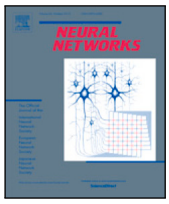
Neural Networks 183 (2025) 106983

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Data-dependent stability analysis of adversarial training

Yihan Wang, Shuang Liu, Xiao-Shan Gao *

Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, China
University of Chinese Academy of Sciences, Beijing, 101408, China

ARTICLE INFO

Keywords:
On-average stability analysis
Generalization bound
Adversarial training
Stochastic gradient descent
Data poisoning attack

ABSTRACT

Stability analysis is an essential aspect of studying the generalization ability of deep learning, as it involves deriving generalization bounds for stochastic gradient descent-based training algorithms. Adversarial training is the most widely used defense against adversarial attacks. However, previous generalization bounds for adversarial training have not included information regarding data distribution. In this paper, we fill this gap by providing generalization bounds for stochastic gradient descent-based adversarial training that incorporate data distribution information. We utilize the concepts of on-average stability and high-order approximate Lipschitz conditions to examine how changes in data distribution and adversarial budget can affect robust generalization gaps. Our derived generalization bounds for both convex and non-convex losses are at least as good as the uniform stability-based counterparts which do not include data distribution information. Furthermore, our findings demonstrate how distribution shifts from data poisoning attacks can impact robust generalization.

1. Introduction

Deep learning models acquire knowledge from training data and generalize to unseen data. Generalization plays a key role in successful machine learning algorithms. On the other hand, a neural network can easily be fooled by adversarial examples (Goodfellow, Shlens, & Szegedy, 2014; Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, & Fergus, 2013). Although adversarial training (Madry, Makelov, Schmidt, Tsipras, & Vladu, 2017) can largely alleviate the adversarial vulnerability of networks, the corresponding robust generalization is more difficult and robust overfitting (Rice, Wong, & Kolter, 2020) harms the robust performance to a very large degree. To understand the generalization ability of adversarial training, an important research direction is to give a theoretical analysis of its generalization bound, i.e., the difference between adversarial population risk and adversarial empirical risk (see Section 3.1).

Stability analysis is a main methodology for obtaining algorithm-dependent generalization bounds (Bousquet & Elisseeff, 2002; Shalev-Shwartz, Shamir, Srebro, & Sridharan, 2010). In standard training, the uniform stability of stochastic gradient descent (SGD) for deep learning was established (Bassily, Feldman, Guzmán, & Talwar, 2020; Hardt, Recht, & Singer, 2016). In Kuzborskij and Lampert (2018), on-average stability is used to provide data-dependent generalization bounds for standard training. The uniform stability of adversarial training was presented in Farnia and Ozdaglar (2021) assuming the inner maximization problem is strongly concave and in Xing, Song, and Cheng (2021) for non-smooth losses. Under the η-approximate β-gradient Lipschitz assumption, generalization bounds for SGD in adversarial training were derived (Xiao, Fan, Sun, Wang, & Luo, 2022).

It is generally believed that the difficulty of robust generalization involves three aspects, including model capacity, training algorithm, and data distribution. The capacity of a strictly robust classifier on a well-separated distribution should be exponential in the data dimension (Li, Jin, Zhong, Hopcroft, & Wang, 2022). The previous generalization bounds based on uniform stability analyses (Xiao et al., 2022; Xing et al., 2021) of adversarial training algorithms did not contain information about data distribution.

In this paper, we provide on-average stability analysis (see Definition 1) of SGD-based adversarial training and derive data-dependent generalization bounds to illustrate robust generalization, that is, the generalization bounds contain information of the data distribution. For convex adversarial losses (see Section 4.3), assuming that the losses are Lipschitz and approximately gradient Lipschitz, we give a generalization bound dependent on the adversarial population risk at the initialization point and the variance of stochastic gradients over the distribution. Assuming the losses are approximately Hessian Lipschitz in addition, we provide a generalization bound for the non-convex adversarial losses (see Section 4.4). Besides the variance of stochastic gradients over the distribution, this bound depends on the curvature (the norm of the Hessian matrix) at the initialization point and the population risk at the output parameters. Our bounds grow

* Corresponding author at: Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100190, China.
E-mail address: xgao@mmrc.iss.ac.cn (X.-S. Gao).

https://doi.org/10.1016/j.neunet.2024.106983
Received 29 February 2024; Received in revised form 10 September 2024; Accepted 26 November 2024
Available online 4 December 2024

with the adversarial training budget and cover the standard training setting when the budget becomes zero. Our bounds for both convex and non-convex losses are no worse than the uniform stability-based counterparts but capture the information about the data distribution and the initialization point.

An additional advantage of our generalization bound over the previous ones is that it describes the effects of distribution shifts caused by data poisoning attacks and hence interprets the shrinkage of generalization gaps in adversarial training under stability attacks since the poisoned distributions can reduce the adversarial population risk over the poisoned data.

The rest of the paper is organized as follows. In Section 3, we revisit the relationship between stability and robust generalization for adversarial training. In Section 4, we provide our main results. In Section 5, we present experimental results to verify the theoretical results.

2. Related work

Robust Generalization. Machine learning models are highly vulnerable to adversarial examples (Biggio, Corona, Maiorca, Nelson, Šrndić, Laskov, Giacinto, & Roli, 2013; Moosavi-Dezfooli, Fawzi, & Frossard, 2016; Nguyen, Yosinski, & Clune, 2015; Szegedy et al., 2013), where crafted and imperceptible perturbations to input data can easily fool a well-trained classifier. A widely adopted illustration attributes adversarial examples to the presence of non-robust features (Ilyas et al., 2019). Among the numerous defenses proposed against adversarial attacks, adversarial training (Goodfellow et al., 2014; Madry et al., 2017; Shaham, Yamada, & Negahban, 2015) has become a major approach to training a robust deep neural network and can achieve optimal robust accuracy if certain loss functions are used (Gao, Liu, & Yu, 2022).

Generalization in adversarial training is much more tricky than that in standard training and requires more data and larger models (Gowal et al., 2021; Li et al., 2022; Schmidt, Santurkar, Tsipras, Talwar, & Madry, 2018; Wang et al., 2023). The robust overfitting phenomenon harms the robustness in a long training procedure (Rice et al., 2020). In recent years, different methods have been proposed to alleviate robust overfitting (Chen, Liu, et al., 2020; Chen, Zhang, Liu, Chang, & Wang, 2020; Chen, Zhang, Wang, Balachandra, Ma, Wang, & Wang, 2022; Wu, Xia, & Wang, 2020; Yu & Gao, 2023; Yu, Han, et al., 2022) .

Algorithmic Stability. Modern stability analysis goes back to the work (Bousquet & Elisseeff, 2002). Stability notations fall into two categories: data-free and data-dependent ones. The first category is usually called uniform stability. Generalization bounds of SGD were first given using uniform stability under Lipschitz and smoothness conditions (Hardt et al., 2016), which was extended to the non-smooth convex case (Bassily et al., 2020). The uniform stability of adversarial training has been reported in Xiao et al. (2022), Xing et al. (2021). The data-dependent stability (Kuzborskij & Lampert, 2018) employing the notion of on-average stability (Shalev-Shwartz et al., 2010) focused on the stability of SGD-based standard training under the data distribution given an initialization point. Later, Lemire Paquin, Chaib-draa, and Giguère (2022) proved generalization bounds for SGD algorithms with normalized losses that appear in linear classifiers and homogeneous neural networks.

Data Poisoning. As defensive strategies against unauthorized exploitation of personal data, availability attacks (Feng, Cai, & Zhou, 2019; Fowl et al., 2021; Huang, Ma, Erfani, Bailey, & Wang, 2021) imperceptibly perturb training data so that trained models learn nothing useful and become futile. Adversarial training can mitigate such availability attacks (Tao, Feng, Yi, Huang, & Chen, 2021). Stability attacks (Fu, He, Liu, Shen, & Tao, 2021; Tao et al., 2022; Wang, Wang, & Wang, 2021; Wen, Zhao, Liu, Backes, Wang, & Zhang, 2023) have been proposed to come through adversarial training and result in a large degradation in the robust test performance.

The shortcut interpretation (Yu, Zhang, Chen, Yin & Liu, 2022) suggests that stability poisoning attacks root "easy-to-learn" features in the poisoned training data. However, these features do not appear in clean data. Our generalization bound can be used to interpret the shrinkage of generalization gaps in adversarial training under stability attacks, since it contains information of the data distribution.

3. Preliminaries

In this section, we revisit the robust generalization gap and the on-average stability analysis.

3.1. Robust generalization gap

Let D be a data distribution over an image classification data space $\mathbb{D} = [0,1]^d \times [m]$, where $[0,1]^d$ contains the image space and $[m] = \{1,\dots,m\}$ is the label set. A dataset S of n samples is drawn i.i.d. from D and is denoted by $S \sim D^n$. Given a network with parameters θ and a non-negative loss function $l(\theta, z) : \mathbb{R}^k \times \mathbb{D} \to \mathbb{R}_{\geq 0}$, the standard training minimizes the *empirical risk* $\mathbb{E}_{z\in S}\, l(\theta, z)$ with SGD.

Adversarial Training. As a major defense approach, adversarial training (Madry et al., 2017) refers to a bi-level optimization, of which the inner maximization iteratively searches for the strongest perturbation inside a L_p-norm ball and the outer minimization optimizes the model via the loss on the perturbed data. Formally, given an adversarial budget ϵ, the adversarial training uses the *adversarial loss*:

$$h(\theta, z) = \max_{z' \in \mathcal{B}_\epsilon(z)} l(z', \theta),$$

where $\mathcal{B}_\epsilon(z) = \{z' \in \mathbb{D} : \|z' - z\|_p \leq \epsilon\}$ and $p \in \mathbb{N} \cup \{\infty\}$. Here, the p-norm is for the image part of z. When $\epsilon = 0$, we have $h = l$ and adversarial training reduces to standard training. The *adversarial population risk* and *adversarial empirical risk* are respectively defined as

$$\mathcal{R}_D(\theta) = \mathbb{E}_{z\sim D}[h(\theta, z)] \quad \text{and} \quad \mathcal{R}_S(\theta) = \mathbb{E}_{z\in S}[h(\theta, z)].$$

We denote the SGD algorithm of adversarial training by $\mathcal{A}$, which inputs a training set S and outputs a parameter set $\mathcal{A}(S)$ of a network through minimizing the adversarial empirical risk $\mathcal{R}_S$.

Robust Generalization Gap. Let $\theta^*, \bar{\theta}$ be the optimal solutions of learning over D and S, namely minimizing $\mathcal{R}_D(\theta)$ and $\mathcal{R}_S(\theta)$, respectively. Then, for the output $\hat{\theta} = \mathcal{A}(S)$ of algorithm $\mathcal{A}$, the excess risk can be decomposed as

$$\begin{aligned}\mathcal{R}_D(\hat{\theta}) - \mathcal{R}_D(\theta^*) = &\underbrace{\mathcal{R}_D(\hat{\theta}) - \mathcal{R}_S(\hat{\theta})}_{\epsilon_{\text{gen}}} + \underbrace{\mathcal{R}_S(\hat{\theta}) - \mathcal{R}_S(\bar{\theta})}_{\epsilon_{\text{opt}}} \\ &+ \underbrace{\mathcal{R}_S(\bar{\theta}) - \mathcal{R}_S(\theta^*)}_{\leq 0} + \underbrace{\mathcal{R}_S(\theta^*) - \mathcal{R}_D(\theta^*)}_{\mathbb{E}=0}.\end{aligned}$$

To control the excess risk, we need to control the *robust generalization gap* ϵ_{gen} and the *robust optimization gap* ϵ_{opt}. The robust optimization gap in adversarial training has been studied a lot theoretically (Nemirovski, Juditsky, Lan, & Shapiro, 2009; Xiao et al., 2022). In addition, empirical results (Madry et al., 2017; Wang et al., 2019; Wu et al., 2020; Zhang et al., 2019) present narrow robust optimization gaps.

On the other hand, robust overfitting (Rice et al., 2020) is a dominant phenomenon in adversarial training that hinders deep neural networks from attaining high robust performance. Hence, we focus on the robust generalization gap ϵ_{gen} in this paper and an upper bound for ϵ_{gen} is called a *generalization bound*.

3.2. On-average stability

In order to analyze the data-dependent stability, we employ the notion of on-average stability. Given a dataset $S = \{z_1, \dots, z_n\} \sim D^n$ and $z \sim D$, replacing z_i in S with z, we denote $S^{i,z} = \{z_1, \dots, z_{i-1}, z, z_{i+1}, \dots, z_n\}$ with $i \in [n]$.

Definition 1 (*On-Average Stability*). A randomized algorithm $\mathcal{A}$ is ε-*on-average stable* if

$$\sup_{i\in[n]} \mathbb{E}_{S,z,\mathcal{A}}[h(\mathcal{A}(S), z) - h(\mathcal{A}(S^{i,z}), z)] \le \varepsilon, \tag{1}$$

where $S \sim \mathcal{D}^n$, $z \sim \mathcal{D}$, and ε can depend on the data distribution $\mathcal{D}$ and the initialization point of $\mathcal{A}$.

