A Closer Look at Curriculum Adversarial Training: From an Online Perspective

Lianghe Shi, Weiwei Liu*
School of Computer Science, Wuhan University
Institute of Artificial Intelligence, Wuhan University
National Engineering Research Center for Multimedia Software, Wuhan University
Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University
{shilianghe007, liuweiwei863}@gmail.com

Abstract

Curriculum adversarial training empirically finds that gradually increasing the hardness of adversarial examples can further improve the adversarial robustness of the trained model compared to conventional adversarial training. However, theoretical understanding of this strategy remains limited. In an attempt to bridge this gap, we analyze the adversarial training process from an online perspective. Specifically, we treat adversarial examples in different iterations as samples from different adversarial distributions. We then introduce the time series prediction framework and deduce novel generalization error bounds. Our theoretical results not only demonstrate the effectiveness of the conventional adversarial training algorithm but also explain why curriculum adversarial training methods can further improve adversarial generalization. We conduct comprehensive experiments to support our theory.

1 Introduction

Although deep neural networks are very effective in classifying natural inputs, Szegedy et al. (2014) show that an adversary is often able to perturb the input with an imperceptible change so that the model produces an incorrect output. This phenomenon has received particular attention in the context of deep neural networks, and there is now a sizable body of work devoting to improving the adversarial robustness of the trained neural network models (Kurakin, Goodfellow, and Bengio 2017; Carlini and Wagner 2017; Madry et al. 2018; Zhang et al. 2019; Rebuffi et al. 2021; Gowal et al. 2021; Pang et al. 2022; Shi and Liu 2023). Adversarial robustness refers to the invariance of a model to small perturbations of its input (Salman et al. 2020).

Adversarial training is one of the most popular and effective defense techniques used to train an adversarially robust model. The objective of adversarial training is to solve a min-max problem (Madry et al. 2018). In practice, adversarial training first generates adversarial examples based on the model under investigation, then uses these examples to train a new model that is robust against these attacks. Therefore, the quality of the generated adversarial examples directly affects the adversarial generalization of the model (Zhang et al. 2020). Interestingly, a series of works (Cai, Liu, and Song 2018; Wang et al. 2019; Zhang et al. 2020; Kinfu and Vidal 2022), called curriculum adversarial training, have demonstrated that gradually increasing the hardness of adversarial examples can help to improve the performance of the trained model. These methods have a similar min-max optimization framework with conventional adversarial training, and control the hardness of the generated adversarial examples in different ways. Empirically, the advantages of curriculum adversarial training to improve the robustness and to alleviate the trade-off between clean accuracy and adversarial accuracy have been verified in prior works (Zhang et al. 2020; Cheng et al. 2022). However, the theoretical motivation of curriculum adversarial training remains unclear.

Remarkable theoretical advances (Chen and Liu 2023; Khim and Loh 2018; Attias, Kontorovich, and Mansour 2019; Montasser, Hanneke, and Srebro 2019; Yin, Ramchandran, and Bartlett 2019; Gao and Wang 2021; Xiao et al. 2022a,c; Xing, Song, and Cheng 2021a; Xu and Liu 2022) have been achieved in adversarial training. Despite this fact, these existing theories focus solely on the fully trained model while ignoring the intermediate objects, such as the adversarial examples generated in each iteration of the training process. Thus, we cannot directly use these existing theoretical results to explain the effectiveness of curriculum adversarial training.

In this paper, we delve deeper into the adversarial training process from an online perspective and propose a novel generalization error bound through the lens of Rademacher complexity. We further introduce the time series prediction framework to propose an improved error bound. Our theoretical results show that the expected adversarial error can be upper-bounded by the average of the optimization objectives of each intermediate iteration, which provides theoretical support for the effectiveness of the two-player iterative training algorithm. Moreover, the error bound contains the distance between the generated adversarial examples of different iterations. We further investigate this term and convert it into an optimizable form that can explain the efficacy of curriculum adversarial training. This paper makes a first step toward a deeper understanding of curriculum adversarial training. Extensive numerical experiments on CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009) datasets verify our theoretical bounds and the explanation we provide regarding curriculum adversarial training.

*Corresponding author.

2 Related Work

Adversarial Training. After (Szegedy et al. 2014) shows that DNNs are fragile to adversarial attacks, a large amount of works have proposed various attack methods (Madry et al. 2018; Kurakin, Goodfellow, and Bengio 2017; Carlini and Wagner 2017) and defense methods (Wang et al. 2023; Goodfellow, Shlens, and Szegedy 2015; Zhang et al. 2019; Li, Xin, and Liu 2022; Gowal et al. 2021; Li and Liu 2023). Adversarial training (Goodfellow, Shlens, and Szegedy 2015) is one of the popular and effective methods that improves adversarial robustness by adding adversarial examples to the training dataset.

From the theoretical perspective, some works focus on the sample complexity (Zhou and Liu 2023; Cullina, Bhagoji, and Mittal 2018) and the generalization of adversarial training. A series of works theoretically analyzes the generalization error bound through the lens of the VC dimension (Attias, Kontorovich, and Mansour 2019; Montasser, Hanneke, and Srebro 2019) and the Rademacher complexity (Khim and Loh 2018; Yin, Ramchandran, and Bartlett 2019; Xiao et al. 2022a; Mustafa, Lei, and Kloft 2022; Gao and Wang 2021). Another line of work studies the generalization performance from the perspective of algorithmic stability (Xing, Song, and Cheng 2021a; Xiao et al. 2022b,c) and feature purification (Allen-Zhu and Li 2021). (Ma, Wang, and Liu 2022) investigates the trade-off between robustness and fairness. Wang and Liu (2022); Zou and Liu (2023) study adversarial robustness under self-supervised learning. However, the impact of the intermediate adversarial examples and hypotheses generated in each iteration of the training process on generalization performance remains unknown.

Experimentally, the properties of adversarial examples have been shown to play important roles in adversarial training. Cai, Liu, and Song (2018) use the strategy of curriculum learning (Bengio et al. 2009) to improve adversarial robustness. Their findings show that gradually increasing the hardness of adversarial examples is beneficial for adversarial robustness. Subsequently, a number of works have proposed different hardness measures of adversarial examples. For example, Wang et al. (2019) use the first-order stationary condition as the hardness measure, Kinfu and Vidal (2022) use the perturbation radius to control the hardness, and Zhang et al. (2020) search for the least adversarial examples that minimize the loss from among those adversarial examples that are confidently misclassified. Notably, however, the aforementioned theoretical works cannot theoretically explain the effectiveness of curriculum adversarial training. This paper accordingly takes some steps towards a deeper understanding of this strategy.

