ADVERSARIAL ROBUSTNESS VIA ADAPTIVE LABEL SMOOTHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training (AT) has become a dominant defense paradigm by enforcing the model's predictions to be locally invariant to adversarial examples. Being a simple technique, Label smoothing (LS) has shown its potential for improving model robustness. However, the prior study shows the benefit of directly combining two techniques together is limited. In this paper, we aim to better understand the behavior of LS and explore new algorithms for more effective LS on improving adversarial robustness. We first show both theoretically and empirically that strong smoothing in AT increases local smoothness of the loss surface which is beneficial for robustness but sacrifices the training loss which influences the accuracy of samples near the decision boundary. Based on this result, we propose *surface smoothing adversarial training* (SSAT). Specifically, much stronger smoothness is used on the perturbed examples farther away from the decision boundary to achieve better robustness, while weaker smoothness is on those closer to the decision boundary to avoid incorrect classification on clean samples. Meanwhile, LS builds a different representation space among data classes in which SSAT differs from other AT methods. We study such a distinction and further propose a cooperative defense strategy termed by Co-SSAT. Experimental results show that our Co-SSAT achieves the state-of-the-art performances on CIFAR-10 with ℓ_∞ adversaries and also has a good generalization ability of unseen attacks, i.e., other ℓ_p norms, or larger perturbations due to the smoothness property of the loss surface.

1 INTRODUCTION

Adversarial examples (Szegedy et al., 2014; Biggio et al., 2013) are ubiquitous in Deep Neural Networks (DNNs), incurring attacking risks across various areas from classification (Goodfellow et al., 2015), object detection (Chen & Martin, 2018) and especially face recognition (Song et al., 2021). Among the techniques for improving robustness of DNNs against adversarial examples (Xu et al., 2018; Samangouei et al., 2018; Meng & Chen, 2017), adversarial training (AT) (Madry et al., 2018) has shown its dominance via solving a min-max game through a conceptually simple process: train the model on adversarial examples rather than clean samples. Since adversarial examples are crafted around the original input, AT trains a robust model whose predictions are locally invariant to a neighborhood of its inputs, i.e. smoothing the local loss surface. Based on the idea of local smoothness, many adversarial training methods are proposed to further improve model robustness, such as TRADES (Zhang et al., 2019) which minimizes the difference between the prediction of the original model and robust model, and LLR (Qin et al., 2019) which designs a novel regularization method to increase the local linearity of the loss surface.

Label smoothing (LS), as a popular way of regularization, recently shows its potential in improving model robustness (Shafahi et al., 2019a; Pang et al., 2021; Stutz et al., 2020). Then it is attractive to combine LS with adversarial training (AT) to achieve even better robustness. However, Pang et al. (2021) shows the limited benefit of such a combination. These intriguing results motivate us to consider: *Why LS **can improve** model robustness? But why does LS in AT **NOT work well?***

For the first question, we theoretically show that *LS can smooth the loss surface through narrowing bounds of local gradients of models w.r.t adversarial perturbations*. Then we design an input-specific adaptive LS strategy to smooth the loss surface efficiently and achieve higher robustness.

For the second question, we discover that LS has a side-effect of lifting the whole loss surface. Since adversarial examples are more likely to be misclassified than clean ones, LS would push those adversarial examples closer to the decision boundary and even cross it. Therefore, we propose a heuristic AT method, i.e., surface smoothing adversarial training (SSAT), applying an adaptive LS to each adversarial example based on its distance towards the decision boundary. Specifically, stronger smoothness is imposed on the perturbed examples closer to the clean samples to achieve better smoothness because it's less likely to find its adversarial perturbation that can mislead model's decision. Second, weaker smoothness is given on the perturbed example closer to the original decision boundary. Otherwise, raw hard labels are kept for the rest of adversarial examples which cross the decision boundary to alleviate the cost of LS.

The recent study (Müller et al., 2019) empirically shows that LS enforces the examples of the same class group in a tighter cluster, separating equidistantly from all the other classes' examples compared to those without LS. Since adversarial examples fool classifiers by perturbing original representations, we assume that our SSAT works fundamentally differently from other AT methods in terms of defense. Our further evidence shows that responses of SSAT to targeted and untargeted attacks are distinct to each other and SSAT shows better robustness against untargeted attacks to targeted attacks. Following this result, we propose cooperative SSAT (Co-SSAT), i.e., SSAT cooperates with another robust classifier against various adversaries together. Experiments show that Co-SSAT fully exploits each of its models to defend against different types of adversarial attacks and Co-SSAT boosts SSAT.

Our contributions are in four-folds: 1) We first analyze the theoretical relationship between LS and loss surface smoothing and demonstrate our statement through statistical analysis; **2)** Based on our theoretical analysis, we propose a new AT method, i.e., SSAT (Surface Smoothing Adversarial Training). **3)** We further study differences between SSAT and other AT methods and propose the cooperative defense strategy, i.e., Co-SSAT against multiple adversaries together. **4)** Empirically, our Co-SSAT achieves the state-of-the-art robustness on CIFAR-10. We show that our model can also successfully defend attacks outside the ℓ_p bounded ball which otherwise can hardly be handled by other AT methods.

2 Backgrounds and Related Works

Adversarial training. Adversarial training (Madry et al., 2018) trains classifiers on the adversarial examples instead of clean data to improve model robustness. Specifically, given a K-class dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$ with input $\mathbf{x}$ and one-hot label $\mathbf{y}$ (and ground truth y^{gt}), the problem can be formulated:

$$\min_{\boldsymbol{\theta}} E_{\mathbf{x},\mathbf{y}} \max_{\|\delta\|_p \leq \epsilon} L\left(f_{\boldsymbol{\theta}}(\mathbf{x}+\delta), \mathbf{y}\right), \tag{1}$$

where ϵ means the radius of the ℓ_p ball centered at the clean training sample $\mathbf{x}$, $f_{\boldsymbol{\theta}}(\cdot)$ represents the neural network with parameters $\boldsymbol{\theta}$, and $L(\cdot)$ is the cross-entropy loss.

We adopt the framework of **free adversarial training** (Shafahi et al., 2019b), in which the training process leverages the adversarial example from every iteration in a PGD attack. This framework naturally fits our goal of building a local smooth loss surface: during the iterative process, these intermediate adversarial examples gradually converge to the decision boundary. Thus we assign an adaptive LS degree to each of the following such optimization trajectories from the original data point to its decision boundary (even across the boundary), resulting in a more smooth loss surface.

