Improving Adversarial Training for Multiple Perturbations through the Lens of Uniform Stability

Jiancong Xiao [1] **Zeyu Qin** [2] **Yanbo Fan** [3] **Baoyuan Wu** [4] **Jue Wang** [3] **Zhi-Quan Luo** [4]

Abstract

In adversarial training (AT), most existing works focus on AT with a single type of perturbation, such as the ℓ_∞ attacks. However, deep neural networks (DNNs) are vulnerable to different types of adversarial examples, necessitating the development of adversarial training for multiple perturbations (ATMP) (Tramèr & Boneh, 2019). Despite the benefits of ATMP, there exists a trade-off between different types of attacks. Furthermore, there is a lack of theoretical analyses of ATMP, which hinders its further development. To address these issues, we conduct a smoothness analysis of ATMP. Our analysis reveals that ℓ_1, ℓ_2, and ℓ_∞ adversaries contribute differently to the smoothness of the loss function in ATMP. Leveraging these smoothness properties, we investigate the improvement of ATMP through the lens of uniform stability. Through our research, we demonstrate that employing an adaptive smoothness-weighted learning rate leads to enhanced uniform stability bounds, thus improving adversarial training for multiple perturbations. We validate our findings through experiments on CIFAR-10 and CIFAR-100 datasets, where our approach achieves competitive performance against various mixtures of multiple perturbation attacks. This work contributes to a deeper understanding of ATMP and provides practical insights for improving the robustness of DNNs against diverse adversarial examples.

[1]University of Pennsylvania [2]Hong Kong University of Science and Technology [3]Tencent AI Lab, China [4]Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China. Correspondence to: Yanbo Fan <fanyanbo0124@gmail.com>.

2^{nd} AdvML Frontiers workshop at 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Deep neural networks (DNNs) have been shown to be vulnerable to adversarial examples (Goodfellow et al., 2014; Szegedy et al., 2013), where small and malicious perturbations can cause incorrect predictions. Adversarial training (AT) (Madry et al., 2017) has emerged as one of the most effective methods for increasing the robustness of DNNs against adversarial attacks. AT involves augmenting training data with ℓ_p norm-bounded adversarial examples. However, existing works primarily focus on adversarial training with a single type of attack, such as the ℓ_∞ attack (Raghunathan et al., 2019; Gowal et al., 2020). Recent research (Tramèr & Boneh, 2019) has experimentally demonstrated that DNNs trained with a single type of adversarial attack may not provide sufficient defense against other types of adversarial examples.

To illustrate this, Figure 1 (a) presents an example using CIFAR-10. The plot shows that ℓ_1 adversarial training fails to defend against ℓ_2 and ℓ_∞ attacks, resulting in a robust accuracy of 0%. On the other hand, ℓ_∞ adversarial training offers partial defense against ℓ_1 and ℓ_2 attacks, achieving accuracy of 17.19% and 53.91%, respectively. However, its performance is not competitive compared to ℓ_1 and ℓ_2 adversarial training, which achieve accuracy of 89.84% and 61.72%, respectively. These findings highlight the limitations of relying on a single type of adversarial training and the need to address multiple types of adversarial attacks to ensure robustness in DNNs.

To enhance robustness against various types of attacks, Tramer et al. (Tramèr & Boneh, 2019) introduce adversarial training for multiple perturbations (ATMP), particularly focusing on the ℓ_1, ℓ_2, and ℓ_∞ attacks. They consider two types of objective functions. The first one is the average of all perturbations (AVG), where the inner maximization problem aims to find adversarial examples for each attack type. The second one is the worst-case perturbation (WST), where the inner maximization problem seeks adversarial examples with the highest loss within the union of ℓ_p norm balls. Several algorithms have been proposed to address these problems. Notable works include multi-steepest descent (MSD) (Maini et al., 2020) and stochastic adversarial training (SAT) (Madaan et al., 2020), which employ dif-

ferent strategies to find adversarial examples within the ℓ_p-norm balls and achieve improvements over the MAX and AVG approaches.

However, there exist several crucial issues that are unsolved with respect to ATMP. Firstly, the optimization process of ATMP is highly unstable compared to that of AT or standard training, as illustrated in Figure 1 (c) and (d). The robust test accuracy exhibits significant fluctuations across different training epochs. Secondly, achieving a satisfying trade-off between different attacks proves to be quite challenging. As shown in Figure 1 (b), none of the methods achieve the best performance against all three attacks. This could be attributed to the issue of robust overfitting (Rice et al., 2020), where the models tend to overfit to one type of ℓ_p adversarial examples, resulting in poor performance on other types. Lastly, there is a noticeable dearth of theoretical studies on ATMP. The exploration of ATMP methods is largely driven by experimental design, lacking strong theoretical guidelines.

In this work, we first study the smoothness and the loss landscape of ATMP. We show that the smoothness of ℓ_1, ℓ_2, and ℓ_∞ adversaries give different contributions to the smoothness of ATMP. It motivates us to study a question:

How to use the smoothness properties of different ℓ_p adversaries to design algorithms for ATMP?

In our research, we investigate this question by employing the concept of uniform stability. Notably, recent studies (Xing et al., 2021; Xiao et al., 2022b;e) have explored the uniform stability of adversarial training and have demonstrated that the resulting bounds align with the observed phenomenon of robust overfitting. These studies have provided evidence that the smoothness of the adversarial loss utilized in adversarial training may be a contributing factor to the problem of robust overfitting. Consequently, this finding serves as motivation to leverage the smoothness properties of different ℓ_p norms in the development of algorithms for Adversarial Training with Multiple Perturbations (ATMP), with the aim of achieving enhanced uniform stability and thereby improving the robust performance of the models.

To utilize the smoothness properties of different ℓ_p adversaries, we consider a smoothness-weighted learning rate. We show that such a learning rate schedule yields better stability bounds. Formally, see our main results in Thm. 2 and Thm. 3.

Given the unknown smoothness, particularly the gradient Lipschitz of each adversarial loss on different ℓ_p adversaries, we propose a novel approach to adaptively estimate the smoothness. Consequently, we introduce adaptive smoothness-weighted adversarial training for multiple perturbations. Through extensive experiments conducted on CIFAR-10 and CIFAR-100, we demonstrate that our technique effectively addresses the aforementioned issues and significantly improves the performance of ATMP. Our solution achieves competitive performance against a mixture of multiple perturbation attacks, highlighting its efficacy in enhancing the robustness of models. The workshop version is a shortened version. For the full version of this paper, please refer to (Xiao et al., 2022c).

Our contributions are listed as follows:

1. Smoothness analysis: We conduct a thorough examination of the smoothness properties of adversarial training for both single and multiple perturbations. This analysis provides valuable insights into the behavior of adversarial training methods.

2. Uniform stability analysis: Building upon the smoothness analysis, we perform a comprehensive uniform stability analysis on adversarial training for multiple perturbations (ATMP). This analysis serves as the foundation for our proposed stability-inspired algorithm, namely adaptive smoothness-weighted adversarial training for multiple perturbations.

3. Theoretical insights: We theoretically demonstrate the advantages of using a smoothness-weighted learning rate, which leads to improved stability bounds. This provides a solid theoretical foundation for our proposed algorithm.

4. Experimental validation: Through extensive experiments conducted on the CIFAR-10 and CIFAR-100 datasets, we demonstrate the effectiveness of our approach. We achieve a notable improvement in robust accuracy and achieve competitive performance compared to existing methods, showcasing the practical relevance of our contributions.

2. Related Work

In this section, we first introduce the standard adversarial training with a single type of perturbation, as well as its theoretical analysis. We then introduce the adversarial training against multiple perturbations.

Adversarial robustness against multiple perturbations models Recently, some works have demonstrated that adversarial training with a single type of perturbation cannot provide well defense against other types of adversarial attacks (Tramèr & Boneh, 2019) and several ATMP algorithms have been proposed accordingly (Maini et al., 2020; Madaan et al., 2020; Zhang et al., 2021; Stutz et al., 2020). The work of (Tramèr & Boneh, 2019) proposed to augment different types of adversarial examples into adversarial training and developed two augmentation strategies, *i.e.*, MAX

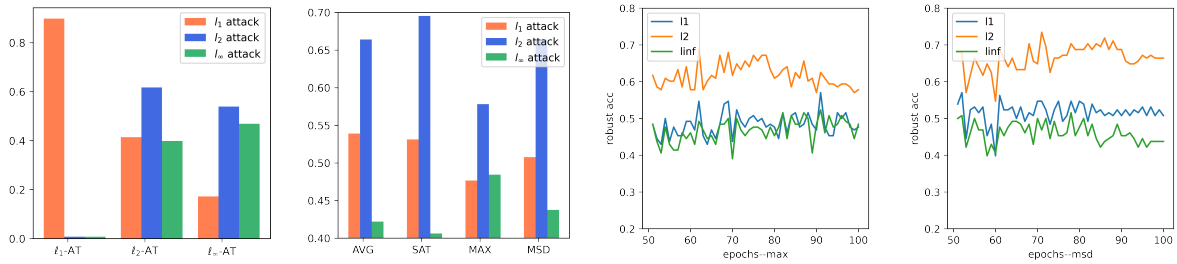


Figure 1. Crucial issues of adversarial training for multiple perturbations. (a) Performance of adversarial training with a single type perturbation against other type attacks. (b) Trade-off between different types of adversaries of four algorithms for ATMP. (c) Robust test accuracy fluctuate between different epochs using MAX. (d) Robust test accuracy fluctuate between different epochs using MSD.

and AVG. The MAX adopts the worst-case adversarial example among different attacks, while the AVG takes all types of adversarial examples into training. Following the above pipeline, some later works developed different aggregation strategies (e.g., the MSD (Maini et al., 2020), and SAT (Madaan et al., 2020)) for better robustness or training efficiency. While these works can boost the adversarial robustness against multiple perturbations to some extent, the training process of ATMP is highly unstable, and there is no theoretical analysis about this. The theoretical understanding of the training difficulty of ATMP is important for the further development of adversarial robustness for multiple perturbations. Besides, there have also been some other works for adversarial robustness against multiple perturbations, such as *Ensemble models* (Maini et al., 2021; Cheng et al., 2021), *Prepossessing* (Nandy et al., 2020) and *Neural architectures search (NAS)* (Liu et al., 2020). The weakness of ensemble models or prepossess methods is that the performance is highly related to the classification quality or detection of different types of adversarial examples. These methods either have lower performance or consider different tasks from the work we considered. Therefore, we mainly compare the algorithms MAX, AVG, MSD, and SAT in this work.