Online Learning and Time Series Prediction. Rakhlin, Sridharan, and Tewari (2010) develop a theory of online learning by defining several complexity measures and demonstrating their connection to online learning. Using the tools of online learning theory, Rakhlin, Sridharan, and Tewari (2015) consider the problem of sequential prediction. Kuznetsov and Mohri (2014) first present the generalization bounds for time series prediction with a non-stationary mixing stochastic process. Kuznetsov and Mohri (2015, 2016, 2020) then subsequently prove the generalization bounds for time series prediction in the general setting of a non-stationary non-mixing stochastic process. This paper provides a new online perspective on adversarial training.

3 Preliminaries

3.1 Standard Statistical Learning Framework

We first consider a standard training framework. Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be a measurable instance space, where $\mathcal{X}$ and $\mathcal{Y}$ denote the feature and label spaces, respectively. The feature space $\mathcal{X}$ is a subset of a d-dimensional space, $\mathcal{X} \subseteq \mathbb{R}^d$. The label space is $\{0, 1\}$ in binary classification, $\{1, \cdots, c\}$ in multiclass classification, and some measurable subset of $\mathbb{R}$ in regression. We assume that the learner is provided with n training samples $Z \coloneqq \{z_i\}_{i=1}^n = \{(x_i, y_i)\}_{i=1}^n$, drawn independent and identically distributed (i.i.d.) according to a fixed but unknown distribution P. We use $\widehat{P}$ to denote the empirical distribution. A hypothesis class $\mathcal{H}$ is defined as a set of functions $h : \mathcal{X} \to \mathcal{Y}$. We use $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ to denote the loss function and $\epsilon_P(h) \coloneqq \mathop{\mathbb{E}}_{(x,y)\sim P}[\ell(h(x), y)]$ to indicate the expected standard error of any hypothesis h under any distribution P over $\mathcal{Z}$. In this paper, we assume that the hypothesis $h \in \mathcal{H}$ is R-Lipschitz, and the loss function ℓ is ρ-Lipschitz and bounded by $\mathcal{M} \geq 0$. These assumptions are widely adopted in prior theoretical works (Yin, Ramchandran, and Bartlett 2019; Awasthi, Frank, and Mohri 2020; Gao and Wang 2021). The detailed formulas of these assumptions can be found in the Appendix. In a standard learning framework, the learner aims to select a hypothesis $h \in \mathcal{H}$ that yields the minimal expected standard error.

3.2 Adversarial Training

Unfortunately, the hypothesis selected by the standard training process is vulnerable. Given a hypothesis h and a natural example (x, y), we can generate the corresponding adversarial example (x^{adv}, y) by adversarially perturbing x in a small neighborhood $B_\delta(x)$ of x, as follows: $x^{adv} = \arg\max_{x' \in B_\delta(x)} \ell(h(x'), y)$. In this paper, we focus on l_p adversarial perturbation $B_\delta(x) \coloneqq \{x' \in \mathcal{X} : \|x - x'\|_p \leq \delta, p \geq 1\}$, which is an l_p-ball around x of radius $\delta \geq 0$ and has been widely studied in existing work. For a vector $x \in \mathbb{R}^d$, we define the l_p-norm of x as $\|x\|_p \coloneqq \left(\sum_{i=1}^d |x_{(i)}|^p\right)^{1/p}$ for $p \in [1, \infty)$, where $x_{(i)}$ is the i-th element of x; for $p = \infty$, we define $\|x\|_\infty \coloneqq \max_{1 \leq i \leq d} |x_{(i)}|$. Following (Tu, Zhang, and Tao 2019), we denote $T_h : \mathcal{Z} \to \mathcal{Z}$ as a measurable function that transports a natural example to an adversarial example according to h. We then obtain the distribution of adversarial examples by pushing forward the original distribution P into a new distribution $P^{adv}(h) \coloneqq T_h \# P$ using the transport function T_h. Next, we provide the formal definition of expected adversarial error to measure the performance of a hypothesis in the presence of adversaries.

Definition 3.1. (Expected Adversarial Error). The expected adversarial error of a hypothesis $h \in \mathcal{H}$ over the

distribution P against the l_p perturbation of radius δ is:

$$\epsilon_P^{adv}(h) \coloneqq \mathop{\mathbb{E}}_{(x,y)\sim P}\left[\max_{x'\in B_\delta(x)} \ell(h(x'),y)\right].$$

Through the lens of pushforward distribution $P^{adv}(h)$, we can degrade the expected adversarial error into expected standard error without the adversarial maximization:

$$\epsilon_P^{adv}(h) = \epsilon_{P^{adv}(h)}(h) = \mathop{\mathbb{E}}_{(x',y)\sim P^{adv}(h)}\left[\ell\left(h(x'),y\right)\right].$$

The goal of adversarial training is to select a hypothesis $h \in \mathcal{H}$ that achieves a small expected adversarial error. However, the distribution P is unknown to the learner, and in practice, we train the model by minimizing the empirical adversarial error.

Definition 3.2. (Empirical Adversarial Error). Given the observed samples $\{x_i\}_{i=1}^n$, the empirical adversarial error of a hypothesis $h \in \mathcal{H}$ against the l_p perturbation of radius δ is:

$$\hat{\epsilon}_P^{adv}(h) = \hat{\epsilon}_{P^{adv}(h)}(h) = \frac{1}{n}\sum_{i=1}^n \max_{x_i'\in B_\delta(x_i)} \ell\left(h(x_i'),y_i\right).$$

The optimization problem of adversarial training can then be written as a natural saddle point (min-max) formulation: $\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \max_{x_i'\in B_\delta(x_i)} \ell\left(h(x_i'),y_i\right))$. To solve this min-max problem, adversarial training iteratively optimizes the inner maximization and the outer minimization in a nested loop. Let T be the total number of iterations of the outer minimization. There are two steps in each iteration of the outer loop. Specifically, for the t-th iteration where $t \in \{0,1,\cdots,T-1\}$, these steps are as follows:

