Smoothed-SGDmax: A Stability-Inspired Algorithm to Improve Adversarial Generalization

Jiancong Xiao[1,*], **Jiawei Zhang**[2,*], **Zhi-Quan Luo**[1], **Asuman Ozdaglar**[2]
[1]The Chinese University of Hong Kong, Shenzhen; [2]Massachusetts Institute of Technology
jiancongxiao@link.cuhk.edu.cn, jwzhang@mit.edu,
luozq@cuhk.edu.cn, asuman@mit.edu

Abstract

Unlike standard training, deep neural networks can suffer from serious overfitting problems in adversarial settings. Recent research [40, 39] suggests that adversarial training can have nonvanishing generalization error even if the sample size n goes to infinity. A natural question arises: can we eliminate the generalization error floor in adversarial training? This paper gives an affirmative answer. First, by an adaptation of information-theoretical lower bound on the complexity of solving Lipschitz-convex problems using randomized algorithms, we establish a minimax lower bound $\Omega(s(T)/n)$ given a training loss of $1/s(T)$ for the adversarial generalization gap, where T is the number of iterations, and $s(T) \to +\infty$ as $T \to +\infty$. Next, by observing that the nonvanishing generalization error of existing adversarial training algorithms comes from the non-smoothness of the adversarial loss function, we employ a smoothing technique to smooth the adversarial loss function. Based on the smoothed loss function, we design a smoothed SGDmax algorithm achieving a generalization bound $\mathcal{O}(s(T)/n)$, which eliminates the generalization error floor and matches the minimax lower bound. Experimentally, we show that our algorithm improves adversarial generalization on common datasets.

1 Introduction

Deep neural networks (DNNs) [22, 20] is successful and rarely suffered overfitting issues [45]. This phenomenon is also called benign overfitting. A well-trained neural network model can generalize well to the test data. However, in adversarial machine learning, overfitting becomes a serious issue [30]. Before the training algorithms converge, the robust test error starts to increase. This special type of overfitting is called *robust overfitting* and can be observed in the experiments on common datasets. See Fig. 1, orange curve. Therefore, mitigating the robust overfitting is important to increase the adversarial robustness of a DNN model. Several recent works tried to figure out the causes of robust overfitting and designed methods to mitigate it. See the discussion in Sec. B.

A recent line of work [40, 39] studied the robust overfitting issue of adversarial training from a theoretical perspective, using the notion of uniform algorithmic stability. Uniform algorithmic stability (UAS) [6] was introduced to bound the generalization gap in machine learning problems. It provides algorithm-specific generalization bounds instead of algorithm-free generalization bounds such as classical results on VC-dimension [37] and Rademacher complexity [5]. Such stability-based generalization bounds provide insight into understanding the generalization ability of neural network models trained by different algorithms.

[*] Equal Contribution.

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 1: Comparison of stability-based generalization bounds of adversarial generalization gap. $c_1(T)$ and $c_2(T)$ are sample size-independent terms. Details of the form of $s(T)$, $c_1(T)$, $c_2(T)$ are discussed in Sec. D and Sec. 2.

	Upper Bounds	Worst-case Lower Bounds	Achieves minimax lower bound $\Omega(s(T)/n)$
SGDmax (AT)	$\mathcal{O}(c_1(T) + \frac{s(T)}{n})$	$\Omega(c_2(T) + \frac{s(T)}{n})$	✗
Smoothed-SGDmax (Ours)	$\mathcal{O}(\frac{s(T)}{n})$	$\Omega(\frac{s(T)}{n})$	✓

Traditional adversarial training is to perform stochastic gradient descent (SGD) on the max function of the standard counterpart, which is also called SGDmax [14]. We will not distinguish two algorithms, "SGDmax" and "adversarial training (AT)", in the paper. The work of [40, 39] both showed that SGDmax incurs a stability-based generalization bound in $\mathcal{O}(c(T) + s(T)/n)$. Here T is the number of iterations, n is the number of samples, $s(T)$ is a function satisfies $s(T) \to +\infty$ as $T \to +\infty$, and $c(T)$ is a sample size-independent term and increase with T. Details of the form of $s(T)$, $c(T)$ are discussed in Sec. D and Sec. 2. They also provided the matching lower bounds to show that the sample size-independent term is unavoidable for SGDmax-based adversarial training algorithms. It provides a possible explanation of robust overfitting: even though we have arbitrarily large number of training samples, the adversarial generalization gap still does not vanish. Therefore, we are motivated to design algorithms to reduce the non-vanishing sample size-independent term. The first question arises: what is the lower bound of the generalization gap for algorithms in adversarial machine learning settings? To answer this question, we develop a minimax lower bound, $\Omega(s(T)/n)$, for the adversarial generalization gap when the adversarial training loss is $1/s(T)$. Clearly, the SGDmax-based adversarial training algorithm does not achieve the lower bound. Therefore, the following main question of our paper arises:

Can we design an algorithm achieving the minimax lower bound of adversarial generalization gap?

It is observed that the term $c(T)$ comes from the non-smoothness of the adversarial loss. Hence, we employ a smoothing technique to smooth the adversarial loss and perform gradient descent to this smooth surrogate. Following the name SGDmax, we propose Smoothed-SGDmax, which is a smoothed version of SGDmax, to improve adversarial generalization. We prove that our algorithm has the same training loss $1/s(T)$ on adversarial loss and achieves the minimax lower bound $\Omega(s(T)/n)$ of the generalization gap. The comparison of the stability-based generalization upper bound and lower bound of our proposed algorithm with the SGDmax-based adversarial training algorithm is given in Table 1.

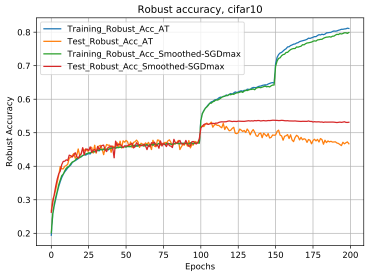


Figure 1: Experiments of adversarial training and Smoothed-SGDmax on CIFAR-10.

Related work are discussed in Appendix B.

2 Proposed Algorithm: Smoothed-SGDmax

In Appendix C, we provide the prelinimaries of stability and generalization gap. In Appendix D, we show that the adversarial generalization bound of SGDmax-based adversarial training algorithm is not optimal. In this section, we will design an algorithm satisfying the following two properties: 1) It has the same training loss as the SGDmax algorithm; 2) Suppose it achieves $1/s(T)$ training loss after T iterations. Then, the generalization bound is bounded by $s(T)/n$.

