Understanding and Improving Fast Adversarial Training against l_0 Bounded Perturbations

Xuyang Zhong, Yixiao Huang, Chen Liu
City University of Hong Kong

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

- Given a model with parameter $\boldsymbol{\theta}$ and input $\boldsymbol{x}$, we aim to find an adversarial perturbation such that

$$\max_{\boldsymbol{\delta} \in \mathcal{S}_p} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{\delta}),$$

where $\mathcal{S}_p = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_p \leq \epsilon, 0 \leq \boldsymbol{x} + \boldsymbol{\delta} \leq 1\}$.

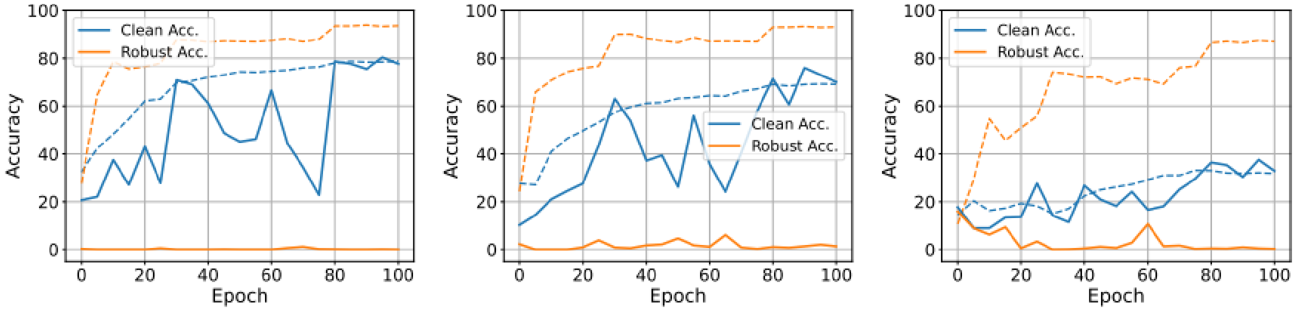
- Adversarial training is to solve a min-max optimization problem to construct a robust model:

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \max_{\boldsymbol{\delta}_i} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}_i + \boldsymbol{\delta}_i), \quad \text{s.t. } ||\boldsymbol{\delta}_i||_p \leq \epsilon,\ 0 \leq \boldsymbol{x}_i + \boldsymbol{\delta}_i \leq 1.$$

- We focus on l_0 bounded perturbations (i.e., $p = 0$) in this work.

Challenges in Fast l_0 Adversarial Training

- While effective, multi-step adversarial training (AT) introduces computational overhead.
- To reduce complexity, 1-step attack is adopted in AT. However, ***catastrophic overfitting*** **(CO)** occurs.



(a) $\epsilon_{train} = 20$ (b) $\epsilon_{train} = 40$ (c) $\epsilon_{train} = 120$

Dashed: training, based on 1-step attack Solid: test, based on Sparse-AutoAttack (sAA) [1]

- Traditional CO-mitigation methods do not work in the l_0 case.

Method	ATTA	Free-AT	GA	Fast-BAT	FLC Pool	N-AAER	N-LAP	NuAT	sTRADES
Robust Acc.	*0.0*	8.9	*0.0*	14.1	*0.0*	*0.1*	*0.0*	51.9	61.7

[1] Zhong et al, "Towards Efficient Training and Evaluation of Robust Models against l_0 Bounded Adversarial Perturbations". ICML 2024