1) Fix the hypothesis h_t, after which an attacker (such as PGD (Madry et al. 2018)) aims to find the adversarial examples $\{x_i^{adv}\}_{i=1}^n$ that maximize the empirical error: $x_i^{adv} = \arg\max_{x_i'\in B_\delta(x_i)} \ell(h_t(x_i'),y_i)$. We assume that the attacker is effective and the adversarial examples achieve the inner maximum. Since $P^{adv}(h_t)$ is the distribution of the adversarial examples, for any $h \in \mathcal{H}$, the following holds:

$$\hat{\epsilon}_{P^{adv}(h_t)}(h_t) - \hat{\epsilon}_{P^{adv}(h)}(h_t) \geq 0. \tag{1}$$

2) Based on the adversarial examples obtained in step 1), the learner calculates the gradient of the error $\hat{\epsilon}_P^{adv}(h_t)$ and updates the hypothesis h_t to h_{t+1} via gradient descent. For any $t \in \{0,\cdots,T-1\}$, the following bound holds:

$$\hat{\epsilon}_{P^{adv}(h_t)}(h_t) - \hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1}) \geq 0. \tag{2}$$

The adversarial training process iterates between step 1) and step 2). We present the relationship between these intermediate terms in Figure 1. Note that the intermediate adversarial examples considered in this paper refer to the output examples of the attacker in each iteration.

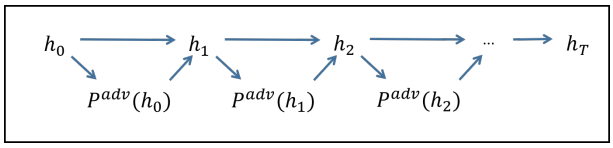

Figure 1: Illustration of the adversarial training process. We use h_0 to denote the randomly initialized hypothesis and h_T to denote the output hypothesis of the last iteration.

3.3 Wasserstein Distance

Wasserstein distance (Wainwright 2019), also known as the earth mover's distance, is a widely used metric to measure the distance between two distributions. We provide the definition of Wasserstein distance below.

Definition 3.3. (Wasserstein Distance). Let $(\mathcal{Z},d)$ be a metric space. For $q \geq 1$, the Wasserstein q-distance between two probability distributions P and Q over $\mathcal{Z}$ with finite q-moments is:

$$W_q(P,Q) \coloneqq \left(\inf_{\gamma\in\Gamma(P,Q)} \mathbb{E}_{(z,z')\sim\gamma} d(z,z')^q\right)^{1/q}, \tag{3}$$

where $\Gamma(P,Q)$ is the set of all couplings of P and Q. A coupling γ is a joint probability distribution over $\mathcal{Z}\times\mathcal{Z}$ whose marginals are P and Q on the first and second factors, respectively. We define $d(z,z') \coloneqq \|x-x'\|_p + |y-y'|$.

4 Theoretical Analysis for Adversarial Training

Adversarial training iteratively solves the inner maximization and the outer minimization problems. At each iteration t, the learner minimizes the empirical error over distribution $P^{adv}(h_t)$ which depends on the current hypothesis h_t. Therefore, different iterations have different optimization objectives. The existing error bounds presented in (Khim and Loh 2018; Yin, Ramchandran, and Bartlett 2019; Gao and Wang 2021) disregard these intermediate adversarial examples and intermediate adversarial errors. In this section, we delve deeper into the adversarial training process and propose an error bound that involves each optimization objective of the corresponding iteration. Our theoretical results can be divided into three parts: (i). We first consider the error difference for each iteration, i.e., $\epsilon_{P^{adv}(h_T)}(h_T) - \hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1})$. We derive an upper bound of the error difference through the lens of Rademacher complexity. The error difference corresponding to each iteration can then be summed over to facilitate the derivation of a generalization error bound that is applicable to the entire training process. However, this bound is somewhat loose, since we bound each error difference of the corresponding iteration separately. (ii). We then adopt tools from the time series prediction and online learning literature to derive a tighter bound for adversarial training. This bound thinks of T iterations as a whole and makes use of all nT examples together. (iii). We further convert the generalization error bound into an optimizable formula. Notably, our theoretical results can explain the effectiveness of existing curriculum adversarial training methods. The proofs can be found in the Appendix.

4.1 Error Bound of Each Iteration

In this subsection, we analyze the error difference, i.e., $\epsilon_{P^{adv}(h_T)}(h_T) - \hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1})$. The trick is to split it into two parts and bound them separately. We first use the Wasserstein distance to bound the error difference of a classifier between a pair of distributions in the following proposition.

Proposition 4.1. *Consider two arbitrary distributions P and Q over $\mathcal{Z}$. Then, for any hypothesis $h \in \mathcal{H}$ and loss function ℓ, the following bound holds:*

$$|\epsilon_P(h) - \epsilon_Q(h)| \leq \rho\sqrt{R^2+1}W_q(P,Q). \tag{4}$$

Based on Proposition 4.1, we connect the empirical errors of the fully trained model and the intermediate model in the following Proposition.

Proposition 4.2. *For an arbitrary iteration $t \in \{0, \cdots, T-1\}$ in the adversarial training process as described in section 3.2, the following bound holds:*

$$\begin{aligned}&\hat{\epsilon}_{P^{adv}(h_T)}(h_T) - \hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1}) \\ &\leq \sum_{t'=t}^{T-1} \rho\sqrt{R^2+1}W_q\left(\hat{P}^{adv}(h_{t'}), \hat{P}^{adv}(h_{t'+1})\right).\end{aligned} \tag{5}$$

Since we focus on the expected adversarial error of the trained model, we then introduce Rademacher complexity (Wainwright 2019) to bound the generalization error.

Definition 4.3. (Rademacher Complexity) Let $\mathcal{F}$ be a set of real-valued functions defined over $\mathcal{Z}$. For any fixed collection of points $Z := (z_1, \cdots, z_n)$, the empirical Rademacher complexity of $\mathcal{F}$ is given by:

$$\widehat{\mathcal{R}}_Z(\mathcal{F}) = \frac{2}{n}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{n}\sigma_i f(z_i)\right]. \tag{6}$$

The expectation is taken over $\sigma = (\sigma_1, \cdots, \sigma_n)$, where σ_is, $i \in \{1, \cdots, n\}$, are independent uniform random variables taking values in $\{-1, +1\}$.