3. Preliminaries of Adversarial Training for Multiple Perturbations

Adversarial training is an approach to train a classifier that minimizes the worst-case loss within a norm-bounded constraint. Let $g(\theta, z)$ be the loss function of the standard counterpart. Given training dataset $\mathcal{S} = \{z_i\}_{i=1\cdots n}$, the optimization problem of adversarial training is

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{\|z_i - z_i'\|_p \le \epsilon_p} g(\theta, z_i'), \tag{3.1}$$

where ϵ_p is the perturbation threshold, $p = 1, 2$ or ∞ for different types of attacks. Usually, g can also be written in the form of $\ell(f_\theta(\boldsymbol{x}), y)$, where f_θ is the neural network to be trained and $(\boldsymbol{x}, y)$ is the input-label pair. Adversarial training aims to train a model against a single type of ℓ_p attack. As AT with a single type of attacks may not be effecting under other types of attacks, adversarial training for multiple perturbations are proposed (Tramèr & Boneh, 2019). Following the aforementioned literature, we consider the case that $p = 1, 2, \infty$. Two formulations can be use to tackle this problem.

Worst-case perturbation (WST). The optimization problem of WST is formulated as follow,

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{p \in \{1,2,\infty\}} \max_{\|z_i - z_i'\|_p \le \epsilon_p} g(\theta, z_i'). \tag{3.2}$$

WST aims to find the worst adversarial examples within the union of the three norm constraints for the inner maximization problem. The outer minimization problem updates model parameters θ to fit these adversarial examples. The MAX strategy (Tramèr & Boneh, 2019) are proposed for the optimization problem in Eq. (3.2). In each inner iteration, MAX takes the maximum loss on these three adversarial examples. Another algorithm for the optimization problem in Eq. (3.2) is multi-steepest descent (MSD) (Maini et al., 2020). In each PGD step in the inner iteration, MSD selects the worst among ℓ_1, ℓ_2, and ℓ_∞ attacks.

Average of all perturbations (AVG). The optimization problem of AVG is formulated as follow

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p \sim \{1,2,\infty\}} \max_{\|z_i - z_i'\|_p \le \epsilon_p} g(\theta, z_i'), \tag{3.3}$$

where $p \sim \{1, 2, \infty\}$ uniformly at random. The goal of the minimax problem in Eq. (3.3) is to train the neural networks using data augmented with all three types of adversarial examples. The AVG strategy (Tramèr & Boneh, 2019) and the stochastic adversarial training (SAT) (Madaan et al., 2020) are two algorithms to solve the problem in Eq. (3.3). In each inner iteration, AVG takes the average loss on these three adversarial examples and SAT randomly chooses one type of adversarial example among ℓ_1, ℓ_2, and ℓ_∞ attacks.

Problem WST and AVG are similar but slightly different problems. WST aims to defend union attacks, *i.e.,* the optimal attack within the union of multiple perturbations. AVG aims to defend mixture attacks, *i.e.,* the attacker randomly pick one ℓ_p attack. In this paper, we mainly focus on problem AVG.

4. Stability-based Excess Risk Analysis

In App. C, we first prove that the smoothness of different ℓ_p adversaries are different. In this section, we focus on the problem AVG.

We use $R^{st}_{\mathcal{D}}(\theta)$ and $R^{st}_{S}(\theta)$ to denote the population and empirical risk for adversarial training with different strategy, *i.e.* st $\in \{1, 2, \infty, \text{wst}, \text{avg}\}$. Assuming that the target model are facing P potential attacks. In the above setting, $P = 3$ and the three attacks are ℓ_1, ℓ_2, and ℓ_∞ attacks. The test and training performance against mixture attacks is

$$R_{\mathcal{D}}(\theta) = \frac{1}{P}\sum_{p=1}^{P} R^p_{\mathcal{D}}(\theta) \text{ and } R_S(\theta) = \frac{1}{P}\sum_{p=1}^{P} R^p_S(\theta),$$

respectively, where $R^p_{\mathcal{D}}(\theta)$ and $R^p_S(\theta)$ are the population and empirical risk against the p^{th} attack.

Risk Decomposition. Let θ^* and $\bar{\theta}$ be the optimal solution of $R_{\mathcal{D}}(\theta)$ and $R_S(\theta)$, respectively. Then for the algorithm output $\hat{\theta} = A(S)$, the excess risk can be decomposed as

$$\begin{aligned} &R_{\mathcal{D}}(\hat{\theta}) - R_{\mathcal{D}}(\theta^*) \\ =&\underbrace{R_{\mathcal{D}}(\hat{\theta}) - R_S(\hat{\theta})}_{\mathcal{E}_{gen}} + \underbrace{R_S(\hat{\theta}) - R_S(\bar{\theta})}_{\mathcal{E}_{opt}} \\ +&\underbrace{R_S(\bar{\theta}) - R_S(\theta^*)}_{\leq 0} + \underbrace{R_S(\theta^*) - R_{\mathcal{D}}(\theta^*)}_{\mathbb{E}=0}. \end{aligned} \tag{4.1}$$

To control the excess risk, we need to control the generalization gap $\mathcal{E}_{gen}$ and the optimization gap $\mathcal{E}_{opt}$. In the rest of the paper, we use $\mathcal{E}_{gen}$ and $\mathcal{E}_{opt}$ to denote the *expectation* of the generalization and optimization gap. The smoothness of the loss function is highly related to the generalization gap $\mathcal{E}_{gen}$ and the optimization gap $\mathcal{E}_{opt}$, we first provide the the optimization error bound (Nemirovski et al., 2009) and stability-based generalization bound[1] (Hardt et al., 2016) for running SGD on Eq. (3.3).

Theorem 1. *Under Assumption 2, assuming in addition that $g(\theta, z)$ is convex in θ for all given $z \in \mathcal{Z}$. Let $D = \|\theta^0 - \theta^*\|$, where θ^0 is the initialization of SGD. Suppose that we run SGD with step sizes $\alpha \leq 1/\beta_{avg}$ for T steps. Then, adversarial training satisfies*

$$\mathcal{E}_{opt} \leq \frac{D^2 + L^2 T\alpha^2}{2T\alpha}, \ \mathcal{E}_{gen} \leq \frac{2L^2 T\alpha}{n}.^{2} \tag{4.2}$$

Therefore, the β-gradient Lipschitz constant of the loss function is related to the choice of stepsize, the optimization and generalization bound. In the loss function of AVG, each of the ℓ_p adversarial loss have different Lipschitz constant β_p. It motivates us to study whether we can assign different stepsize to different ℓ_p adversarial loss to improve the excess risk.

4.1. Smoothness-weighted Adversarial Training for Multiple Perturbations

Considering the algorithm

$$\theta^{t+1} = \theta^t - \frac{1}{P}\left[\alpha^t_1 \nabla R^1_S(\theta^t) + \cdots + \alpha^t_P \nabla R^P_S(\theta^t)\right], \tag{4.3}$$

In each of the iterations t, we assign stepsize α^t_p to the p^{th}-tasks.

Properties of Update Rules. We define $G_z(\theta) = \theta - \frac{1}{P}\sum_{p=1}^{P} \alpha_p \nabla h^p(\theta, z)$ be the update rule. The following lemma holds.

Lemma 1 (Non-expansive)**.** *Assuming that the function $h^p(\theta, z)$ is β_p-gradient Lipschitz, convex for all $z \in \mathcal{Z}$. Then, $\forall \theta_1, \theta_2$ and $\forall z \in \mathcal{Z}$, for $\alpha_p \leq 1/\beta_p$, we have $\|G_z(\theta_1) - G_z(\theta_2)\| \leq \|\theta_1 - \theta_2\|$.*

Proof of Lemma 1 is deferred to App. A. Based on Lemma 1, we have the following generalization guarantee for problem AVG.

Theorem 2 (Generalization error bounds of smoothness-weighted learning rate)**.** *Under Assumption 2, assume in addition that $h^p(\theta, z)$ is convex in θ for all given $z \in \mathcal{Z}$. Suppose that we run Algorithm 1 with step sizes $\alpha_p \leq 1/\beta_p$ for T steps. Then, adversarial training satisfies uniform stability with*

$$\mathcal{E}_{gen} \leq \frac{2L^2 T \sum_{p=1}^{P} \alpha_p / P}{n}.$$

Proof: The proof is based on Lemma 1 and defer to App. A.

Let $\alpha^t_{sw} = (\alpha^t_1 + \cdots + \alpha^t_P)/P$, we have $\mathcal{E}_{gen} \leq 2L^2 T\alpha_{sw}/n$.

Theorem 3 (Optimization error bounds of smoothness-weighted learning rate)**.** *Under Assumption 2, assume in addition that $h^p(\theta, z)$ is convex in θ for all given $z \in \mathcal{Z}$.*

[1] We refer the readers to (Hardt et al., 2016) for the preliminaries of uniform stability.

[2] For varing stepsize, we can replace $T\alpha$ and $T\alpha^2$ by $\sum_{t=1}^{T} \alpha_t$ and $\sum_{t=1}^{T} \alpha^2_t$, respectively.