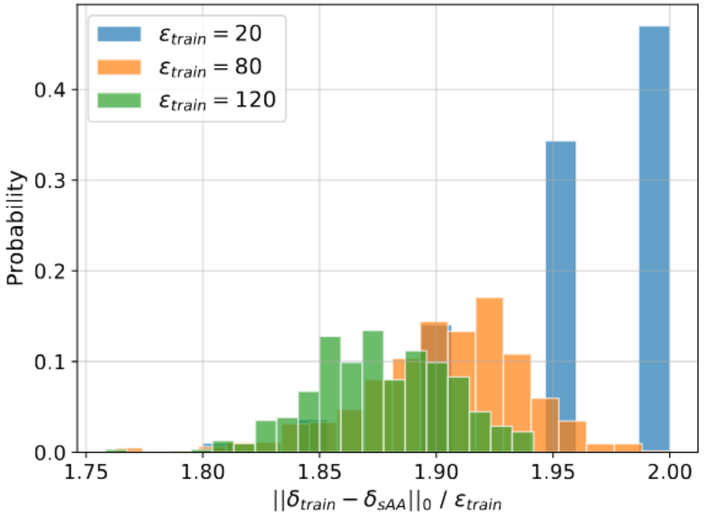
CO in l_0 Adversarial Training

Compared to the l_2 and l_∞ cases, CO in l_0 adversarial training is attributed to **sub-optimal perturbation locations** rather than sub-optimal perturbation magnitudes.

1. Successful adversarial examples cannot be completely found through simple interpolations
2. Perturbations generated by 1-step attack during training are almost completely different from those generated by sAA in **location**.

Table 2: Robust accuracy of the models obtained by 1-step sAT with different ϵ_{train} against the interpolation between perturbations generated by 1-step sPGD ($\epsilon = 20$) and their corresponding clean examples, where α denotes the interpolation factor, i.e., $\boldsymbol{x}_{interp} = \boldsymbol{x} + \alpha \cdot \boldsymbol{\delta}$. The results of sAA are also reported.

α	0.0	0.1	0.2	0.3	0.4	0.6	0.8	1.0	**sAA**
$\epsilon_{train} = 20$	77.5	69.8	**69.1**	73.7	80.4	88.0	90.2	90.4	**0.0**
$\epsilon_{train} = 40$	70.2	**63.1**	64.3	70.9	79.8	87.4	89.6	89.6	**0.0**
$\epsilon_{train} = 120$	32.5	26.5	**24.5**	29.4	41.5	65.2	72.8	67.6	**0.0**



Loss Landscape Analysis

Sub-optimal location issue can be mitigated to some extent by multi-ϵ strategy. However, a larger ϵ_{train} in turn, leads to unstable training and degraded clean accuracy. In this regard, We investigate the **loss landscape in l_0 AT**.

From theoretical perspective, we prove:

1. Lipschitz continuity of adversarial loss function can be guaranteed.
2. Adversarial loss function is no longer smooth, **larger ϵ aggravates the non-smoothness**.
3. **The loss landscape in l_0 adversarial training can be more craggy than other cases.**

Lemma 3.2. ***(Lipschitz continuity of adversarial loss)*** *If Assumption 3.1 holds, we have:*

$$\forall \boldsymbol{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \quad \|\mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_1) - \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_2)\| \leq A_{\boldsymbol{\theta}} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|, \tag{4}$$

The Lipschitz constant $A_{\boldsymbol{\theta}} = 2\sum_{i \in \mathcal{S}_+} y_i L_{\boldsymbol{\theta}}$ *where* $\mathcal{S}_+ = \{i \mid y_i > 0, h_i(\boldsymbol{x} + \boldsymbol{\delta}_1, \boldsymbol{\theta}_2) > h_i(\boldsymbol{x} + \boldsymbol{\delta}_1, \boldsymbol{\theta}_1)\}$, $\boldsymbol{\delta}_1 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta})$ *and* $\boldsymbol{\delta}_2 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta})$.