Rademacher complexity is used to measure the complexity of a hypothesis set. Let $\widetilde{\ell} \circ \mathcal{H}$ be a class of real-valued functions defined over the feature space $\mathcal{Z}$, i.e.:

$$\widetilde{\ell} \circ \mathcal{H} := \{(x,y) \to \max_{x' \in B_\delta(x)} \ell(h(x'), y) : h \in \mathcal{H}\}. \tag{7}$$

We can use the Rademacher complexity to connect the population and empirical error (Bartlett and Mendelson 2002) as follows.

Proposition 4.4. *Let $\widetilde{\ell} \circ \mathcal{H}$ be the function class in Eq. (7). Then, for any $\alpha \geq 0$, with probability of at least $1-\alpha$ over samples Z of size n, the following holds for any $h_T \in \mathcal{H}$:*

$$\epsilon_P^{adv}(h_T) - \hat{\epsilon}_P^{adv}(h_T) \leq \widehat{\mathcal{R}}_Z(\widetilde{\ell} \circ \mathcal{H}) + 3\mathcal{M}\sqrt{\frac{\log\frac{2}{\alpha}}{2n}}. \tag{8}$$

By combining Proposition 4.2 with Proposition 4.4, we can derive the error difference bound for each iteration. We then sum up all the error differences over all iterations to get the following Theorem.

Theorem 4.5. *Consider the adversarial training process as described in section 3.2. For any $\alpha \geq 0$, with probability of at least $1-\alpha$ over samples Z, the following bound holds:*

$$\begin{aligned}\epsilon_P^{adv}(h_T) \leq \frac{1}{T}\sum_{t=0}^{T-1}\hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1}) + \mathcal{O}\left(\frac{1}{T}\sum_{t=1}^{T} t\Delta_{t-1,t}\right) \\ + \widehat{\mathcal{R}}_Z(\widetilde{\ell} \circ \mathcal{H}) + \mathcal{O}\left(\sqrt{\frac{\log\frac{2}{\alpha}}{2n}}\right),\end{aligned} \tag{9}$$

where $\Delta_{t-1,t} = W_q(\hat{P}^{adv}(h_{t-1}), \hat{P}^{adv}(h_t))$.

Remark 4.6. According to this theorem, the expected adversarial error of the output hypothesis of the adversarial training algorithm is upper-bounded by three parts: (i). The average empirical adversarial error of the hypothesis in each iteration. Note that the hypothesis h_{t+1} is trained to minimize the empirical error on distribution $P^{adv}(h_t)$. The first part is thus the optimization objectives of each iteration. Unlike existing work, our bound involves every intermediate iteration and accordingly shows that the conventional adversarial training algorithm can indeed help to improve the robustness of the hypothesis from a dynamic training perspective. (ii). A summation of the Wasserstein distance between adjacent adversarial distributions. (iii). Rademacher complexity of the hypothesis class and a term converging to 0 as the sample size n increases to infinity.

4.2 An Online Perspective for Adversarial Training

In the last subsection, we derive an error bound by separately considering the error difference of each iteration. From the last term in the bound (9), i.e., $\sqrt{\frac{\log\frac{2}{\alpha}}{2n}}$, we know that the bound does not make use of all nT training samples, and the convergence rate is $\mathcal{O}(n^{-1/2})$. The question is, can we derive a tighter bound involving terms with convergence rate $\tilde{\mathcal{O}}((nT)^{-1/2})$?

To answer this question, we adopt tools from online learning and time series prediction. We will first introduce some definitions and lemmas. In time series prediction theory, we can view adversarial training from an online perspective, and the training (adversarial) samples in all iterations can be seen as a realization of a stochastic process. Similar to the VC dimension and Rademacher complexity in standard supervised learning, there are some sequential complexities used to measure the structural complexity of a class function in an online learning scenario, such as the Littlestone dimension, expected sequential covering number, sequential metric entropy, and sequential Rademacher complexity (Rakhlin, Sridharan, and Tewari 2010). In this paper, we adopt the following definition of a complete binary tree and sequential Rademacher complexity.

Definition 4.7. (Complete Binary Tree (Shalev-Shwartz and Ben-David 2014)). A $\mathcal{Z}$-valued complete binary tree $\mathfrak{Z}$ is a sequence $(\mathfrak{Z}_0, \cdots, \mathfrak{Z}_{T-1})$ of T mappings $\mathfrak{Z}_t : \{\pm 1\}^t \to \mathcal{Z}, t \in \{0, \cdots, T-1\}$. A path in the tree is denoted by $\sigma = (\sigma_0, \cdots, \sigma_{T-2}) \in \{\pm 1\}^{T-1}$. For simplicity, we use $\mathfrak{Z}_t(\sigma)$ to represent $\mathfrak{Z}_t(\sigma_0, \cdots, \sigma_{t-1})$.

Definition 4.8. (Sequential Rademacher Complexity (Rakhlin, Sridharan, and Tewari 2010)). Let $\lambda = (\lambda_0, \cdots, \lambda_{T-1})$ be an arbitrary sequence of real numbers forming a probability distribution. The sequential Rademacher complexity $\mathcal{R}_T^{seq}(\mathcal{F})$ of a function class $\mathcal{F}$ is defined as follows:

$$\mathcal{R}_T^{seq}(\mathcal{F}) = \sup_{\mathfrak{Z}} \mathbb{E}_{\sigma} \left[\sup_{f \in \mathcal{F}} \sum_{t=0}^{T-1} \sigma_t \lambda_t f(\mathfrak{Z}_t(\sigma)) \right], \tag{10}$$

where the supremum is taken over all $\mathcal{Z}$-valued complete binary trees of depth T, and the expectation is taken over a sequence of uniform random variables $\sigma = \{\sigma_0, \cdots, \sigma_{T-1}\}$ taking values in $\{-1, +1\}$.

The sequential Rademacher complexity has been well studied in the online learning literature. In the below, we introduce an example of neural networks.