The on-average stability considers the expected difference between the losses of algorithm outputs on S and its replace-one-example version for all replacement index i. The on-average stability derives the generalization bound as follows.

Theorem 2 (*Kuzborskij & Lampert, 2018*). *If $\mathcal{A}$ is ε-on-average stable, then the robust generalization gap of $\mathcal{A}$ is bounded by ε: $\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_{\mathcal{D}}(\mathcal{A}(S)) - \mathcal{R}_S(\mathcal{A}(S))] \le \varepsilon$, that is, the generalization bound of $\mathcal{A}$ is ϵ.*

4. Theoretical results

In this section, we give the data-dependent stability analysis of adversarial training for both convex and non-convex adversarial losses. We provide proof sketches of our results, and the full proofs are placed in Appendix A.

4.1. Lipschitz conditions

Stability analysis always relies on some Lipschitz conditions. The loss function is assumed to be L-Lipschitz and β-gradient Lipschitz, which is called β-smooth in the work (Hardt et al., 2016). For adversarial training, we need the adversarial loss $h(\theta, z)$ to satisfy some Lipschitz conditions. It is not reasonable to directly endow h with Lipschitz conditions, since $h(\theta, z)$ takes the maximum of $l(\theta, z')$ with $z' \in \mathcal{B}_\epsilon(z)$. Instead, we assume that the original loss function $l(\theta, z)$ satisfies the following Lipschitz conditions. Let $\|\cdot\|_p$ be the p-norm of vectors or matrices and we write $\|\cdot\|$ instead of $\|\cdot\|_2$ for brevity. In this paper, ∇ is the abbreviation for ∇_θ.

Assumption 3. The loss l is L-Lipschitz in θ:

$$\|l(\theta_1, z) - l(\theta_2, z)\| \le L\|\theta_1 - \theta_2\|.$$

Assumption 4. The loss l is L_θ-gradient Lipschitz in θ and L_z-gradient Lipschitz in z:

$$\|\nabla l(\theta_1, z) - \nabla l(\theta_2, z)\| \le L_\theta\|\theta_1 - \theta_2\|,$$
$$\|\nabla l(\theta, z_1) - \nabla l(\theta, z_2)\| \le L_z\|z_1 - z_2\|_p.$$

Assumption 5. The loss l is H_θ-Hessian Lipschitz in θ and H_z-Hessian Lipschitz in z:

$$\|\nabla^2 l(\theta_1, z) - \nabla^2 l(\theta_2, z)\| \le H_\theta\|\theta_1 - \theta_2\|,$$
$$\|\nabla^2 l(\theta, z_1) - \nabla^2 l(\theta, z_2)\| \le H_z\|z_1 - z_2\|_p.$$

Remark 6. For commonly used losses and ReLU-based networks, Assumption 3 is valid (Gao et al., 2022). The gradient Lipschitz conditions (Lipschitz smoothness) are often used in robustness analysis (Liu, Salzmann, Lin, Tomioka, & Süsstrunk, 2020; Sinha, Namkoong, Volpi, & Duchi, 2017; Xiao et al., 2022). Lipschitz Hessians are used in the analysis of SGD (Ge, Huang, Jin, & Yuan, 2015; Kuzborskij & Lampert, 2018). Assumptions 4 and 5 are valid for networks based on smooth activation functions such as Sigmoid and smooth loss functions such as cross-entropy (CE) and mean squared error (MSE). Related works on ReLU-based networks were given in Allen-Zhu, Li, and Song (2019), Du, Lee, Li, Wang, and Zhai (2019).

Note that the adversarial vulnerability of deep networks is rooted in the explosion of the Lipschitz constant of $l(\theta, z)$ in z. However, the zero-order Lipschitz constant in θ can be directly inherited by $h(\theta, z)$ (Liu et al., 2020). Additional Lipschitz conditions in z imply approximate gradient and Hessian Lipschitz conditions in θ which are needed for stability analysis.

Definition 7. Let $\eta, \beta, \nu, \rho > 0$ and $h(\theta)$ be a second-order differentiable function.

1. h is η-approximately β-gradient Lipschitz, if

$$\|\nabla h(\theta_1) - \nabla h(\theta_2)\| \le \beta\|\theta_1 - \theta_2\| + \eta.$$

2. h is ν-approximately ρ-Hessian Lipschitz, if

$$\|\nabla^2 h(\theta_1) - \nabla^2 h(\theta_2)\| \le \nu\|\theta_1 - \theta_2\| + \rho.$$

Lemma 8. *The adversarial loss $h(\theta, z)$ inherits (approximate) Lipschitz properties from the original loss $l(\theta, z)$.*

1. *Under Assumption 3, h is L-Lipschitz with respect to θ:*

$$\|h(\theta_1, z) - h(\theta_2, z)\| \le L\|\theta_1 - \theta_2\|.$$

2. *Under Assumption 4, h is $2\epsilon L_z$-approximately L_θ-gradient Lipschitz with respect to θ:*

$$\|\nabla h(\theta_1, z) - \nabla h(\theta_2, z)\| \le L_\theta\|\theta_1 - \theta_2\| + 2\epsilon L_z.$$

3. *Under Assumption 5, h is $2\epsilon H_z$-approximately H_θ-Hessian Lipschitz with respect to θ:*

$$\|\nabla^2 h(\theta_1, z) - \nabla^2 h(\theta_2, z)\| \le H_\theta\|\theta_1 - \theta_2\| + 2\epsilon H_z.$$

4.2. Preliminaries for analysis

We consider the SGD without replacement, that is, given a training set $S \sim \mathcal{D}^n$, algorithm $\mathcal{A}$ chooses a random permutation π over $[n] = \{1, \dots, n\}$ and cycles through S in the order determined by the permutation. If not mentioned otherwise, our analyses focus on the on-average stability of adversarial training in a single pass.

Suppose the update of $\mathcal{A}$ starts from an initialization point θ_1 and for $t \in [n]$,

$$\theta_{t+1} = G_{\mathcal{A}}(\theta_t, z_{\pi(t)}, \alpha_t),$$

where the permutation π depends on $\mathcal{A}$ and α_t is the tth step size. We update T steps in a single pass for $T \in [n]$ and analyze the on-average stability of the algorithm output $\mathcal{A}(S) = \theta_{T+1}$. We assume that the variance of stochastic gradients in $\mathcal{A}$ obeys

$$\mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_{\mathcal{D}}(\theta_t)\|^2] \le \sigma^2 \tag{2}$$

for all $t \in [T]$. The variance σ describes the distance between the stochastic gradient and the optimal gradient. Indeed, σ will change if the distribution $\mathcal{D}$ changes.

4.3. Convex adversarial losses

For convex adversarial losses, our analysis requires the approximate gradient Lipschitz assumption.

Theorem 9. *Assume that the adversarial loss $h(\theta, z)$ is non-negative, convex in θ, L-Lipschitz and η-approximately β-gradient Lipschitz with respect to θ. Let the step sizes $\alpha_t \le 1/\beta$. Then algorithm $\mathcal{A}$ is $\epsilon(\mathcal{D}, \theta_1)$-on-average stable with*

$$\epsilon(\mathcal{D}, \theta_1) = \Big(\frac{2\sigma L}{n} + L\eta\Big)\sum_{t=1}^{T}\alpha_t + \frac{4L}{n}\sqrt{\sum_{t=1}^{T}\alpha_t}$$

$$\cdot \sqrt{\mathcal{R}_{\mathcal{D}}(\theta_1) - \mathcal{R}_{\mathcal{D}}(\theta^*) + \frac{\beta\sigma^2}{2}\sum_{t=1}^{T}\alpha_t^2 + \eta L\sum_{t=1}^{T}\alpha_t}, \tag{3}$$

where θ_1 is the initialization point.

Proof sketch. Given a dataset $S = \{z_1, \ldots, z_n\} \sim \mathcal{D}^n$, an example $z \sim \mathcal{D}$, and an index $i \in [n]$, we denote $S^{i,z} = \{z'_1, \ldots, z'_n\}$ with $z'_j = z_j$ for $j \neq i$ and $z'_i = z$. Let θ_t, θ'_t be the tth outputs of $\mathcal{A}(S)$ and $\mathcal{A}(S^{i,z})$, respectively. Denote the distance of two trajectories at step t by $\delta_t(S, z, i, \mathcal{A}) = \|\theta_t - \theta'_t\|$. As both updates start from θ_1, we have $\delta_1(S, z, i, \mathcal{A}) = 0$. Denote $\Delta_t(S, z, i) = \mathbb{E}_{\mathcal{A}}[\delta_t(S, z, i, \mathcal{A})|\delta_{t_0}(S, z, i, \mathcal{A}) = 0]$. Lemma 17 (Lemma 5 in Kuzborskij & Lampert, 2018) tells us that

$$\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_t, z) - h(\theta'_t, z)] \leq L\mathbb{E}_{S,z}[\Delta_t(S, z, i)].$$

According to whether the algorithm meets the different sample with index i at step t, we derive the following recursion formula involving the adversarial budget $\eta = 2\epsilon L_z$,

$$\Delta_{t+1}(S, z, i) \leq \Delta_t(S, z, i) + (1 - \frac{1}{n})\alpha_t\eta + \frac{\alpha_t}{n}\mathbb{E}_{\mathcal{A}}[\|\nabla h(\theta_t, z_{\pi(t)})\| + \|\nabla h(\theta'_t, z'_{\pi(t)})\|].$$

By repeatedly applying Jensen's inequality, both expectations $\mathbb{E}_S[\sum_{t=1}^{T}\alpha_t\|\nabla h(\theta_t, z_{\pi(t)})\|]$ and $\mathbb{E}_{z,S}[\sum_{t=1}^{T}\alpha_t\|\nabla h(\theta'_t, z'_{\pi(t)})\|]$ have the same upper bound (Lemma 19)

$$\sum_{t=1}^{T}\sigma\alpha_t + 2\sqrt{\sum_{t=1}^{T}\alpha_t}\cdot\sqrt{r + \frac{\beta}{2}\sum_{t=1}^{T}\sigma^2\alpha_t^2 + \eta L\sum_{t=1}^{T}\alpha_t},$$

where $r = \mathcal{R}_{\mathcal{D}}(\theta_1) - \mathcal{R}_{\mathcal{D}}(\theta^*)$. Then, we can recursively bound $\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_t, z) - h(\theta'_t, z)]$ and prove the theorem.

Remark 10. By Theorem 2, Theorem 9 gives an upper bound $\epsilon(\mathcal{D}, \theta_1)$ for the robust generalization gap of algorithm $\mathcal{A}$, that is, $\epsilon(\mathcal{D}, \theta_1)$ is a stability generalization bound, which is also the case for Theorems 12 and 13.

When the step size is $\alpha_t = \mathcal{O}(\frac{1}{\sqrt{t}}) \leq \frac{1}{\beta}$ and the adversarial budget is $\epsilon = 0$, this bound reduces to the result in Kuzborskij and Lampert (2018). Now we fix step sizes to be constant and bound the adversarial loss gap between the initialization point and the optima via the Lipschitz condition.

Corollary 11. *Let the step size α_t be a constant $\alpha \leq 1/\beta$ and $r = \mathcal{R}_{\mathcal{D}}(\theta_1) - \mathcal{R}_{\mathcal{D}}(\theta^*)$. Then algorithm $\mathcal{A}$ is $\epsilon(\mathcal{D}, \theta_1)$-on-average stable with*

$$\epsilon(\mathcal{D}, \theta_1) = \eta\alpha LT + \frac{2\alpha LT}{n}(\sigma + \sqrt{2}\sigma + 2\sqrt{\eta L}) + \frac{4L\sqrt{\alpha rT}}{n}. \tag{4}$$

Comparison. We compare our result with existing results for adversarial training with convex adversarial losses in a single pass. For clarity, we take a constant step size α and use the $\mathcal{O}$ notation.

- Result of Xing et al. (2021):
$$\mathcal{O}(\alpha L^2\sqrt{T} + \frac{\alpha L^2 T}{n}). \tag{5}$$
- Result of Xiao et al. (2022):
$$\mathcal{O}(\eta\alpha LT + \frac{\alpha L^2 T}{n}). \tag{6}$$
- Our result:
$$\mathcal{O}(\eta\alpha TL + \frac{\alpha\sigma LT + \alpha\sqrt{\eta}L^{1.5}T + L\sqrt{\alpha rT}}{n}) \tag{7}$$

The smoothness of h is not required for the result of Xing et al. (2021). The bound (5) remains unchanged under changes in the adversarial training budget ϵ. Thus, this result does not capture the empirical observations that the robust overfitting phenomenon deteriorates as ϵ grows. The approximate smoothness of h is required for the result of Xiao et al. (2022). The bound (6) takes into account ϵ, i.e. $\eta = 2\epsilon L_z$ by the second statement in Lemma 8. However, this bound stays unchanged whenever the distribution shifts or the initialization point changes. Detailed discussion is shown in Appendix B.

