018), inequality (ii) is due to the Lipschitz property of \( \phi \). Now, define \( f^k(x, y) = \inf_{\|x - x'\| \leq \epsilon} (\|f(x')\|_y - f(x')_k) \), we have \( h^k(x, y) = f^k(x, y) + \gamma \mathbb{1}(y = k) \). Define the function class

\[
\mathcal{F}^k = \{ f^k(x, y) = \inf_{\|x - x'\| \leq \epsilon} (\|f(x')\|_y - f(x')_k) | f \in \mathcal{F} \}.
\]

We have

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

\[
= \frac{1}{n} \mathbb{E}_\sigma \sup_{h^k \in H^k} \sum_{i=1}^n \sigma_i \left[ f^k(x_i, y_i) + \gamma \mathbb{1}(y = k) \right]
\]

\[
= \frac{1}{n} \mathbb{E}_\sigma \sup_{h^k \in H^k} \sum_{i=1}^n \sigma_i f^k(x_i, y_i) + \frac{1}{n} \mathbb{E}_\sigma \sum_{i=1}^n \sigma_i \mathbb{1}(y = k)
\]

\[
= \frac{1}{n} \mathbb{E}_\sigma \sup_{f \in \mathcal{F}^k} \sum_{i=1}^n \sigma_i f^k(x_i, y_i)
\]

\[
= \mathcal{R}_S(\mathcal{F}^k)
\]

Finally, we need to bound the Rademacher complexity of \( \mathcal{R}_S(\mathcal{F}^k) \). Notice that

\[
[f(x)]_y - [f(x)]_k = (W^y_l - W^k_l) \rho(W_{l-1} \rho(\cdots W_1(x) \cdots)),
\]

and we have \( \|W^y_l - W^k_l\|_F \leq 2M_j \). By Theorem 4 (the results in binary classification case), we have

\[
\mathcal{R}_S(\mathcal{F}^k) \leq \frac{48}{\sqrt{n}} \max \{1, d^{\frac{1}{2} - \frac{1}{p}}\} (\|X\|_{p, \infty} + \epsilon)L_{\rho}^{l-1} \sum_{j=1}^{l} h_j h_{j-1} \log(3l) \prod_{j=1}^{l} M_j. \quad (25)
\]

Combining Eq. (25) and (24), we obtain that

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

**B. Discussion on Existing Methods for Rademacher Complexity**

In this section, we discuss the related work, discuss the existing methods in calculating Rademacher complexity, and identify the difficulty of analyzing adversarial Rademacher complexity.

**B.1 Existing Bounds for (Standard) Rademacher Complexity**

**Layer Peeling.** The main idea of calculating the Rademacher complexity of multi-layers neural networks is the ‘peeling off’ technique (Neyshabur et al. 2015). We denote \( g \circ \mathcal{F} \) as the function class.
By Talagrand’s Lemma, we have \( \mathcal{R}_S(g \circ f) \leq L_g \mathcal{R}_S(F) \). Based on this property, we can obtain \( \mathcal{R}_S(F_l) \leq 2L_{\rho} M_j \mathcal{R}_S(F_{l-1}) \), where \( F_l \) is the function class of \( l \)-layers neural networks. Since the Rademacher complexity of linear function class is bounded by \( O(B M_1 / \sqrt{n}) \), we can get the upper bound \( O(B^2 l \sum_{j=1}^{l} M_j / \sqrt{n}) \) by induction. We can remove the \( L_{\rho} \) by assuming that \( L_{\rho} = 1 \) (e.g. Relu activation function).

(Golowich et al., 2018) improves the dependence on depth-\( d \) from \( 2^l \) to \( \sqrt{d} \). The main idea is to rewrite the Rademacher complexity \( E_\sigma [\cdot] \) as \( E_\sigma \exp \ln [\cdot] \). Then, we can peel off the layer inside the \( \ln (\cdot) \) function and the \( 2^l \) now appears inside the \( \ln (\cdot) \).

**Covering Number.** (Bartlett et al., 2017) uses a covering numbers argument to show that the generalization gap scale as

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

where \( \| \cdot \| \) is the spectral norm. The proof is based on the induction on layers. Let \( W_j \) be the weight matrix of the present layer and \( X_j \) be the output of \( X \) pass through the first to the \( l-1 \) layer. Then, one can compute the matrix covering number \( \mathcal{N}(\{W_j X_j\}, \| \cdot \|_2, \epsilon) \) by induction.