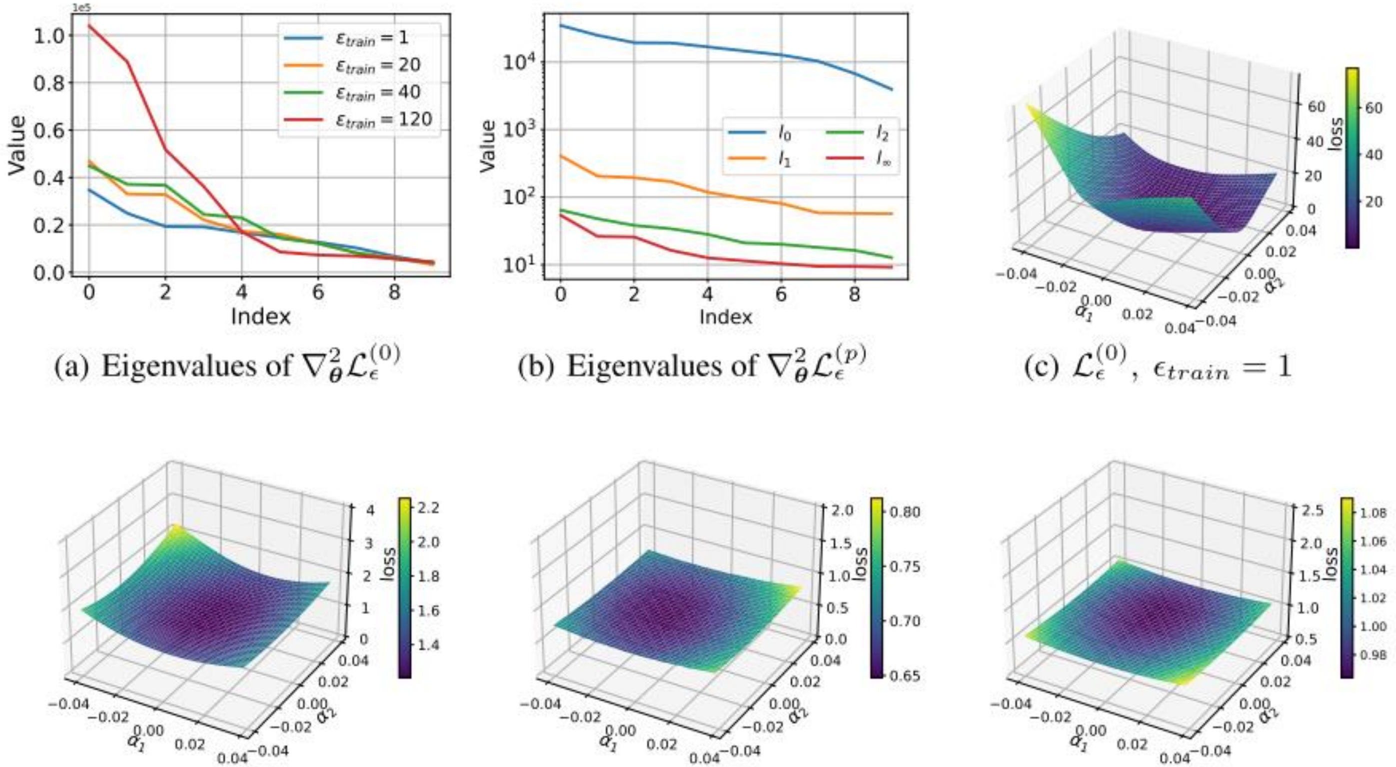
Lemma 3.4. ***(Lipschitz smoothness of adversarial loss)*** *If Assumption 3.1 and 3.3 hold, we have:*

$$\forall \boldsymbol{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \quad \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_1) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_2)\| \leq A_{\boldsymbol{\theta\theta}} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| + B_{\boldsymbol{\theta\delta}}. \tag{7}$$

The Lipschitz constant $A_{\boldsymbol{\theta\theta}} = L_{\boldsymbol{\theta\theta}}$ *and* $B_{\boldsymbol{\theta\delta}} = L_{\boldsymbol{\theta x}} \|\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2\| + 4L_{\boldsymbol{\theta}}$ *where* $\boldsymbol{\delta}_1 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}_1)$ *and* $\boldsymbol{\delta}_2 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}_2)$.

Loss Landscape Analysis

Numerical results further demonstrate the craggy loss landscape in the l_0 AT



(a) Eigenvalues of $\nabla^2_{\theta}\mathcal{L}^{(0)}_{\epsilon}$ (b) Eigenvalues of $\nabla^2_{\theta}\mathcal{L}^{(p)}_{\epsilon}$ (c) $\mathcal{L}^{(0)}_{\epsilon}$, $\epsilon_{train} = 1$

(d) $\mathcal{L}^{(1)}_{\epsilon}$, $\epsilon_{train} = 24$ (e) $\mathcal{L}^{(2)}_{\epsilon}$, $\epsilon_{train} = 0.5$ (f) $\mathcal{L}^{(\infty)}_{\epsilon}$, $\epsilon_{train} = 8/255$

Recipe

We propose to leverage **soft labels** and **trade-off loss function** to provably improve Lipschitz continuity and Lipschitz smoothness, respectively.