Our bound grows with the adversarial training budget ϵ as well. In the general case, (6) and (7) are both $\mathcal{O}(T)$. When $\eta = 0$, our bound reduces to $\mathcal{O}(\frac{\alpha\sigma LT + L\sqrt{\alpha rT}}{n})$ which is the case for standard training. In the case where $\eta = 0$ and σ is negligible, our bound is dominated by the term $\frac{L\sqrt{\alpha rT}}{n}$ and becomes tighter than $\mathcal{O}(T)$ in Eqs. (5) and (6). Since r relies on θ_1 and $\mathcal{D}$, our result implies that a properly selected initialization point matters for robust generalization, and a potential distribution shift caused by some poisoning attack may affect robust generalization.

4.4. Non-convex adversarial losses

For non-convex adversarial losses, our analysis requires both approximate gradient Lipschitz and approximate Hessian Lipschitz assumptions.

Theorem 12. *Suppose the adversarial loss $h(\theta, z)$ is non-negative, L-Lipschitz, η-approximately β-gradient Lipschitz and ν-approximately ρ-Hessian Lipschitz with respect to θ. Let the step sizes $\alpha_t = \frac{c}{t}$ with $c \leq \min\{\frac{1}{\beta}, \frac{1}{4\beta\ln T}, \frac{1}{8(\beta\ln T)^2}\}$. Then $\mathcal{A}$ is $\epsilon(\mathcal{D}, \theta_1)$-on-average stable with*

$$\epsilon(\mathcal{D}, \theta_1) = \frac{1 + \frac{1}{c\gamma}}{n}(2cL^2 + nc\eta L)^{\frac{1}{1+c\gamma}}\cdot(\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_{\mathcal{D}}(\mathcal{A}(S))]T)^{\frac{c\gamma}{1+c\gamma}}, \tag{8}$$

where

$$\gamma = \min\{\beta, \tilde{\mathcal{O}}(\mathbb{E}_z[\|\nabla^2 h(\theta_1, z)\|] + \nu + \Delta^*)\},$$
$$\Delta^* = \rho(\sqrt{(\mathcal{R}_{\mathcal{D}}(\theta_1) - \mathcal{R}_{\mathcal{D}}(\theta^*))c} + c\sigma + c\sqrt{\eta L}).$$

Proof sketch. By Lemma 17 (Lemma 5 in Kuzborskij & Lampert, 2018), $\forall t_0 \in [n+1]$,

$$\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta'_{T+1}, z)] \leq L\mathbb{E}_{S,z}[\Delta_{T+1}(S, z, i)] + \frac{t_0 - 1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_{\mathcal{D}}(\theta_{T+1})].$$

The key is to recursively bound $\Delta_{T+1}(S, z, i)$. When the algorithm meets the different sample with index i at step t with probability $\frac{1}{n}$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta'_t)\| \leq \delta_t(S, z, i, \mathcal{A}) + 2\alpha_t L.$$

Otherwise, the second statement in Lemma 16 (from Xiao et al., 2022) implies

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta'_t)\| \leq (1 + \alpha_t\beta)\delta_t(S, z, i, \mathcal{A}) + \alpha_t\eta.$$

Additionally, in this case, Lemma 20 starts from Taylor expansion with integral remainder and exploits the approximate Hessian Lipschitz condition, deriving another bound as

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta'_t)\| \leq (1 + \alpha_t\xi_t(S, z, i, \mathcal{A}))\delta_t(S, z, i, \mathcal{A}),$$

where

$$\mathbb{E}_{S,z}[\xi_t(S, z, i, \mathcal{A})] = \tilde{\mathcal{O}}(\mathbb{E}_z[\|\nabla^2 h(\theta_1, z)\|] + \nu + \Delta^*).$$

Let $\psi_t(S, z, i) = \mathbb{E}_{\mathcal{A}}[\min\{\xi_t(S, z, i, \mathcal{A}), \beta\}]$ and we have

$$\Delta_{t+1}(S, z, i) \leq \frac{1}{n}(\Delta_t(S, z, i) + 2\alpha_t L) + (1 - \frac{1}{n})((1 + \alpha_t\psi_t(S, z, i))\Delta_t(S, z, i) + \alpha_t\eta).$$

Assigning proper step sizes α_t and leveraging Lemma 21, the on-average stability is given as

$$\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta'_{T+1}, z)] \leq (\frac{2L^2 + \eta nL}{2n\gamma})(\frac{T}{t_0 - 1})^{2c\gamma} + \frac{t_0 - 1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_{\mathcal{D}}(\theta_T)].$$

Then we take the optimal t_0 and prove the theorem.

From Eq. (8), we see that smaller γ yields higher stability. Note that γ is controlled by η and ν, the adversarial population risk at the initialization point, and the average Hessian norm of adversarial loss at the initialization point over the distribution.

Since SGD in a single pass is considered, we take $T \approx n$ and obtain that $\varepsilon(D,\theta_1) = \mathcal{O}(n^{-\frac{1}{1+c\gamma}})$ which can be improved to a more optimistic result $\mathcal{O}(n^{-1})$ when the adversarial empirical risk $\mathcal{R}_S(\mathcal{A}(S))$ becomes negligible according to Kuzborskij and Lampert (2018). Due to $\eta = 2\epsilon L_z$ and $\nu = 2\epsilon H_z$, a large adversarial budget ϵ makes the algorithm unstable and setting $\epsilon = 0$ derives the result for standard training. The gradient and Hessian Lipschitz constants L_z and H_z amplify the effect of ϵ and this explains why adversarial training appears to be more tricky than standard training and requires more training data (Gowal et al., 2021; Schmidt et al., 2018; Wang et al., 2023).

The initialization point is another factor that affects robust generalization. Intuitively, adversarial training prefers an initialization point naturally with low adversarial population risk which is close to the global optima. Furthermore, our result suggests that a proper selection of the initialization point should better have a low curvature over the distribution.

Comparison. Assume that the adversarial loss $h(\theta, z)$ is bounded in $[0, B]$ and $\alpha_t = \frac{c}{t}$ with $c \le \frac{1}{\beta}$. The result[1] of Xiao et al. (2022) for the non-convex case is

$$\frac{1+\frac{1}{c\beta}}{n}(2cL^2 + ncL)^{\frac{1}{1+c\beta}}(BT)^{\frac{c\beta}{1+c\beta}}. \tag{9}$$

Observe that (8) and (9) have similar forms. Nevertheless, (9) remains unchanged under data poisoning attacks. Our result replaces β with γ which captures much more information dependent on the initialization point, the loss function, and data distribution. Besides η, the approximation ν emphasizes the effect of ϵ again in our bound. Moreover, γ and c are bounded by β and $\frac{1}{\beta}$ respectively in (8). During training, the size of the dataset n is fixed and the term involving the training step T dominates the bound in Eq. (8), namely smaller γ means smaller $(\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\mathcal{A}(S))]T)^{\frac{c\gamma}{1+c\gamma}}$, and then a tighter bound. Thus, our result is no worse than (9).

Multiple-pass Case. Note that Eq. (8) holds within one pass through the training set. If we loosen some data-dependency requirements, say γ, the on-average stability analysis provides a result for the multiple-pass case.

Theorem 13 (*Multiple-pass Case*). *Assume the adversarial loss $h(\theta, z)$ is non-negative, convex in θ, L-Lipschitz and η-approximately β-gradient Lipschitz with respect to θ. Let the step sizes $\alpha_t \le \frac{c}{t}$ with $c \le \frac{1}{\beta}$. Then algorithm $\mathcal{A}$ is $\varepsilon(D,\theta_1)$-on-average stable with*

$$\varepsilon(D,\theta_1) = \frac{1+\frac{1}{c\beta}}{n}(2cL^2 + ncL)^{\frac{1}{1+c\beta}}(\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\mathcal{A}(S))]T)^{\frac{c\beta}{1+c\beta}}. \tag{10}$$

Both (8) and (10) contain the data-dependent factor $\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\mathcal{A}(S))]$ which can be much smaller than B in (9).

4.5. Poisoned generalization gap

To take a closer look at how changes in data distribution can affect robust generalization, we consider the distribution shift caused by a poisoning attack. A data poisoning attack $\mathcal{P}$ maps a distribution D to the poisoned distribution $\mathcal{P}_\# D$. Poisoning is usually constrained by a given *poisoning budget* ϵ' such that $\sup_{z\in\mathbb{D}} \|\mathcal{P}(z) - z\|_p \le \epsilon'$. The poisoned version of an algorithm $\mathcal{A}$ is denoted by $\mathcal{A}_\mathcal{P}$ which inputs $S \sim D^n$ and outputs $\mathcal{A}_\mathcal{P}(S) = \mathcal{A}(\mathcal{P}(S))$ by minimizing $\mathcal{R}_{\mathcal{P}(S)}(\theta)$. The robust generalization gap of $\mathcal{A}_\mathcal{P}(S)$ over the poisoned distribution $\mathcal{P}_\# D$ is called the *poisoned generalization gap*, denoted by $\varepsilon_\mathcal{P}$. That is,

$$|\mathbb{E}_{S,\mathcal{A}_\mathcal{P}}[\mathcal{R}_{\mathcal{P}_\# D}(\mathcal{A}_\mathcal{P}(S)) - \mathcal{R}_{\mathcal{P}(S)}(\mathcal{A}_\mathcal{P}(S))]| \le \varepsilon_\mathcal{P}. \tag{11}$$

[1] They reported a conservative result in the paper. Here we place their optimal result for comparison.

Influence of poisoning. Our data-dependent bounds in Eqs. (3) and (8) embody the influence of poisoning. When the distribution D is poisoned by $\mathcal{P}$, the bound $\varepsilon(D,\theta_1)$ becomes $\varepsilon(\mathcal{P}_\# D,\theta_1)$. The expected curvature at the initialization point becomes $\mathbb{E}_z[\|\nabla^2 h(\theta_1, \mathcal{P}(z))\|]$. The initial population risk gap becomes $\mathcal{R}_{\mathcal{P}_\# D}(\theta_1) - \mathcal{R}_{\mathcal{P}_\# D}(\theta^*_\mathcal{P})$, in which $\theta^*_\mathcal{P}$ is optimal with respect to $\mathcal{R}_{\mathcal{P}_\# D}$. Besides, the variance σ also depends on the poisoning and becomes $\sigma_\mathcal{P}$. Additionally, the adversarial population risk $\mathbb{E}_{S,\mathcal{A}_\mathcal{P}}[\mathcal{R}_{\mathcal{P}_\# D}(\mathcal{A}_\mathcal{P}(S))]$ in the poisoned counterparts of Eqs. (8) and (10) can be significantly influenced by poisoning. For non-convex losses, we specify the poisoned robust generalization bound in the following corollary:

Corollary 14. *With the same notations in Theorem 12, in the presence of poisoning $\mathcal{P}$, the poisoned algorithm $\mathcal{A}_\mathcal{P}$ is $\varepsilon(\mathcal{P}_\# D,\theta_1)$-on-average stable with*

$$\varepsilon(\mathcal{P}_\# D,\theta_1) = \frac{1+\frac{1}{c\gamma'}}{n}(2cL^2 + ncL)^{\frac{1}{1+c\gamma'}} \cdot (\mathbb{E}_{S,\mathcal{A}_\mathcal{P}}[\mathcal{R}_{\mathcal{P}_\# D}(\mathcal{A}_\mathcal{P}(S))]T)^{\frac{c\gamma'}{1+c\gamma'}}, \tag{12}$$

where

$$\gamma' = \min\{\beta, \tilde{\mathcal{O}}(\mathbb{E}_z[\|\nabla^2 h(\theta_1, \mathcal{P}(z))\|] + \nu + \Delta^{*\prime})\},$$
$$\Delta^{*\prime} = \rho(\sqrt{(\mathcal{R}_{\mathcal{P}_\# D}(\theta_1) - \mathcal{R}_{\mathcal{P}_\# D}(\theta^*_\mathcal{P}))c} + c\sigma_\mathcal{P} + c\sqrt{\eta L}).$$

5. Experiments

In this section, experiments are used to demonstrate the data-dependent stability of adversarial training and the advantages of our theoretical results. We adopt the L_∞ norm as constraints of imperceptible perturbations. The experimental setups and details are presented in Appendix C.

5.1. Robust generalization

We adversarially train ResNet-18 (He, Zhang, Ren, & Sun, 2016) on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton, et al., 2009), SVHN (Netzer et al., 2011), and Tiny-ImageNet (Le & Yang, 2015). Fig. 1 shows that the robust generalization is more difficult than the standard generalization, i.e. $\epsilon = 0$ as shown by Eqs. (4) and (8). The effect of even a small ϵ such as $2/255$ is amplified by the gradient and Hessian Lipschitz constants in z, namely L_z and H_z, and results in a large generalization gap. Moreover, the robust generalization gap increases with ϵ which implies that it is harder to ensure robustness in a broader area. Notice that Fig. 1 does not mean adversarial training with a larger budget leads to a less robust network, since here training and evaluation leverage the same PGD attack to approximate the adversarial loss with a given budget. Fig. 2 presents the robust overfitting phenomenon on the four datasets. When training errors converge to zero, the robust generalization gaps (blue lines) grow throughout the whole training procedure, while the robust test accuracy (red lines) increases in the first 100 epochs, decreases from the first learning rate decay at the 100-th epoch, and then jumps a little at the 150-th epoch before stabilizes.