**B.2 Existing Bounds for (Variants) of Adversarial Rademacher Complexity**

In linear cases, the upper bounds can be directly derived by the definition of ARC (Khim and Loh, 2018; Yin et al., 2019). Below we discuss the attempts of analyzing adversarial Rademacher complexity in multi-layers cases. We start from the work using modified adversarial loss.

**Tree Transformation Loss.** The work of (Khim and Loh, 2018) introduced a tree transformation \( T \) and showed that \( \max_{\|x-x'\| \leq \epsilon} \ell(f(x), y) \leq \ell(T f(x), y) \). Then, we have the following upper bound for the adversarial population risk. For \( \delta \in (0, 1) \),

\[
\tilde{R}(f) \leq R(T f) \leq R_n(T f) + 2L \mathcal{R}_S(T \circ F) + \sqrt{\log \frac{Z}{2n}}.
\]

It gives an upper bound of the robust population risk by the empirical risk and the standard Rademacher complexity of \( T \circ f \). \( \mathcal{R}_S(T \circ F) \) can be viewed as an approximation of adversarial Rademacher complexity. However, the empirical risk \( R_n(T f) \) is not the objective in practice. This analysis does not provide a guarantee for robust generalization gaps.

**SDP Relaxation Surrogate Loss.** In the work of (Yin et al., 2019), the authors defined the SDP surrogate loss as

\[
\tilde{\ell}(f(x), y) = \phi_\gamma \left( M(f(x), y) - \frac{\epsilon}{2} \max_{k \in [K], z_{x=\pm 1}} \max_{P \succeq 0, \text{diag}(P) \leq 1} \langle z Q(w_{2,k}, W_1), P \rangle \right)
\]

to approximate the adversarial loss for two-layer neural nets. Therefore, the adversarial Rademacher complexity is approximated by the Rademacher complexity on this loss function.
**FGSM Attack Loss.** The work of (Gao and Wang, 2021) tries to provide an upper bound for adversarial Rademacher complexity. To deal with the max operation in the adversarial loss, they consider FGSM adversarial examples. By some assumptions on the gradient, they provide an upper bound for adversarial Rademacher complexity using the loss \( \ell(f(x_{FGSM}), y) \). However, the bound includes some parameters of the assumptions on the gradients, it is not a clean bound.

The following work provided a bound using original adversarial loss.

**Massert’s Lemma Bound.** The work of (Awasthi et al., 2020) showed that the ARC is bounded by

\[
O\left( \frac{(B + \epsilon)\sqrt{h_1 q} \sqrt{\log m_1 m_2}}{\sqrt{n}} \right)
\]

in two-layers cases.

### B.3 Difficulty of Using Layer Peeling to Bound Adversarial Rademacher Complexity

We first take a look at the layer peeling technique.

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

\[
= \mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{h' \in \mathcal{H}_{l-1}, \|W_i\| \leq M_l} \sum_{i=1}^n \sigma_i W_i \rho(h'(x_i)) \right]
\]

\[
\leq M_l \mathbb{E}_\sigma \left[ \frac{1}{n} \left\| \sum_{i=1}^n \sigma_i \rho(h'(x_i)) \right\| \right]
\]

\[
\leq 2M_l L_\rho \mathbb{E}_\sigma \left[ \frac{1}{n} \sum_{i=1}^n \sigma_i h'(x_i) \right]
\]

\[
= 2M_l L_\rho \mathbb{R}_S(\mathcal{H}_{l-1})
\]

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

\[
\mathcal{R}_S(\tilde{\mathcal{H}}) = \mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{h \in \mathcal{H}} \sum_{i=1}^n \sigma_i \max_{\|x_i - x'_i\| \leq \epsilon} h(x'_i) \right]
\]

\[
= \mathbb{E}_\sigma \left[ \frac{1}{n} \sup_{h' \in \mathcal{H}_{l-1}, \|W_i\| \leq M_l} \sum_{i=1}^n \sigma_i W_i \rho(h'(x_i^*(h))) \right]
\]

\[
\leq M_l \mathbb{E}_\sigma \left[ \frac{1}{n} \left\| \sum_{i=1}^n \sigma_i \rho(h'(x_i^*(h))) \right\| \right]
\]

\[
\leq 2M_l L_\rho \mathbb{E}_\sigma \left[ \frac{1}{n} \sum_{i=1}^n \sigma_i h'(x_i^*(h)) \right]
\]

\[
\neq 2M_l L_\rho \mathbb{E}_\sigma \left[ \frac{1}{n} \sum_{i=1}^n \sigma_i h'(x_i^*(h')) \right]
\]

\[
= 2M_l L_\rho \mathbb{R}_S(\tilde{\mathcal{H}}_{l-1})
\]
where $x^*_i(h)$ is the optimal adversarial example given an $l$-layers neural networks, $x^*_i(h')$ is the optimal adversarial example given a $l - 1$-layers neural networks. $x^*_i(h) \neq x^*_i(h')$ is the main reason why layer peeling cannot be directly extended to the adversarial settings.