Theorem 4.1. ***(Soft label improves Lipschitz continuity)*** *Based on Lemma 3.2, given a hard label vector* $\boldsymbol{y}_h \in \{0,1\}^K$ *and a soft label vector* $\boldsymbol{y}_s \in (0,1)^K$*, we have* $A_{\boldsymbol{\theta}}(\boldsymbol{y}_s) \leq A_{\boldsymbol{\theta}}(\boldsymbol{y}_h)$*.*

Trade-off loss function: $\mathcal{L}_{\epsilon,\alpha}(\boldsymbol{x},\boldsymbol{\theta}) = (1-\alpha)\mathcal{L}(\boldsymbol{x},\boldsymbol{\theta}) + \alpha \max_{\boldsymbol{\delta}\in\mathcal{S}_\epsilon(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{\theta})$

Theorem 4.2. ***(Trade-off loss function improves Lipschitz smoothness)*** *If Assumption 3.1 and 3.3 hold, we have:*

$$\|\nabla_\theta \mathcal{L}_{\epsilon,\alpha}(\boldsymbol{x},\boldsymbol{\theta}_1) - \nabla_\theta \mathcal{L}_{\epsilon,\alpha}(\boldsymbol{x},\boldsymbol{\theta}_2)\| \leq A_{\boldsymbol{\theta}\boldsymbol{\theta}}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| + B'_{\boldsymbol{\theta}\boldsymbol{\delta}} \tag{9}$$

The Lipschitz constant $A_{\boldsymbol{\theta}\boldsymbol{\theta}} = L_{\boldsymbol{\theta}\boldsymbol{\theta}}$ *and* $B'_{\boldsymbol{\theta}\boldsymbol{\delta}} = \alpha L_{\boldsymbol{\theta}\boldsymbol{x}}\|\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2\| + 2(1+\alpha)L_{\boldsymbol{\theta}}$ *where* $\boldsymbol{\delta}_1 \in \arg\max_{\boldsymbol{\delta}\in\mathcal{S}_\epsilon(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{\theta}_1)$ *and* $\boldsymbol{\delta}_2 \in \arg\max_{\boldsymbol{\delta}\in\mathcal{S}_\epsilon(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{\theta}_2)$*.*

Experiments

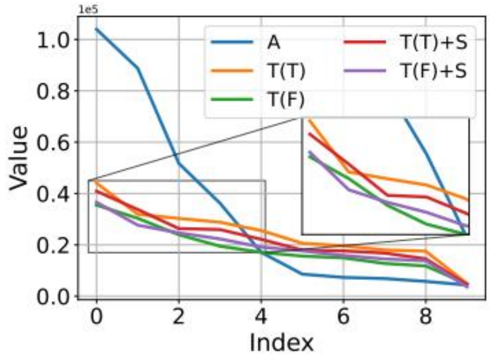
- Evaluating different combinations of techniques incorporating soft labels or/and trade-off loss function. We name the best combination **Fast-LS-** $\boldsymbol{l_0}$.

Table 3: Comparison of different approaches and their combinations in robust accuracy (%) by sAA. The target sparsity level $\epsilon = 20$. We compare PreAct ResNet-18 (He et al., 2016a) models trained on CIFAR-10 (Krizhevsky et al., 2009) with 100 epochs. The *italic numbers* indicate catastrophic overfitting (CO) happens.

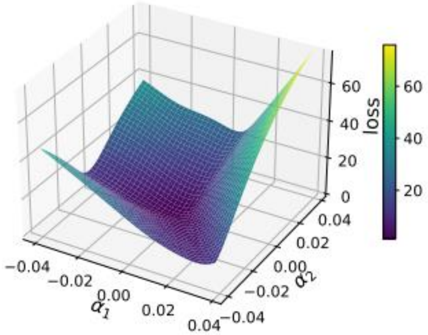
Method	sAT	Tradeoff	sTRADES (T)	sTRADES (F)
1-step	*0.0*	*2.6*	31.0	55.4
+ N-FGSM	*0.3*	*17.5*	46.9	55.9
+ SAT	29.3	30.3	61.4	59.4
+ SAT & N-FGSM	**43.8**	**39.2**	**63.0**	**62.6**