Label smoothing for robustness. LS converts 'one-hot' label vectors into 'soft' vectors with a low-confidence classification, as a common technique to reduce over-fitting on general classifications. For a K-class classification problem, the one-hot label vector $\mathbf{y}$ can be smoothed by:

$$\mathbf{y}_s = \mathbf{y} - s \times (\mathbf{y} - 1/K), \tag{2}$$

where $s \in [0, 1]$ controls the smoothing level. We can get a hard decision vector without smoothing when $s = 0$, while $s = 1$ represents the uniform choice of labels. To our knowledge, (Shafahi et al., 2019a) first uses LS to mimic the mechanism of AT, which shows the great potential for improved robustness. Later, (Pang et al., 2021) explores a bag of techniques of AT and finds only a narrow range of smoothing degrees has limited benefit for robustness, but in our experiments, a slightly larger or smaller degree is ineffective or even failed, (see Fig. 6). (Cheng et al., 2020) design an adaptive strategy in adjusting the decision boundary and adopts simple LS with it together to get good results. However, few studies have explored why LS is effective to improve robustness, and why

simply combining LS with AT cannot work. In this paper, we try to fill these two gaps and propose our model and theory to improve the robustness.

3 THEORETICAL ANALYSIS

In this section, we present a theoretical analysis of the relationship between label smoothing(LS) and model robustness. We first define the loss surface smoothness, then prove that LS builds a smoother loss surface, and thus improves model robustness.

3.1 LOSS SURFACE SMOOTHNESS

Given a loss function $L(\cdot)$ and $\forall\ (\mathbf{x}, \mathbf{y}) \in \mathcal{D}$, we use the bound of local gradients with respect to $\mathbf{x}$ to represent the smoothness of the loss surface. Suppose a small perturbation in the neighborhood of the original inputs, the bounded local gradient indicates the bounded change of the loss. That is to say, even the attacker has the access to the gradients, it becomes harder for the attacker to fool the classifier if the bound of the local gradient is tighter. Conceptually, it is consistent with the argument that smoothness is an indispensable property of robust models (Cisse et al., 2017).

In the next subsection, we will illustrate that given an $(\mathbf{x}, \mathbf{y})$, two classifiers f_θ and $f_{\tilde{\theta}}$ trained by the cross-entropy (CE) loss $L(\cdot)$ without and with LS respectively, the loss surface with LS is smoother than that without LS, which is specified as:

$$|\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}_i}| < |\frac{\partial L(\mathbf{x}, \mathbf{y}, \theta)}{\partial \mathbf{x}_i}| \tag{3}$$

for the dimension $i = 1, 2, \ldots, n$. Here n is the total number of dimensions for $\mathbf{x}$.

3.2 RELATIONSHIP BETWEEN LS AND LOSS SURFACE SMOOTHNESS

Let $\tilde{\mathbf{p}}$ denote the prediction confidence of $f_{\tilde{\theta}}$, $\tilde{\mathbf{z}}$ denote its logit representation (before softmax activation). Likewise, $\mathbf{p}$ denotes the prediction confidence of f_θ. Based on Eq. 2, s is the label smoothing degree, t is the index of the ground-truth label and K is the number of classes.

Lemma 1. *Optimizing CE loss $L(\cdot)$ with the soft label $\mathbf{y}_s$ is a logit regularization problem:*

$$\min_{\tilde{\theta}} L(\mathbf{x}, \mathbf{y}_s, \tilde{\theta}) = \min_{\tilde{\theta}} L(\mathbf{x}, \mathbf{y}, \tilde{\theta}) + s \cdot R_K(\tilde{\theta}) \tag{4}$$

where the regularization term is $R_K(\tilde{\theta}) = \frac{1}{K}\sum_{i \neq t}(\tilde{z}_i - \tilde{z}_t)$.

Lemma 1 shows that LS acts as a regularizer, constraining the margin between any logit $\tilde{z}_i$ $(i \neq t)$ and $\tilde{z}_t$. As such, a strong label smoothing degree s refrains the bound of $\{\nabla_{\mathbf{x}}\tilde{z}_i, i \in [K]\}$ from being enlarged. We further corroborate it by empirically estimating local Lipschitzness of our classifier $f_{\tilde{\theta}}$ based on (Yang et al., 2020). The detailed results and analyses are given in Appendix B.

Remark: Lemma 1 also reveals that such a regularization term penalizes model's over-fitting on ground-truth label, i.e., lowering confidence on true label. As such, under the same perturbation δ, $\mathbf{x} + \delta$ is easier to cross the boundary of $f_{\tilde{\theta}}$ than that of f_θ. That is to say, LS may harm the accuracy of samples near the decision boundary. This guides the design of adaptive LS (see Eq. 6 and 7) to alleviate such cost.

Theorem 1. *Label smoothing improves loss surface smoothness by narrowing bounds of loss gradients w.r.t inputs, as stated in Eq. 3.*

According to the chain rule, the gradients of $L(\cdot)$ w.r.t $\mathbf{x}$ can be derived as: $\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}} = \sum_i (\tilde{p}_i - y_i) \cdot \nabla_{\mathbf{x}}\tilde{z}_i = (\tilde{\mathbf{p}} - \mathbf{y}) \cdot \nabla_{\mathbf{x}}\tilde{\mathbf{z}}$, where $\sum_i \tilde{p}_i = 1$, $y_t = 1$. We analyze the relationship between $\nabla_{\mathbf{x}}\tilde{\mathbf{z}}$ and $\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}}$ and prove that if the bound of $\nabla_{\mathbf{x}}\tilde{\mathbf{z}}$ shrinks as stated in Lemma 1, we can obtain a tighter bound of $\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}, \tilde{\theta})$ than that of $\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}, \theta)$, and as a result, achieve a smoother loss surface. The proof is given in Appendix A.

4 METHODOLOGY: SURFACE SMOOTHING ADVERSARIAL TRAINING

Based on the above analysis, label smoothing is used for AT to obtain a smooth loss surface at the cost of lower confidence on the true label. As a result, we design a novel adaptive LS technology for adversarial training, i.e., Surface Smoothing Adversarial Training (SSAT), as detailed in Alg. 1. In this section, we show the learning objective of SSAT as well as its algorithmic implementation.

4.1 OBJECTIVE DESIGN

Let $\tilde{\mathbf{y}}(\delta^*)$ be the adaptive LS function on the hard label $\mathbf{y}$. We modify the framework of vanilla AT and propose our objective of SSAT as follows:

$$\begin{aligned} &\min_{\theta} E_{(\mathbf{x},\mathbf{y})} L\left(f_\theta(\mathbf{x}+\delta^*), \tilde{\mathbf{y}}(\delta^*)\right) \\ &\delta^* = \arg\max_{\|\delta\|_p \leq \epsilon} L\left(f_\theta(\mathbf{x}+\delta), \mathbf{y}\right) \end{aligned} \tag{5}$$

where $\tilde{\mathbf{y}}(\delta^*)$ depends on the current perturbation δ^*. In general, the constraint ensures that $\tilde{\mathbf{y}}(\delta^*)$ applies weaker LS with increasing 'size' of δ. We defer the definition of $\tilde{\mathbf{y}}(\delta)$ to Eq. 6.