B.4 Comparison of Different Adversarial Generalization Bounds

VC-Dimension Bounds. A classical approach in statistical learning is to use VC dimension to bound the generalization gap. It is thus natural to apply the VC-dim framework to adversarial setting, as (Cullina et al., 2018; Montasser et al., 2019; Attias et al., 2021) did. However, these works did not provide a computable bound on the adversarial generalization gap, as explained next. Let $\mathcal{H}$ be the hypothesis class (e.g. the set of neural networks with a given architecture).

In the work of (Cullina et al., 2018), the authors defined adversarial VC-dim (AVC) and gave an bound on adversarial generalization gap with respect to $AVC(\mathcal{H})$. However, they did not show how to calculate $AVC$ of neural works. Therefore, their paper did not provide a computable bound for adversarial generalization gap.

In the work of (Montasser et al., 2019), the authors defined the adversarial function class as $L^U\mathcal{H}$, where $L$ is the loss and $U$ is the uncertainty set. They bound the adversarial generalization gap by $L^U\mathcal{H}$, which is different from $AVC(\mathcal{H})$ of (Cullina et al., 2018). However, the authors did not provide a computable bound of as well, which means that their paper did not provide a computable bound of the adversarial adversarial generalization gap.

In the work of (Attias et al., 2021), the authors assume that the perturbation set $U(x)$ is finite, i.e., for each sample $x$, there are only $k$ adversarial examples that can be chosen. They showed that the adversarial generalization gap can be bounded by

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

Note that there is a computable bound of $VC(\mathcal{H})$, which is the number of parameters, thus in terms of ”computable”, this bound is stronger than the previous two. However, this comes at a price: their bound depends on $k$, the number of allowed examined perturbed samples. This is a deviation from the original notion of adversarial generalization, where $U(x)$ is assumed to be an infinite set ($k \neq +\infty$). In contrast, our bound is for the ”original” adversarial generalization gap, which allows $k = +\infty$.

Adversarial Generalization Bounds in Other Settings. The work of (Javanmard et al., 2020) study the generalization properties in the setting of linear regression. Gaussian mixture models are used to analyze adversarial generalization (Taheri et al., 2020; Javanmard et al., 2020; Dan et al., 2020). Uniform stability analysis on adversarial training Xing et al. (2021); Xiao et al. (2022a b) showed that the poor generalization of adversarial training might be due to the non-smoothness of adversarial loss.

Certified robustness. A series of works study the certified robustness within the norm constraint around the original data. (Cohen et al., 2019) provides an analysis on certified robustness via random smoothing. (Lecuyer et al., 2019) studies certified robustness through the lens of differential privacy.

Other Theoretical Studies on Adversarial Examples. A series of works (Gilmer et al., 2018; Khoury and Hadfield-Menell, 2018) study the geometry of adversarial examples. The off-manifold assumption tells us that the adversarial examples leave the underlying data manifold (Szegedy et al.,
ADVERSARIAL RADEMACHER COMPLEXITY OF DEEP NEURAL NETWORKS

2013). Pixeldefends (Song et al., 2017) uses a generative model to show that adversarial examples lie in a low probability region of the data distribution. The work of (Ma et al., 2018) uses Local Intrinsic Dimensionality (LID) to argues that the adversarial subspaces are of low probability, and lie off the data submanifold.

C. Additional Experiments

In this section, we provide additional experiments.

C.1 Experiments on VGG-11 and VGG-13

In Figure 2, we show the experiments on VGG-11 and VGG-13. In Figure 3, we show the experiments on VGG-16 and VGG-19. The gap of product of Frobenius norm between standard and adversarial training is large, which yields bad generalization.

Figure 2: Product of the Frobenius norm in the experiments on VGG networks. The red lines are the results of standard training. The blue lines are the results of adversarial training. The first row are the experiments on VGG-11. The second row are the experiments on VGG-13. (a) and (e): Generalization gap. (b) and (f): Margin $\gamma$ over training set. (c) and (g): $\prod_{j=1}^l \|W_j\|_F$ of the neural networks. (d) and (h): $\prod_{j=1}^l \|W_j\|_{1,\infty}$ of the neural networks.