2.1 Smooth Surrogate Adversarial Loss

In Thm. D.2, the non-smoothness of h leads to a poor generalization bound. This motivates us to construct smooth surrogate loss functions to improve adversarial generalization. Inspired by the work of [46], we use the Moreau envelope function to smooth the adversarial loss. Let

$$K(w, u; z) = h(w; z) + \frac{p}{2}\|w - u\|^2. \tag{2.1}$$

If h is l-weakly convex, we can choose $p > l$ to insure that $K(w, u; z)$ is strongly convex with respect to w. In the case that h is convex, we only need $p > 0$. We define the Moreau envelope function:

$$
\begin{aligned}
M(u; S) &= \min_{w \in W} K(w, u; S) = \min_{w \in W} \frac{1}{n} \sum_{z \in S} K(w, u; z), && (2.2)\\
w(u; S) &= \arg \min_{w \in W} K(w, u; S). && (2.3)
\end{aligned}
$$

Then, $M(u; S)$ is a smooth function. Formally, we state the theoretical results in Lemma D.1. Depending on whether we solve the subproblems exactly or not, we have the exact approach and inexact approach.

2.2 Exact approach

We first consider the exact approach, which is the gradient descent to $M(u; S)$.

Theorem 2.1. *Assume h is a convex, L-Lipschitz function. Suppose we run GD on the smoothed surrogate adversarial loss $M(u; S)$ defined in Eq. (2.2) with step size $\alpha_t \leq 1/\sqrt{T}$ for $T \geq 4p^2$ steps. Then, the optimization and generalization gap satisfies*

$$
\mathcal{E}_{opt} \leq \mathcal{O}(1/T\alpha) \quad \textit{and} \quad \mathcal{E}_{gen} \leq \left(\frac{2L^2}{n}\right) \sum_{t=1}^{T} \alpha_t. \tag{2.4}
$$

Therefore, the exact approach achieves the minimax lower bounds of the generalization gap. However, the exact approach requires the exact minimization of $K(w, u; S)$, which is sometimes computationally intractable. To address this issue, we consider the inexact approach below.

2.3 The Inexact approach

The inexact approach is to estimate $\nabla_u M(u; S)$ by inexactly solving $\min_w K(w, u; S)$. To this aim, we perform multiple steps of SGD to the subproblem $\min_w K(w, u; S)$, attaining an estimate $\bar{w}(u)$ of the true $w(u)$, and then use $\bar{w}(u)$ to estimate $\nabla_u M(u; S)$.

Algorithm 1 Smoothed-SGDMax

```
 1: Initialize $w^0$, $u^0$;
 2: Choose stepsize $c_s^t > 0$ and $\alpha_t > 0$;
 3: for $t = 0, 1, 2, \ldots, T$ do
 4:     Let $w_0^t = w^t$;
 5:     for $s = 0, 1, 2, \cdots, N$ do
 6:         Draw a sample $z_s^t$ from $S$ uniformly;
 7:         $w_{s+1}^t = P_W(w_s^t - c_s^t \nabla_w K(w_s^t, u^t; z_s^t))$;
 8:     end for
 9:     $w^{t+1} = w_N^t$;
10:     $u^{t+1} = u^t + \alpha_t p(w^{t+1} - u^t)$;
11: end for
```

In Step 7 in Alg. 1, we run SGD on $K(w, u, S)$ *w.r.t* w to find a solution given u. In step 10, we run GD on $K(w, u, S)$ *w.r.t* u. To provide the upper bounds of the optimization gap and generalization gap of Alg. 1, we need the following Lemma for the inner optimization.

Lemma 2.1. *Given t and u^t, suppose we run SGD on $K(w, u^t, S)$ w.r.t. w with stepsize $c_s^t \leq 1/(p - l)s$ for N steps. w_N^t is approximately the minimizer with an error C_1^2/N, i.e.,*

$$
E\|w_N^t - w(u^t)\|^2 \leq \frac{C_1^2}{N},
$$

where $C_1 = (L + pD_W)/(p - l)$.

In convex case, *i.e.*, $l = 0$, we have $C_1 = L/p + D_W$. Lemma 2.1 provides the optimization error of the inner loop. In words, if we run the inner loop for sufficient steps, we can approximate the smoothed loss $M(u; S)$. Below we provide the training loss and uniform stability of Smoothed-SGDmax with sufficient steps for the inner loop.

Theorem 2.2 (Training Loss of Smoothed-SGDmax)**.** *Suppose h is convex and L-Lipschitz. In Alg. 1, if we choose inner stepsize $c_s^t \leq 1/ps$, number of steps in inner loop $N = T$, outer stepsize $\alpha_t \leq 1/\sqrt{T}$, $T \geq 4p^2$, the optimization gap satisfies*

$$\mathcal{E}_{opt} \leq \frac{\|u^0 - u^*\|^2 + 2pC_1D_W + (L + pD_W)^2}{2T\alpha} = \frac{C_2}{T\alpha}, \tag{2.5}$$

where $C_2 = \|u^0 - u^*\|^2/2 + pC_1D_W + (L + pD_W)^2/2$.

Theorem 2.3 (Generalization bound of Smoothed-SGDmax)**.** *Assume that h is convex and L-Lipschitz. In Alg. 1, if we choose inner stepsize $c_s^t \leq 1/ps$, number of steps in inner loop $N = n^2$, outer stepsize $\alpha_t \leq 1/\sqrt{T}$, $T \geq 4p^2$, the generalization gap satisfies*

$$\mathcal{E}_{gen} \leq L\left(\frac{2C_1p}{n} + \frac{2L}{n}\right)\sum_{t=1}^{T}\alpha_t = \frac{C_3}{n}\sum_{t=1}^{T}\alpha_t, \tag{2.6}$$

where $C_3 = L(4L + 2pD_W)$.

Thm. 2.2 and 2.3 are the main results of our paper. It shows that Alg. 1 has training loss $\mathcal{O}(1/T\alpha)$ and has optimal generalization bound in $\mathcal{O}(T\alpha/n)$.