Different from vanilla AT using the same loss for both adversarial example generation (inner maximization) and weight updating (outer minimization), our training scheme leverages the loss based on LS in the outer minimization but keeps the hard label based loss in the adversarial generation step. The reason is that the adversary does not know the mechanism of how to smooth labels in training. In another word, the potential strong attacks would be performed on hard labels on the attacker's side. As a result, we design the same data generation process on the defender's side as the potential attacks.

4.2 IMPLEMENTATIONS OF SSAT

The key idea of SSAT is to adopt different LS strategies for the adversarial examples generated in each iteration. This is based on the analysis of Lemma 1 and Theorem 1 on the premise of correct classification. We first divide adversarial examples into two **subsets**: inside the decision boundary $\mathcal{B}_I$ and outside: $\mathcal{B}_O$. The set of adversarial examples in $\mathcal{B}_I$ is defined as:

$$\mathcal{B}_I(\mathbf{x}) = \{\delta | f_{\boldsymbol{\theta}}(\mathbf{x}+\delta) = \mathbf{y}, \| \delta \|_p \leq \epsilon\},$$

which represents the predicted label of the adversarial example is the same as its ground truth label. $\mathcal{B}_O(\mathbf{x})$ is the complement of $\mathcal{B}_I(\mathbf{x})$. The strategies applied on these two subsets are as follows:

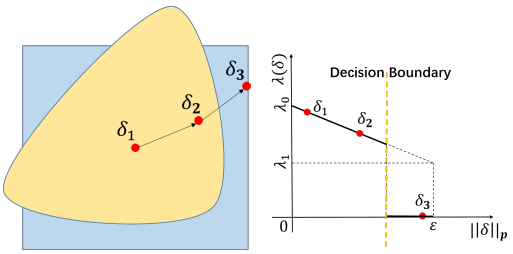
$$\tilde{\mathbf{y}}(\delta) = \begin{cases} \mathbf{y} & \delta \in \mathcal{B}_O(\mathbf{x}) \\ \mathbf{y} - \lambda(\delta) \times (\mathbf{y} - \frac{1}{K}) & \delta \in \mathcal{B}_I(\mathbf{x}) \end{cases} \tag{6}$$

where hard labels are used to those outside the decision boundary, and smooth labels, which are adaptive to the current perturbations with $\lambda(\delta)$, are applied to those inside the decision boundary. In detail, for adversarial samples near the original data point, strong LS is used to improve the smoothness of the loss surface. Meanwhile, we adopt weak LS for adversarial samples near the decision boundary. Moreover, $\lambda(\delta)$ is a non-decreasing function w.r.t the distance of adversarial examples towards the decision boundary. It is not trivial to compute the distance of an example to its decision boundary. For simplicity, we calculate the size of δ to measure such distance. Our intuition is that when δ is gradually far away from the original point with more iterations, i.e., its size gets larger, the input $\mathbf{x}$ perturbed by δ is more easily misclassified, which means its distance towards the decision boundary becomes smaller. Here we propose a heuristic function for $\delta \in \mathcal{B}_I(\mathbf{x})$ as:

$$\lambda(\delta) = (\lambda_1 - \lambda_0) \times \frac{\| \delta \|_p}{\epsilon} + \lambda_0 \tag{7}$$

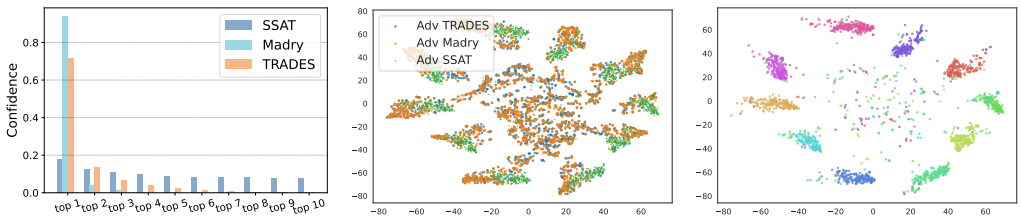
Algorithm 1 Surface Smoothing Adversarial Training (SSAT).

Require: Training samples $\boldsymbol{X}$, perturbation bound ϵ, learning rate τ, hop steps m, clip ratio γ, two smoothing levels α_0 and α_ϵ, training epoch # N;
for all epoch$= 1, \ldots, N/m$ **do**
  for all minibatch $B \subset \boldsymbol{X}$ **do**
    $\delta \sim$ Uniform$(-\gamma * \epsilon, \gamma * \epsilon)$;
    for all $step = 1, \ldots, m$ **do**
      * Outer Minimization to update θ:
        Obtain adaptive label $\tilde{\boldsymbol{y}}$ by Eq. 6 for B;
        $g_\theta \leftarrow E_{(x,y)\in B} \nabla_\theta L(f_\theta(\mathbf{x}+\delta), \tilde{\boldsymbol{y}}(\delta))$;
        $\theta \leftarrow \theta - \tau g_\theta$;
      * Inner Maximization to update δ:
        $g_{adv} \leftarrow E_{(x,y)\in B} \nabla_{\mathbf{x}} L(f_\theta(\mathbf{x}+\delta), \mathbf{y})$;
        $\delta \leftarrow Proj\Big(\delta + \beta \cdot sign(g_{adv})\Big)$;
    end for
  end for
end for



(a) Adversarial Perturb (b) Adaptive Degree

Figure 1: With the iterative generation of adversarial perturbations, the degree of LS decreases with a piecewise linear function.