Suppose that we run SGD with step sizes $\alpha_p^t \leq 1/\beta_p$ for T steps. Then, adversarial training satisfies

$$\mathcal{E}_{opt} \leq \frac{D^2 + L^2 T \alpha_{sw}^2}{2T\alpha_{sw}} + B\frac{\sum_{p=1}^P |\alpha_{sw} - \alpha_p|}{\alpha_{sw}}.$$

The proof is deferred to App. A. The first term has the same form as Theorem 1, the second term is an additional bias term introduced by the smoothness-weighted learning rate. Combining the $\mathcal{E}_{opt}$ and $\mathcal{E}_{gen}$, we have

$$\begin{aligned}&\mathcal{E}_{gen} + \mathcal{E}_{opt}\\ \leq &\underbrace{\frac{2L^2 T\alpha_{sw}}{n} + \frac{D^2 + L^2 T\alpha_{sw}^2}{2T\alpha_{sw}}}_{\text{The same as Thm. 1 with different } \alpha} + \underbrace{B\frac{\sum_{p=1}^P |\alpha_{sw} - \alpha_p|}{\alpha_{sw}}}_{\text{bias term}}.\end{aligned}$$

Optimizing the first two terms with respective to α, we have

$$\alpha^* = \frac{D\sqrt{n}}{L\sqrt{T(n+2T)}}.$$

In adversarial training, T cannot be too large because of robust overfitting. Then, The right-hand-side may be too large and we may not choose α^* as the learning rate. We need a larger α to reduce the first two term. From the previous dicussion, we have

$$\alpha_{avg} \leq \frac{P}{\beta_1 + \cdots + \beta_P} \text{ and } \alpha_{sw} \leq \frac{\frac{1}{\beta_1} + \cdots + \frac{1}{\beta_P}}{P}.$$

Therefore, α_{sw} can be view as the inverse of the harmonic mean of β_p and α_{avg} can be view as the inverse of the arithmetic mean of β_p. α_{sw} is larger and reduce the first two terms when T is small.

Overall, we need smaller T and carefully chosen learning rates to speed up adversarial training to avoid robust overfitting. Smoothness-weighted learning rate gives us a way to increase the learning rate. As a side-effect, it introduce an additional bias term. In experiments, we will show that α_{sw} can improve the test performance.

4.2. Adaptive Smoothness Estimation

In practice, β_p are unknown. We adaptively estimate β_p in each iterations. The main idea to estimate the smoothess comes from the descent Lemma, which is

$$\begin{aligned}&h^p(\theta_t) - h^p(\theta^*)\\ \leq &\nabla h^p(\theta^*)\langle \theta_t - \theta^*\rangle + \frac{\beta_p}{2}\|\theta_t - \theta^*\|^2 = \frac{\beta_p}{2}\|\theta_t - \theta^*\|^2.\end{aligned}$$

Organize the terms, we have

$$\frac{1}{\beta_p} \leq \frac{\|\theta_t - \theta^*\|^2}{2(h^p(\theta_t) - h^p(\theta^*))}.$$

Therefore, we use the right-hand-side of the above inequality to estimate the proportion between different α_p^t, *i.e.,*

$$\alpha_p^t \propto \frac{1}{\beta_p} \propto \frac{\|\theta_t - \theta^*\|^2}{h^p(\theta_t) - h^p(\theta^*)} \propto \frac{1}{h^p(\theta_t) - h^p(\theta^*)},$$

where $\|\theta_t - \theta^*\|^2$ is omitted because it does not depended on p. Assume that $h^p(\theta^*) = 0$, we can use $\sum_p h^p(\theta_t)/(Ph^p(\theta_t))$ as the weight of α_p^t. Given the initial learning rate schedule $\alpha^1, \cdot, \alpha^T$, the following Algorithm 1 is the adaptive smoothness-weighted ATMP.

Algorithm 1 Adaptive Smoothness-Weighted ATMP

Inputs: classifier $f_\theta(\boldsymbol{x}, y)$, dataset $\{\boldsymbol{x}_i, y_i\}_{i=1,\cdots,n}$.
Initialize learning rate schedule $\alpha^1, \cdot, \alpha^T$.
for $t = 1$ to T **do**
  for $p = 1$ to P **do**

$$\mathcal{L}_p = \frac{1}{n}\sum_{i=1}^n \max_{\|\delta_i\|_p \leq \epsilon_p} \ell(f_{\theta_t}(\boldsymbol{x}_i + \delta_i, y_i).$$

  end for
  Define $\mathcal{L} = \mathcal{L}_1 + \cdots + \mathcal{L}_P$.
  Update $\alpha_p^t = \alpha^t \times \mathcal{L}/P\mathcal{L}_p$, $p = 1, 2, \cdots, P$.
  Update

$$\theta^{t+1} = \theta^t - \frac{1}{P}\left[\alpha_1^t \nabla R_S^1(\theta^t) + \cdots + \alpha_P^t \nabla R_S^P(\theta^t)\right],$$

end for

5. Experiments

5.1. Performance of ADT

Datasets and Classification Models. We conduct experiments on two widely used benchmark datasets: CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). CIFAR-10 includes 50k training images and 10k test images with 10 classes. CIFAR-100 includes 50k training images and 10k test images with 100 classes. For classification models, we use PreActRes-18 (He et al., 2016). Code is available at `https://github.com/JiancongXiao/Adaptive-Smoothness-Weighted-AT`.

Evaluation Protocol. We consider two formulations, WST in Eq. (3.2) and AVG in Eq. (3.3). We use ADT to stand for our proposed algorithm. Since ADT is designed for defending mixture attacks, we mainly use Mix to evaluate the performance. We also provide the performance against union attacks.

Comparison of AVG, SAT, and ADT. We mainly compare the three methods for the problem in Eq. (3.3). The

Table 1. Test accuracy (%) of different algorithms (MAX, AVG, MSD, SAT, and ADT) against ℓ_1, ℓ_2, and ℓ_∞ attacks on CIFAR-10.

Dataset		CIFAR-10					
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞	Union	Mix
AT	ℓ_1	93.22	89.81	0.00	0.00	0.00	29.98
	ℓ_2	88.66	41.41	61.72	39.84	18.41	47.66
	ℓ_∞	84.94	17.19	53.91	46.88	40.11	39.32
WST (Eq. 3.2)	MAX	84.96	52.63	64.74	46.93	46.08±0.43	54.77±0.22
	MSD	83.51	54.92	67.68	49.88	46.99±0.23	57.49±0.11
AVG (Eq. 3.3)	AVG	85.28	58.78	68.08	43.87	43.28±0.55	56.91±0.34
	SAT	85.23	58.68	67.77	43.59	43.12±1.89	56.68±1.01
	ADT	85.87	**61.81**	**69.61**	**46.64**	**46.05**±0.31	**59.16**±0.09

Table 2. Test accuracy (%) of different algorithms (AVG, SAT, and ADT) against ℓ_1, ℓ_2, and ℓ_∞ AutoAttacks on CIFAR-10/100.

CIFAR-10	ℓ_1	ℓ_2	ℓ_∞	Mix
AVG	46.81	64.01	42.82	51.21
SAT	47.72	64.21	42.17	51.37
ADT	49.07	65.21	44.43	52.90
CIFAR-100	ℓ_1	ℓ_2	ℓ_∞	Mix
AVG	29.72	38.31	20.19	29.40
SAT	29.09	39.32	22.32	30.24
ADT	31.22	39.45	22.86	31.18

highest numbers are in bold. Mixture attack (average of all attacks) is the main index to evaluate the performance. On the CIFAR-10 dataset, we observe that ADT achieves the highest robust accuracy at 59.16%. Similarly, on CIFAR-100, ADT demonstrates a robust accuracy of 35.39%. These results indicate that ADT has the capability to automatically adapt to cases that may not have been well-optimized. Furthermore, ADT exhibits a smaller deviation compared to SAT and AVG. To further analyze the effectiveness of the techniques employed, ablation studies are conducted. Specifically, we compare AVG, SAT, and ADT with and without early stopping and SWA techniques, as shown in Table 4. The results demonstrate that ADT consistently outperforms the baseline algorithms, even without the inclusion of SWA and early stopping techniques. Finally, we evaluate the performance against AutoAttack, and the corresponding results are presented in Table 2.

5.2. Discussion: different goals of WST and AVG

Comparison of ADT and MSD. The formulations of Eq. (3.3) and (3.2) have similar but slightly different goals. WST tries to fit the adversarial examples who have the largest loss within the union of the three norms. AVG is designed to defend the mixture attacks. The difference in ADT and MSD is the difference in the optimization problems AVG and WST. In Table 1, ADT achieves competitive performance, 46%, on union of all the attacks. In terms of mixture attack, ADT achieves 59% robust accuracy, while the robust accuracy of MSD is 57%.

Overall, ℓ_∞ adversarial examples induce larger norm within the union of the three norms, and MSD tends to find and fit them. ADT pays more attention to ℓ_1 adversarial examples. Comparing the overall performance, ADT achieves better robustness trade-off against mixture adversarial attacks.

Solutions for WST. Our paper mainly focus on problem AVG. We also discuss some solutions to improve the performance of WST. In Fig. 2, we plot the robust accuracy against ℓ_1, ℓ_2, and ℓ_∞ adversarial attacks of four different strategies (MAX, AVG, MSD and SAT) on CIFAR-10. It shows that SWA is an effective methods to improve the performance of ATMP. For instance, in subplot (a), the l1 / l2 / linf denotes the robust accuracy of ATMP trained with AVG strategy against the ℓ_1 / ℓ_2 / $\ell_{\inf}$ adversarial attacks. While the SWA_l1 / SWA_l2 / SWA_linf relates to the ATMP model that trained by AVG strategy coupled with SWA. From the plots, we observe that the test accuracy is highly unstable among different training epochs without SWA. When coupling with SWA, the tendency curves of all four ATMP strategies are largely stabilized. Using early stopping, we could find the checkpoint for the best performance. On CIFAR-10, the improvement of SWA is 1.82%, 3.39%, 1.56%, and 2.67% using MSD, SAT, AVG, and MAX, respectively.