Interpretation of Number of Steps. In practice, if we use batch size 1 and go through the whole dataset in each epoch, T can be viewed as the number of epochs, and N can be viewed as the number of samples. Let $T\alpha = \sqrt{C_2n/C_3}$, we obtain the optimal excess risk with respect to T and α, *i.e.*, $\mathcal{E}_{opt} + \mathcal{E}_{gen} \leq 2\sqrt{\frac{C_2C_3}{n}}$. Further analysis are deferred to Appendix E.

3 Experiments

Table 2: Robust test accuracy of our proposed algorithm. $\epsilon = 8/255$. Model: WideResNet-28 × 10 with Swish activation function. Training data to unlabeled data ratio: 3:7.

Dataset	Loss	Algorithm	Clean	AutoAttack
CIFAR-10	AT Loss	SGDmax	90.93±0.25%	58.41±0.25%
		Smooth-SGDmax	91.51±0.20%	59.14±0.18%
	TRADES Loss	SGDmax	88.36%	59.45%
		Smooth-SGDmax	85.33±0.13%	62.41±0.11%
CIFAR-100	TRADES Loss	SGDmax	59.38%	26.07%
		Smooth-SGDmax	59.25±0.22%	28.54±0.19%

Experiments setting and experiments on sample complexity are provided in Appendix F. In Table 2, we provide the robust test performance of our proposed algorithms. The baseline performance on CIFAR-10 are reported in [18]. We can see that the performance of our proposed algorithms is comparable in the same settings used in [18]. Notice that the state-of-the-art performance of adversarial robustness is obtained using large models (*e.g.*, WideResNet-106 × 16) and DDPM-generated data [29]. We do not have enough resources to run large models.

4 Conclusion

In this paper, we study a question: can we design an algorithm to achieve the minimax lower bound of the adversarial generalization gap? We propose Smoothed-SGDmax and prove that it has the same convergence guarantee as adversarial training and attains the minimax lower bound of the adversarial generalization gap. We hope our work can lead to a better understanding of adversarial machine learning theory.

References

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

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

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

[4] Peter L Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE transactions on Information Theory*, 44(2):525–536, 1998.

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

[6] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

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

[8] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11190–11201, 2019.

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

[10] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021.

[11] Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

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

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

[14] Farzan Farnia and Asuman Ozdaglar. Train simultaneously, generalize better: Stability of gradient-based minimax learners. In *International Conference on Machine Learning*, pages 3174–3185. PMLR, 2021.

[15] Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. *arXiv preprint arXiv:1812.09859*, 2018.

[16] Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pages 1270–1279. PMLR, 2019.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[18] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

[19] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

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

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

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

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

[24] Omar Montasser, Steve Hanneke, and Nathan Srebro. Vc classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory*, pages 2512–2530. PMLR, 2019.

[25] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[26] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[27] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

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

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

[30] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[31] William H Rogers and Terry J Wagner. A finite sample distribution-free performance bound for local discrimination rules. *The Annals of Statistics*, pages 506–514, 1978.

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

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

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

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

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

[37] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

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

[39] Jiancong Xiao, Yanbo Fan, Ruoyu Sun, Jue Wang, and Zhi-Quan Luo. Stability analysis and generalization bounds of adversarial training. In *Advances in Neural Information Processing Systems*, 2022.

[40] Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

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

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

[43] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning*, pages 25595–25610. PMLR, 2022.

[44] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[46] Jiawei Zhang and Zhi-Quan Luo. A proximal alternating direction method of multiplier for linearly constrained nonconvex minimization. *SIAM Journal on Optimization*, 30(3):2272–2302, 2020.

[47] Yonggang Zhang, Xinmei Tian, Ya Li, Xinchao Wang, and Dacheng Tao. Principal component adversarial example. *IEEE Transactions on Image Processing*, 29:4804–4815, 2020.

A Proof of Theorems

A.1 Proof of Theorem D.1

The proof is adopted from the proof of minimax lower bound of optimization error from the work of [11]. We define the excess risk as $R_{\mathcal{D}}(w) - \min_{w \in W} R_{\mathcal{D}}(w)$. A minimax lower bound of the excess risk for the function class $\mathcal{H}$ is given in [26].

$$\min_{w} \max_{\mathcal{D}} \mathbb{E}_{S \sim \mathcal{D}^n}[R_{\mathcal{D}}(w) - \min_{w \in W} R_{\mathcal{D}}(w)] \geq \frac{LD_W}{C_4 \sqrt{n}}. \tag{A.1}$$

Assume that $\mathcal{E}_{opt}(w^T) \leq \mathcal{O}(1/s(T))$. Then

$$\min_{A \in \mathcal{A}} \max_{\mathcal{D}} \mathcal{E}_{gen}(w^T) \geq \Omega\bigg(\frac{LD_W}{\sqrt{n}} - \frac{1}{s(T)}\bigg). \tag{A.2}$$

where C_4 is a universal constant. Since

$$\frac{LD_W}{\sqrt{n}} - \frac{1}{s(T)} = -\bigg(\frac{1}{\sqrt{s(T)}} - \frac{LD_W\sqrt{s(T)}}{2n}\bigg)^2 + \frac{L^2 D_W^2 s(T)}{4n}, \tag{A.3}$$

By choosing T such that the first term is closed to zero, we obtain that

$$\min_{A \in \mathcal{A}} \max_{h \in \mathcal{H}} \mathcal{E}_{gen} \geq \Omega\bigg(\frac{s(T)}{n}\bigg). \tag{A.4}$$

□

A.2 Proof of Lemma D.1

To simplify the notation, we use $M(u)$ as a short hand notation of $M(u; S)$. Similar to $h(u)$, $K(u)$, and $w(u)$.

1. Let $w^* \in \arg\min R_S(w)$, since we have

$$R_S(w^*) = K(w^*, u = w^*, S) \geq K(w(u), u = w^*, S) \geq R_S(w(u = w^*)).$$

The equality holds. Therefore, $w = u = w^*$ is the optimal solutions of both $R_S(w)$ and $M(u; S)$.