(a) Confidence Difference (b) Transfer and White-box Attack (c) White-box Attack

Figure 2: **(a)** Prediction confidence difference of SSAT. Compared with other AT-based methods(e.g. Madry and TRADES), SSAT has a much lower mean confidence on the Top-1 category but much higher mean confidence on Top-2 to Top-10 on CIFAR10. For visualization simplicity, the prediction confidence of all images is sorted in its own decreasing order and averaged together. **(b)** t-SNE results of PGD-based transfer and white-box attacks on SSAT. Yellow and blue dots mean transfer attacks generated by other AT models and the green star represents the white-box attacks based on SSAT. For each kind of attack, 2000 data points are randomly sampled over the whole dataset CIFAR-10. **(c)** t-SNE results of white-box attacks on SSAT with the same data points as **(b)**. Adversarial data points with different labels are colored differently for clarity.

where $\lambda_0, \lambda_1 \in [0,\ 1]$ satisfying $\lambda_0 \geq \lambda_1$. Eq. 7 suggests that for a correctly predicted clean sample, when it is added with a perturbation generated by adversarial attack, the perturbed sample will be closer to the decision boundary even the prediction is correct. As a result, a stronger LS can be applied on the clean sample to achieve better smoothness, while a weaker one can be applied on the perturbed sample to alleviate the side-effect of label smoothing on model confidence as stated in Lemma 1. As shown in Fig. 1(b), the adaptive smoothing degree is a piecewise linear function to guide the model to obtain a smoother loss surface on the premise of correct classification. Then, we train the model with the adaptive smoothing label, which can be specified as

$$\min_{\theta} E_{(\mathbf{x},\mathbf{y})} L\left(f_\theta(\mathbf{x}+\delta), \tilde{\mathbf{y}}(\delta)\right) \tag{8}$$

where δ is the generated adversarial perturbation with the framework of Free AT (Shafahi et al., 2019b). Besides, the above method with LS focuses more on the smoothness of CE loss surface, which may ignore the attack of target adversarial example. So we add a regularization term of a target attack loss to increase the smoothness of the target loss surface. We discuss it in detail in Appendix C.

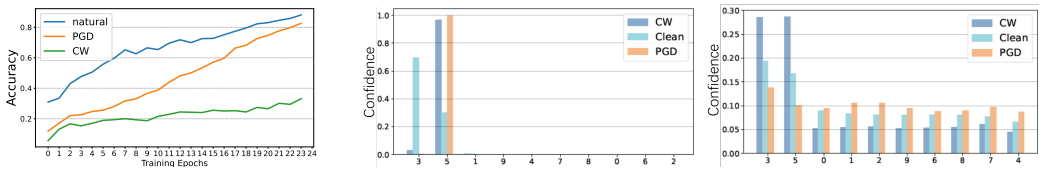
Why Free AT? In maximization of Eq. 5, we follow free AT (Shafahi et al., 2019b) instead of standard AT (Madry et al., 2018) to optimize δ, where we take all of its intermediate examples from each iteration as inputs and assign adaptive LS values to them for the outer minimization. As discussed in Lemma 1, LS can help to improve the surface smoothness but sacrifice model confidence on the true label. If we apply strong LS to adversarial examples generated with many iterations, since those adversarial examples are closer (or even across) the decision boundary, LS may harm clean accuracy. Thus, we apply strong LS to the intermediate examples at the beginning of perturbation optimization and gradually apply weaker LS to them with more iterations to achieve the goal of smoothing loss surface on the premise of correct classification. As shown in Fig. 1(a), in one PGD-like attack, the optimization trajectory of those intermediate examples, $\delta_1, \delta_2, \ldots, \delta_m$, is towards the decision boundary, i.e., with more iteration steps, the attack becomes stronger where weaker LS is then applied (see Fig. 1(b)). In other words, we focus more on the intrinsic robustness of different adversarial examples. For the adversarial example which is located far from the decision boundary, we adopt stronger LS while we use weaker LS or hard label for adversarial examples near or across the decision boundary to avoid misclassification, as one of the key differences compared with other methods e.g. CAT (Cheng et al., 2020). A detailed comparison of existing works is given in Appendix D, which hopefully sheds more insights on the novelty and motivation of this work.

5 Modeling Cooperation for SSAT

As discussed above, SSAT can improve robust accuracy with a smooth CE loss surface. In this section, we show that there exists great differences between SSAT and other AT-based models. This allows us to better understand SSAT and gives us motivation for devising a cooperative model.

5.1 Difference between SSAT and AT-based models

The difference in prediction confidence. As discussed in Sec. 4.2, SSAT requires a strong LS near the clean images, which makes the prediction confidence of SSAT distribute more uniformly on every



(a) Accuracy (b) Madry (c) SSAT

Figure 4: Accuracy with training epoch(Fig. 4(a)) and prediction confidence of clean and adversarial examples based on Madry and SSAT(Fig. 4(b) and Fig. 4(c)). The example in Fig. 4(b) and Fig. 4(c) is randomly selected from CIFAR10 and its adversarial examples are generated by PGD and CW attacks.

label than that of AT-based methods. Fig. 2(a) shows the mean prediction confidence of different models on the whole data of CIFAR10. As Lemma 1 stated, compared with Madry and TRADES, SSAT has a lower confidence in the top-1 category, which is sacrificed to improve the smoothness of the loss surface.

The difference on transfer and white-box attacks. The recent study (Müller et al., 2019) empirically shows that LS enforces the examples of the same class group in a tighter cluster, separating equidistantly from all the other classes' examples compared to those without LS. The distinction in embedding space drives us to study different responses of SSAT to various adversarial attacks. As shown in Fig. 2(b), transfer attacks based on Madry and TRADES exhibit different distribution compared with white-box attacks based on SSAT. Many of their adversarial examples(yellow and blue dots) are around the center, separating from the surrounding clusters. Even around one cluster, their data points still gather differently from those of SSAT. For better comparison, we visualize data points of SSAT in Fig. 2(c). Fig. 2(c) also reveals good robustness of SSAT: adversarial examples with different labels are still grouped into different clusters, increasing the difficulty of adversarial attacks. For a fair comparison, we apply transfer and white-box attacks on Madry and visualize the t-SNE results. The figures and analyses are given in Appendix E.

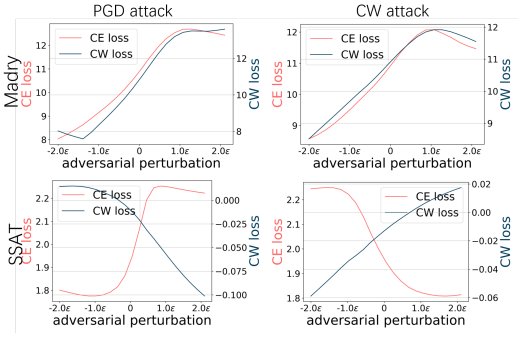


Figure 3: The different loss trends for Madry and SSAT by varying the perturbation of PGD and CW attacks. Each row represents one defense model, and each column means one type of the attack. Each figure represents one attack applied on one defense model with different perturbations. The perturbation δ is varied within the range of $(-2.0\epsilon, 2.0\epsilon)$. Note that negative value means the reversed direction.