Example 4.9 (Neural Networks (Rakhlin, Sridharan, and Tewari 2015)). Let $\mathcal{H}$ be R-Lipschitz L-Layer fully connected neural networks with 1-Lipschitz transformation function. The sequential Rademacher complexity can be bounded as:

$$\mathcal{R}_T^{seq}(\mathcal{H}) \leq \mathcal{O}\left(R\sqrt{\frac{(\log T)^{3(L-1)}}{T}} \right). \tag{11}$$

Since the distributions of each sample are different in the context of time series prediction, we need a metric to measure the discrepancy between the target distribution and the distributions of the intermediate samples. We adopt the following notion of discrepancy used in (Kuznetsov and Mohri 2020).

Definition 4.10. (Discrepancy (Kuznetsov and Mohri 2020)). For a stochastic process $\{z_t\}_{t \in \{0, \cdots, T-1\}}$ and an arbitrary sequence $\lambda = (\lambda_0, \cdots, \lambda_{T-1})$ of real numbers forming a probability distribution, the discrepancy $disc_T(\lambda)$ with respect to a hypothesis class $\mathcal{F}$ is defined as:

$$disc_T(\lambda) \coloneqq \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{z_T}[f(z_T)|z_0^{T-1}] - \sum_{t=0}^{T-1} \lambda_t \mathbb{E}_{z_t}[f(z_t)|z_0^{t-1}] \right), \tag{12}$$

where $z_0^t = \{z_0, \cdots, z_t\}$ is a sequence of samples.

In the following Proposition, we bound the discrepancy in the setting of adversarial training as described in section 3.2. Specifically, $\mathcal{F}$ is the family of loss functions associated to $\mathcal{H}$: $\mathcal{F} = \{(x, y) \to \ell(h(x), y) : h \in \mathcal{H}\}$ and z_t follows the distribution $P^{adv}(h_t)$.

Proposition 4.11. *Let $\lambda = (\frac{1}{T}, \cdots, \frac{1}{T})$ be a uniform sequence. For the adversarial training process defined in section 3.2, the discrepancy can be bounded as follows:*

$$disc_T(\lambda) \leq \frac{1}{T} \rho \sqrt{R^2 + 1} \sum_{t=1}^{T} t W_q \left(\hat{P}^{adv}(h_{t-1}), \hat{P}^{adv}(h_t) \right). \tag{13}$$

Based on these introduced concepts, we now present an important generalization error bound for forecasting time series. The detailed results can be found in (Kuznetsov and Mohri 2020).

Theorem 4.12 (Bounds for Time Series Prediction (Kuznetsov and Mohri 2020)). *Let $\{z_t\}_{t \in \{0, \cdots, T-1\}}$ be a stochastic process, and let $\lambda = (\lambda_0, \cdots, \lambda_{T-1})$ be a sequence of real numbers forming a probability distribution. Then, for any $\alpha \geq 0$, with probability of at least $1 - \alpha$, the following inequality holds for all $f \in \mathcal{F}$:*

$$\begin{aligned} \mathbb{E}[f(z_T)|z_0^{T-1}] \leq \sum_{t=0}^{T-1} \lambda_t f(z_t) + disc_T(\lambda) + \|\lambda\|_2 \\ + 6\mathcal{M}\sqrt{4\pi \log T} \mathcal{R}_T^{seq}(\mathcal{F}) + \mathcal{M}\|\lambda\|_2 \sqrt{8 \log \frac{1}{\alpha}}. \end{aligned} \tag{14}$$

However, if we directly apply this theorem in an off-the-shelf way, then we would obtain a generalization bound involving terms with dependence only on T but no dependence on n, in the form $\mathcal{O}(T^{-\frac{1}{2}})$, which means the term does not vanish to 0 even with an infinite dataset size n. In order to derive a generalization error bound with term $\tilde{\mathcal{O}}((nT)^{-\frac{1}{2}})$, we apply Theorem 4.12 from a sample-level perspective and then derive the following result of our generalization bounds for adversarial training.

Theorem 4.13. ***(Generalization Error Bound for Adversarial Training).*** *Let $\mathcal{H}$ be R-Lipschitz L-Layer fully connected neural networks with 1-Lipschitz transformation function. For any $\alpha \geq 0$, with probability of at least $1 - \alpha$, we can bound the adversarial error of the output hypothesis h_T of adversarial training algorithm as follows:*

$$\begin{aligned} \epsilon^{adv}(h_T) \leq \frac{1}{T} \sum_{t=0}^{T-1} \hat{\epsilon}_{P^{adv}(h_t)}(h_{t+1}) + \mathcal{O}\left(\frac{1}{T} \sum_{t=1}^{T} t \Delta_{t-1,t} \right) \\ + \mathcal{O}\left(\mathcal{M}\sqrt{\frac{-8 \log \alpha}{nT}} + \frac{1}{\sqrt{nT}} + \mathcal{M}\rho R \sqrt{\frac{(\log nT)^{3L+1}}{nT}} \right), \end{aligned} \tag{15}$$

where $\Delta_{t-1,t} = W_q\left(\hat{P}^{adv}(h_{t-1}), \hat{P}^{adv}(h_t) \right)$.

Remark 4.14. Although this bound focuses on fully connected neural networks, the theorem can be easily extended to any other hypothesis class by replacing the sequential Rademacher complexity of the corresponding hypothesis class.

Comparison with Eq. (9). A salient improvement of the bound in Eq. (15) compared with the bound in Eq. (9) is that the asymptotic term has the convergence rate $\tilde{\mathcal{O}}((nT)^{-\frac{1}{2}})$. This term characterizes the overall sample size nT used by the algorithm as the iterations progress; i.e., the algorithm updates the model on n adversarial examples in each iteration, and there are T iterations during training.

Comparison with Previous Work. Xiao et al. (2022b) and Xing, Song, and Cheng (2021a) derive generalization error bounds through stability analysis. Their bounds focus on $\epsilon_P^{adv}(h_T) - \hat{\epsilon}_P^{adv}(h_T)$. (Mustafa, Lei, and Kloft 2022) uses Rademacher complexity to bound the generalization error of any model, i.e., $\epsilon_P^{adv}(h) - \hat{\epsilon}_P^{adv}(h)$ for any $h \in \mathcal{H}$. Xing, Song, and Cheng (2021b) study the generalization performance of adversarial training from a statistical estimation perspective. They prove that the adversarial error of

the trained model converges to the minimal adversarial error, i.e., $\epsilon_P^{adv}(h_T) \to \min_{h\in\mathcal{H}} \epsilon_P^{adv}(h)$, with probability tending to 1. However, these bounds do not analyze the intermediate objects and the adversarial examples. Our bounds in (9) and (15) fill this gap.