2. Since $K(w, u)$ is a $(p - l)$-strongly convex function, $w(u)$ is unique. Then

$$M(u) = h(w(u)) + \frac{p}{2}\|w(u) - u\|^2.$$

Then, take the derivative of $M(u)$ with respect to u, we have

$$\begin{aligned} \nabla_u M(u) &= \left[\frac{\partial w(u)}{\partial u}\right]^T \cdot \nabla_{w(u)} h(w(u)) + \left[\frac{\partial w(u)}{\partial u} - I\right]^T \cdot p(w(u) - u). && \text{(A.5)} \\ &= \left[\frac{\partial w(u)}{\partial u}\right]^T \cdot (\nabla_{w(u)} h(w(u)) + p(w(u) - u)) + p(u - w(u)). && \text{(A.6)} \end{aligned}$$

The first term is equal to zero. It is because $w(u)$ is the optimal solution of $K(w, u)$. It satisfies the first order condition,

$$\nabla_{w(u)} K(w(u), u) = \nabla_{w(u)} h(w(u)) + p(w(u) - u) = 0. \tag{A.7}$$

Therefore, we have $\nabla_u M(u) = p(u - w(u))$.

3. In Eq. (A.7), take the derivatives with respect to u on both sides, we have

$$\left[\frac{\partial w(u)}{\partial u}\right]^T \nabla_w^2 h(w) + p(\left[\frac{\partial w(u)}{\partial u}\right]^T - I) = 0. \tag{A.8}$$

Organizing the terms, we have

$$\left[\frac{\partial w(u)}{\partial u}\right]^T (\nabla_w^2 h(w) + pI) = pI. \tag{A.9}$$

Since $h(w)$ is l-weakly convex, $\nabla_w^2 h(w) + pI$ is positive definite. Then,

$$\left[\frac{\partial w(u)}{\partial u}\right]^T \prec \frac{p}{p-l} I. \tag{A.10}$$

Then,

$$\nabla_u^2 M(u) = [\frac{\partial}{\partial u} p(u - w(u))]^T = p(I - \left[\frac{\partial w(u)}{\partial u}\right]^T) \succ p(1 - \frac{p}{p-l})I. \tag{A.11}$$

Therefore, $M(u)$ is a $pl/(p-l)$-weakly convex function.

4. By Eq. (A.10), we have

$$\|\nabla M(u_1) - \nabla M(u_2)\| = p\|u_1 - w(u_1) - u_2 - w(u_2)\| \leq p(1 + \frac{p}{p-l})\|u_1 - u_2\|. \tag{A.12}$$

Therefore, $M(u; S)$ is $(2p^2 - pl)/(p-l)$-gradient Lipschitz continuous.

5. By Eq. (A.7),

$$\|\nabla_u M(u)\| = \|p(u - w(u))\| = \|\nabla_w h(w)\| \leq L. \tag{A.13}$$

□

A.3 Proof of Thm. 2.1

The training loss is a standard result of runing GD on smooth objective function. Before we provide the proof of the generalization bound in Thm. 2.1, we first introduce the following Lemma.

Lemma A.1. *In weekly-convex case, for neighbouring S and S', we have*

$$\|w(u; S) - w(u; S')\| \leq 2L/(n(p - \ell)).$$

Proof. By the $(p-l)$-strongly convexity of $K(w, u; S)$, we have

$$\begin{aligned}
& (p-l)\|w(u;S) - w(u;S')\| \\
\leq\ & \|\nabla K(w(u;S), u; S) - \nabla K(w(u;S'), u; S)\| \\
\leq\ & \|\nabla K(w(u;S), u; S) - \nabla K(w(u;S'), u; S')\| \\
& \frac{1}{n}\|\nabla h(w(u;S'), z_i)\| + \frac{1}{n}\|\nabla h(w(u;S'), z_i')\| \\
=\ & \frac{1}{n}\|\nabla h(w(u;S'), z_i)\| + \frac{1}{n}\|\nabla h(w(u;S'), z_i')\| \\
\leq\ & \frac{2L}{n},
\end{aligned}$$

where the second inequality is due to the definition of $K(w, u; S)$, the third one is due to the first-order optimally condition, and the last inequality is because of the bounded gradient of $h(w; z)$. □

Next, we move to the proof of Thm. 2.1.

$$\begin{aligned}
& \|u_S^{t+1} - u_{S'}^{t+1}\| \\
=\ & \|u_S^t - u_{S'}^t - \alpha_t(\nabla M(u_S^t; S) - \nabla M(u_{S'}^t; S'))\| \\
\leq\ & \|u_S^t - u_{S'}^t - \alpha_t(\nabla M(u_S^t; S) + \nabla M(u_{S'}^t; S))\| + \alpha_t\|\nabla M(u_{S'}^t; S') - \nabla M(u_{S'}^t; S)\| \\
\leq\ & \|u_S^t - u_{S'}^t\| + \alpha^t\|\nabla M(u_{S'}^t; S') - \nabla M(u_{S'}^t; S)\| \\
=\ & \|u_S^t - u_{S'}^t\| + \alpha^t p\|u_{S'}^t - u_{S'}^t - w(u_{S'}^t, S) + w(u_{S'}^t, S')\| \\
\leq\ & \|u_S^t - u_{S'}^t\| + \frac{2L\alpha_t}{n},
\end{aligned}$$

where the second inequality is due to the non-expansive [19] of convex function $M(u; S)$, the last inequality is due to Lemma A.1. unwind the recursive, we have

$$\|u_S^T - u_{S'}^T\| \leq \frac{2L\sum_{t=1}^T \alpha_t}{n}.$$

□

A.4 Proof of Lemma 2.1

Lemma 2.1 can be obtained from classical strong-convex optimization results. Since

$$\|\nabla_w K(w,u;z)\| = \|\nabla_w h(w;z) + p(w-u)\| \le L + pD_W,$$

$K(w,u;z)$ has bounded gradient $L_K = L + pD_W$. By [25], running SGD on $K(w,u;S)$ with stepsize $c_s \le 1/s(p-l)$ iccurs an optimization error in

$$E\|w_N - w(u)\|^2 \le \frac{C_1^2}{N},$$

where $C_1 = (L + pD_W)/(p-l)$.