The difference on the targeted/untargeted loss trends. Most AT-based methods have similar trends for CE and CW losses. CW loss is given as follows (Carlini & Wagner, 2017):

$$L_{CW} = \max\{[\max_{i \neq t} z_i - z_t], -k\} \tag{9}$$

where k is a positive hyperparameter, which controls how confident the adversarial example is misclassified. In Fig. 3, we find that SSAT exhibits different responses to adversarial attacks in terms of CW and CE losses compared with Madry. For example, by varying perturbations from PGD attack, CE loss increases but CW loss decreases for SSAT. For CW attack, CE and CW loss of SSAT also shows the opposite direction of change. It reveals that SSAT and Madry have completely different defense mechanisms for targeted and untargeted adversarial examples.

5.2 Cooperation between SSAT and AT models

The three aspects of difference above reveals that SSAT has inherently different defense mechanisms from other AT methods. Especially, in terms of different attack types, the distinguishing difference between SSAT and other AT models motivates us to propose a cooperative defense strategy to work together against both targeted and untargeted attacks, achieving better robustness.

The idea is that given two models(i.e., SSAT and another AT model, a.k.a, defense partner) and an input, the cooperative model can judge which model is more convincing when the two models make different predictions. A direct way is judging by the prediction confidence, in which the predicted

label with higher confidence of the two models is chosen. However, it does not work well as shown in Table 11 in Appendix H.2. By making use of the distinguishing features of different attack types, we design a detector to decide the attack type and propose a novel cooperative strategy: different adversarial examples are assigned to SSAT and the partner model respectively, such that one of them defends successfully against those examples with a higher confidence than the other model.

Targeted attack detection. As shown in Fig. 4(a), we find the difficulty for SSAT to defend against the white-box targeted attacks. Given the analysis of Fig. 3, we assume that targeted and untargeted attack can be detected based on SSAT's response to them. We further analyze difference between targeted and untargeted attacks in terms of the prediction confidence as shown in Fig. 4(b) and Fig. 4(c) and observe that for SSAT, the targeted attack works in a different way from the untargeted attack. Specifically, under CW attack, the confidences of top-1 and top-2 label grow together, driving the whole distribution more steep while under PGD attack, the confidence distribution tends to be more uniform. This is a distinct characteristic of SSAT by comparing with the other AT methods. In order to quantify this characteristic, we modify the Difference of Logit Ratio (DLR) (Croce & Hein, 2020a) to distinguish the two attack types by:

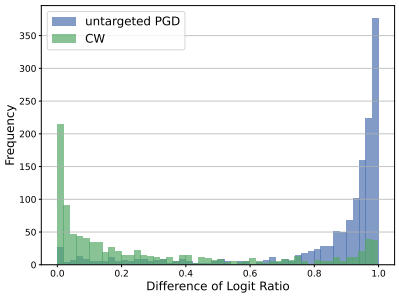


Figure 5: Frequency of DLR defined in Eq. 10 which is used to distinguish the targeted or untargeted attack.

$$DLR(\mathbf{x}) = \frac{z_{\pi_1} - z_{\pi_2}}{z_{\pi_2} - z_{\pi_3}} \tag{10}$$

where π_i is the rank i-th component in the descending order of $\mathbf{z}$ w.r.t $\mathbf{x}$. As shown in Fig. 5, most DLR values from the targeted attack are close to 0 and overall, the DLR of an untargeted attack is greater than that of a targeted attack. Thus, we can distinguish the targeted and untargeted attacks with a threshold τ. That is to say, when $DLR(\mathbf{x}) \leq \tau$, $\mathbf{x}$ belongs to targeted attacks. Besides, we propose a simple yet effective voting strategy combined with several similar threshold-based methods for better detection (see Appendix G.2).

Cooperative defense Given the SSAT model f_{SSAT} and the defense partner f_{base}, our cooperative defense model $f_{Co-SSAT}$ can be formulated as

$$f_{Co-SSAT}(\mathbf{x}) = d \cdot f_{base}(\mathbf{x}) + (1 - d) \cdot f_{SSAT}(\mathbf{x}) \tag{11}$$

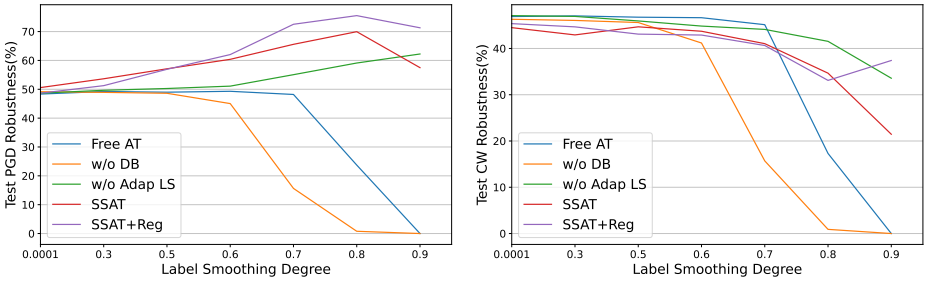
where $d = \mathbf{1}_{DLR(\mathbf{x}) \leq \tau}$ is the characteristic function for detection. After making a distinction between targeted and untargeted attacks, we cooperate SSAT with other AT methods such as Madry (Madry et al., 2018). When SSAT believes an adversarial example is from an untargeted attack, we choose the SSAT's prediction directly. Otherwise, Madry is used for prediction. For clean images, we make predictions in the same way. Note that the lower bound of clean accuracy of Co-SSAT depends on the lower bound of the two models in cooperation. Pseudocode is given in Alg. 2. Details for how to determine the threshold are given in Appendix G.1.

Algorithm 2 Cooperative Surface Smoothing Adversarial Training (Co-SSAT).

Input: A set of examples $\mathbf{X}$, well-trained model f_{SSAT}, another well-trained robust model as defense partner f_{base}, threshold τ;
Output: The prediction results S
Initialize the NULL set S;
for all $\mathbf{x} \in \mathbf{X}$ **do**
  Obtain the logit $\mathbf{z}$ based on model f_{SSAT} w.r.t $\mathbf{x}$;
  if $DLR(\mathbf{x}) \leq \tau$ **then**
    Predict $\mathbf{x}$ with f_{base} as y and append it into S;
  else
    Predict $\mathbf{x}$ with f_{SSAT} as y and append it into S;
  end if
end for

6 Experiments and Discussion

Dataset and model structure. Experiments are performed on CIFAR10 (Krizhevsky et al., 2009). We use (PreAct)ResNet-18 (He et al., 2016) and Wide ResNet. For Wide ResNet, we follow the same architecture with (Shafahi et al., 2019b; Zhang et al., 2019). For reproducibility, we use the checkpoints of those AT models collected by RobustBench (Croce et al., 2020). We perform attacks both inside and beyond the l_∞ norm constrained ball, such as l_2 norm attack. Experiments run on



(a) PGD robust accuracy (b) CW robust accuracy

Figure 6: Robust accuracy under CW and PGD attacks for ablation study. SSAT is compared with SSAT model without considering subset dividend(w/o DB), without adaptive label(w/o Adap LS), without both DB and adaptive LS(Free AT), and with target regularization by varying the label smoothing degree.