4.3 Analysis of the Generated Adversarial Examples

The effect of adversarial examples on the robustness of the trained model is widely studied in previous work (Madry et al. 2018). Recently, based on the idea of curriculum learning (Bengio et al. 2009), some authors claim that training with mild adversarial examples in the early stage of training is beneficial to the model (Zhang et al. 2020). These works control the strength of adversarial examples in various ways, such as by constraining the number of iterative steps (Cai, Liu, and Song 2018) and the convergence quality of adversarial examples (Wang et al. 2019). All of these methods can be regarded as different forms of controlling the perturbation radius. We now use our theoretical results in Theorem 4.5 and Theorem 4.13 to explain the effectiveness of this strategy.

In both Eq. (9) and Eq. (15), the Wasserstein distance between adversarial distributions of adjacent iterations plays an important role. In this subsection, we do not consider the constant coefficient and denote the Wasserstein distance term in Eq. (15) as follows:

$$J \coloneqq \frac{1}{T}\sum_{t=1}^{T} t W_q\left(\hat{P}^{adv}_{\delta_{t-1}}(h_{t-1}), \hat{P}^{adv}_{\delta_t}(h_t)\right),$$

where $\hat{P}^{adv}_{\delta_t}(h_t)$ denotes the distribution of adversarial examples under the constraint set B_{δ_t}. The next question is, how can we schedule δ_t to minimize J? Note that δ_T is fixed to be the perturbation radius δ_{test} of the test stage. The visualization of the radius constraint can be found in the Appendix.

To better understand the term J, we convert it into an optimizable form. In curriculum adversarial training, the overall trend of $W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P})$ is non-decreasing as t increases, which has been experimentally verified (see Figure 2). Since the Wasserstein distance is proven to satisfy the triangle inequality in information theory literature (Clement and Desch 2008), we use the original distribution $\hat{P}$ to connect the adjacent adversarial distributions, as follows: $W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}) - W_q(\hat{P}^{adv}_{\delta_{t-1}}(h_{t-1}), \hat{P}) \le W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}^{adv}_{\delta_{t-1}}(h_{t-1})) \le W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}) + W_q(\hat{P}^{adv}_{\delta_{t-1}}(h_{t-1}), \hat{P})$. Using a coefficient term k_t, we further rewrite the inequality as $W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}^{adv}_{\delta_{t-1}}(h_{t-1})) = W_q(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}) + k_{t-1} W_q(\hat{P}^{adv}_{\delta_{t-1}}(h_{t-1}), \hat{P})$, where $k_{t-1} \in [-1, +1]$ reflects the volatility of the adversarial distributions and the model. k_t tends to $+1$ when the distance between the adversarial distributions of adjacent iterations is relatively large, which means that the model fluctuates. In contrast, k_t tends to -1 when the adversarial distributions and the model tend to converge. We visualize the triangle inequalities with different values of k_t in the Appendix. By substituting this into J, we can rewrite J as: $J = W_q\left(\hat{P}^{adv}_{\delta_T}(h_T), \hat{P}\right) + \frac{1}{T} k_0 W_q\left(\hat{P}^{adv}_{\delta_0}(h_0), \hat{P}\right) + \frac{1}{T}\sum_{t=1}^{T-1}\left((t+1)k_t + t\right) W_q\left(\hat{P}^{adv}_{\delta_t}(h_t), \hat{P}\right)$.

To minimize the term J, we can adaptively control each Wasserstein distance according to the value of k_t. We observe in the experiments that the adversarial examples are generated very close to the border of the perturbation set, i.e. $\|x - x^{adv}\|_p \approx \delta$. According to the definition of Wasserstein distance, the following formula holds for any $h \in \mathcal{H}$: $W_q\left(\hat{P}^{adv}_{\delta_t}(h), \hat{P}\right) \approx \delta_t$ (Lemma 2 of (Tu, Zhang, and Tao 2019)). So the term J can be further converted as:

$$J \approx \delta_T + \frac{1}{T}\sum_{t=1}^{T-1}((t+1)k_t + t)\delta_t + \frac{1}{T}k_0\delta_0. \tag{16}$$

Since J is associated with δ_t, we can control the radius δ_t to minimize J in Eq. (16):

1) when $k_t > -\frac{t}{t+1}$, which means $(t+1)k_t + t > 0$, reducing the radius δ_t reduces J. As verified in later experiments, k_t is large during the early stage; hence, constraining the radius at the beginning improves the robustness of the trained model.
2) when $k_t < -\frac{t}{t+1}$, which means $(t+1)k_t + t < 0$, increasing the radius δ_t reduces J. A small k_t indicates that the model is nearly convergent with radius δ_t, which implies that, after the model tends to converge, we should increase the radius δ_t.

In the works of curriculum adversarial training (Cai, Liu, and Song 2018; Wang et al. 2019; Kinfu and Vidal 2022), the researchers find that gradually increasing the hardness of the adversarial examples is beneficial to the training. Our theory provides a theoretical explanation for the efficacy of this strategy.

5 Experiments

From the theoretical results in Section 4.3, we can observe that the terms W_q, k_t and J play important roles in adversarial training. In this section, we use the CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009) datasets to track the changes in these terms during the training process and verify our theoretical results. Our code is attached to the supplementary material.

5.1 Setup

Baselines. The baselines used in the experiments are as follows: conventional adversarial training with PGD attacker (Madry) (Madry et al. 2018) and Friendly Adversarial Training (FAT) (Zhang et al. 2020). FAT is a typical curriculum adversarial training method that achieves better performance than Madry. The details of this method can be found in the Appendix. We compare the term J between these methods and account for the effectiveness of curriculum adversarial training methods using our theory.