5.2. Poisoned robust generalization

A poisoning attack is called a *stability attack* if the attack aims at destroying the robustness of a model, trained on the poisoned training set $\mathcal{P}(S) \sim \mathcal{P}_\# D^n$, on the original distribution D, i.e. $\mathcal{R}_D(\mathcal{A}_\mathcal{P}(S))$. Stability attacks employed in this paper include the error-minimizing noise (EM) (Huang et al., 2021), the robust error-minimizing noise (REM) (Fu et al., 2021), the adversarial poisoning (ADV) (Fowl et al., 2021), the hypocritical perturbation (HYP) (Tao et al., 2022) and the class-wise random noise (RAN). We poison both training and test sets

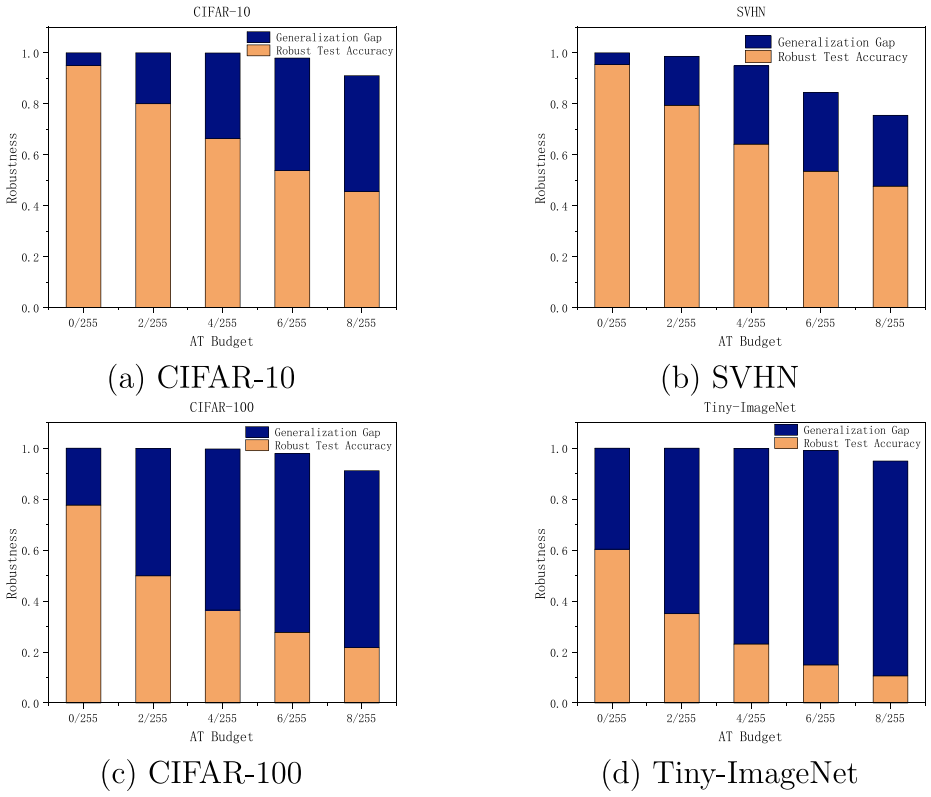


Fig. 1. The robust performance of adversarial training with different AT budgets ϵ ranging from 0 to 8/255. The adversarial training and robust evaluation leverage PGD-10 attack with the same budget ϵ.

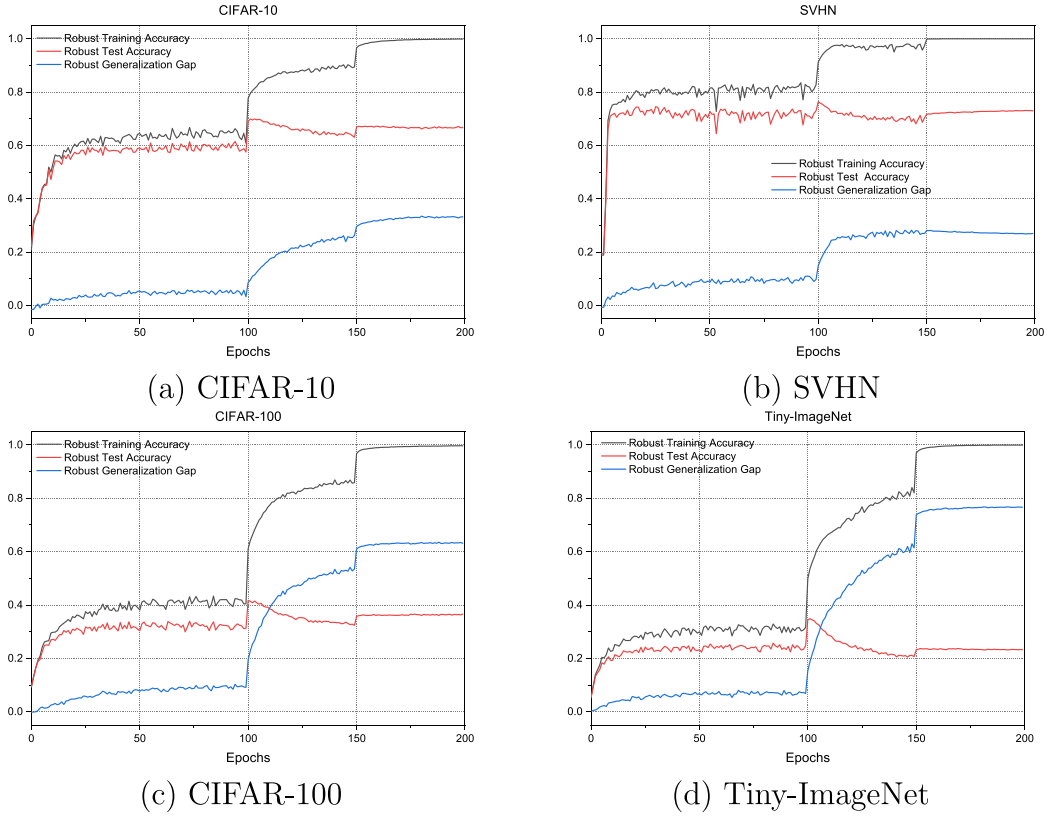
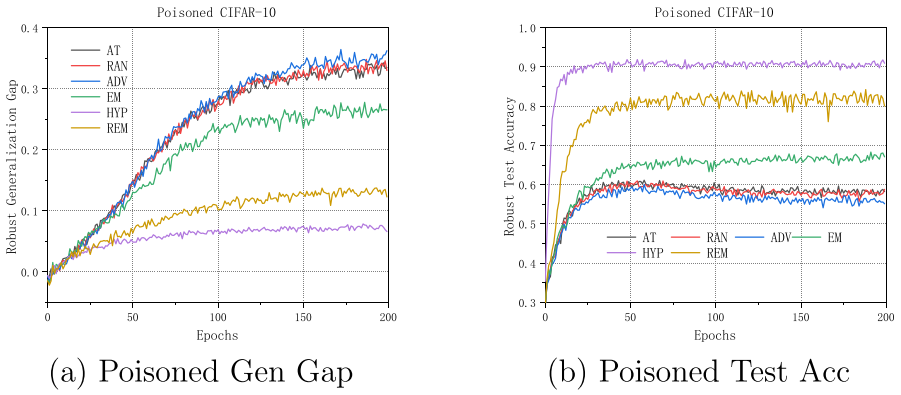
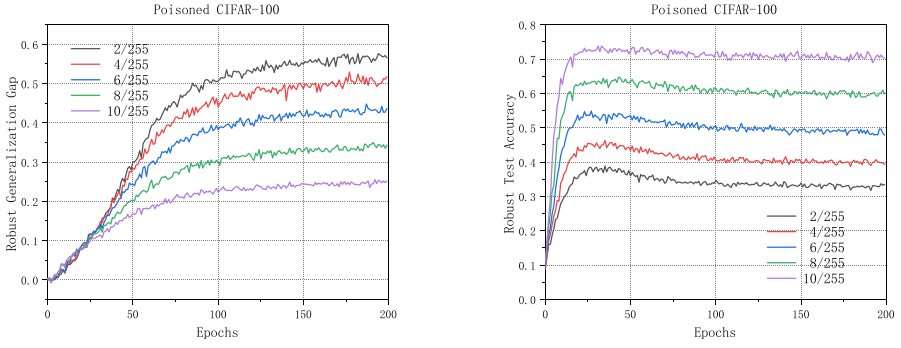


Fig. 2. The robust overfitting phenomenon. With AT budget $\epsilon = 4/255$, we adversarially train a ResNet-10 on four datasets for 200 epochs. The blue line shows the gap between robust training accuracy and robust test accuracy.



(a) Poisoned Gen Gap (b) Poisoned Test Acc

Fig. 3. The robust generalization and robust test accuracy on poisoned CIFAR-10 under different stability attacks. The adversarial training budget $\epsilon = 4/255$ and the poisoning budget $\epsilon' = 8/255$.



(a) Poisoned Gen Gap (a) Poisoned Test Acc

Fig. 4. The robust generalization and robust test accuracy on the poisoned data under HYP attack with different poisoning budgets. The adversarial training budget $\epsilon = 4/255$ and the poisoning budget ϵ' varies.

to simulate the poisoned distribution $\mathcal{P}_{\#}D$. Detailed poisoning settings are given in Appendix C.

Our bounds reflect the influence of data poisoning on the poisoned robust generalization. First, effective stability attacks such as EM, HYP, and REM, indeed result in the shrinkage of robust generalization gaps on CIFAR-10 and ResNet-18 in Fig. 3. Comparing (a) and (b) of Fig. 3, we see that robust generalization gaps present correlated trends to the test performance as pointed out by our results, i.e. Eqs. (8) and (10). We further study the robust generalization under the HYP attack with various intensities, i.e., the poisoning budget ϵ', on CIFAR-100. A larger budget leads to a stronger stability attack. Fig. 4 shows that a stronger stability attack results in a lower robust test accuracy as well as a narrower robust generalization gap on the poisoned data distribution, which confirms the principle stated by our results again.

6. Conclusion

Motivated by the need to analyze the generalization ability for adversarial training under data poisoning attacks, we present a data-dependent stability analysis of adversarial training. Precisely, under certain reasonable smoothness conditions on the loss functions, we prove that SGD-based adversarial training is an $\epsilon(D, \theta_1)$-on-average stable randomized algorithm, and thus give an upper bound $\epsilon(D, \theta_1)$ for the robust generalization gap of the training algorithm.

Our theoretical results provide three main insights for practical adversarial training: (1) From the perspective of generalization bounds, adversarial training is more challenging to generalize than standard training; the larger the robustness budget, the more difficult it is to generalize. Specifically speaking, in Theorems 9 and 12, the derived bounds increase with η, ν which are proportional to the budget ϵ as shown in Lemma 8. (2) Generalization heavily depends on the data distribution. Our bounds can characterize the change of robust generalization in the presence of a poisoning attack as discussed in Section 4.5. Our experiments show that even minor perturbations to the data distribution can cause changes in the generalization bounds. (3) Moreover, our results show that finding better initialization points can help improve robust generalization.

Limitations and future works. Our theoretical results provide the first attempt to analyze the influence of distribution shifts on robust generalization bounds, but only partial solutions are given. More refined generalization bounds for adversarial training to capture more relationships between robust generalization and distribution are a future research problem. In particular, establishing relationships between stability generalization bound and data modification methods such as data sampling (Wang, Liu, et al., 2021) is also a desired topic. Furthermore, alternative forms of Assumptions 4 and 5 for ReLU-based networks need to be further studied.

CRediT authorship contribution statement

Yihan Wang: Writing – original draft, Software, Formal analysis, Data curation, Conceptualization. **Shuang Liu:** Visualization, Validation, Software, Data curation. **Xiao-Shan Gao:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the NKRDPC grant No. 2018YFA0704705 and NSFC grant No. 11688101.

Appendix A. Proofs

A.1. Proof of Lemma 8

Proof.