6. Conclusion

In this paper, we study the smoothness of adversarial loss on different ℓ_p adversaries and try to use this property to improve the performance of adversarial training for multiple perturbations. To this end, we provide a uniform stability analysis and propose adaptive smoothness-weighted adversarial training for multiple perturbations, which achieves better excess risk bound and achieve better performance. Our framework might also be possible to extend to other multi-task learning problems with the following two properties or issues. 1, Each task should be equally important. 2, Training epochs cannot be too large.

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Awasthi, P., Frank, N., and Mohri, M. Adversarial learning guarantees for linear hypotheses and neural networks. In *International Conference on Machine Learning*, pp. 431–441. PMLR, 2020.

Cheng, H., Xu, K., Wang, C., Lin, X., Kailkhura, B., and Goldhahn, R. Mixture of robust experts (more): A flexible defense against multiple perturbations. *arXiv preprint arXiv:2104.10586*, 2021.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Farnia, F. and Ozdaglar, A. Train simultaneously, generalize better: Stability of gradient-based minimax learners. In *International Conference on Machine Learning*, pp. 3174–3185. PMLR, 2021.

Gao, R., Cai, T., Li, H., Hsieh, C.-J., Wang, L., and Lee, J. D. Convergence of adversarial training in overparametrized neural networks. *Advances in Neural Information Processing Systems*, 32:13029–13040, 2019.

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

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

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

Liu, A., Tang, S., Liu, X., Chen, X., Huang, L., Tu, Z., Song, D., and Tao, D. Towards defending multiple adversarial perturbations via gated batch normalization. *arXiv preprint arXiv:2012.01654*, 2020.

Madaan, D., Shin, J., and Hwang, S. J. Learning to generate noise for robustness against multiple perturbations. *arXiv preprint arXiv:2006.12135*, 2020.

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

Maini, P., Wong, E., and Kolter, Z. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pp. 6640–6650. PMLR, 2020.

Maini, P., Chen, X., Li, B., and Song, D. Perturbation type categorization for multiple ℓ_p bounded adversarial robustness, 2021. URL https://openreview.net/forum?id=Oe2XI-Aft-k.

Nandy, J., Hsu, W., and Lee, M. L. Approximate manifold defense against multiple adversarial perturbations. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020.

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

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pp. 5014–5026, 2018.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2, 2017.

Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

Stutz, D., Hein, M., and Schiele, B. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, pp. 9155–9166. PMLR, 2020.

Stutz, D., Hein, M., and Schiele, B. Relating adversarially robust generalization to flat minima. *arXiv preprint arXiv:2104.04448*, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems*, pp. 5858–5868, 2019.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., and Gu, Q. On the convergence and robustness of adversarial training. In *ICML*, volume 1, pp. 2, 2019.

Xiao, J., Fan, Y., Sun, R., and Luo, Z.-Q. Adversarial rademacher complexity of deep neural networks. *arXiv preprint arXiv:2211.14966*, 2022a.

Xiao, J., Fan, Y., Sun, R., Wang, J., and Luo, Z.-Q. Stability analysis and generalization bounds of adversarial training. *Advances in Neural Information Processing Systems*, 35:15446–15459, 2022b.

Xiao, J., Qin, Z., Fan, Y., Wu, B., Wang, J., and Luo, Z.-Q. Adaptive smoothness-weighted adversarial training for multiple perturbations with its stability analysis. *arXiv preprint arXiv:2210.00557*, 2022c.

Xiao, J., Yang, L., Fan, Y., Wang, J., and Luo, Z.-Q. Understanding adversarial robustness against on-manifold adversarial examples. *arXiv preprint arXiv:2210.00430*, 2022d.

Xiao, J., Zhang, J., Luo, Z.-Q., and Ozdaglar, A. E. Smoothed-sgdmax: A stability-inspired algorithm to improve adversarial generalization. In *NeurIPS ML Safety Workshop*, 2022e.

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

Yin, D., Kannan, R., and Bartlett, P. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pp. 7085–7094. PMLR, 2019.

Zhang, X., Zhang, Z., and Wang, T. Composite adversarial training for multiple adversarial perturbations and beyond, 2021. URL https://openreview.net/forum?id=H92-E4kFwbR.

A. Proof of Theorem

In this section, we provide the detailed proof.

A.1. Proof of Proposition 1

Since

$$\begin{aligned} & \|(\boldsymbol{X}+\delta)\theta - y\|^2 \\ \leq\ & [\|\boldsymbol{X}\theta - y\| + \|\delta\theta\|]^2 \\ =\ & [\|\boldsymbol{X}\theta - y\| + \sqrt{\sum |\delta_i\theta|^2}]^2 \\ \leq\ & [\|\boldsymbol{X}\theta - y\| + \sqrt{\sum [\|\delta_i\|_p \|\theta\|_{p*}]^2}]^2 \\ \leq\ & [\|\boldsymbol{X}\theta - y\| + \sqrt{n\epsilon_p^2 \|\theta\|_{p*}^2}]^2 \\ =\ & [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2, \end{aligned}$$

where the first inequality is due to triangle inequality, the second inequality is due to Cauchy-Schwarz inequality, and the last inequality is due to the constraint $\|\delta\|_{p,\infty} \leq \epsilon_p$. Choosing δ_i to satisfy the aforementioned three inequalities, we obtain

$$R_S^p(\theta) = \max_{\|\delta\|_{p,\infty} \leq \epsilon_p} \|(\boldsymbol{X}+\delta)\theta - y\|^2 = [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2.$$

Based on this, we directly have

$$\begin{aligned} R_S^{wst}(\theta) &= \max_{p\in\{1,2,\infty\}} \max_{\|\delta\|_{p,\infty} \leq \epsilon_p} \|(\boldsymbol{X}+\delta)\theta - y\|^2 \\ &= \max_{p\in\{1,2,\infty\}} [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2, \end{aligned}$$

$$\begin{aligned} R_S^{avg}(\theta) &= \mathbb{E}_{p\in\{1,2,\infty\}} \max_{\|\delta\|_{p,\infty} \leq \epsilon_p} \|(\boldsymbol{X}+\delta)\theta - y\|^2 \\ &= \mathbb{E}_{p\in\{1,2,\infty\}} [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2. \end{aligned}$$

□

A.2. Proof of Lemma 2

Proof:
Case 1: st $\in \{1, 2, \infty\}$. The proof can be found in (Sinha et al., 2017; Wang et al., 2019).
Case 2: st $= wst$. Since $g(\theta, z)$ is locally μ_p-strongly concave for all $z \in \mathcal{Z}$ in ℓ_p-norm, $g(\theta, z)$ is locally $\min \mu_p$-strongly concave within the union of ℓ_p norm ball for all $z \in \mathcal{Z}$.
Case 3: st $= avg$. Since $h^{avg}(\theta, z) = \mathcal{E}h^p(\theta, z)$, we have

$$\|\nabla_\theta h^{avg}(\theta_1, z) - \nabla_\theta h^{avg}(\theta_2, z)\| = \|\mathbb{E}(\nabla_\theta h^p(\theta_1, z) - \nabla_\theta h^p(\theta_2, z))\| \leq \mathbb{E}\beta_p \|\theta_1 - \theta_2\|.$$

□

A.3. Discussion on Non-Strongly-Convex Cases

Assumption 1. *The function g satisfies the following Lipschitzian smoothness conditions:*

$$\begin{aligned} &\|g(\theta_1, z) - g(\theta_2, z)\| \leq L\|\theta_1 - \theta_2\|, \\ &\|\nabla_\theta g(\theta_1, z) - \nabla_\theta g(\theta_2, z)\| \leq L_\theta \|\theta_1 - \theta_2\|, \\ &\|\nabla_\theta g(\theta, z_1) - \nabla_\theta g(\theta, z_2)\| \leq L_{\theta z}^p \|z_1 - z_2\|_p, \\ &\|\nabla_z g(\theta_1, z) - \nabla_\theta g(\theta_2, z)\| \leq L_{z\theta} \|\theta_1 - \theta_2\|. \end{aligned}$$

Assumption 1 assumes that the gradient Lipschitz in different ℓ_p-norm are $L^p_{\theta z}$, which can be verified by the relation between norms.

Lemma 1. *Under Assumption 1, $\forall \theta_1, \theta_2$ and $\forall z \in \mathcal{Z}$, the following properties hold.*

1. *(Lipschitz function.)* $\|h^{st}(\theta_1, z) - h^{st}(\theta_2, z)\| \leq L\|\theta_1 - \theta_2\|$.
2. *(Non-gradient Lipschitz.)* $\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \leq L_\theta\|\theta_1 - \theta_2\| + \eta_{st}$, *where* $\eta_p = 2L^p_{\theta z}\epsilon_p$, $\eta_{wst} = 2\max\{L^p_{\theta z}\epsilon_p\}$, *and* $\eta_{avg} = 2\mathbb{E}_p L^p_{\theta z}\epsilon_p$.

Lemma 1.2 and 2.2 show that adversarial surrogate loss in different ℓ_p adversaries have different smoothness in general non-concave case.