Experiments

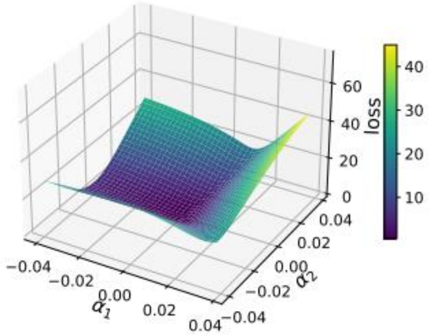
- Our method smooths the loss landscape



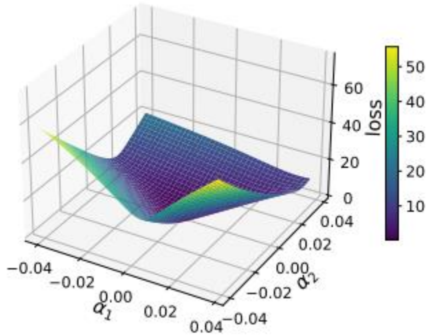
(a) Eigenvalues of $\nabla^2_{\boldsymbol{\theta}}\mathcal{L}^{(0)}_{\epsilon}$



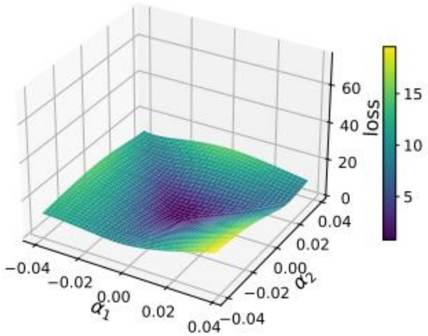
(b) 1-step sAT



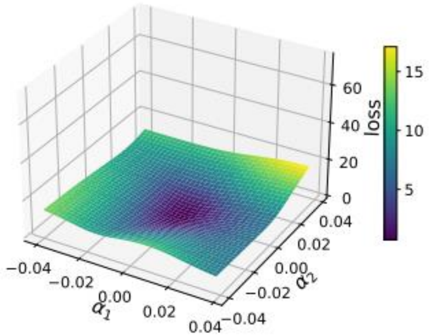
(c) 1-step sTRADES (T)



(d) 1-step sTRADES (F)



(e) 1-step sTRADES (T) + SAT



(f) 1-step sTRADES (F) + SAT

Experiments

- Our method also benefits multi-step AT

(a) **CIFAR-10,** $\epsilon = 20$

Model	Time Cost	Clean	Black CS	Black RS	White SAIF	White σ-zero	White sPGD$_p$	White sPGD$_u$	sAA
Multi-step									
sAT	5.3 h	84.5	52.1	36.2	76.6	79.8	75.9	75.3	36.2
+S&N	5.5 h	80.8	64.1	61.1	76.1	78.7	76.8	75.1	61.0
sTRADES	5.5 h	89.8	69.9	61.8	84.9	85.9	84.6	81.7	61.7
+S&N	5.4 h	82.2	66.3	66.1	77.1	77.0	74.1	72.2	**65.5**
One-step									
Fast-LS-l_0	0.8 h	82.5	69.3	65.4	75.7	73.7	67.2	67.7	**63.0**

(b) **ImageNet-100,** $\epsilon = 200$

Model	Time Cost	Clean	Black RS	White SAIF	White σ-zero	White sPGD$_p$	White sPGD$_u$	sAA
Multi-step								
sAT	325 h	86.2	61.4	69.0	78.6	78.0	77.8	61.2
+S&N	336 h	83.0	75.0	76.4	80.8	78.8	79.2	74.8
sTRADES	359 h	84.8	76.0	77.4	81.6	80.6	81.4	75.8
+S&N	360 h	82.4	78.2	79.2	80.0	78.2	79.8	**77.8**
One-step								
Fast-LS-l_0	44 h	82.4	76.8	75.4	74.0	74.6	74.6	**72.4**

Thanks!