A.5 Proof of Thm. 2.2

Proof. Let $A_{t+1} = \frac{1}{2}\|u^{t+1} - u^*\|^2$ and $a_{t+1} = \frac{1}{2}\mathbb{E}\|u^{t+1} - u^*\|^2$.

$$\begin{aligned}
A_{t+1} &= \frac{1}{2}\|u^{t+1} - u^*\|^2 \\
&\le \frac{1}{2}\|u^t - \alpha_t \nabla_u K(w_N^t, u^t; S) - u^*\|^2 \\
&\le A_t + \frac{1}{2}\alpha_t^2 L_K^2 - \alpha_t \langle \nabla_u K(w_N^t, u^t; S), u^t - u^* \rangle \\
&= A_t + \frac{1}{2}\alpha_t^2 L_K^2 - \alpha_t \langle \nabla_u M(u^t; S), u^t - u^* \rangle \\
&\quad + \alpha_t \langle \nabla_u M(u^t; S) - \nabla_u K(w_N^t, u^t; S), u^t - u^* \rangle.
\end{aligned}$$

Taking expectation on both side, rearranging the terms, we have

$$\begin{aligned}
&\alpha_t \mathbb{E}[M(u^t) - M(u^*)] \\
\le\ & a_t - a_{t+1} + \frac{1}{2}\alpha_t^2 L_K^2 + \alpha_t \mathbb{E}\langle \nabla_u M(u^t; S) - \nabla_u K(w_N^t, u^t; S), u^t - u^* \rangle
\end{aligned} \tag{A.14}$$

Since

$$\begin{aligned}
&\mathbb{E}\langle \nabla_u M(u^t; S) - \nabla_u K(w_N^t, u^t; S), u^t - u^* \rangle \\
\le\ & \|\nabla_u M(u^t; S) - \nabla_u K(w_N^t, u^t; S)\| \mathbb{E}\|u^t - u^*\| \\
\le\ & \frac{pC_1 D_W}{\sqrt{N}},
\end{aligned}$$

Eq. (A.14) becomes

$$\begin{aligned}
&\alpha_t \mathbb{E}[M(u^t) - M(u^*)] \\
\le\ & a_t - a_{t+1} + \frac{1}{2}\alpha_t^2 L_K^2 + \frac{\alpha_t p C_1 D_W}{\sqrt{N}}.
\end{aligned}$$

Let $N \ge T$, taking the summation over t, we obtain that

$$\begin{aligned}
&\sum_{t=1}^{T} \alpha_t \mathbb{E}[M(u^t) - M(u^*)] \\
\le\ & a_0 + \frac{1}{2}\sum_{t=1}^{T} \alpha_t^2 L_K^2 + \frac{\sum_{t=1}^{T} \alpha_t p C_1 D_W}{\sqrt{T}}.
\end{aligned}$$

There exists $t \le T$, such that

$$\mathbb{E}[M(u^t) - M(u^*)] \le \frac{a_0 + \frac{1}{2}\sum_{t=1}^{T} \alpha_t^2 L_K^2 + \frac{\sum_{t=1}^{T} \alpha_t p C_1 D_W}{\sqrt{T}}}{\sum_{t=1}^{T} \alpha_t}.$$

Considering constant step $\alpha \leq 1/\sqrt{T}$, we have $\alpha \leq 1/T\alpha$ and $\alpha\sqrt{T} \leq 1$. Therefore,

$$\begin{aligned}\mathbb{E}[M(u^t) - M(u^*)] &\leq \frac{2a_0 + T\alpha^2 L_K^2 + 2\alpha\sqrt{T} p C_1 D_W}{2T\alpha} \\ &\leq \frac{\|u^0 - u^*\|^2 + L_K^2 + 2pC_1 D_W}{2T\alpha} \\ &= \frac{C_2}{T\alpha}.\end{aligned}$$

Since $M(u; S)$ and $R_S(w)$ have the same global solutions, we can use both of them to measure the optimization error. Above is the optimization error measure defined in $M(u; S)$. Below we provide the optimization error defined in $R_S(w)$.

$$\mathbb{E}[R_S(w(u^t)) - R_S(w^*)] \leq \mathbb{E}[M(u^t) - M(u^*)] \leq \frac{C_2}{T\alpha}.$$

Notice that the choices of algorithm output are slightly different. Therefore, we have

$$\mathcal{E}_{opt} \leq \frac{C_2}{T\alpha},$$

where $C_2 = \|u^0 - u^*\|^2/2 + pC_1 D_W + (L + pD_W)^2/2$. □

A.6 Proof of Thm. 2.3

Proof. We decompose $\|u_S^{t+1} - u_{S'}^{t+1}\|$ as

$$\begin{aligned}& \mathbb{E}\|u_S^{t+1} - u_{S'}^{t+1}\| \\ = \; & \mathbb{E}\|u_S^t - \alpha_t \nabla_u K(w_{N,S}^t, u_S^t; S) - u_{S'}^t + \alpha_t \nabla_u K(w_{N,S'}^t, u_{S'}^t; S')\| \\ \leq \; & \mathbb{E}\|u_S^t - \alpha_t \nabla_u M(u_S^t; S) - u_{S'}^t + \alpha_t \nabla_u M(u_{S'}^t; S')\| \\ + \; & 2\alpha_t \mathbb{E}\|\nabla_u K(w_{N,S}^t, u_S^t; S) - \nabla_u M(u_S^t; S)\| \\ \leq \; & \mathbb{E}\|u_S^t - u_{S'}^t\| + \frac{2L\alpha_t}{n} + 2\alpha_t p \mathbb{E}\|w_N^t - w(u^t)\| \\ \leq \; & \mathbb{E}\|u_S^t - u_{S'}^t\| + \frac{2L\alpha_t}{n} + 2\alpha_t p \frac{C_1}{\sqrt{N}}.\end{aligned}$$

Let $N \geq n^2$, unwind the recursive and let u^T be the output of the algorithm, we have

$$\begin{aligned}\mathcal{E}_{gen} &\leq L\mathbb{E}\|u_S^T - u_{S'}^T\| \\ &\leq \frac{L(2L + 2C_1 p)\sum_{t=1}^T \alpha_t}{n} \\ &= \frac{C_3 \sum_{t=1}^T \alpha_t}{n}.\end{aligned}$$

If we choose $w(u^T)$ as algorithm output, since $\nabla M(u; S) = p(u - w(u))$, we have

$$\begin{aligned}\mathcal{E}_{gen} &\leq L\mathbb{E}\|w(u_S^T; S) - w(u_{S'}^T; S')\| \\ &= L\mathbb{E}\|u_S^T - \frac{1}{p}\nabla M(u_S^T, S) - u_{S'}^T - \frac{1}{p}\nabla M(u_{S'}^T; S'))\| \\ &\leq L\mathbb{E}\|u_S^T - u_{S'}^T\| + \frac{2L^2}{np} \\ &\leq \frac{L(2L + 2C_1 p)\sum_{t=1}^T \alpha_t}{n} + \frac{2L^2}{np} \\ &= \mathcal{O}\left(\frac{\sum_{t=1}^T \alpha_t}{n}\right). \end{aligned} \tag{A.15}$$

where the second inequality is due to the non-expansive of $M(u; S)$. □

B Related Work

Adversarial Attacks and Defense. Starting from the work of [34], it has now been well known that deep neural networks trained via standard gradient descent based algorithms are highly susceptible to imperceptible corruptions to the input data [17, 9, 7, 23]. This has led to a series of work aimed at training neural networks robust to such perturbations [38, 18] and works aimed at designing more sophisticated attacks to attack the classifiers [2, 36, 9].