$\|\cdot\|_{1,\infty}$-Norm Bounds. The $\|\cdot\|_{1,\infty}$-norm bounds are shown in Figure 4. 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 of Margins

In Figure 5, we show the results of the margins in 1$^{th}$, 3$^{th}$, and 5$^{th}$-percentile of the training dataset. Since the (robust) training accuracy is 100%, the choice of percentile will not affect the results. As
Figure 3: Product of the Frobenius norm in the experiments on CIFAR-10. The red lines are the results of standard training. The blue lines are the results of adversarial training. The first row is the experiments on VGG-16. The second row is the experiments on VGG-19. (a) and (e): Generalization gap. (b) and (f): Margin. (c) and (g): $\prod_{j=1}^{l} \|W_j\|_F$ of the neural networks. (d) and (h): $\prod_{j=1}^{l} \|W_j\|_F/\gamma$ of the neural networks.

Figure 4: Product of the $\|\cdot\|_{1,\infty}$-Norm in the experiments on CIFAR-10. The red lines are the results of standard training. The blue lines are the results of adversarial training. (a) $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$ of VGG-16 networks. (b) $\prod_{j=1}^{l} \|W_j\|_{1,\infty}/\gamma$ of VGG-16 networks. (c) $\prod_{j=1}^{l} \|W_j\|_{1,\infty}$ of VGG-19 networks. (d) $\prod_{j=1}^{l} \|W_j\|_{1,\infty}/\gamma$ of VGG-19 networks.
we can see in the Figure, in all the cases, the margins of standard training are larger than the margins of adversarial training. Since the margins appear in the divider in the upper bound of Rademacher complexity, the margins of the training dataset have some small effects on the bad generalization of adversarial training.

C.3 Experiments on CIFAR-100

Performance. In Table 3 we show the performance of standard training and adversarial training on CIFAR-100 using VGG-16 and 19 networks. We can see that using smaller number of training samples is unable to train an acceptable VGG-networks on CIFAR-100. Therefore it is hard use only 50000 training samples to study the trends of the weight norm using the experiments on CIFAR-100. We compare the product of weight norm between standard and adversarial training.

Figure 5: Ablation study of margins. The first to the 4\textsuperscript{th} rows are the experiments on VGG-11, 13, 16, and 19, respectively.

Product of Weight Norms. In Figure 6, we show the results of on training VGG-19-16 and VGG-19 on CIFAR-100. Similar to the experiments on CIFAR-10, we can see that the adversarially trained models have larger weight norm that that of the standard trained model.
Figure 6: Product of the Frobenius norm in the experiments on VGG networks on CIFAR-100. The red lines are the results of standard training. The blue lines are the results of adversarial training. The first row are the experiments on VGG-16. The second row are the experiments on VGG-19. (a) and (e): Generalization gap. (b) and (f): Margin $\gamma$ over training set. (c) and (g): $\prod_{j=1}^{l} \|W_j\|_F$ of the neural networks. (d) and (h): $\prod_{j=1}^{l} \|W_j\|_F / \gamma$ of the neural networks.

Figure 7: Experiments on the effects of weight decay. (a) Robust generalization gap with or without weight decay on VGG-16. (b) Frobenius norm with or without weight decay on VGG-16. (c) Robust generalization gap with or without weight decay on VGG-19. (d) Frobenius norm with or without weight decay on VGG-19.
Table 3: Accuracy of standard and adversarial training on CIFAR-100 using VGG-16 and 19 networks. For standard training model, we show the clean accuracy. For adversarial training model, we show the robust accuracy against PGD attacks.

| 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 8: Experiments on the effects of weight decay ranging from $1 \times 10^{-3}$ to $9 \times 10^{-3}$. (a) Training error. (b) Frobenius norm. (c) Robust generalization gap.

C.4 Weight Decay

The upper bounds of adversarial Rademacher complexity suggest adding a regularization term on the weights to improve generalization, which is essentially weight decay. In Figure 7, we provide the experiments of adversarial training with and without weight decay. In Figure 7 (a) and (c), we can see that adversarial training with weight decay has a smaller robust generalization gap. In Figure 7 (b) and (d), adversarial training with weight decay have a smaller product of weight norms. These experiments show the relationship between the robust generalization gap and the product of weight norms.

In Figure 8, we increase the weight decay ranging from $1 \times 10^{-3}$ to $9 \times 10^{-3}$. In this range, the training becomes bad. From 6e-3, the margin becomes negative (which means that training error is large). Then, weight norm/margin becomes negative. Form 9e-3, training totally fail, training error = test error = 90%. Therefore, the ‘optimal’ choice of weight decay (for weight norm) is from [1-5]e-3. We see the smallest one is 6.00848997e+12 (in the experiments using wd = 4e-3). This is still large than that of standard training, which is 1.8961088e+12.