Table 1: White box robust accuracy (%) for seen and unseen attacks under WideResNet-34-10. All adversarial examples are generated using the cross-entropy (CE) loss.

defense \ attack	clean	seen ($\epsilon = 8/255$)			unseen ($\epsilon = 12/255$)		unseen ($\epsilon = 16/255$)		ℓ_2 ($\epsilon = 2$)	$\ell_1(\epsilon = 24)$
		FGSM	PGD10	PGD20	FGSM	PGD20	FGSM	PGD20	PGD20	PGD20
Madry (2018)	87.3	67.68	64.99	56.34	64.03	42.74	**61.00**	30.51	15.84	63.68
Free AT (2019b)	81.35	54.87	60.61	48.09	41.46	28.38	31.82	14.72	05.14	68.08
MMA (2020)	84.36	62.05	61.39	54.72	57.51	46.29	54.48	39.73	32.09	67.11
TRADES (2019)	84.22	61.26	65.23	56.92	50.17	39.53	41.64	25.51	7.18	62.95
MART (2020)	84.17	68.89	72.07	64.43	58.99	49.43	49.99	34.85	13.31	72.01
SSAT (ours)	**88.5**	**77.28**	**85.31**	**84.2**	**66.91**	**81.19**	58.64	**76.07**	**75.22**	**86.14**

Intel(R) Xeon(R) CPU E5-2678 v3 CPUs (2.50GHz) and 8 GTX 2080 TI GPUs. More training details of SSAT are given in Appendix J.

Measuring adversarial robustness via white-box and black-box attacks. We measure robustness under PGD attacks (Madry et al., 2018). Specifically, each model is evaluated by white-box PGD attacks and transfer-based black-box attacks.

Measuring robustness via unseen attacks. Typically robustness is only reasonable in the given threat model, i.e., the ℓ_p norm bound and the gained robustness does not extrapolate to unseen attacks such as larger ℓ_∞ norm attacks or ℓ_1 norm attacks. However, since we obtain a relatively smoother loss surface, we will test it on the unseen perturbation tolerances ϵ and unseen attacks with different norms.

6.1 Experimental Results of SSAT

White box attack results. For CIFAR10, we evaluate all the models under different perturbation tolerances and step numbers of PGD attacks. Specifically, we use untargeted PGD_n (n-step PGD with 5 random-starts) and targeted attack CW under l_∞ to measure robustness. Note that Free AT is the baseline of our model, which shares the same training hyperparameters with SSAT.

We show that our model performs well for both seen and unseen attacks. In Table 1, we train the model with $\epsilon = 8/255$ and test it with $\epsilon = 12/255$ and $16/255$. For larger perturbation tolerances, our model achieves higher robust accuracy than other models. Besides, SSAT outperforms mostly with the highest clean accuracy under FGSM attack, PGD_{10} and PGD_{20}. For other norm bounds such as ℓ_1 and ℓ_2, SSAT still achieves the highest robustness, showing its generalization ability. Results with much stronger attacks e.g. PGD_{100} are given in Appendix H.1.

Table 2: Robust accuracy (%) of Co-SSAT.

models \ attack	white-box attack		half-white-box attack	
	PGD-20	CW-20	PGD-20	CW-20
SSAT	84.2	32.16	84.2	32.16
Madry (2018)	56.34	47.81	56.34	47.81
Co-SSAT (Madry)	**73.28**	**48.13**	**87.77**	**78.2**
TRADES (2019)	56.71	**54.13**	56.71	56.58
Co-SSAT (TRADES)	**71.37**	53.55	**85.82**	**78.95**
MART (2020)	62.91	**59.15**	62.91	59.15
Co-SSAT (MART)	**73.79**	58.71	**87.66**	**83.09**

Black box attack results. Following the criterion of evaluating transfer attacks by (Athalye et al., 2018), $10,000$ adversarial examples are generated from standard training model ResNet-50 to evaluate the robustness of the targeted model. Table 3 shows that SSAT overwhelms other baselines on both targeted and untargeted transfer attacks, except for FGSM attack. Note that for other AT methods, FGSM attack is still stronger than iterative gradient-based attacks, which shows FGSM attack has better transferability. For the performance gap under FGSM attacks, this is

Table 3: Black box robust accuracy (%) for seen and unseen attacks under WideResnet-34-10. Adversarial examples with FGSM and PGD20 attacks are generated by CE loss while CW10 is generated by CW loss.

defense \ attack	seen attack ($\epsilon = 8/255$)			unseen attack ($\epsilon = 12/255$)			unseen attack ($\epsilon = 16/255$)			ℓ_2 ($\epsilon = 2$)	ℓ_1 $\epsilon = 24$
	FGSM	CW20	PGD20	FGSM	CW20	PGD20	FGSM	CW20	PGD20	PGD20	PGD20
Madry (2018)	82.77	83.84	83.32	81.64	83.63	83.09	80.50	83.64	82.90	82.46	83.91
MMA (2020)	81.74	83.34	82.89	79.73	82.59	81.97	77.14	81.71	80.77	81.10	83.58
TRADES (2019)	82.93	84.20	83.69	81.53	83.92	83.31	79.80	83.52	82.74	82.59	84.51
MART (2020)	**85.37**	86.62	86.17	**83.52**	86.16	85.62	**81.03**	85.78	84.68	84.82	86.86
Free AT (2019b)	84.93	86.81	86.27	83.24	86.64	85.81	80.84	86.22	84.72	84.82	86.88
SSAT (ours)	84.80	**87.14**	**86.28**	79.02	**87.02**	**85.61**	74.79	**86.63**	**85.11**	**85.15**	**87.34**
Co-SSAT(Madry)	84.32	86.15	85.68	82.42	86.08	85.39	80.33	86.05	84.87	84.52	86.13

perhaps because LS sacrifices the confidence of our model on the true label, and the gap between true label and other labels is naturally smaller than other AT methods. As such, the single FGSM attack especially with larger ℓ_p bound ($\epsilon = \frac{12}{255}, \frac{16}{255}$), would more easily push the adversarial example across the decision boundary while such weakness can be remedied by Co-SSAT as shown in Table 3.