Adversarial Generalization. The work of [32, 28, 44] have shown that in some scenarios achieving adversarial generalization requires more data. The work of [3, 24] explains generalization in adversarial settings using VC-dimension. [12] studies PAC-learning guarantees in the adversarial setting via VC-dimension. VC-dimension usually depends on the number of parameters in the model, while Rademacher complexity usually depends on the weight matrices. Rademacher complexity usually provides tighter generalization bounds [4]. [27] uses a PAC-Bayesian approach to provide a generalization bound for neural networks. [33] study the generalization of an adversarial training algorithm in terms of distributional robustness. The work of [41, 42, 21] study the generalization properties in the setting of linear regression. Gaussian mixture models are used to analyze adversarial generalization [35, 21, 13]. The work of [1] explains adversarial generalization through the lens of feature purification.

Robust Overfitting. Starting from the work of [30], a series of work studied the causes of robust overfitting. [43] studied robust overfitting from the perspective of adversarial distribution. [10] leveraged knowledge distillation and self-training to mitigate robust overfitting.

Uniform Stability. Stability can be traced back to the work of [31]. In statistical learning problems, it was well developed in analyzing the generalization bounds [6]. These bounds have been significantly improved in a recent sequence of works [15, 16]. The work of [11] discussed the optimal trade-off between stability and convergence.

C Preliminaries: Stability analysis for generalization gap

Let $\mathcal{D}$ be an unknown distribution over examples from space $\mathcal{Z}$. Let $S = \{z_1, \dots, z_n\} \sim \mathcal{D}^n$ be the sample dataset drawn i.i.d. from $\mathcal{D}$. Our goal is to find a model w with small population risk, defined as:

$$R_{\mathcal{D}}(w) = \mathbb{E}_{z\sim\mathcal{D}} h(w, z),$$

where $h(\cdot, \cdot)$ is the loss function. Since we cannot minimize the objective $R_{\mathcal{D}}(w)$ directly, we instead minimize the empirical risk, defined as

$$R_S(w) = \frac{1}{n}\sum_{i=1}^{n} h(w, z_i).$$

Let $\bar{w}$ be the optimal solution of $R_S(w)$. Then, for the algorithm output $\hat{w} = A(S)$, we define the expected generalization gap as

$$\mathcal{E}_{gen}(A, h, n, \mathcal{D}) = \mathbb{E}_{S\sim\mathcal{D}^n, A}[R_{\mathcal{D}}(A(S)) - R_S(A(S))]. \tag{C.1}$$

We define the the expected optimization gap as

$$\mathcal{E}_{opt}(A, h, n, \mathcal{D}) = \mathbb{E}_{S\sim\mathcal{D}^n, A}[R_S(A(S)) - R_S(\bar{w})]. \tag{C.2}$$

We use $\mathcal{E}_{gen}$ and $\mathcal{E}_{opt}$ as short hand notations of the above definition. To bound the generalization gap of a model $\hat{w} = A(S)$ trained by a randomized algorithm A, we employ the following notion of *uniform stability*.

Definition C.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*

$$\sup_z \mathbb{E}_A \left[h(A(S); z) - h(A(S'); z)\right] \le \varepsilon\,. \tag{C.3}$$

The following theorem shows that expected generalization gap can be attained from uniform stability.

Theorem C.1 (Generalization in expectation [19])**.** *Let A be ε-uniformly stable. Then, the expected generalization gap satisfies*

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

D Minimax Lower Bound

Adversarial Loss. In adversarial training, we consider the following adversarial loss

$$h(w; z) = \max_{\|z-z'\|\leq\epsilon} g(w; z'),$$

where $g(w; z)$ is the loss function of the standard counterpart. In practice, w is usually the parameter of neural networks. As discussed in [40, 39], even if g is a smooth function, h is not necessarily smooth. Therefore, we mainly consider the following function class of convex, non-smooth, Lipschitz functions throughout the paper.

$$\mathcal{H} = \{h : W \times \mathcal{Z} \to \mathbb{R} \mid h \text{ is convex, } L\text{-Lipshitz in } w, |W| = D_W\}. \tag{D.1}$$

If we assume that g is convex and L-Lipschitz, *i.e.,* $|g(w_1; z) - g(w_2; z)| \leq L\|w_1 - w_2\|$, it is easy to prove that h is also convex and L-Lipschitz [39]. L-Lipschitz is a standard assumption in uniform stability analysis since [19]. The assumption of convexity is to compare with the existing results and to develop the following the minimax lower bound.

Definition D.1 (Training Loss)**.** *We say an algorithm class $\mathcal{A}$ has training loss $1/s(T)$ on a function class $\mathcal{H}$, if for all $A \in \mathcal{A}$ and $h \in \mathcal{H}$, running A on h for T iterations, we have*

$$\mathcal{E}_{opt}(A, h, n, \mathcal{D}) \leq \mathcal{O}\left(\frac{1}{s(T)}\right),$$

where $\lim_{T\to+\infty} s(T) = +\infty$.

Theorem D.1 (Minimax lower bound of generalization gap)**.** *Let $\mathcal{H}$ be the function class defined in Eq. (D.1). Let $\mathcal{A}$ be the class of randomized algorithms using n samples with training loss $1/s(T)$ on $\mathcal{H}$. For all n, there exists T, s.t. the following lower bound holds.*

$$\min_{A\in\mathcal{A}} \max_{\mathcal{D}} \mathcal{E}_{gen}(A, h, n, \mathcal{D}) \geq \Omega\left(\frac{s(T)}{n}\right). \tag{D.2}$$

The proof of Thm. D.1 is based on a lower bound of the complexity of Lipschitz-convex problems ([26], Ch.4), see Appendix A.1.