Networks. Following (Zhang et al. 2020), we use ResNet-18 (He et al. 2016) and Wide ResNet (WRN-34-10)

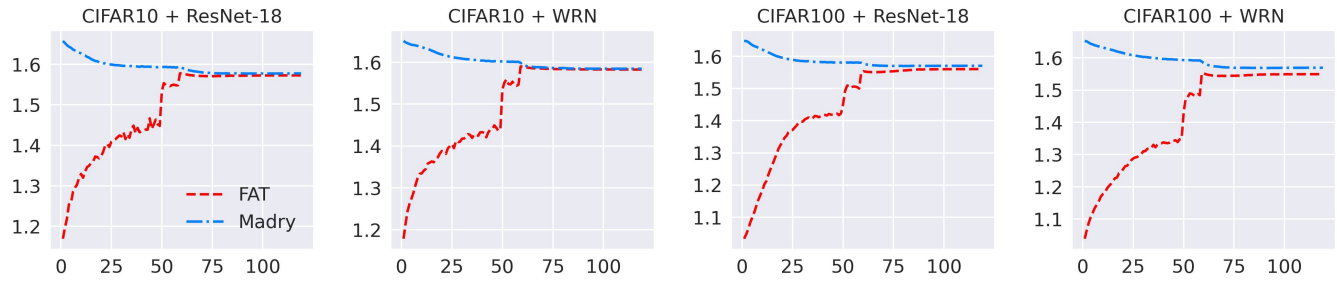

Figure 2: Changes in the term $W_1(\hat{P}^{adv}(h_t), \hat{P})$ for the (curriculum) adversarial training methods during the training process.

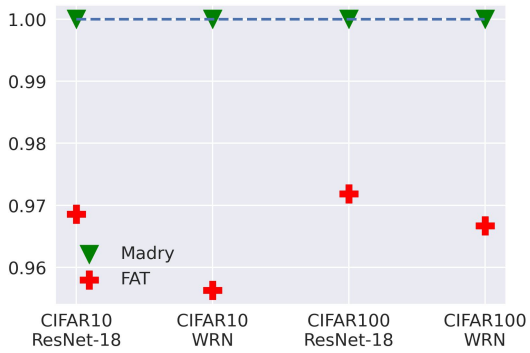

Figure 3: The values of term J for the three methods at the end of the training process. Since the values for different groups (different networks and datasets) have different ranges, we normalize the values for each group by dividing by the maximum value in the group.

(Zagoruyko and Komodakis 2016) for both CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009) datasets.

Parameters. For all baselines, we run projected gradient descent (PGD) as our adversary, with a step size of 0.007. The maximum step of PGD is 20, and the maximum radius of the l_∞-norm bounded perturbation is $\delta = 0.031$. Following Zhang et al. (2020), the models are trained using stochastic gradient descent (SGD) with momentum of 0.9 for 120 epochs. The initial learning rate is 0.1, reduced to 0.01, 0.001, and 0.0005 at epoch 60, 90, and 110, respectively. The batch size is 128. For parameters unique to each method, we use the default values in their papers.

Calculation of Wasserstein Distance. The Wasserstein distance is defined in Definition 3.3. Since the optimal transport solution is difficult to find, researchers often use Kantorovich-Rubinstein duality to approximately calculate the Wasserstein distance between two distributions (Arjovsky, Chintala, and Bottou 2017). However, this method is both time-consuming and unstable (Gulrajani et al. 2017). Fortunately, we can calculate the Wasserstein distance directly, since the mapping T_h defined in Section 3.2 is indeed the optimal transport between distributions $\hat{P}$ and $\hat{P}^{adv}(h)$. Hence, we obtain $W_1(\hat{P}^{adv}(h), \hat{P}) = \frac{1}{n}\sum_{i=1}^{n} \|x_i^{adv} - x_i\|$, where x_i^{adv} is the adversarial example of x_i according to h.

5.2 Results and Analysis

The term k_t reflects the volatility of the training process. Due to space limitations, we present the changes in the term k_t during the training process in the Appendix. As can be seen from the figures, the k_t values of all three methods decrease gradually in most cases. As analyzed in Section 4.3, when k_t is large at the beginning of the training process, we can control the radius δ_t to reduce J. On the other hand, when k_t decreases at the end of the training process, increasing the radius δ_t will reduce J. In practice, the curriculum adversarial training method FAT gradually moves the adversarial examples further away from the natural examples, making them more difficult to learn. To visualize this strategy, we track the changes in $W_1(\hat{P}^{adv}(h_t), \hat{P})$ during training in Figure 2, since $W_q\left(\hat{P}_{\delta_t}^{adv}(h), \hat{P}\right)$ approximates δ_t (Lemma 2 of Tu, Zhang, and Tao (2019)). As the figure shows, FAT keeps the Wasserstein distance relatively small in the early stage of the training process, which is consistent with our analysis at the end of Section 4.3.

We compare the values of the term J at the end of the training process for the methods in Figure 3. Compared to the Madry, FAT achieves smaller J, no matter which dataset and neural network we use. According to our theoretical results in Theorem 4.5 and Theorem 4.13, a smaller J improves generalization performance. Our theoretical analysis thus explains the efficacy of curriculum adversarial training methods, which gradually increases the hardness of the adversarial examples. More experiments of SVHN (Netzer et al. 2011) and Tiny-ImageNet (Le and Yang 2015) datasets can be found in the Appendix, which consistently verify our theory.

6 Conclusion

In this work, we study the generalization performance of adversarial training from an online perspective. We first propose a generalization error bound via Rademacher complexity, then introduce the time series prediction framework to derive an improved error bound. Our theoretical results further explain why curriculum adversarial training methods can improve generalization performance. Extensive experiments verify our theoretical findings.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant 2023YFC3604702, the National Natural Science Foundation of China under Grant 61976161, the Fundamental Research Funds for the Central Universities under Grant 2042022rc0016.

References

Allen-Zhu, Z.; and Li, Y. 2021. Feature Purification: How Adversarial Training Performs Robust Deep Learning. In *FOCS*, 977–988.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

Attias, I.; Kontorovich, A.; and Mansour, Y. 2019. Improved Generalization Bounds for Robust Learning. In *ALT*, 162–183.

Awasthi, P.; Frank, N.; and Mohri, M. 2020. Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks. In *ICML*, volume 119, 431–441.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3: 463–482.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *ICML*, 41–48.

Cai, Q.; Liu, C.; and Song, D. 2018. Curriculum Adversarial Training. In *IJCAI*, 3740–3747.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *SP*, 39–57.

Chen, Y.; and Liu, W. 2023. A Theory of Transfer-Based Black-Box Attacks: Explanation and Implications. In *NeurIPS*.