Ablation study. Fig. 6 shows the necessity of every component of SSAT. Recall that the adaptive LS range of SSAT is $[\lambda_1, \lambda_0]$, (see Fig. 1(b)). In Fig. 6, we fix $\lambda_0 = 1$ and vary λ_1 on x-axis with stronger LS. In Fig. 6(a), with a larger smoothing degree, the robustness of SSAT with regularization performs the best among others. Without considering the decision boundary, SSAT quickly decreases to 0. In Fig. 6(b), robustness of all methods decays by increasing the smoothing degree, but SSAT with regularization decreases relatively slower. Thus, we adopt SSAT with regularization as the final model. We also analyze the effect of varying λ_0 with $\lambda_1 = 0$ and find adjusting the magnitude of λ_1 is more effective in building local smoothness of SSAT with $\lambda_0 = 0$. See more results in Appendix I.

6.2 Experimental Results of Co-SSAT.

White Box results. Co-SSAT contains three parts: the detector, SSAT and the partner. We design a white-box attack on both detector and the two classifiers to evaluate its robustness by Eq. 11. The attacker has the knowledge of decisions made by the detector, such that the attacker can craft adversarial examples based on the model selected by the detector. In Table 2, under white-box attacks, Co-SSAT still achieves better performance under PGD attacks and has similar performance against CW attacks compared with other baselines.

Half-white Box results. White-box attacks are usually used to estimate the worst-case robustness of networks, which is less practical. We propose a new attack protocol by fusing both white and black attacks, which is more of practical use in real world. In particular, the attacker only has access to SSAT without knowledge of the detector as well as the partner. Table 2 shows that under the half-white box setting, Co-SSAT further boosts robustness of both of the two models under two attack types.

Black Box results. Black-box results are shown in Table 3. Exactly Co-SSAT does not significantly improve robustness against black-box attacks except the FGSM based attack, which confirms that our detector fully exploits the advantages of the two models in cooperation to achieve a better robustness.

The results of autoattack. We evaluate our model with stronger attacks: autoattack (AA) (Croce & Hein, 2020b) to show the superiority of Co-SSAT. The results are given in Appendix H.2. In case when our detector is known, Co-SSAT with Madry achieves better robustness than Madry alone. While the robustness of Co-SSAT degrades a little when combined with TRADES and MART. In our analysis, this is because the attacker knows the decisions of the detector and then perturb inputs to fool Co-SSAT when our detector makes a wrong decision. Moreover, if the detector is inaccessible by the attacker, Co-SSAT consistently boosts the robustness of our two models, especially obtaining an impressive accuracy over about 19% on Madry.

7 Conclusion

In this paper, we have taken a deep dive into the mechanism and effect of LS on AT, whose underlying theory and empirical study are still in their relatively early stage. We give both experimental and theoretical justification to our proposed sample-aware adaptive LS method, i.e., surface smoothing adversarial training (SSAT). SSAT builds a smoother loss surface via assigning stronger (or weaker) LS degree to adversarial data which is inherently farther away from (or closer to) the decision boundary, which can be readily reused either as a plugin or standalone model for AT.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

B. Biggio, I. Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. *ArXiv*, abs/1708.06131, 2013.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

Shang-Tse Chen and J. Martin. Physical adversarial attack on object detectors (extended abstract). 2018.

Minhao Cheng, Qi Lei, Pin-Yu Chen, I. Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *ArXiv*, abs/2002.06789, 2020.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020a.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020b.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv:2010.09670*, 2020.

Gavin Weiguang Ding, Yash Sharma, Kry Yik-Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

I. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

A. Madry, Aleksandar Makelov, Ludwig Schmidt, D. Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *NeurIPS*, 2019.

Tianyu Pang, Xian Yang, Y. Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *ICLR*, 2021.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and P. Kohli. Adversarial robustness through local linearization. In *NeurIPS*, 2019.

Pouya Samangouei, Maya Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ArXiv*, abs/1805.06605, 2018.

A. Shafahi, Amin Ghiasi, F. Huang, and T. Goldstein. Label smoothing and logit squeezing: A replacement for adversarial training? *ArXiv*, abs/1910.11585, 2019a.

A. Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, L. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! In *NeurIPS*, 2019b.

Qing Song, Y. Wu, and L. Yang. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia Tools and Applications*, 80:855–875, 2021.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *ICML*, 2020.

Christian Szegedy, W. Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2014.

Jianyu Wang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6628–6637, 2019.

Yisen Wang, Difan Zou, Jinfeng Yi, J. Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Eric Wong, Leslie Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Weilin Xu, David Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *ArXiv*, abs/1704.01155, 2018.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In *NeurIPS*, 2020.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, abs/1901.08573, 2019.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

APPENDIX

A PROOF ABOUT THEOREM 1

For simplicity, we write the Lemma 1 and Theorem 1 here first.

Lemma 1. *Optimizing CE loss $L(\cdot)$ with the soft label $\mathbf{y}_s$ is a logit regularization problem:*

$$\min_{\tilde{\theta}} L(\mathbf{x}, \mathbf{y}_s, \tilde{\theta}) = \min_{\tilde{\theta}} L(\mathbf{x}, \mathbf{y}, \tilde{\theta}) + s \cdot R_K(\tilde{\theta}) \tag{12}$$

where the regularization term is $R_K(\tilde{\theta}) = \frac{1}{K}\sum_{i \neq t}(\tilde{z}_i - \tilde{z}_t)$.

Theorem 1. *Label smoothing improves loss surface smoothness by narrowing bounds of loss gradients w.r.t inputs, as stated in Eq. 3.*

Given a K-class problem, two robust classifiers f_θ and $f_{\tilde{\theta}}$ trained with the cross-entropy loss $L(\cdot)$ by the hard label and the soft label with a degree of λ, let $\mathbf{p}$ denote the prediction probability of f_θ w.r.t $\mathbf{x}$, y^{gt} denote the ground-truth label and y^{att} denote the misclassified label. Likewise $\tilde{\mathbf{p}}$ represents the prediction probability of $f_{\tilde{\theta}}$ w.r.t $\mathbf{x}$. Assume that $\mathbf{p}^{gt} = \alpha$ and $\tilde{\mathbf{p}}^{gt} = \beta$.