Proof: Notice that $R^{st}_S(\theta) = \sum_{i=1}^n h^{st}(\theta, z_i)/n$, we only need to prove that $\forall \boldsymbol{x}$, we have

$$\begin{aligned} &\|h^{st}(\theta_1, z) - h^{st}(\theta_1, z)\| \leq L_\theta\|\theta_1 - \theta_2\|, \\ &\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \leq L_{\theta\theta}\|\theta_1 - \theta_2\| + \eta_{st}, \end{aligned} \tag{A.1}$$

where st $\in \{1, 2, \infty, \text{wst}, \text{avg}\}$ with $\eta_p = 2L^p_{\theta \boldsymbol{x}}\epsilon_p$, $\eta_{wst} = 2\max\{L^p_{\theta \boldsymbol{x}}\epsilon_p\}$, and $\eta_{avg} = 2\mathbb{E}_p L^p_{\theta \boldsymbol{x}}\epsilon_p$.

Case 1: st $\in \{1, 2, \infty\}$:

Let the adversarial examples for parameter θ_1 and θ_2 be

$$\boldsymbol{x}_1 = \arg\max_{\|\delta\| \leq \epsilon_p} g(\boldsymbol{x} + \delta, \theta_1)$$

$$\boldsymbol{x}_2 = \arg\max_{\|\delta\| \leq \epsilon_p} g(\boldsymbol{x} + \delta, \theta_2),$$

then we have

$$\begin{aligned} &\|h^{st}(\theta_1, z) - h^{st}(\theta_1, z)\| \\ =&|g(\theta_1, z_1) - g(\theta_2, z_2)| \\ \leq&\max\{|g(\theta_1, z_1) - g(\theta_2, z_1)|, |g(\theta_1, z_2) - g(\theta_2, z_2)|\} \\ \leq&L_\theta\|\theta_1 - \theta_2\|, \end{aligned}$$

where the first inequality is based on the fact that $g(\theta_1, z_1) \geq g(\theta_1, z_2)$ and $g(\theta_2, z_2) \geq g(\theta_2, z_1)$, the second inequality is based on Assumption 11. This proves the first inequality in equation (A.1) in this case. For the second one in equation (A.1), we have

$$\begin{aligned} &\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \\ =&\|\nabla_\theta h^{st}(\theta_1, z_1) - \nabla_\theta g(\theta_2, z_2)\| \\ \leq&\|\nabla_\theta h^{st}(\theta_1, z_1) - \nabla_\theta g(\theta_2, z_1)\| + \|\nabla_\theta g(\theta_2, z_1) - \nabla_\theta g(\theta_2, z_2)\| \\ \leq&L_{\theta\theta}\|\theta_1 - \theta_2\| + L^p_{\theta \boldsymbol{x}}\|z_1 - z_2\|_p \\ \leq&L_{\theta\theta}\|\theta_1 - \theta_2\| + L^p_{\theta \boldsymbol{x}}[\|z_1 - z\|_p + \|z - z_2\|_p] \\ \leq&L_{\theta\theta}\|\theta_1 - \theta_2\| + 2L^p_{\theta \boldsymbol{x}}\epsilon_p \\ =&L_{\theta\theta}\|\theta_1 - \theta_2\| + \eta_{st}, \end{aligned}$$

where the first and the third inequality is triangle inequality, the second inequality is based on Assumption 1. This proves the second inequality.

Case 2: st = **wst**:

Let the adversarial examples for parameter θ_1 and θ_2 be

$$\boldsymbol{x}_1 = \arg\max_{p \in \{1,2,\infty\}} \max_{\|\delta\| \leq \epsilon_p} g(\boldsymbol{x} + \delta, \theta_1)$$

$$\boldsymbol{x}_2 = \arg\max_{p \in \{1,2,\infty\}} \max_{\|\delta\| \leq \epsilon_p} g(\boldsymbol{x} + \delta, \theta_2),$$

the prove of the first inequality in equation (A.1) is the same as the proof in Case 1. For the second inequality in equation (A.1), we have

$$\begin{aligned}
&\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \\
\leq& L_{\theta\theta}\|\theta_1 - \theta_2\| + L^p_{\theta\boldsymbol{x}}[\|z_1 - z\|_p + \|z - z_2\|_p] \\
\leq& L_{\theta\theta}\|\theta_1 - \theta_2\| + 2\max_{p\in\{1,2,\infty\}}[L^p_{\theta\boldsymbol{x}}\epsilon_p] \\
=& L_{\theta\theta}\|\theta_1 - \theta_2\| + \eta_{st}.
\end{aligned}$$

This proves the second inequality.

Case 3: st = **avg**:

Let the adversarial examples for parameter θ_1 and θ_2 and $p = 1, 2, \infty$ be

$$\boldsymbol{x}_1^p = \arg\max_{\|\delta\|\leq\epsilon_p} g(\boldsymbol{x} + \delta, \theta_1)$$

$$\boldsymbol{x}_2^p = \arg\max_{\|\delta\|\leq\epsilon_p} g(\boldsymbol{x} + \delta, \theta_2).$$

For the first inequality in equation (A.1), we have

$$\begin{aligned}
&\|h^{st}(\theta_1, z) - h^{st}(\theta_1, z)\| \\
=&|\mathbb{E}_{p\sim\{1,2,\infty\}} g(\boldsymbol{x}_1^p, \theta_1) - \mathbb{E}_{p\sim\{1,2,\infty\}} g(\boldsymbol{x}_2^p, \theta_2)| \\
\leq&\mathbb{E}_{p\sim\{1,2,\infty\}}|g(\boldsymbol{x}_1^p, \theta_1) - g(\boldsymbol{x}_2^p, \theta_2)| \\
\leq&\mathbb{E}_{p\sim\{1,2,\infty\}}\max\{|g(\boldsymbol{x}_1^p, \theta_1) - g(\boldsymbol{x}_1^p, \theta_2)|, |g(\boldsymbol{x}_2^p, \theta_1) - g(\boldsymbol{x}_2^p, \theta_2)|\} \\
\leq&\mathbb{E}_{p\sim\{1,2,\infty\}} L_\theta\|\theta_1 - \theta_2\| \\
\leq& L_\theta\|\theta_1 - \theta_2\|,
\end{aligned}$$

where the first inequality is Jensen's inequality. This proves of the first inequality in equation (A.1) in this case. For the second inequality in equation (A.1), we have

$$\begin{aligned}
&\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \\
=&\|\nabla_\theta \mathbb{E}_{p\sim\{1,2,\infty\}} g(\boldsymbol{x}_1^p, \theta_1) - \nabla_\theta \mathbb{E}_{p\sim\{1,2,\infty\}} g(\boldsymbol{x}_2^p, \theta_2)\| \\
\leq&\mathbb{E}_{p\sim\{1,2,\infty\}}\|\nabla_\theta g(\boldsymbol{x}_1^p, \theta_1) - \nabla_\theta g(\boldsymbol{x}_2^p, \theta_2)\| \\
\leq&\mathbb{E}_{p\sim\{1,2,\infty\}}[L_{\theta\theta}\|\theta_1 - \theta_2\| + 2L^p_{\theta\boldsymbol{x}}\epsilon_p] \\
=& L_{\theta\theta}\|\theta_1 - \theta_2\| + \eta_{st},
\end{aligned}$$

where the first inequality is the Jensen's inequality, the second one is the result in Case 1. the This proves the second inequality in equation (A.1) in this case. □

A.4. Proof of Lemma 1

$$\begin{aligned}
\|G_z(\theta_1) - G_z(\theta_2)\| &= \|\theta_1 - \theta_2 - \frac{1}{P}\sum_{p=1}^P \alpha^p h^p(\theta_1, z) + \frac{1}{P}\sum_{p=1}^P \alpha^p h^p(\theta_2, z)\| \\
&\leq \frac{1}{P}\sum_{p=1}^P \|\theta_1 - \theta_2 - \alpha^p h^p(\theta_1, z) + \alpha^p h^p(\theta_2, z)\| \\
&\leq \frac{1}{P}\sum_{p=1}^P \|\theta_1 - \theta_2\| \\
&= \|\theta_1 - \theta_2\|,
\end{aligned}$$

where the first inequality is due to triangular inequality, the second inequality is due to the co-coercive propertiy of convex function. □

A.5. Proof of Theorem 2

To bound the generalization gap of a model, we employ the following notion of uniform stability.

Definition 1. *A randomized algorithm A is* ε-uniformly stable *if for all data sets $S, S' \in \mathcal{Z}^n$ such that S and S' differ in at most one example, we have*

$$\sup_z \mathbb{E}_A \left[h(A(S); z) - h(A(S'); z) \right] \leq \varepsilon \,. \tag{A.2}$$

Here, the expectation is taken over the randomness of A. Uniform stability implies generalization in expectation (Hardt et al., 2016).

Theorem 4 (Generalization in expectation)**.** *Let A be ε-uniformly stable. Then, the expected generalization gap satisfies*

$$|\mathcal{E}_{gen}| = |\mathbb{E}_{S,A}[R_{\mathcal{D}}[A(S)] - R_S[A(S)]]| \leq \varepsilon \,.$$

Let S and S' be two samples of size n differing in only a single example. Consider two trajectories $\theta_1^1, \ldots, \theta_1^T$ and $\theta_2^1, \ldots, \theta_2^T$ induced by running an algorithm on sample S and S', respectively. Let $\delta_t = \|\theta_1^t - \theta_2^t\|$.