1. Assume $h(\theta_1, z) = l(\theta_1, z_1)$ and $h(\theta_2, z) = l(\theta_2, z_2)$. We have
$$\|h(\theta_1, z) - h(\theta_2, z)\| = \|l(\theta_1, z_1) - l(\theta_2, z_2)\|.$$
Note that $l(\theta_1, z_1) \geq l(\theta_1, z_2)$ and $l(\theta_2, z_2) \geq l(\theta_2, z_1)$.
If $l(\theta_1, z_1) \geq l(\theta_2, z_2)$, then
$$\|l(\theta_1, z_1) - l(\theta_2, z_2)\| \leq l(\theta_1, z_1) - l(\theta_2, z_1) \leq L\|\theta_1 - \theta_2\|.$$
If $l(\theta_1, z_1) \leq l(\theta_2, z_2)$, then
$$\|l(\theta_1, z_1) - l(\theta_2, z_2)\| \leq l(\theta_2, z_2) - l(\theta_1, z_2) \leq L\|\theta_1 - \theta_2\|.$$

2. Assume that $h(\theta_1, z) = l(\theta_1, z_1)$ and $h(\theta_2, z) = l(\theta_2, z_2)$.
$$\begin{aligned}
&\|\nabla h(\theta_1, z) - \nabla h(\theta_2, z)\| \\
=&\|\nabla l(\theta_1, z_1) - \nabla l(\theta_2, z_2)\| \\
\leq&\|\nabla l(\theta_1, z_1) - \nabla l(\theta_1, z_2)\| + \|\nabla l(\theta_1, z_2) - \nabla l(\theta_2, z_2)\| \\
\leq&L_\theta\|\theta_1 - \theta_2\| + L_z\|z_1 - z_2\|_p \\
\leq&L_\theta\|\theta_1 - \theta_2\| + 2\epsilon L_z.
\end{aligned}$$

3. Assume that $h(\theta_1, z) = l(\theta_1, z_1)$ and $h(\theta_2, z) = l(\theta_2, z_2)$.
$$\begin{aligned}
&\|\nabla^2 h(\theta_1, z) - \nabla^2 h(\theta_2, z)\| \\
=&\|\nabla^2 l(\theta_1, z_1) - \nabla^2 l(\theta_2, z_2)\| \\
\leq&\|\nabla^2 l(\theta_1, z_1) - \nabla^2 l(\theta_1, z_2)\| + \|\nabla^2 l(\theta_1, z_2) - \nabla^2 l(\theta_2, z_2)\| \\
\leq&H_\theta\|\theta_1 - \theta_2\| + H_z\|z_1 - z_2\|_p \\
\leq&H_\theta\|\theta_1 - \theta_2\| + 2\epsilon H_z. \quad \square
\end{aligned}$$

A.2. Proof of Theorem 9

We first prove several lemmas.

A core technique in stability analysis is to give the expansion properties of update rules.

Definition 15 (*Expansion*). The update rule $G_{\mathcal{A}}$ is ι-approximately κ-expansive, if $\forall z \in \mathbb{D}$
$$\|G_{\mathcal{A}}(\theta_1, z, \alpha) - G_{\mathcal{A}}(\theta_2, z, \alpha)\| \leq \kappa\|\theta_1 - \theta_2\| + \iota.$$

If the original loss $l(\theta, z)$ is β-gradient Lipschitz in θ, then the update rule in standard training is 1-expansive in the convex case and $(1 + \alpha\beta)$-expansive in the non-convex case (Hardt et al., 2016). If the adversarial loss $h(\theta, z)$ is η-approximately β-gradient Lipschitz in θ, then the expansion coefficients in the update rule $G_{\mathcal{A}}$ remain unchanged in both convex and non-convex cases, while the approximation parameter η leads to an additional term $\alpha\eta$ in each update (Xiao et al., 2022).

Lemma 16 (*Xiao et al., 2022*). *Suppose that the adversarial loss $h(\theta, z)$ is η-approximately β-gradient Lipschitz in θ.*

1. *(η-approximate descent.)*
$$h(\theta_1, z) - h(\theta_2, z) \leq \nabla h(\theta_2, z)^\top(\theta_1 - \theta_2) + \frac{\beta}{2}\|\theta_1 - \theta_2\|^2 + \eta\|\theta_1 - \theta_2\|.$$

2. *The update rule $G_{\mathcal{A}}$ is η-approximately $(1 + \alpha\beta)$-expansive:*
$$\|G_{\mathcal{A}}(\theta_1, z, \alpha) - G_{\mathcal{A}}(\theta_2, z, \alpha)\| \leq (1 + \alpha\beta)\|\theta_1 - \theta_2\| + \alpha\eta.$$

3. *Assume in addition that $h(\theta, z)$ is convex in θ, for $\alpha \leq 1/\beta$, we have*
$$\|G_{\mathcal{A}}(\theta_1, z, \alpha) - G_{\mathcal{A}}(\theta_2, z, \alpha)\| \leq \|\theta_1 - \theta_2\| + \alpha\eta.$$

Given a dataset $S = \{z_1, \ldots, z_n\} \sim \mathcal{D}^n$, an example $z \sim \mathcal{D}$, and an index $i \in [n]$, we denote $S^{i,z} = \{z'_1, \ldots, z'_n\}$ with $z'_j = z_j$ for $j \neq i$ and $z'_i = z$. Let θ_t, θ'_t be the tth outputs of $\mathcal{A}(S)$ and $\mathcal{A}(S^{i,z})$ respectively. Denote the distance of two trajectories at step t by $\delta_t(S, z, i, \mathcal{A}) = \|\theta_t - \theta'_t\|$. As both updates start from θ_1, we have $\delta_1(S, z, i, \mathcal{A}) = 0$. Since the on-average stability in Definition 1 takes the supremum over the index $i \in [n]$, the stability analysis aims at providing a unified bound for all $i \in [n]$. Thus, we will not point out the selection of i in later statements for brevity.

We restate Lemma 5 in Kuzborskij and Lampert (2018) on which the data-dependent stability analysis relies. Note that this lemma holds for SGD without replacement in both a single pass and multiple passes through the training set. The multiple-pass case cycles through S repeatedly in a fixed order determined by $\mathcal{A}$.

Lemma 17. *Assume that the adversarial loss $h(\theta, z)$ is non-negative and L-Lipschitz in θ. Then, $\forall t_0 \in [n + 1]$,*
$$\begin{aligned}
\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_t, z) - h(\theta'_t, z)] \leq L\mathbb{E}_{S,z}[\mathbb{E}_{\mathcal{A}}[\delta_t(S, z, i, \mathcal{A})|\delta_{t_0}(S, z, i, \mathcal{A}) = 0]] \\
+ \frac{t_0 - 1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_{\mathcal{D}}(\theta_t)].
\end{aligned}$$

Due to the change in notation, we repeat the proof here.

Proof. By the Lipschitz condition and non-negativeness of h, we have
$$\begin{aligned}
&h(\theta_t, z) - h(\theta'_t, z) \\
=&(h(\theta_t, z) - h(\theta'_t, z))\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} + (h(\theta_t, z) - h(\theta'_t, z)) \\
&\times \mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) \neq 0\} \\
\leq&L\delta_t(S, z, i, \mathcal{A})\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} + h(\theta_t, z)\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) \neq 0\}.
\end{aligned}$$
Take expectation w.r.t. $\mathcal{A}$ and we have
$$\begin{aligned}
\mathbb{E}_{\mathcal{A}}[h(\theta_t, z) - h(\theta'_t, z)] \leq L\mathbb{E}_{\mathcal{A}}[\delta_t(S, z, i, \mathcal{A})|\delta_{t_0}(S, z, i, \mathcal{A}) = 0] \\
+ \mathbb{E}_{\mathcal{A}}[h(\theta_t, z)\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) \neq 0\}].
\end{aligned} \tag{A.1}$$
Note that the first time $\mathcal{A}$ selects the different example is $\pi^{-1}(i)$. Since that $\pi^{-1}(i) \geq t_0$ implies $\delta_{t_0}(S, z, i, \mathcal{A}) = 0$, we have $\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) \neq 0\} \leq \mathbb{I}\{\pi^{-1}(i) < t_0\}$. It follows that
$$\begin{aligned}
&\mathbb{E}_{S,z}[\mathbb{E}_{\mathcal{A}}[h(\theta_t, z)\mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) \neq 0\}]] \\
&\leq \mathbb{E}_{S,z}[\mathbb{E}_{\mathcal{A}}[h(\theta_t, z)\mathbb{I}\{\pi^{-1}(i) < t_0\}]] \\
&= \mathbb{E}_{z,\mathcal{A}}[\mathbb{E}_S[h(\theta_t, z)]\mathbb{I}\{\pi^{-1}(i) < t_0\}].
\end{aligned} \tag{A.2}$$
Recall that a realization of $\mathcal{A}$ is a permutation π of $[n]$. Thus, with a fixed π, taking over $S \sim \mathcal{D}^n$ equals taking over both $S \sim \mathcal{D}^n$ and $\mathcal{A}$. That is, $\mathbb{E}_S[h(\theta_t, z)] = \mathbb{E}_{\mathcal{A},S}[h(\theta_t, z)]$. As a consequence, we have
$$\begin{aligned}
&\mathbb{E}_{z,\mathcal{A}}[\mathbb{E}_S[h(\theta_t, z)]\mathbb{I}\{\pi^{-1}(i) < t_0\}] \\
=&\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_t, z)]\mathbb{E}_{\mathcal{A}}[\mathbb{I}\{\pi^{-1}(i) < t_0\}] \\
\leq&\frac{t_0 - 1}{n}\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_t, z)].
\end{aligned} \tag{A.3}$$
Combining Eqs. (A.1) (A.2) and (A.3), we get the statement. $\square$

Lemma 18. *Suppose that the adversarial loss $h(\theta, z)$ is L-Lipschitz and η-approximately β-gradient Lipschitz in θ. Then, in a single pass such that $T \in [n]$, we have that*
$$\sum_{t=1}^{T}(\alpha_t - \frac{\beta\alpha_t^2}{2})\mathbb{E}_S[\|\nabla\mathcal{R}_{\mathcal{D}}(\theta_t)\|^2] \leq$$

$$\mathcal{R}_D(\theta_1) - \mathbb{E}_S[\mathcal{R}_D(\theta_T)] + \eta L \sum_{t=1}^{T} \alpha_t + \frac{\beta}{2} \sum_{t=1}^{T} \alpha_t^2 \mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|^2].$$

Proof. From the first statement in Lemma 16, we have

$$\begin{aligned}
&\mathcal{R}_D(\theta_{t+1}) - \mathcal{R}_D(\theta_t) \\
\le &\nabla \mathcal{R}_D(\theta_t)^\top (\theta_{t+1} - \theta_t) + \frac{\beta \alpha_t^2}{2} \|\nabla h(\theta_t, z_{\pi(t)})\|^2 + \eta \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\| \\
\le &(\beta \alpha_t^2 - \alpha_t) \nabla \mathcal{R}_D(\theta_t)^\top \nabla h(\theta_t, z_{\pi(t)}) + \eta \alpha_t L + \frac{\beta \alpha_t^2}{2} \|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|^2 \\
&- \frac{\beta \alpha_t^2}{2} \|\nabla \mathcal{R}_D(\theta_t)\|^2.
\end{aligned}$$

Since θ_t is determined by $z_{\pi(1)}, \dots, z_{\pi(t-1)}$ and $\mathbb{E}_{z_{\pi(t)}}[h(\theta_t, z_{\pi(t)})] = \mathcal{R}_D(\theta_t)$, we have that

$$\mathbb{E}_S[\nabla \mathcal{R}_D(\theta_t)^\top \nabla h(\theta_t, z_{\pi(t)})] = \mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2].$$

Take expectation w.r.t. S and rearrange terms,

$$\begin{aligned}
&(\alpha_t - \frac{\beta \alpha_t^2}{2}) \mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2] \le \\
&\mathbb{E}_S[\mathcal{R}_D(\theta_t) - \mathcal{R}_D(\theta_{t+1}) + \frac{\beta \alpha_t^2}{2} \|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|^2] + \eta \alpha_t L.
\end{aligned}$$

Sum the above over $t = 1, \dots, T$ and get the statement. □

Lemma 19. *Suppose that the adversarial loss $h(\theta, z)$ is L-Lipschitz and η-approximately β-gradient Lipschitz with respect to θ, and the step sizes $\alpha_t \le 1/\beta$. Assume the variance of stochastic gradients in $\mathcal{A}$ obeys for all $t \in [T]$*

$$\mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|^2] \le \sigma_t^2.$$

We have

$$\mathbb{E}_S[\sum_{t=1}^{T} \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\|] \le \sum_{t=1}^{T} \sigma_t \alpha_t + 2 \sqrt{\sum_{t=1}^{T} \alpha_t} \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta}{2} \sum_{t=1}^{T} \sigma_t^2 \alpha_t^2 + \eta L \sum_{t=1}^{T} \alpha_t}$$

Proof. Repeatedly applying Jensen's inequality, we have

$$\begin{aligned}
&\mathbb{E}_S[\sum_{t=1}^{T} \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\|] \\
\le &\sum_{t=1}^{T} \alpha_t \mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|] + \sum_{t=1}^{T} \alpha_t \mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|] \\
\le &\sum_{t=1}^{T} \alpha_t \sqrt{\mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla \mathcal{R}_D(\theta_t)\|^2]} + \sum_{t=1}^{T} \alpha_t \sqrt{\mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2]} \\
\le &\sum_{t=1}^{T} \sigma_t \alpha_t + \sum_{t=1}^{T} \alpha_t \sqrt{\mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2]} \\
\le &\sum_{t=1}^{T} \sigma_t \alpha_t + 2 \sum_{t=1}^{T} (\alpha_t - \frac{\beta \alpha_t^2}{2}) \sqrt{\mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2]} \\
\le &\sum_{t=1}^{T} \sigma_t \alpha_t + 2 \sqrt{\sum_{t=1}^{T} (\alpha_t - \frac{\beta \alpha_t^2}{2})} \sqrt{\sum_{t=1}^{T} (\alpha_t - \frac{\beta \alpha_t^2}{2}) \mathbb{E}_S[\|\nabla \mathcal{R}_D(\theta_t)\|^2]} \\
\le &\sum_{t=1}^{T} \sigma_t \alpha_t + 2 \sqrt{\sum_{t=1}^{T} \alpha_t} \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta}{2} \sum_{t=1}^{T} \sigma_t^2 \alpha_t^2 + \eta L \sum_{t=1}^{T} \alpha_t}.
\end{aligned}$$

The penultimate inequality is by Lemma 18. □

We now prove Theorem 9.