Given this lower bound, a natural question is whether the usual adversarial training algorithms can achieve this lower bound. For example, can we attain a solution with training loss $1/s(T) = 1/\sqrt{n}$ and generalization gap $s(T)/n = 1/\sqrt{n}$ simultaneously? In the literature, the SGDmax-based adversarial training algorithms are the most popular ones. However, we will see in the following subsection that SGDmax can not achieve the minimax lower bound.

D.1 SGDmax does not Achieve the Minimax Lower Bound

The following theorem shows that SGDmax does not achieve the minimax lower bound.

Theorem D.2 (Uniform stability for SGDmax [39])**.** *Suppose $g(w, z)$ is L-Lipschitz* w.r.t *w, L_w and L_z-gradient Lipschitz* w.r.t *w and z, respectively. Suppose in addition that $g(w, z)$ is convex in w for all given $z \in \mathcal{Z}$. If we run SGD on $h(w; z)$ with fixed step sizes $\alpha_t \leq 1/\sqrt{T}$ for $T \geq L_w^2$ steps. Then, SGDmax satisfies*

$$\mathcal{E}_{opt} \leq \mathcal{O}(1/T\alpha) \quad \text{and} \quad \mathcal{E}_{gen} \leq L\mathbb{E}[w_S^T - w_{S'}^T] \leq 2L(L_z\epsilon + L/n)T\alpha. \tag{D.3}$$

A worst-case lower bound, $\mathbb{E}[w_S^T - w_{S'}^T] \geq 2L_z\epsilon\sqrt{T}\alpha + \frac{LT\alpha}{n}$, is also given in the aforementioned work. In Thm. D.2, SGD on adversarial loss have training loss $s(T) = T\alpha$. However, the generalization bound is in the order of $\mathcal{O}(T\alpha + \frac{T\alpha}{n})$, which has a $c(T)$ gap compared to the minimax lower bound. The existence of the n-independent term $c(T)$ might be a reason for robust overfitting: even though we have arbitrarily large number of training samples, the generalization error still increases with T. From the stability analysis, the additional term $LL_z\epsilon T\alpha$ comes from the non-smoothness of the adversarial loss $h(w;z)$. It motivates us to design new algorithms to overcome the non-smooth issue. We introduce our proposed algorithm, Smoothed-SGDmax, in the next section.

Lemma D.1. *Assume h is l-weakly convex. Let $p > l$. Then, $M(u;S)$ satisfies*

1. $\min_u M(u;S)$ *has the same global solutions as* $\min_w R_S(w)$.
2. *The gradient of $M(u;S)$ is $\nabla_u M(u;S) = p(u - w(u;S))$.*
3. *$M(u;S)$ is $pl/(p-l)$-weakly convex.*
4. *$M(u;S)$ is $(2p^2 - pl)/(p-l)$-gradient Lipschitz continuous.*
5. *$M(u;S)$ has bounded gradient norm L.*

Remark: We focus on the case where h is convex in the main text. Then, Lemma D.1.3 and D.1.4 reduce to $M(u;S)$ is convex and $2p$-gradient Lipschitz. Lemma D.1 is stated in general l-weekly convex cases for further theoretical studies. Since $M(u;S)$ has the same global solutions as $R_S(w)$, we can do adversarial training using this smooth objective $M(u;S)$. A natural way is to perform gradient descent to $M(u;S)$. By Lemma D.1, the estimate of the gradient requires the estimate of the solution of the minimization problem $\min_w K(w,u;S)$.

E Further Comparison with Existing Algorithms

In Alg. 1, Step 7 is just to run SGD on $K(w,u;z) = h(w;z) + p\|w-u\|^2/2$ instead of $h(w;z)$. The additional term can be viewed as a regularization term similar to weight decay. Step 10 is a model averaging step similar to stochastic weight averaging (SWA). Therefore, we discuss the similarity of our Algorithm 1 in detail. The summary of the comparison is provided in Table 3.

Table 3: Comparison of adversarial training, adversarial training with weight decay, adversarial training with stochastic weight averaging, and our proposed algorithm, Smoothed-SGDmax.

	Operation on w	Operation on u
SGDmax (AT)	Mnimizing w on $R_S(w)$	No operation on u
AT with Weight decay	Mnimizing w on $K(w,u;S)$	Set $u = 0$
AT with SWA	Mnimizing w on $R_S(w)$	Minimizing u on $K(w,u;S)$
Smoothed-SGDmax (Ours)	Mnimizing w on $K(w,u;S)$	Minimizing u on $K(w,u;S)$

Weight Decay. Weight decay (WD) is to add a ℓ_2 regularization to the empirical loss. The loss function with WD is $h(w;z) + p\|w\|^2/2$. Therefore, if we replace Step 10 by $u = 0$ in Alg. 1, the proposed algorithm reduces to a simple weight decay regularization technique. Following the analysis in Thm. D.2, it is easy to see that adversarial training with weight decay incurs a generalization bound in

$$\mathcal{E}_{gen} \leq 2L(L_z\epsilon + L/n)T\alpha, \tag{E.1}$$

where the step size $\alpha \leq 1/(L_w - p)$. Therefore, weight decay is not guaranteed to reduce the additional n-independent term.

Stochastic Weight Averaging. Stochastic weight averaging suggests using the weighted average of the iterates rather than the final one for inference. The update rules of SWA is $u^{t+1} = \tau^t u^t + (1 - \tau^t)w^{t+1}$. In the work of [39], they provide a generalization bound for SWA in the case that u is the average of the iterates, which is equivalent to using the step size $u^t = (t-1)/t$. The generalization bound in this case is

$$\mathcal{E}_{gen}(SWA) \leq (LL_z\epsilon + 2L^2/n)T\alpha. \tag{E.2}$$

The n-independent term is one-half of the one without SWA. However, the additional term is still unavoidable in the analysis. SWA is still not guaranteed to achieve the minimax lower bound in this analysis.

Optimal Generalization Bound of SWA in our Regime. In Alg. 1, if we denote $\tau^t = 1 - \alpha^t p$, Step 10 can be view as a weight averaging step. In Thm. 2.3, it is required that $\alpha^t \leq 1/2p$. Then, $\tau^t = (1 - \alpha^t p) \geq 1/2$. Therefore, by fixing $\alpha^t p$ to be constant and letting $p \to 0$, our proposed algorithm is reduced to SWA. In other words, our proposed algorithm can be viewed as a general form of SWA. Also, we provide an optimal generalization bound of SWA in the regime that $\tau \in [1/2, 1]$ and $p \to 0$.