Fixing an example $z \in Z$ and apply the Lipschitz condition on $h(\cdot\,; z)$, we have

$$\mathbb{E}\left| h(\theta_1^T; z) - h(\theta_2^T; z) \right| \leq L\, \mathbb{E}\left[\delta_T\right] \,. \tag{A.3}$$

Observe that at step t, with probability $1 - 1/n$, the example selected by the randomized algorithms is the same in both S and S'. With probability $1/n$ the selected example is different. Based on Lemma 1, we have

$$\mathbb{E}\left[\delta_{t+1}\right] \leq \left(1 - \frac{1}{n}\right)\left(\mathbb{E}[\delta_t]\right) + \frac{1}{n}\,\mathbb{E}\left[\delta_t\right] + \frac{2\frac{1}{P}\sum_{p=1}^P \alpha_t^p L}{n} \tag{A.4}$$

$$\leq \mathbb{E}\left[\delta_t\right] + \left(\eta + \frac{2L}{n}\right)\frac{1}{P}\sum_{p=1}^P \alpha_t^p \,. \tag{A.5}$$

Unraveling the recursion, we have

$$\mathbb{E}\left[\delta_T\right] \leq \left(\frac{2L}{n}\right)\sum_{t=1}^T \frac{1}{P}\sum_{p=1}^P \alpha_t^p, \text{ and } \mathcal{E}_{gen} \leq L\left(\frac{2L}{n}\right)\sum_{t=1}^T \frac{1}{P}\sum_{p=1}^P \alpha_t^p \,.$$

Since this bounds holds for all S, S' and z, we obtain the desired bound on the uniform stability. □

A.6. Proof of Theorem 3

$$\|\theta^{t+1} - \theta^*\|^2 = \left\| \theta^t - \theta^* - \frac{1}{P}\sum_{p=1}^P \alpha_p^t \nabla h^p(\theta^t, z) \right\|^2$$

$$= \|\theta^t - \theta^*\|^2 + \left\| \frac{1}{P}\sum_{p=1}^P \alpha_p^t \nabla h^p(\theta^t, z) \right\|^2 - 2\left\langle \frac{1}{P}\sum_{p=1}^P \alpha_p^t \nabla h^p(\theta^t, z), \theta^{t+1} - \theta^* \right\rangle.$$

Take expectation over z, we have

$$\mathbb{E}\|\theta^{t+1} - \theta^*\|^2$$

$$\leq \mathbb{E}\|\theta^t - \theta^*\|^2 + \left(\frac{1}{P}\sum_{p=1}^P \alpha_p^t L\right)^2 - \frac{2}{P}\sum_{p=1}^P \alpha_p^t \mathbb{E}[h^p(\theta^t) - h^p(\theta^*)],$$

Then,

$$\alpha_{sw}^t \frac{2}{P}\sum_{p=1}^{P}\mathbb{E}[h^p(\theta^t)-h^p(\theta^*)] \leq \mathbb{E}\|\theta^t-\theta^*\|^2-\mathbb{E}\|\theta^{t+1}-\theta^*\|^2$$
$$+(\alpha_{sw}^t L)^2+\frac{2}{P}\sum_{p=1}^{P}(\alpha_{sw}^t-\alpha_p^t)\mathbb{E}[h^p(\theta^t)-h^p(\theta^*)]$$

Considering constant step size and expand the recursive, we have

$$T\alpha_{sw}\frac{2}{P}\sum_{p=1}^{P}\mathbb{E}[h^p(\theta^T)-h^p(\theta^*)]$$
$$\leq \mathbb{E}\|\theta^T-\theta^*\|^2+T(\alpha_{sw}L)^2+\frac{2T}{P}\sum_{p=1}^{P}|\alpha_{sw}^t-\alpha_p^t|B.$$

Therefore, we obtain the optimization error bound

$$\mathcal{E}_{opt} \leq \frac{D^2+L^2T\alpha_{sw}^2}{2T\alpha_{sw}}+B\frac{\sum_{p=1}^{P}|\alpha_{sw}-\alpha_p|}{\alpha_{sw}}.$$

□

B. Additional Experiments

Training settings. We adopt popular training techniques and three widely considered types of adversarial examples mentioned above: ℓ_1, ℓ_2, and ℓ_∞ attacks in the inner maximization. For ℓ_1 attack, we adopt the attack method used in (Maini et al., 2020). For ℓ_2 and ℓ_∞, we utilize the multi-step PGD attack methods (Madry et al., 2017). The perturbation budgets are set as 12, 0.5, 0.03. For better convergence performance of the inner maximization problem (Tramer et al., 2020), we set the number of steps as 50 and further increase it to 100 in the testing phase. For the stepsize in the inner maximization, we set it as 1, 0.05, and 0.003. respectively. *Cyclic Learning Rates:* in the outer minimization, we use the SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}, along with a variation of learning rate schedule from (Smith, 2018), which is piece-wise linear from 0 to 0.1 over the first 40 epochs, down to 0.005 over the next 40 epochs, and finally back down to 0 in the last 20 epochs. *Stochastic Weight Averaging and Early Stopping:* following the state-of-the-art training techniques for adversarial training, we incorporate stochastic weight averaging (SWA) (Izmailov et al., 2018) and early stopping in ATMP. It is shown that SWA could find flat local minima and yields performance (Stutz et al., 2021). The update of SWA is $\theta_{swa}^t=\gamma\theta_{swa}^{t-1}+(1-\gamma)\theta^{t-1}$, where γ is a hyper-parameter and the final θ_{swa}^T is used for evaluation. We follow the setting of (Izmailov et al., 2018) and start SWA from the 60-th epoch for all the methods we compare. For all the methods, we repeat five runs.

Table 3. Test accuracy (%) of different algorithms (MAX, AVG, MSD, SAT, and ADT) against ℓ_1, ℓ_2, and ℓ_∞ attacks on CIFAR-100.

Dataset		CIFAR-100					
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞	Union	Mix
AT	ℓ_1	70.98	73.44	00.82	00.51	0.04	24.92
	ℓ_2	63.76	21.88	43.75	20.31	12.07	28.64
	ℓ_∞	58.86	11.02	39.22	28.01	9.41	26.08
WST (Eq. 3.2)	MAX	57.82	30.36	40.71	26.08	25.03±0.39	32.38±0.18
	MSD	57.33	32.08	41.90	27.06	26.21±0.22	34.02±0.08
AVG (Eq. 3.3)	AVG	59.75	35.55	41.03	24.61	24.31±0.68	33.73±0.41
	SAT	59.25	35.60	42.33	25.01	24.78 ±1.41	34.31±0.88
	ADT	59.41	**37.64**	**42.82**	**25.70**	**25.29**±0.15	**35.39**±0.07

In the additional experiments, the test accuracy are evaluated using the first batch (128 samples) to save the computational cost.

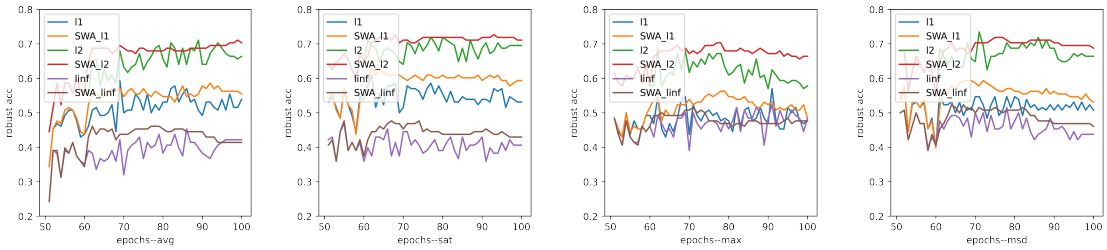


Figure 2. Tendency curves of robust accuracy against different types of adversarial attacks. The models are trained by ATMP using four different strategies, with and without SWA.

Table 4. Test accuracy (%) of AVG and ADT w/o early stopping and w/o SWA against ℓ_1, ℓ_2, and ℓ_∞ attacks on CIFAR-10.

	ℓ_1	ℓ_2	ℓ_∞	Union	Mix
AVG	53.32	60.93	39.76	39.03	51.33
AVG+ES	57.37	66.48	43.33	43.02	55.72
AVG+ES+SWA	57.78	68.08	43.87	43.28	56.91
ADT	55.43	65.72	41.43	41.21	54.19
ADT+ES	61.11	67.88	44.67	44.31	57.89
ADT+ES+SWA	**61.81**	**69.61**	**46.64**	**46.05**	**59.16**

B.1. Ablation Study of SWA

We give more ablation study of SWA in this subsection.

SWA on CIFAR-100 In Fig. 3, we we plots the robust accuracy of of ATMP using AVG, SAT, MAX, and MSD respectively on CIFAR-100, which are the same experiments we show in Figure 3. From the plots, we observe that without SWA, the test accuracy are highly unstable among different training epochs. There exists a trade-off between different types of adversaries. The increase in robust accuracy against one kind of adversaries may accompany with the decrease in robust accuracy on another adversarial examples. On the other side, when coupling with SWA, the tendency curves of all four ATMP strategies are largely stabilized. Besides, SWA (red, orange, and brown line) will increase the test accuracy in most of the case. The experiments in Figure 3 give the same conclusion as that of Figure 2 in the main paper.

Gradient norm of SWA In Figure 4, we show the gradient norm of all the batches with and without SWA using AVG and MAX. On training set, the gradient norm with and without SWA is similar. On test set, the gradient norm of using SWA is smaller than that without SWA. This proves that SWA can find flatter minima with smaller gradient norm, and have better generalization.

When to start SWA In Table 5 we try to start SWA at the 60th, 70th, and 80th epochs. We use MSD for our experiments. We can see that the test accuracy against ℓ_∞ attack is higher when we start later. BUt the test accuracy against ℓ_1 attack is higher when we start earlier. It is hard to say which is better but using SWA is always better than no SWA. The accuracy of AVG, MAX MSD, and SAT with and without SWA are provided in Table 6.

Table 5. Test accuracy (%) of MSD with SWA started from the 60th,70th, and 80th epochs.