Proof. Denote $\Delta_t(S, z, i) = \mathbb{E}_{\mathcal{A}}[\delta_t(S, z, i, \mathcal{A}) | \delta_{t_0}(S, z, i, \mathcal{A}) = 0]$. By Lemma 17, $\forall t_0 \in \{1, \dots, n, n+1\}$ we have

$$\begin{aligned}
&\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta'_{T+1}, z)] \\
\le &L \mathbb{E}_{S,z}[\Delta_{T+1}(S, z, i)] + \frac{t_0 - 1}{n} \mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_{T+1})].
\end{aligned}$$

At step t, $\mathcal{A}$ selects the example $\pi(t) = i$ with probability $1/n$ and $\pi(t) \ne i$ with probability $1 - 1/n$. When $\pi(t) \ne i$, by the third statement in Lemma 16, we have

$$\begin{aligned}
&\delta_{t+1}(S, z, i, \mathcal{A}) \cdot \mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} \\
\le &\delta_t(S, z, i, \mathcal{A}) \cdot \mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} + \alpha_t \eta.
\end{aligned}$$

When $\pi(t) = i$, we have

$$\begin{aligned}
&\delta_{t+1}(S, z, i, \mathcal{A}) \cdot \mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} \\
\le &\delta_t(S, z, i, \mathcal{A}) \cdot \mathbb{I}\{\delta_{t_0}(S, z, i, \mathcal{A}) = 0\} + \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\| + \alpha_t \|\nabla h(\theta'_t, z'_{\pi(t)})\|.
\end{aligned}$$

Take expectation w.r.t. $\mathcal{A}$ and we have

$$\begin{aligned}
&\Delta_{t+1}(S, z, i) \\
\le &\frac{1}{n} (\Delta_t(S, z, i) + \alpha_t \mathbb{E}_{\mathcal{A}}[\|\nabla h(\theta_t, z_{\pi(t)})\| + \|\nabla h(\theta'_t, z'_{\pi(t)})\|]) \\
&+ (1 - \frac{1}{n}) (\Delta_t(S, z, i) + \alpha_t \eta) \\
= &\Delta_t(S, z, i) + (1 - \frac{1}{n}) \alpha_t \eta + \frac{\alpha_t}{n} \mathbb{E}_{\mathcal{A}}[\|\nabla h(\theta_t, z_{\pi(t)})\| + \|\nabla h(\theta'_t, z'_{\pi(t)})\|].
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta'_{T+1}, z)] \\
\le &\frac{L}{n} \sum_{t=t_0}^{T} \alpha_t \mathbb{E}_{z,\mathcal{A}}[\mathbb{E}_S[\|\nabla h(\theta_t, z_{\pi(t)})\| + \|\nabla h(\theta'_t, z'_{\pi(t)})\|]] + (1 - \frac{1}{n}) L \eta \sum_{t=t_0}^{T} \alpha_t \\
&+ \frac{t_0 - 1}{n} \mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_T)].
\end{aligned}$$

Here we take $t_0 = 1$. By Lemma 19, we have

$$\begin{aligned}
&\mathbb{E}_S[\sum_{t=1}^{T} \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\|] \\
\le &\sum_{t=1}^{T} \sigma \alpha_t + 2 \sqrt{\sum_{t=1}^{T} \alpha_t} \cdot \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta}{2} \sum_{t=1}^{T} \sigma^2 \alpha_t^2 + \eta L \sum_{t=1}^{T} \alpha_t},
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}_{z,S}[\sum_{t=1}^{T} \alpha_t \|\nabla h(\theta'_t, z'_{\pi(t)})\|] \\
= &\mathbb{E}_S[\sum_{t=1}^{T} \alpha_t \|\nabla h(\theta_t, z_{\pi(t)})\|] \\
\le &\sum_{t=1}^{T} \sigma \alpha_t + 2 \sqrt{\sum_{t=1}^{T} \alpha_t} \cdot \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta}{2} \sum_{t=1}^{T} \sigma^2 \alpha_t^2 + \eta L \sum_{t=1}^{T} \alpha_t},
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta'_{T+1}, z)] \le \frac{2L}{n} \sum_{t=1}^{T} \sigma \alpha_t \\
&+ L \eta \sum_{t=1}^{T} \alpha_t + \frac{4L}{n} \sqrt{\sum_{t=1}^{T} \alpha_t} \cdot \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta}{2} \sum_{t=1}^{T} \sigma^2 \alpha_t^2 + \eta L \sum_{t=1}^{T} \alpha_t}.
\end{aligned}$$

□

A.3. Proof of Corollary 11

Proof.

$$(\frac{2\sigma L}{n} + L\eta) \sum_{t=1}^{T} \alpha_t + \frac{4L}{n} \sqrt{\sum_{t=1}^{T} \alpha_t}.$$

$$\times \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta\sigma^2}{2}\sum_{t=1}^{T}\alpha_t^2 + \eta L\sum_{t=1}^{T}\alpha_t}$$

$$=(\frac{2\sigma L}{n} + L\eta)\alpha T + \frac{4L}{n}\sqrt{\alpha T(\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta\sigma^2\alpha^3T^2}{2} + \eta L\alpha^2T^2)}$$

$$\le(\frac{2\sigma L}{n} + L\eta)\alpha T + \frac{4L}{n}\sqrt{\alpha Tr + \frac{\sigma^2\alpha^2T^2}{2} + \eta L\alpha^2T^2}$$

$$\le\eta\alpha TL + \frac{2\sigma\alpha TL}{n} + \frac{4L}{n}(\sqrt{\alpha Tr} + \frac{\sigma\alpha T}{\sqrt{2}} + \sqrt{\eta L}\alpha T)$$

$$=\eta\alpha TL + \frac{2\alpha TL}{n}(\sigma + \sqrt{2}\sigma + 2\sqrt{\eta L}) + \frac{4L\sqrt{\alpha Tr}}{n}. \quad \square$$

A.4. Proof of Theorem 12

We first prove several lemmas.

Lemma 20. *Suppose the adversarial loss $h(\theta, z)$ is ν-approximately ρ-Hessian Lipschitz with respect to θ. At step t with $\pi(t) \neq i$, we have*

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta_t')\| \le (1 + \alpha_t\xi_t(S, z, i, \mathcal{A}))\delta_t(S, z, i, \mathcal{A}),$$

where

$$\xi_t(S, z, i, \mathcal{A}) = \|\nabla h(\theta_1, z_{\pi(t)})\| + \frac{\rho}{2}\sum_{k=1}^{t-1}\alpha_k(\|\nabla h(\theta_k, z_{\pi(k)})\| + \|\nabla h(\theta_k', z_{\pi(k)}')\|) + \nu.$$

Furthermore, when $\alpha_k = \frac{c}{k}$ with $c \le \frac{1}{\beta}$, we have

$$\mathbb{E}_{S,z}[\xi_t(S, z, i, \mathcal{A})]$$
$$\le\mathbb{E}_z[\|\nabla^2 h(\theta_1, z)\|] + \nu + 2\rho\sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*)c(1 + \ln t)}$$
$$+ 2\rho\sigma c\sqrt{\beta c(1 + \ln t)} + \rho c(\sigma + 2\sqrt{\eta L})(1 + \ln t).$$

Proof. For $\pi(t) \neq i$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta_t')\| \le \|\theta_t - \theta_t'\| + \alpha_t\|\nabla h(\theta_t, z_{\pi(t)}) - \nabla h(\theta_t', z_{\pi(t)})\|.$$

For brevity, we denote $h_t(\theta) = h(\theta, z_{\pi(t)})$. By Taylor expansion with integral remainder, we have

$$\nabla h_t(\theta_t) - \nabla h_t(\theta_t')$$
$$= \int_0^1 \nabla^2 h_t(\theta_t + \tau(\theta_t' - \theta_t))d\tau \cdot (\theta_t - \theta_t')$$
$$= \int_0^1 (\nabla^2 h_t(\theta_t + \tau(\theta_t' - \theta_t)) - \nabla^2 h_t(\theta_1))d\tau \cdot (\theta_t - \theta_t') + \nabla^2 h_t(\theta_1) \cdot (\theta_t - \theta_t').$$

Since h is ν-approximately ρ-Hessian Lipschitz,

$$\|\nabla h_t(\theta_t) - \nabla h_t(\theta_t')\|$$
$$\le (\rho\int_0^1 \|\theta_t + \tau(\theta_t' - \theta_t) - \theta_1\|d\tau + \nu + \|\nabla^2 h_t(\theta_1)\|) \cdot \|\theta_t - \theta_t'\|.$$

Note that

$$\theta_t + \tau(\theta_t' - \theta_t) - \theta_1$$
$$= (1 - \tau)(\theta_t - \theta_1) + \tau(\theta_t' - \theta_1)$$
$$= (1 - \tau)\sum_{k=1}^{t-1}(\theta_{k+1} - \theta_k) + \tau\sum_{k=1}^{t-1}(\theta_{k+1}' - \theta_k')$$
$$= (1 - \tau)\sum_{k=1}^{t-1}\alpha_k\nabla h(\theta_k, z_{\pi(k)}) + \tau\sum_{k=1}^{t-1}\alpha_k\nabla h_k(\theta_k', z_{\pi(k)}').$$

Therefore,

$$\int_0^1 \|\theta_t + \tau(\theta_t' - \theta_t) - \theta_1\|d\tau$$
$$\le \frac{1}{2}\sum_{k=1}^{t-1}\alpha_k(\|\nabla h(\theta_k, z_{\pi(k)})\| + \|\nabla h(\theta_k', z_{\pi(k)}')\|).$$

Taking $\alpha_k = \frac{c}{k}$ and by Lemma 19, we have

$$\frac{1}{2}\sum_{k=1}^{t-1}\alpha_k(\|\nabla h(\theta_k, z_{\pi(k)})\| + \|\nabla h(\theta_k', z_{\pi(k)}')\|)$$
$$\le\sigma\sum_{k=1}^{t-1}\alpha_k + 2\sqrt{\sum_{k=1}^{t-1}\alpha_k} \cdot \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \frac{\beta\sigma^2}{2}\sum_{k=1}^{t-1}\alpha_k^2 + \eta L\sum_{k=1}^{t-1}\alpha_k}$$
$$\le c\sigma(1 + \ln t) + 2\sqrt{c(1 + \ln t)} \cdot \sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*) + \beta c^2\sigma^2 + c\eta L(1 + \ln t)}$$
$$\le 2\sqrt{\mathcal{R}_D(\theta_1) - \mathcal{R}_D(\theta^*)c(1 + \ln t)} + 2\sigma c\sqrt{\beta c(1 + \ln t)} + c(\sigma + 2\sqrt{\eta L})(1 + \ln t).$$

The penultimate inequality is due to

$$\sum_{k=1}^{t}\frac{1}{k} \le 1 + \ln t, \text{ and } \sum_{k=1}^{t}\frac{1}{k^2} \le 2 - \frac{1}{t}. \quad \square$$

Lemma 21 (*Bernstein-type Inequality Kuzborskij & Lampert, 2018*). *Let Z be a zero-mean real-valued random variable such that $|Z| \le b$ and $\mathbb{E}[Z^2] \le \sigma^2$. Then for all $|c| \le \frac{1}{2b}$, we have that $\mathbb{E}[e^{cZ}] \le e^{c^2\sigma^2}$.*

We now prove Theorem 12.