Comparison of the Algorithms. We list the theoretical results of our proposed algorithm and the existing algorithms in Table 4. For adversarial training, adversarial training with weight decay, and with SWA, the generalization upper bound is $\mathcal{O}(1 + 1/n)$ in terms of sample complexity. It is not guaranteed to be optimal. As for our proposed algorithm, Smoothed-SGDmax (including SWA with particular stepsizes), the generalization bound is $\mathcal{O}(1/n)$, achieving the minimax lower bound.

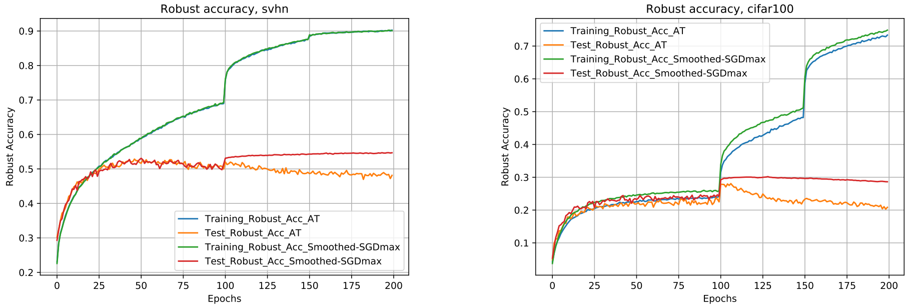
Table 4: Comparison of the stability-based generalization bounds of our proposed algorithm with the generalization bounds of adversarial training, adversarial training with weight decay, and adversarial training with stochastic weight averaging.

	Stepsize	Upper Bounds	Optimal Bounds
SGDmax (AT)	$\alpha \leq \frac{1}{L_w}$	$\mathcal{O}(L_z \epsilon T\alpha + \frac{T\alpha}{n})$	✗
AT with weight decay	$\alpha \leq \frac{1}{(L_w - p)}$	$\mathcal{O}(L_z \epsilon T\alpha + \frac{T\alpha}{n})$	✗
AT with SWA	$\tau = \frac{t-1}{t}, \alpha \leq \frac{1}{L_w}$	$\mathcal{O}(L_z \epsilon T\alpha + \frac{T\alpha}{n})$	✗
Smoothed-SGDmax (Ours)	$c_s \leq \frac{1}{ps}, \alpha \leq \frac{1}{2p}$	$\mathcal{O}(\frac{T\alpha}{n})$	✓

F Experiments

Training Procedure of Smoothed-SGDmax. To have a first glance of how Smoothed-SGDmax mitigates robust overfitting, we consider the experiments on a lightweight model, PreActResNet-18, on CIFAR-10, CIFAR-100, and SVHN to plot the training procedure.

Training Settings. For the attack algorithms, we use ℓ_∞-PGD-10 [23], $\epsilon = 8/255$. The step size is set to be $\epsilon/4$. For adversarial training, we use piece-wise learning rates, which are equal to $0.1, 0.01, 0.001$ for epochs 1 to 100, 101 to 150, and 151 to 200, respectively. For Smoothed-SGDmax, we keep the piece-wise learning rate (for the choice of c_s^t in Alg. 1) for comparison. Because of the similarity of ℓ_2 regularization term of weight decay and the proximal term in $K(w, u; z)$, we set $p = 5 \times 10^{-4}$, which is a common choice of weight decay. The step size α_t of updating u is set to be 50, then $\tau = 1 - \alpha p = 0.995$.



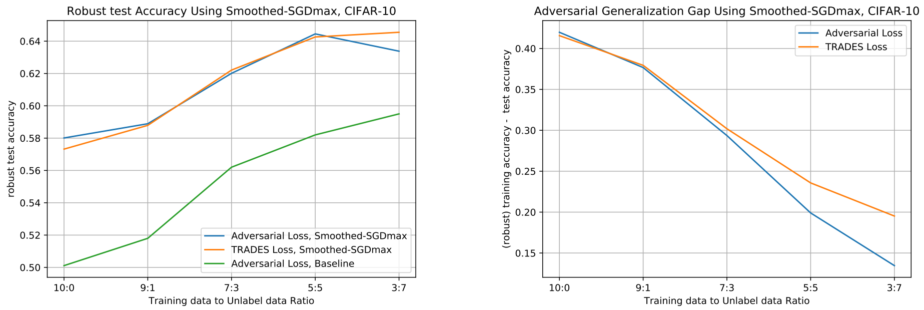
(a)

(b)

Figure 2: Robust test accuracy of adversarial training and Smoothed-SGDmax on SVHN and CFAR-100.

The training procedure of the experiments on CIFAR-10 is already provided in Introduction, Fig. 1. The experiments on SVHN and CIFAR-100 are provided in Fig. 2. For adversarial training, the robust test accuracy starts to decrease at around the 100^{th} epoch, which is called robust overfitting [30]. Using Smooth-SGDmax, the robust overfitting issue is much milder. These experiments verify the generalization bounds. The bound of Smoothed-SGDmax (which is $\mathcal{O}(T\alpha/n)$) is much better than the bound of adversarial training ($\mathcal{O}(T\alpha + T\alpha/n)$).

Sample Complexity. Secondly, we study the sample complexity provided in Thm. 2.3. We use Wide-ResNet-28 $\times$ 10 with Swish activation function for better test accuracy instead of ResNet-18. The training setting mainly follows the work of [18]. We consider two losses, adversarial loss [23] and TRADES loss [47] for the choice of $h(w; z)$. The total number of epochs is 400. Other training settings are similar to the experiments on ResNet-18.



(a) (b)

Figure 3: Robust test accuracy and generalization gap in the experiments of training CIFAR-10 using Smoothed-SGDmax.

Adversarial Generalization Gap. CIFAR-10 only contains 50K training samples. We adopt the pseudo-label data introduced in [8] to study the sample complexity. Increasing the percentage of pseudo-label data is an approximation of increasing the training data. In Fig. 3, we show the robust test accuracy (a) and adversarial generalization gap (b). The results are consistent with the theorem that Smoothed-SGDmax reduces a term in the generalization bounds.