Dataset		CIFAR-10			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞
MSD with SWA	no SWA	83.51	50.78	66.41	43.75
	the 60th epochs	81.99	53.12	68.75	46.09
	the 70th epochs	82.01	49.22	69.53	47.66
	the 80th epochs	81.95	50.78	69.54	48.44

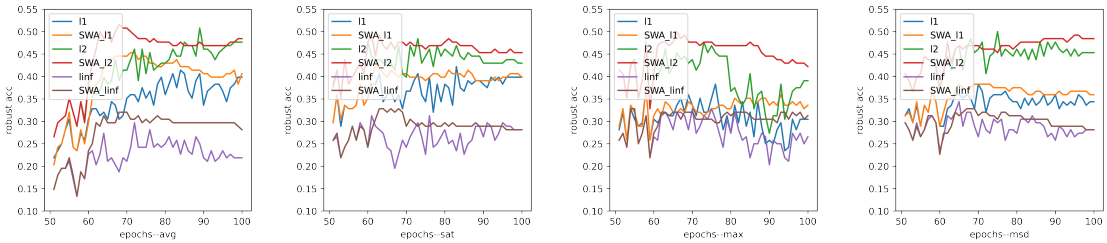


Figure 3. Results of robust accuracy for each type perturbation over epochs of multiple perturbation AT in CIFAR-100 dataset:(a) Avg and Avg-SWA, (b) SAT and SAT-SWA, (c) Max and Max-SWA, (d) MSD and MSD-SWA.

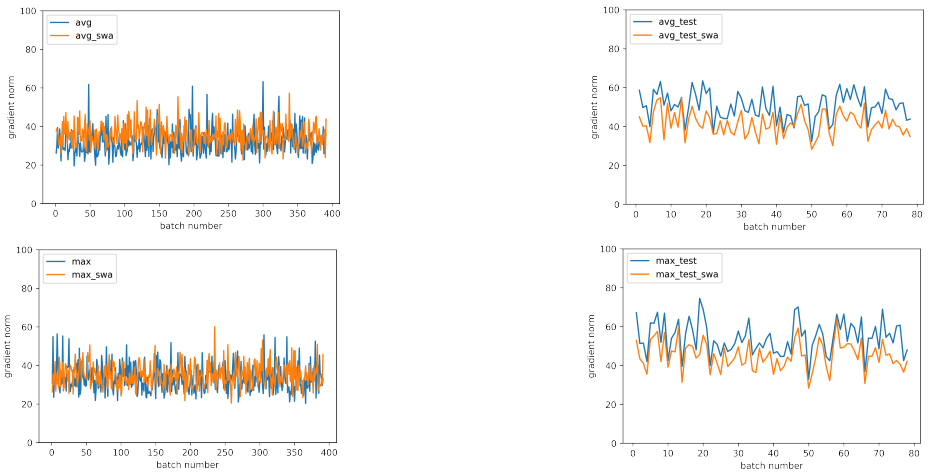


Figure 4. Gradient norm with and without SWA.

B.2. Other Tricks

Label smoothing (LS) Label smoothing is introduced as regularization to improve generalization by replacing the one-hot labels to soft labels in the cross-entropy loss. It is shown that it relates to flat minima and yields better generalization. In the cross-entropy loss, using soft labels other than one-hot label can increase the smoothness of the loss function. Given probability $\gamma \in [0, 1]$ and number of classes K, for each sample $(\boldsymbol{x}_i, y_i)$, let y_i keeps the true label with probability $1 - \gamma$, y_i is replace by a wrong label in one of the other $K - 1$ classes with equal probability $\gamma/(K - 1)$. The performance of ATMP with label smoothing is provided in Table 7. We can see that label smoothing provides some improvement in these four strategies.

Label noise Label noise is a alternative choose of label smoothing. Given probability $\gamma \in [0, 1]$ and number of classes K, for each sample $(\boldsymbol{x}_i, y_i)$, let y_i keeps the true label with probability $1 - \gamma$, y_i is replace by a wrong label in one of the other $K - 1$ classes with equal probability $\gamma/(K - 1)$. The difference between label noise and label smoothing is that label noise use hard label while label smoothing use soft label.

In Table 8, we provide the experiments of incorporating label noise in ATMP. We can see that label noise can give some improvements. But the improvements are not competitive to label smoothing. The reason behind is that label noise cannot increase the smoothness of the loss function of ATMP.

Silu activation function Silu activation function is proposed to deal with the non-smoothness of Relu activation function. In Table 9, we can see that Silu has some small improvement on ATMP.

Mixup We use adversarial examples in the form $\boldsymbol{x}_{adv} = \boldsymbol{x} + (\delta_1 + \delta_2 + \delta_\infty)/3$ for adversarial training. In Table 10, we can see that the robust accuracy is not comparable to MAX and AVG. It is because this training procedure focuses more on the mixup adversarial examples. It fails to fit the three types of adversarial examples.

Table 6. Test accuracy (%) of different algorithms (MAX, AVG, MSD and SAT, with and without SWA) against ℓ_1, ℓ_2, and ℓ_∞ attacks on CIFAR-10 and CIFAR-100.

Dataset		CIFAR-10				CIFAR-100			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞	clean	ℓ_1	ℓ_2	ℓ_∞
AT	ℓ_1	93.19	89.84	0.00	0.00	70.98	73.44	00.78	00.78
	ℓ_2	88.66	41.41	61.72	39.84	63.76	21.88	43.75	20.31
	ℓ_∞	84.94	17.19	53.91	46.88	58.86	11.72	39.06	30.47
ATMP	AVG	**85.28**	50.78	64.84	38.28	59.71	39.84	47.66	21.88
	MAX	84.96	43.75	49.22	41.41	57.90	30.47	39.06	26.56
	MSD	83.51	50.78	66.41	43.75	57.33	34.38	45.31	28.12
	SAT	85.23	53.12	69.53	40.62	59.25	39.84	42.97	28.12
ATMP with SWA	AVG	83.19	57.03	69.53	42.97	58.04	**40.62**	**48.44**	28.12
	MAX	83.78	42.19	60.16	**47.66**	58.29	33.59	42.19	**31.25**
	MSD	81.99	53.12	68.75	46.09	57.33	35.94	**48.44**	28.12
	SAT	84.76	**59.38**	**71.09**	42.97	**60.18**	39.84	45.31	28.12

Table 7. Test accuracy (%) of different algorithms (MAX, AVG, MSD and SAT, with label smoothing) against ℓ_1, ℓ_2, and ℓ_∞ attacks on CIFAR-10 and CIFAR-100.

Dataset		CIFAR-10				CIFAR-100			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞	clean	ℓ_1	ℓ_2	ℓ_∞
ATMP with LS	AVG	**85.57**	55.47	68.75	39.06	**60.58**	38.28	46.88	24.22
	MAX	84.49	45.31	57.03	45.31	58.25	33.59	45.31	29.69
	MSD	83.49	50.78	69.53	45.30	58.61	35.94	**48.44**	32.03
	SAT	85.70	53.91	67.97	37.50	60.17	39.84	44.53	26.56
ATMP with LS & SWA	AVG	83.42	**57.03**	71.09	**48.44**	59.09	**42.19**	45.31	28.12
	MAX	83.05	47.66	63.28	46.09	57.76	39.84	47.66	31.25
	MSD	82.15	50.78	**71.88**	46.09	57.70	38.28	46.88	**35.94**
	SAT	84.96	56.25	66.41	41.41	60.51	39.06	47.66	29.69

Table 8. Test accuracy (%) of ATMP with label noise.

Dataset		CIFAR-10			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞
ATMP with LN	MAX	84.18	43.75	57.03	44.53
	AVG	84.67	57.03	70.31	41.41
	MSD	82.79	52.34	6562	47.66
	SAT	85.36	56.25	69.53	42.19
ATMP with LN & SWA	MAX	81.66	50.78	63.28	47.66
	AVG	81.06	57.81	70.31	42.97
	MSD	81.03	51.56	64.84	50.00
	SAT	84.00	59.38	67.19	46.88

Table 9. Test accuracy (%) of ATMP with Silu activation function.

Dataset		CIFAR-10			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞
ATMP with Silu	MAX	80.91	53.91	64.84	46.09
	AVG	81.61	57.03	70.31	42.19
	MSD	80.44	57.03	64.84	48.44
	SAT	83.89	54.69	64.84	44.53
ATMP with Silu & SWA	MAX	78.48	50.78	67.19	49.22
	AVG	78.20	58.59	68.75	42.98
	MSD	78.39	55.47	66.41	49.99
	SAT	82.02	57.81	67.19	45.31

Table 10. Test accuracy (%) of ATMP using mixup.

Dataset		CIFAR-10			
Attack methods		clean	ℓ_1	ℓ_2	ℓ_∞
Other tricks	Mixup	84.73	51.56	67.19	39.06

C. Smoothness Analysis

We first study the smoothness of the minimax problems in Eq. (3.2) and (3.3). To simplify the notation, let

$$\begin{aligned} h^p(\theta, z) &= \max_{\|z-z'\|_p \le \epsilon_p} g(\theta, z'), \\ h^{avg}(\theta, z) &= \mathbb{E}_{p\sim\{1,2,\infty\}} \max_{\|z-z'\|_p \le \epsilon_p} g(\theta, z'), \\ h^{wst}(\theta, z) &= \max_{p\in\{1,2,\infty\}} \max_{\|z-z'\|_p \le \epsilon_p} g(\theta, z') \end{aligned}$$

be the loss function of standard adversarial training, worst-case multiple perturbation adversarial training, and average of all perturbations adversarial training, respectively. The population and empirical risks are the expectation and average of $h^{st}(\cdot)$, respectively. We use $R^{st}_{\mathcal{D}}(\theta)$ and $R^{st}_S(\theta)$ to denote the population and empirical risk for adversarial training with different strategy, *i.e.* st $\in \{1, 2, \infty, \text{wst}, \text{avg}\}$.

Case study: Linear regression. We use a simple case, adversarial linear regression, to illustrate the smoothness of the optimization problem of (3.2) and (3.3). Let $f_\theta(\boldsymbol{x}) = \theta^T \boldsymbol{x}$ and $\ell(\theta^T x, y) = |\theta^T \boldsymbol{x} - y|^2$, we have the following proposition.