Proof. Let $\Delta_t(S, z, i) = \mathbb{E}_{\mathcal{A}}[\delta_t(S, z, i, \mathcal{A})|\delta_{t_0}(S, z, i, \mathcal{A}) = 0]$. By Lemma 17, $\forall t_0 \in [n + 1]$,

$$\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1}, z) - h(\theta_{T+1}', z)] \le L\mathbb{E}_{S,z}[\Delta_{T+1}(S, z, i)] + \frac{t_0 - 1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_{T+1})].$$

When $\pi(t) = i$ with probability $\frac{1}{n}$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta_t')\| \le \delta_t(S, z, i, \mathcal{A}) + 2\alpha_t L.$$

When $\pi(t) \neq i$ with probability $1 - \frac{1}{n}$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta_t')\| \le (1 + \alpha_t\beta)\delta_t(S, z, i, \mathcal{A}) + \alpha_t\eta,$$

by the second statement in Lemma 16 and

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t) - \mathcal{G}_{\mathcal{A}}(\theta_t')\| \le (1 + \alpha_t\xi_t(S, z, i, \mathcal{A}))\delta_t(S, z, i, \mathcal{A}),$$

by Lemma 20. Let

$$\psi_t(S, z, i) = \mathbb{E}_{\mathcal{A}}[\min\{\xi_t(S, z, i, \mathcal{A}), \beta\}]$$

and we have

$$\Delta_{t+1}(S, z, i)$$
$$\le\frac{1}{n}(\Delta_t(S, z, i) + 2\alpha_t L) + (1 - \frac{1}{n})((1 + \alpha_t\psi_t(S, z, i))\Delta_t(S, z, i) + \alpha_t\eta)$$
$$=(1 + (1 - \frac{1}{n})\alpha_t\psi_t(S, z, i))\Delta_t(S, z, i) + \frac{2\alpha_t L + (n - 1)\alpha_t\eta}{n}$$
$$\le\exp((1 - \frac{1}{n})\alpha_t\psi_t(S, z, i))\Delta_t(S, z, i) + \frac{2\alpha_t L}{n} + \alpha_t\eta.$$

Note that $\Delta_{t_0}(S, z, i) = 0$ and $\alpha_t = \frac{c}{t}$. We have

$$\Delta_{T+1}(S, z, i)$$
$$\le\sum_{t=t_0}^{T}(\prod_{k=t+1}^{T}\exp(\frac{(n - 1)c\psi_k(S, z, i)}{nk}))(\frac{2cL}{nt} + \frac{c\eta}{t})$$
$$=\sum_{t=t_0}^{T}\exp(\frac{(n - 1)c}{n}\sum_{k=t+1}^{T}\frac{\psi_k(S, z, i)}{k})(\frac{2cL}{nt} + \frac{c\eta}{t}).$$

Let $\mu_k = \mathbb{E}_{S,z}[\psi_k(S, z, i)]$. We have $|\psi_k(S, z, i) - \mu_k| \le 2\beta$ and

$$\mathbb{E}_{S,z}[\exp(c\sum_{k=t+1}^{T}\frac{\psi_k(S, z, i)}{k})]$$
$$= \mathbb{E}_{S,z}[\exp(c\sum_{k=t+1}^{T}\frac{\psi_k(S, z, i) - \mu_k}{k})]\exp(c\sum_{k=t+1}^{T}\frac{\mu_k}{k}).$$

Since

$$|\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)-\mu_k}{k}|\le 2\beta\ln T,$$

we assume $c\le\min\{\frac{1}{2(2\beta\ln T)^2},\frac{1}{2(2\beta\ln T)}\}$. By Lemma 21, we have

$$\begin{aligned}
&\mathbb{E}_{S,z}[\exp(c\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)-\mu_k}{k})]\\
&\le\exp(c^2\mathbb{E}_{S,z}[(\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)-\mu_k}{k})^2])\\
&\le\exp(\frac{c}{2}\mathbb{E}_{S,z}[(\frac{1}{2\beta\ln T}\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)-\mu_k}{k})^2])\\
&\le\exp(\frac{c}{2}\mathbb{E}_{S,z}[|\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)-\mu_k}{k}|])\\
&\le\exp(\frac{c}{2}\sum_{k=t+1}^{T}\frac{\mathbb{E}_{S,z}[|\psi_k(S,z,i)-\mu_k|]}{k})\\
&\le\exp(c\sum_{k=t+1}^{T}\frac{\mu_k}{k}).
\end{aligned}$$

It follows that

$$\mathbb{E}_{S,z}[\exp(c\sum_{k=t+1}^{T}\frac{\psi_k(S,z,i)}{k})]\le\exp(c\sum_{k=t+1}^{T}\frac{2\mu_k}{k}).$$

Note that

$$\mu_k\le\min\{\mathbb{E}_{\mathcal{A}}[\mathbb{E}_{S,z}[\xi_k(S,z,i,\mathcal{A})]],\beta\}.$$

Assuming in addition that $c\le\frac{1}{\beta}$, we have $\mathbb{E}_{S,z}[\xi_k(S,z,i,\mathcal{A})]$ is bounded by γ by Lemma 20. We have that

$$\begin{aligned}
&\mathbb{E}_{S,z}[\Delta_{T+1}(S,z,i)]\\
&\le\sum_{t=t_0}^{T}\exp(2c\gamma(1-\frac{1}{n})\sum_{k=t+1}^{T}\frac{1}{k})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&\le\sum_{t=t_0}^{T}\exp(2c\gamma(1-\frac{1}{n})\ln\frac{T}{t})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&\le\sum_{t=t_0}^{T}\exp(2c\gamma\ln\frac{T}{t})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&=(\frac{2cL}{n}+c\eta)T^{2c\gamma}\sum_{t=t_0}^{T}t^{-2c\gamma-1}\\
&\le(\frac{2L+\eta n}{2n\gamma})(\frac{T}{t_0-1})^{2c\gamma}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1},z)-h(\theta'_{T+1},z)]&\le(\frac{2L^2+\eta nL}{2n\gamma})(\frac{T}{t_0-1})^{2c\gamma}\\
&+\frac{t_0-1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_T)].
\end{aligned}\tag{A.4}$$

Let $q=2c\gamma$ and $r=\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_{T+1})]$. Setting

$$t_0=((2L^2+n\eta L)\frac{cT^q}{r})^{\frac{1}{1+q}}+1$$

minimizes Eq. (A.4) and we have

$$\epsilon(\mathcal{D},\theta_1)\le\frac{1+1/q}{n}(2cL^2+nc\eta L)^{\frac{1}{1+q}}(Tr)^{\frac{q}{1+q}}.\quad\square$$

A.5. Proof of Theorem 13

Proof. Let $\Delta_t(S,z,i)=\mathbb{E}_{\mathcal{A}}[\delta_t(S,z,i,\mathcal{A})|\delta_{t_0}(S,z,i,\mathcal{A})=0]$. By Lemma 17, $\forall t_0\in[n+1]$,

$$\begin{aligned}
\mathbb{E}_{S,z,\mathcal{A}}[h(\theta_{T+1},z)-h(\theta'_{T+1},z)]&\le L\mathbb{E}_{S,z}[\Delta_{T+1}(S,z,i)]\\
&+\frac{t_0-1}{n}\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_{T+1})].
\end{aligned}$$

When $\pi(t)=i$ with probability $\frac{1}{n}$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t)-\mathcal{G}_{\mathcal{A}}(\theta'_t)\|\le\delta_t(S,z,i,\mathcal{A})+2\alpha_tL.$$

When $\pi(t)\ne i$ with probability $1-\frac{1}{n}$, we have

$$\|\mathcal{G}_{\mathcal{A}}(\theta_t)-\mathcal{G}_{\mathcal{A}}(\theta'_t)\|\le(1+\alpha_t\beta)\delta_t(S,z,i,\mathcal{A})+\alpha_t\eta,$$

by the second statement in Lemma 16. We have

$$\begin{aligned}
&\Delta_{t+1}(S,z,i)\\
&\le\frac{1}{n}(\Delta_t(S,z,i)+2\alpha_tL)+(1-\frac{1}{n})((1+\alpha_t\beta)\Delta_t(S,z,i)+\alpha_t\eta)\\
&=(1+(1-\frac{1}{n})\alpha_t\beta)\Delta_t(S,z,i)+\frac{2\alpha_tL+(n-1)\alpha_t\eta}{n}\\
&\le\exp((1-\frac{1}{n})\alpha_t\beta)\Delta_t(S,z,i)+\frac{2\alpha_tL}{n}+\alpha_t\eta.
\end{aligned}$$

Let $\alpha_t=\frac{c}{t}$ with $c\le\frac{1}{\beta}$. It follows that

$$\begin{aligned}
\Delta_{T+1}(S,z,i)&\le\sum_{t=t_0}^{T}(\prod_{k=t+1}^{T}\exp\frac{((n-1)c\beta)}{nk})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&=\sum_{t=t_0}^{T}\exp(\frac{(n-1)c}{n}\sum_{k=t+1}^{T}\frac{\beta}{k})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&\le\sum_{t=t_0}^{T}\exp(\frac{(n-1)c\beta}{n}\ln\frac{T}{t})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&\le\sum_{t=t_0}^{T}\exp(2c\beta\ln\frac{T}{t})(\frac{2cL}{nt}+\frac{c\eta}{t})\\
&=(\frac{2cL}{n}+c\eta)T^{2c\beta}\sum_{t=t_0}^{T}t^{-2c\beta-1}\\
&\le(\frac{2L+\eta n}{2n\beta})(\frac{T}{t_0-1})^{2c\beta}.
\end{aligned}$$

Let $q=2c\beta$ and $r=\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\theta_{T+1})]$. Setting

$$t_0=((2L^2+n\eta L)\frac{cT^q}{r})^{\frac{1}{1+q}}+1,$$

we have

$$\epsilon(\mathcal{D},\theta_1)\le\frac{1+1/q}{n}(2cL^2+nc\eta L)^{\frac{1}{1+q}}(Tr)^{\frac{q}{1+q}}.\quad\square$$

Appendix B. Discussions of uniform stability-based counterparts

Uniform stability analysis employs the following notion:

Definition 22 (*Uniform Stability*). A randomized algorithm $\mathcal{A}$ is ϵ-*uniformly stable* if for all $S,S'\in\mathbb{D}^n$ such that S and S' differ in at most one element, we have

$$\sup_{z\in\mathbb{D}}\mathbb{E}_{\mathcal{A}}[h(\mathcal{A}(S),z)-h(\mathcal{A}(S'),z)]\le\epsilon.\tag{B.1}$$

Thus, uniform stability bounds the expected difference between the losses of algorithm outputs on two adjacent training sets. The uniform stability is distribution-free since S and S' are independent of $\mathcal{D}$. A generalization bound was given as follows.

Theorem 23 (*Hardt et al., 2016*). *If $\mathcal{A}$ is ϵ-uniformly stable, then the robust generalization gap of $\mathcal{A}$ is bounded by ϵ:*

$$|\mathbb{E}_{S,\mathcal{A}}[\mathcal{R}_D(\mathcal{A}(S))-\mathcal{R}_S(\mathcal{A}(S))]|\le\epsilon.$$

The following theorem gives an upper bound of the robust generalization gap.

Theorem 24 (*Xiao et al., 2022*). *Let $h(\theta,z)$ be convex, L-Lipschitz and η-approximately β-gradient Lipschitz in θ. Let the step sizes $\alpha_t=\alpha\le\frac{1}{\beta}$.*

Then the generalization gap of algorithm $\mathcal{A}$ using the training set of size n after T steps of the SGD update has an upper bound

$$\epsilon_{\text{gen}} = (\eta + \frac{2L}{n})\alpha T L,$$

where η is a parameter proportional to the adversarial training budget ϵ and the approximate gradient Lipschitz condition will be defined in Lemma 8.

B.1. Data poisoning

A poisoned algorithm $\mathcal{A}_{\mathcal{P}}$ uses the gradient on a poisoned datum $\nabla h(\theta, \mathcal{P}(z))$ instead of $\nabla h(\theta, z)$ which may result in totally different update trajectories dependent on $\mathcal{P}$. Nevertheless, the expansion properties of $\mathcal{G}_{\mathcal{A}_{\mathcal{P}}}$ are not affected by the attack $\mathcal{P}$ at all. Indeed, due to

$$\mathcal{G}_{\mathcal{A}_{\mathcal{P}}}(\theta, z, \alpha) = \mathcal{G}_{\mathcal{A}}(\theta, \mathcal{P}(z), \alpha),$$

if $\mathcal{G}_{\mathcal{A}}$ is ι-approximately κ-expansive, the poisoned update rule $\mathcal{G}_{\mathcal{A}_{\mathcal{P}}}$ is ι-approximately κ-expansive as well. Therefore, uniform stability analysis provides the same upper bound of $\|\mathcal{A}_{\mathcal{P}}(S) - \mathcal{A}_{\mathcal{P}}(S')\|$. Thus, Theorem 24 implies the following proposition.

Proposition 25. *The poisoned generalization gap $\epsilon_{\mathcal{P}}$ based on the uniform stability analysis remains unchanged, i.e.*

$$\epsilon_{\mathcal{P}} = (\eta + \frac{2L}{n})\alpha T L.$$

Similar consequences also hold for the results in Hardt et al. (2016), Xing et al. (2021), and the results for non-convex and strongly-convex cases in Xiao et al. (2022).