Cheng, M.; Lei, Q.; Chen, P.; Dhillon, I. S.; and Hsieh, C. 2022. CAT: Customized Adversarial Training for Improved Robustness. In *IJCAI*, 673–679.

Clement, P.; and Desch, W. 2008. An Elementary Proof of the Triangle Inequality for the Wasserstein Metric. *Proceedings of the American Mathematical Society*, 136(1): 333–339.

Cullina, D.; Bhagoji, A. N.; and Mittal, P. 2018. PAC-learning in the presence of adversaries. In *NeurIPS*, 228–239.

Gao, Q.; and Wang, X. 2021. Theoretical investigation of generalization bounds for adversarial learning of deep neural networks. *Journal of Statistical Theory and Practice*, 15(2): 51.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

Gowal, S.; Rebuffi, S.; Wiles, O.; Stimberg, F.; Calian, D. A.; and Mann, T. A. 2021. Improving Robustness using Generated Data. In *NeurIPS*, 4218–4233.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved Training of Wasserstein GANs. In *NeurIPS*, 5767–5777.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Khim, J.; and Loh, P. 2018. Adversarial Risk Bounds for Binary Classification via Function Transformation. *arXiv preprint arXiv:1810.09519*.

Kinfu, K. A.; and Vidal, R. 2022. Analysis and Extensions of Adversarial Training for Video Classification. In *CVPR*, 3416–3425.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial examples in the physical world. In *ICLR*.

Kuznetsov, V.; and Mohri, M. 2014. Generalization Bounds for Time Series Prediction with Non-stationary Processes. In *ALT*, 260–274.

Kuznetsov, V.; and Mohri, M. 2015. Learning Theory and Algorithms for Forecasting Non-stationary Time Series. In *NeurIPS*, 541–549.

Kuznetsov, V.; and Mohri, M. 2016. Time series prediction and online learning. In *COLT*, 1190–1213.

Kuznetsov, V.; and Mohri, M. 2020. Discrepancy-Based Theory and Algorithms for Forecasting Non-Stationary Time Series. *Annals of Mathematics and Artificial Intelligence*, 88(4): 367–399.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, B.; and Liu, W. 2023. WAT: Improve the Worst-Class Robustness in Adversarial Training. In *AAAI*, 14982–14990.

Li, X.; Xin, Z.; and Liu, W. 2022. Defending Against Adversarial Attacks via Neural Dynamic System. In *NeurIPS*.

Ma, X.; Wang, Z.; and Liu, W. 2022. On the Tradeoff Between Robustness and Fairness. In *NeurIPS*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

Montasser, O.; Hanneke, S.; and Srebro, N. 2019. VC Classes are Adversarially Robustly Learnable, but Only Improperly. In *COLT*, 2512–2530.

Mustafa, W.; Lei, Y.; and Kloft, M. 2022. On the Generalization Analysis of Adversarial Learning. In *ICML*, 16174–16196.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Pang, T.; Lin, M.; Yang, X.; Zhu, J.; and Yan, S. 2022. Robustness and Accuracy Could Be Reconcilable by (Proper) Definition. In *ICML*, 17258–17277.

Rakhlin, A.; Sridharan, K.; and Tewari, A. 2010. Online Learning: Random Averages, Combinatorial Parameters, and Learnability. In *NeurIPS*, 1984–1992.

Rakhlin, A.; Sridharan, K.; and Tewari, A. 2015. Online learning via sequential complexities. *Journal of Machine Learning Research*, 16: 155–186.

Rebuffi, S.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. A. 2021. Fixing Data Augmentation to Improve Adversarial Robustness. *arXiv preprint arXiv:2103.01946*.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do Adversarially Robust ImageNet Models Transfer Better? In *NeurIPS*.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shi, L.; and Liu, W. 2023. Adversarial Self-Training Improves Robustness and Generalization for Gradual Domain Adaptation. In *NeurIPS*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

Tu, Z.; Zhang, J.; and Tao, D. 2019. Theoretical Analysis of Adversarial Learning: A Minimax Approach. In *NeurIPS*, 12259–12269.

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

Wang, Y.; Ma, X.; Bailey, J.; Yi, J.; Zhou, B.; and Gu, Q. 2019. On the Convergence and Robustness of Adversarial Training. In *ICML*, 6586–6595.

Wang, Z.; and Liu, W. 2022. Robustness Verification for Contrastive Learning. In *ICML*, 22865–22883.

Wang, Z.; Pang, T.; Du, C.; Lin, M.; Liu, W.; and Yan, S. 2023. Better Diffusion Models Further Improve Adversarial Training. In *ICML*, 36246–36263.

Xiao, J.; Fan, Y.; Sun, R.; and Luo, Z. 2022a. Adversarial Rademacher Complexity of Deep Neural Networks. *arXiv preprint arXiv:2211.14966*.

Xiao, J.; Fan, Y.; Sun, R.; Wang, J.; and Luo, Z. 2022b. Stability Analysis and Generalization Bounds of Adversarial Training. *arXiv preprint arXiv:2210.00960*.

Xiao, J.; Qin, Z.; Fan, Y.; Wu, B.; Wang, J.; and Luo, Z. 2022c. Adaptive Smoothness-weighted Adversarial Training for Multiple Perturbations with Its Stability Analysis. *arXiv preprint arXiv:2210.00557*.

Xing, Y.; Song, Q.; and Cheng, G. 2021a. On the Algorithmic Stability of Adversarial Training. In *NeurIPS*, 26523–26535.

Xing, Y.; Song, Q.; and Cheng, G. 2021b. On the Generalization Properties of Adversarial Training. In *AISTATS*, 505–513.

Xu, J.; and Liu, W. 2022. On Robust Multiclass Learnability. In *NeurIPS*.

Yin, D.; Ramchandran, K.; and Bartlett, P. L. 2019. Rademacher Complexity for Adversarially Robust Generalization. In *ICML*, 7085–7094.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *ICML*, 7472–7482.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. S. 2020. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *ICML*, 11278–11287.

Zhou, Z.; and Liu, W. 2023. Sample Complexity for Distributionally Robust Learning under chi-square divergence. *Journal of Machine Learning Research*, 1–27.

Zou, X.; and Liu, W. 2023. Generalization Bounds for Adversarial Contrastive Learning. *Journal of Machine Learning Research*, 114:1–114:54.