Proposition 1. *Let* $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n]^T$, $y = [y_1, \cdots, y_n]^T$, *and* $\delta = [\delta_1, \cdots, \delta_n]^T$, *we have*

$$\begin{aligned} R^p_S(\theta) &= [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2, \\ R^{wst}_S(\theta) &= \max_{p\in\{1,2,\infty\}} [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2, \\ R^{avg}_S(\theta) &= \mathbb{E}_{p\in\{1,2,\infty\}} [\|\boldsymbol{X}\theta - y\|_2 + \sqrt{n}\epsilon_p \|\theta\|_{p^*}]^2. \end{aligned}$$

The proof is deferred to Appendix A. From Proposition 1, the loss landscape of adversarial training is non-smooth because of the term $\|\theta\|_{p^*}$. Specifically, the loss function of ℓ_2 adversarial training is non-smooth at $\theta = 0$. For ℓ_1 adversarial training, the loss function is non-smooth at $|\theta_i| = |\theta_j|, \forall i, j$. For ℓ_∞ adversarial training, the loss function is non-smooth at $\theta_i = 0, \forall i$. For adversarial training for multiple perturbations, the loss function is non-smooth at both $\theta_i = 0, \forall i$ and $|\theta_i| = |\theta_j|, \forall i, j$. The non-smooth region of the loss function of ATMP is the union of that of the single perturbation cases. Different ℓ_p adversaries give different contribution to the smoothness of the loss function of ATMP.

In Fig. 5, we give a numerical simulation and demonstrate the loss landscape in a two-dimensional case. In ℓ_2 adversarial training, the loss landscape is smooth almost everywhere, except the original point. In ℓ_1 and ℓ_∞ cases, the non-smooth region is a 'cross'. In the cases of WST and AVG, the non-smooth region is the union of two 'crosses'.

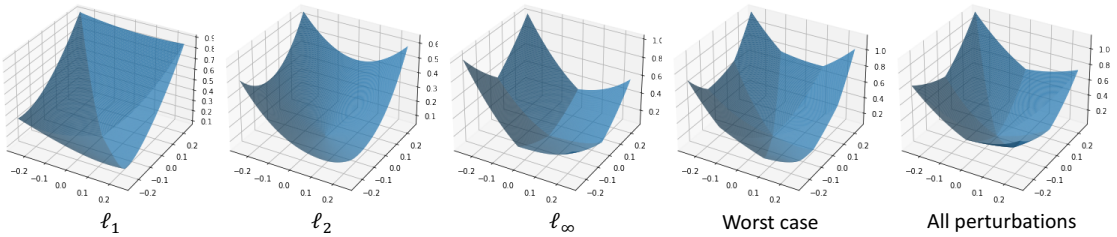
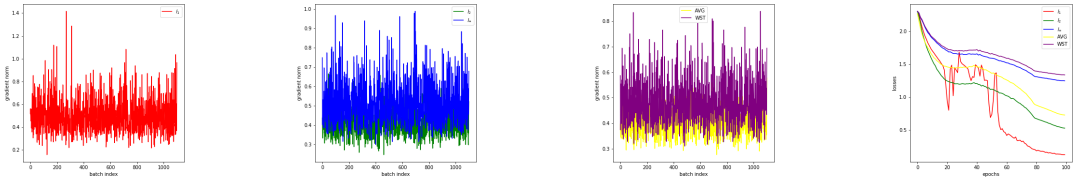


Figure 5. Loss landscape of adversarial linear regression for single and multiple perturbations.



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

Figure 6. Gradient norms and loss value of adversarial training for single and multiple perturbations. (a) $\|\nabla R_S^1(\theta)\|$, last 1173 batch. (b) $\|\nabla R_S^2(\theta)\|$ (green) and $\|\nabla R_S^\infty(\theta)\|$ (blue), last 1173 batch. (c) $\|\nabla R_S^{avg}(\theta)\|$ (yellow) and $\|\nabla R_S^{wst}(\theta)\|$ (purple), last 1173 batch. (d) Training loss for the total 100 epochs.

General nonlinear model. Now let us consider general nonlinear models. Following the work of (Sinha et al., 2017), without loss of generality, let us assume

Assumption 2. *The function g satisfies the following Lipschitzian smoothness conditions:*

$$\begin{aligned}
&\|g(\theta_1, z) - g(\theta_2, z)\| \leq L\|\theta_1 - \theta_2\|,\\
&\|\nabla_\theta g(\theta_1, z) - \nabla_\theta g(\theta_2, z)\| \leq L_\theta\|\theta_1 - \theta_2\|,\\
&\|\nabla_\theta g(\theta, z_1) - \nabla_\theta g(\theta, z_2)\| \leq L_{\theta z}\|z_1 - z_2\|,\\
&\|\nabla_z g(\theta_1, z) - \nabla_\theta g(\theta_2, z)\| \leq L_{z\theta}\|\theta_1 - \theta_2\|.
\end{aligned}$$

Assumption 2 assumes that the loss function is smooth (in zeroth-order and first-order). While ReLU activation function is non-smooth, recent works (Allen-Zhu et al., 2019; Du et al., 2019) showed that the loss function of overparamterized DNNs is semi-smooth. It helps justify Assumption 2. Under Assumption 2, the following Lemma Provide the smoothness of ATMP.

Lemma 2. *Under Assumption 2, assuming in addtion that $g(\theta, z)$ is locally μ_p-strongly concave for all $z \in \mathcal{Z}$ in ℓ_p-norm. $\forall\theta_1, \theta_2$ and $\forall z \in \mathcal{Z}$, the following properties hold.*

1. *(Lipschitz function.)* $\|h^{st}(\theta_1, z) - h^{st}(\theta_2, z)\| \leq L\|\theta_1 - \theta_2\|$.
2. *(Gradient Lipschitz.) If we Then,* $\|\nabla_\theta h^{st}(\theta_1, z) - \nabla_\theta h^{st}(\theta_2, z)\| \leq \beta_{st}\|\theta_1 - \theta_2\|$, *where*

$$\beta_{st} = \begin{cases} L_{\theta z}L_{z\theta}/\mu_{st} + L_\theta & st \in \{1, 2, \infty\},\\ L_{\theta z}L_{z\theta}/\min_p \mu_p + L_\theta & st = wst,\\ \mathbb{E}_{p\sim\{1,2,\infty\}}\beta_p & st = avg. \end{cases}$$

Proof: see Appendix A. Lemma 2 shows that adversarial surrogate loss in different ℓ_p adversaries have different smoothness in strongly-concave case. This property is not specific for strongly-concave case. The analysis of non-strongly-concavity case is provided in App. A.3. Lemma 2 motivates us to study a question:

How to achieve better performance on adversarial robustness against different ℓ_p adversarial attacks utilizing the smoothness properties?

In the next section, we first discuss the stability analysis of ATMP. Then, we discuss how to utilize the smoothness-properties of different ℓ_p adversaries to obtain smaller generalization bound.

C.1. Gradient Norm Analysis

Since the gradient Lipschitz $L_{\theta x}^p$ is unknown in practice, we provide a numerical simulation of the convergence error in this subsection on CIFAR-10 to help justify our Theoretical results. In Fig. 6, we show the gradient norm $\|\nabla R_S^{st}(\theta)\|$ of the last layer for the last 3 epochs ($3 \times 391 = 1173$ batchs) for $st \in \{1, 2, \infty, avg, wst\}$ as well as the training loss[3] for the total 100 epochs.

[3]We should use optimality gap (training loss - optimal loss) to evaluate convergence, but the optimal loss is unknown, we use training loss as a substitude.

Comparison of ℓ_1, ℓ_2, and ℓ_∞ In Fig. 6, We can see that $\|\nabla R_S^1(\theta)\|$ is the largest one accompany with the largest variance among these three. In (d), the training loss of $R_S^1(\theta)$ is unstable. It is because the top-k ℓ_1 attack is inefficient and sparse (Tramèr & Boneh, 2019). In the middle stage, the fluctuation is large since the success rate of the top-k attack is small. In the final stage, the top-k attack cannot find adversarial examples. Therefore, the loss is small. Comparing the $R_S^2(\theta)$ and $R_S^\infty(\theta)$, we can see that ℓ_∞ adversarial training has higher gradient norm and larger training loss. In conclusion, ℓ_1 and ℓ_∞ give less contributions to the smoothness of ATMP.

D. Other related work

Adversarial training Adversarial training (AT) has been demonstrated to be one of the most effective ways to increase the adversarial robustness (Szegedy et al., 2013). The key idea of AT is to augment the training set with adversarial examples during training. Currently, most AT-based methods are trained with a single type of adversarial examples, and the ℓ_p (p=1, 2, or ∞) is commonly used to generate adversarial examples during training (Madry et al., 2017). It is shown that AT overfits the adversarial examples on the training set and generalizes badly on the test sets. Many approaches have been proposed to increase the adversarial generalization (Raghunathan et al., 2019; Schmidt et al., 2018). Meanwhile, there have been some attempts for the theoretical understanding of adversarial training, mainly focusing on the convergence properties and generalization bound. For example, the work of (Gao et al., 2019) studies the convergence of adversarial training in the neural tangent kernel (NTK) regime. In terms of generalization bound, the work of (Yin et al., 2019; Awasthi et al., 2020; Xiao et al., 2022a) study the generalization bound in terms of Rademacher complexity. The work of (Xiao et al., 2022d) considers manifold properties of adversarial examples.

Uniform Stability The work of (Hardt et al., 2016) introduced uniform stability to study the generalization-optimization trade-off in machine learning problem. The work of (Farnia & Ozdaglar, 2021; Xing et al., 2021; Xiao et al., 2022b) extended the analysis to adversarial training and emphasize the severe of robust overfitting issues.