Proof. By Theorem 23 and the Lipschitz assumption, it suffices to prove

$$\mathbb{E}_{\mathcal{A}_P}[\|\mathcal{A}_P(S) - \mathcal{A}_P(S')\|] \leq (\eta + \frac{2L}{n})\alpha T. \tag{B.2}$$

Assume that the trajectories of $\mathcal{A}_P(S)$ and $\mathcal{A}_P(S')$ are $\theta_1, \ldots, \theta_T$ and $\theta'_1, \ldots, \theta'_T$ respectively. Let $\delta_t = \|\theta_t - \theta'_t\|$. By the third statement in Lemma 16, the update rule $\mathcal{G}_{\mathcal{A}}$ is $\alpha\eta$-approximately 1-expansive. Since we have

$$\mathcal{G}_{\mathcal{A}_P}(\theta, z, \alpha) = \mathcal{G}_{\mathcal{A}}(\theta, P(z), \alpha),$$

the update rule $\mathcal{G}_{\mathcal{A}_P}$ is $\alpha\eta$-approximately 1-expansive as well due to Definition 15. Note that at step t the algorithm $\mathcal{A}_P$ selects the example that S and S' differ with probability $\frac{1}{n}$. In this case,

$$\delta_{t+1} \leq \delta_t + 2\alpha L.$$

In the other case,

$$\delta_{t+1} \leq \delta_t + \alpha\eta.$$

It follows that

$$\begin{aligned}\mathbb{E}_{\mathcal{A}_P}[\delta_{t+1}] &\leq \frac{1}{n}(\mathbb{E}_{\mathcal{A}_P}[\delta_t] + 2\alpha L) + (1 - \frac{1}{n})(\mathbb{E}_{\mathcal{A}_P}[\delta_t] + \alpha\eta) \\ &\leq \mathbb{E}_{\mathcal{A}_P}[\delta_t] + (\eta + \frac{2L}{n})\alpha.\end{aligned}$$

Therefore, Eq. (B.2) follows. □

Appendix C. Experiments setups

We conduct experiments by adversarially training a ResNet-18 (He et al., 2016) on common datasets and their poisoned counterparts under different data poisoning attacks.

C.1. Data augmentation

For CIFAR-10 and CIFAR-100, we perform RandomHorizontalFlip, RandomCrop(32, 4) on the training set and Normalize on both the training set and test set. For Tiny-ImageNet, we perform RandomHorizontalFlip on the training set and Normalize on both the training set and test set. For SVHN, we perform only Normalize on both the training set and test set.

C.2. Adversarial training

We use the cross entropy loss as the original loss l. We adopt the 10-step projection gradient descent (PGD-10) (Madry et al., 2017) to generate adversarial examples. The adversarial budget is ϵ and the step size is $\epsilon/4$ in L_∞-norm. We report robust accuracy as the ratio of correctly classified adversarial examples generated by PGD-10, and the robust generalization gap as the gap between robust training accuracy and robust test accuracy.

We use the SGD optimizer in PyTorch and set the momentum and weight decay to be 0.9 and 5×10^{-4} respectively. For all four datasets, the batch sizes of data loaders are set to 128. In Figs. 1 and 2, to illustrate the robust overfitting phenomenon, we run SGD with an initial learning rate of 0.1 that decays by a factor of 0.1 at the 100 and 150 epochs. In other experiments, we adopt the constant learning rate of 0.01. We run AT for 50 epochs on SVHN and for 200 epochs on CIFAR-10, CIFAR-100 and Tiny-ImageNet.

C.3. Poisoning details

We introduce different poisoning attacks used in our experiments on CIFAR-10 and CIFAR-100. In order to simulate the poisoned distribution $\mathcal{P}_{\#}D$, we generate the poisoned training set and test set simultaneously.

EM (error minimizing noise). Huang et al. (2021) proposed a min-min bi-level optimization to generate error-minimizing noises on the training set. Such noises prevent deep learning models from learning information about the clean distribution from the poisoned training data. Formally:

$$\min_\theta \frac{1}{n}\sum_{i=1}^{n} \min_{\|\delta_i\| \leq \epsilon'} l(f_\theta(x_i + \delta_i), y_i),$$

where ϵ' is the poisoning budget. The trained noise generator f_θ generates an unlearnable example (x_{em}, y) with respect to the clean datum (x, y) such that $x_{em} = x + \arg\min_{\|\delta_i\| \leq \epsilon'} l(f_\theta(x + \delta), y)$. PGD-10 is employed for solving the minimization problem. It is worth noting that in our experiments, we combine the training set and test set together to train noise generator f_θ in order to obtain the poisoned training set and poisoned test set coming from the same shifted distribution.

REM (robust error minimizing noise). Fu et al. (2021) further proposed robust minimizing noise in order to protect data from adversarial training, which also can degrade the test robustness. Formally:

$$\min_\theta \frac{1}{n}\sum_{i=1}^{n} \min_{\|\delta_i^u\| \leq \epsilon'} \max_{\|\delta_i^a\| \leq \rho_a} l(f_\theta(x + \delta_i^u + \delta_i^a), y),$$

where ϵ' and ρ_a are poisoning budget and adversarial perturbation budget. ρ_a controls the protection level against adversarial training. For REM, we set $\rho_a = 2/255$. The trained noise generator f_θ generates a robust unlearnable example (x_{rem}, y) with respect to the clean datum (x, y) such that $x_{rem} = x + \arg\min_{\|\delta_i^u\| \leq \epsilon'} \max_{\|\delta_i^a\| \leq \rho_a} l(f_\theta(x + \delta^u + \delta^a), y)$. It is worth noting that in our experiments, we combine the training set and test set together to train noise generator f_θ in order to obtain the poisoned training set and poisoned test set coming from the same shifted distribution.

Following Fu et al. (2021), the source model is trained with SGD for 5000 iterations, with batch size of 128, momentum of 0.9, weight decay of 5×10^{-4}, an initial learning rate of 0.1, and a learning rate scheduler that decays the learning rate by a factor of 0.1 every 2000 iterations. The inner minimization and maximization use PGD-10 to approximate. For EOT, the data transformation T is set as the data augmentation of the corresponding dataset, and the repeated sampling number for expectation estimation is set as 5.

ADV (adversarial perturbation). Tao et al. (2021) and Fowl et al. (2021) both proposed that adding adversarial perturbations to the training data is effective to degrade the test performance of a naturally

trained model. In our experiments, we follow their class-targeted adversarial attack. Let K be the number of data classes. We choose fixed target permutation $t = (y+1) \mod K$ according to source label y. Then add a small adversarial perturbation to x in order to force a naturally trained model to classify it as the wrong label t. Formally:

$$x_{adv} = \arg\min_{\|\delta\| \le \epsilon'} l(f_\theta(x+\delta), t),$$

where f_θ is a classifier naturally trained on the combination of the training set and test set, and ϵ' is the poisoning budget. For the minimization problem, we adopt PGD-100 which is enough to generate strong poisons.

HYP (hypocritical perturbation). Tao et al. (2022) proposed adding hypocritical perturbation on training data to degrade the test robustness of an adversarially trained model. Before generating the poisons, a crafting model is adversarially trained with a crafting budget $\epsilon = 2/255$ for 10 epochs. Then generate hypocritical noises within the perturbation budget ϵ' which can mislead the learner by reinforcing the non-robust features. Formally:

$$x_{hyp} = \arg\min_{\|\delta\| \le \epsilon'} l(f_\theta(x+\delta), y),$$

where f_θ is the crafting model. Like in ADV, we choose PGD-100 to solve the minimization problem. It is worth mentioning that we trained the crafting model on the combination of the training set and test set.

RAN (class-wise random perturbation).

In our experiment, we generate a random perturbation $p_y \in B(0, \epsilon')$ for each label y according to the uniform distribution. Then we have poisoned pairs $(x + p_y, y)$. It is important that we choose the same class-wise random perturbation for the training set and test set.

Data availability

Data will be made available on request.

References

Allen-Zhu, Z., Li, Y., & Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International conference on machine learning* (pp. 242–252).

Bassily, R., Feldman, V., Guzmán, C., & Talwar, K. (2020). Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, *33*, 4381–4391.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., et al. (2013). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 387–402). Springer.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, *2*, 499–526.

Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., & Wang, Z. (2020). Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 699–708).

Chen, T., Zhang, Z., Liu, S., Chang, S., & Wang, Z. (2020). Robust overfitting may be mitigated by properly learned smoothening. In *International conference on learning representations*.

Chen, T., Zhang, Z., Wang, P., Balachandra, S., Ma, H., Wang, Z., et al. (2022). Sparsity winning twice: Better robust generaliztion from more efficient training. arXiv preprint arXiv:2202.09844.

Du, S., Lee, J., Li, H., Wang, L., & Zhai, X. (2019). *Gradient descent finds global minima of deep neural networks*, *International conference on machine learning* (pp. 1675–1685).

Farnia, F., & Ozdaglar, A. (2021). Train simultaneously, generalize better: Stability of gradient-based minimax learners. In *International conference on machine learning* (pp. 3174–3185).

Feng, J., Cai, Q.-Z., & Zhou, Z.-H. (2019). Learning to confuse: generating training time adversarial data with auto-encoder. *Advances in Neural Information Processing Systems*, *32*.

Fowl, L., Goldblum, M., Chiang, P.-y., Geiping, J., Czaja, W., & Goldstein, T. (2021). Adversarial examples make strong poisons. arXiv preprint arXiv:2106.10807.

Fu, S., He, F., Liu, Y., Shen, L., & Tao, D. (2021). Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International conference on learning representations*.

Gao, X.-S., Liu, S., & Yu, L. (2022). Achieving optimal adversarial accuracy for adversarial deep learning using stackelberg games. *Acta Mathematica Scientia*, *42B*(6), 2399–2418.

Ge, R., Huang, F., Jin, C., & Yuan, Y. (2015). Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory* (pp. 797–842).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., & Mann, T. A. (2021). Improving robustness using generated data. *Advances in Neural Information Processing Systems*, *34*, 4218–4233.

Hardt, M., Recht, B., & Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning* (pp. 1225–1234).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Huang, H., Ma, X., Erfani, S. M., Bailey, J., & Wang, Y. (2021). Unlearnable examples: Making personal data unexploitable. arXiv preprint arXiv:2101.04898.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, *32*.

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*: *Technical report TR-2009*.

Kuzborskij, I., & Lampert, C. (2018). Data-dependent stability of stochastic gradient descent. In *International conference on machine learning* (pp. 2815–2824).

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

Lemire Paquin, A., Chaib-draa, B., & Giguère, P. (2022). Stability analysis of stochastic gradient descent for homogeneous neural networks and linear classifiers. Available at SSRN 4247146.

Li, B., Jin, J., Zhong, H., Hopcroft, J., & Wang, L. (2022). Why robust generalization in deep learning is difficult: Perspective of expressive power. *Advances in Neural Information Processing Systems*, *35*, 4370–4384.

Liu, C., Salzmann, M., Lin, T., Tomioka, R., & Süsstrunk, S. (2020). On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, *33*, 21476–21487.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

Moosavi-Dezfooli, S.-M., Fawzi, A., & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2574–2582).

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, *19*(4), 1574–1609.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS workshop on deep learning and unsupervised feature learning*.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 427–436).

Rice, L., Wong, E., & Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *International conference on machine learning* (pp. 8093–8104).

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., & Madry, A. (2018). Adversarially robust generalization requires more data. *Advances in neural information processing systems*, *31*.

Shaham, U., Yamada, Y., & Negahban, S. (2015). Understanding adversarial training: Increasing local stability of neural nets through robust optimization. arXiv preprint arXiv:1511.05432.

Shalev-Shwartz, S., Shamir, O., Srebro, N., & Sridharan, K. (2010). Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, *11*, 2635–2670.

Sinha, A., Namkoong, H., Volpi, R., & Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. arXiv preprint arXiv:1710.10571.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., et al. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Tao, L., Feng, L., Wei, H., Yi, J., Huang, S.-J., & Chen, S. (2022). Can adversarial training be manipulated by non-robust features? *Advances in Neural Information Processing Systems*, *35*, 26504–26518.

Tao, L., Feng, L., Yi, J., Huang, S.-J., & Chen, S. (2021). Better safe than sorry: Preventing delusive adversaries with adversarial training. *Advances in Neural Information Processing Systems*, *34*, 16209–16225.

Wang, Q., Liu, F., Han, B., Liu, T., Gong, C., Niu, G., et al. (2021). Probabilistic margins for instance reweighting in adversarial training. In *Proceedings of advances in neural information processing systems*.

Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., & Yan, S. (2023). Better diffusion models further improve adversarial training. arXiv preprint arXiv:2302.04638.

Wang, Z., Wang, Y., & Wang, Y. (2021). Fooling adversarial training with inducing noise. arXiv preprint arXiv:2111.10130.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., & Gu, Q. (2019). Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*.

Wen, R., Zhao, Z., Liu, Z., Backes, M., Wang, T., & Zhang, Y. (2023). Is adversarial training really a silver bullet for mitigating data poisoning? In *International conference on learning representations*.

Wu, D., Xia, S.-T., & Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, *33*, 2958–2969.

Xiao, J., Fan, Y., Sun, R., Wang, J., & Luo, Z.-Q. (2022). Stability analysis and generalization bounds of adversarial training. In *36th conference on neural information processing systems*.

Xing, Y., Song, Q., & Cheng, G. (2021). On the algorithmic stability of adversarial training. *Advances in Neural Information Processing Systems*, *34*, 26523–26535.

Yu, L., & Gao, X. S. (2023). Improve the robustness and accuracy of deep neural network with $L_{2,\infty}$ normalization. *Journal of Systems Science and Complexity*, *36*(1), 3–28.

Yu, C., Han, B., Shen, L., Yu, J., Gong, C., Gong, M., et al. (2022). Understanding robust overfitting of adversarial training and beyond. In *International conference on machine learning* (pp. 25595–25610).

Yu, D., Zhang, H., Chen, W., Yin, J., & Liu, T.-Y. (2022). Availability attacks create shortcuts. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2367–2376).

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning* (pp. 7472–7482).