Assumption 1. *For the robust classifier f_θ without label smoothing, based on the overconfidence property of neural networks (Guo et al., 2017), the probability condifence of f_θ after being attacked becomes $(\alpha, 0, \ldots, 1-\alpha, \ldots, 0)$, i.e., $\mathbf{p}^{gt} = \alpha$ and $\mathbf{p}^{att} = 1 - \alpha$ with $\alpha \approx 0$.*

Assumption 2. *For the robust classifier $f_{\tilde{\theta}}$ with label smoothing, based on Lemma 1 and statistical analysis on Table 6, the probability confidence of $f_{\tilde{\theta}}$ satisfies smooth constraint due to the logit regularization, i.e., $\tilde{\mathbf{p}} = (\frac{1-2\beta}{K-2}, \ldots, \beta, \frac{1-2\beta}{K-2}, \ldots, \frac{1-2\beta}{K-2}, \beta, \ldots, \frac{1-2\beta}{K-2})$, where $\tilde{\mathbf{p}}^{gt} \approx \tilde{\mathbf{p}}^{att} = \beta$ and $\tilde{\mathbf{p}}_{i, i \neq gt, att} = \frac{1-2\beta}{K-2}$.*

Proof. We'd like to prove that:

$$|\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}_i}| < |\frac{\partial L(\mathbf{x}, \mathbf{y}, \theta)}{\partial \mathbf{x}_i}|. \tag{13}$$

for the dimension $i = 1, 2, \ldots, n$. Here n is the dimension number of $\mathbf{x}$.

The gradients of $L(\cdot)$ w.r.t $\mathbf{x}$ can be derived as $\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}} = \sum_i (\tilde{p}_i - y_i) \cdot \nabla_{\mathbf{x}} \tilde{z}_i = (\tilde{\mathbf{p}} - \mathbf{y}) \cdot \nabla_{\mathbf{x}} \tilde{\mathbf{z}}$, where $\sum_i \tilde{p}_i = 1$, $y^{gt} = 1$. For any $i \in [n]$, to obtain the bound of $|\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}_i}|$, we have:

$$\begin{aligned} \nabla_{\mathbf{x}_i} \tilde{\mathbf{z}} &= \arg\max_{\mathbf{v}} (\tilde{\mathbf{p}} - \mathbf{y}) \cdot \mathbf{v} \\ &= c_i * (\tilde{\mathbf{p}} - \mathbf{y}) \end{aligned} \tag{14}$$

where c_i represents the positive bound constant w.r.t. $|\nabla_{\mathbf{x}_i} \tilde{\mathbf{z}}|$.

Let $\tilde{c}_i$ denote the bound w.r.t. $|\nabla_{\mathbf{x}_i} \tilde{\mathbf{z}}|$ with label smoothing while c_i represents that bound without label smoothing. Lemma 1 tells that label smoothing constrains the bound $\tilde{c}_i$ with a smaller magnitude, i.e., $\tilde{c}_i < c_i$. Based on Eq. 14, we have:

$$|\frac{\partial L(\mathbf{x}, \mathbf{y}, \tilde{\theta})}{\partial \mathbf{x}_i}| = c_i * ||\tilde{\mathbf{p}} - \mathbf{y}||_2^2 \tag{15}$$

From Eq. 15, we next prove that $\tilde{c}_i * ||\tilde{\mathbf{p}} - \mathbf{y}||_2^2 < c_i * ||\mathbf{p} - \mathbf{y}||_2^2$. Since we have $\tilde{c}_i < c_i$, we next prove

$$||\tilde{\mathbf{p}} - \mathbf{y}||_2 < ||\mathbf{p} - \mathbf{y}||_2 \tag{16}$$

From Assumption 1, we have:

$$\begin{aligned} ||\tilde{\mathbf{p}} - \mathbf{y}||_2 &= \sqrt{(\beta - 1)^2 + \beta^2 + \frac{(1-2\beta)^2}{(K-2)^2} * (K-2)} \\ &< (1 - 2\beta)\sqrt{\frac{K-1}{K-2}} \end{aligned} \tag{17}$$

From Assumption 2, we have:

$$\begin{aligned} ||\mathbf{p}-\mathbf{y}||_2 &= \sqrt{(\alpha-1)^2+(1-\alpha)^2} \\ &= \sqrt{2}(1-\alpha) \end{aligned} \tag{18}$$

To prove Eq. 16, we prove the upper bound of Eq. 17 is smaller than Eq.18:

$$\beta > \frac{1}{2}(1-\sqrt{\frac{2(K-2)}{K-1}}(1-\alpha)) \tag{19}$$

Assumption 1 tells that $\mathbf{p}^{att} = 1-\alpha$ is close to 1, so Eq. 19 naturally holds since we only need $\beta > 0$. Finally based on Eq. 19, we prove Eq. 13.

□

B EMPIRICAL STUDY ABOUT LEMMA 1

We start from the definition of local Lipschitz continuity of a classifier f:

Definition: Let $(\mathcal{X}, dist)$ be a metric space. A function $f{:}\mathcal{X} \to \mathbb{R}^C$ is L-locally Lipschitz at radius r if for each $i \in [C]$, we have $|f(x)_i - f(x')_i| \leq L * dist(x, x')$ for all x' with $dist(x, x') \leq r$.

As f is locally bounded by L so its gradient is locally bounded by L as well.

We demonstrate that **with stronger LS, the bound of $\nabla_{\mathbf{x}}\tilde{z}_i, i = 1, \ldots, n$ would shrink into a smaller range, indicating a smoother loss surface** through local Lipschitz continuity. In specific, we use the *empirical Lipschitz constant* proposed by Yang et al (Yang et al., 2020) to evaluate such bound.

$$\frac{1}{n}\sum_{i=1}^{n} \max_{x_i' \in \mathcal{B}_\infty(x_i,\epsilon)} \frac{\parallel f(x_i') - f(x_i) \parallel_1}{\parallel x_i' - x_i \parallel_\infty} \tag{20}$$

A lower value of the empirical Lipschitz constant implies a smoother classifier. We also calculate the lower bound of this constant by minimizing the above formula.

We perform two experiments to verify local smoothness brought by our adaptive LS. First, we compare SSAT with other baselines on this metric and SSAT has the highest degree of local smoothness among them as the table 4 below shows.

Table 4: Local Lipschitz Bound of baselines and SSAT on CIFAR-10 dataset

Defense	Local Lipschitz Lower Bound	Local Lipschitz Upper Bound
Madry	67.197	395.6494
MMA	67.4459	374.3057
TRADES	22.6532	134.7732
Free AT	40.4731	232.6561
SSAT(ours)	**14.5584**	**76.1627**

Furthermore, we dive deeply into LS itself. As table 5 shows, with varying λ_0 from 0.9 to 0, the smoothing degree gets lower and the corresponding local Lipschitz constant gets larger, which demonstrates that our proposed LS strategy indeed improves loss surface smoothness.

C A TARGET REGULARIZATION ON SSAT

Regularizing the optimization with targeted loss. We regularize the adversarial optimization with targeted loss $L\left(f_\theta(\mathbf{x}+\delta^*), \mathbf{y}'\right)$:

$$\min_\theta E_{(\mathbf{x},\mathbf{y})}(1+\beta) * L\left(f_\theta(\mathbf{x}+\delta^*), \tilde{\mathbf{y}}\right) - \beta * L\left(f_\theta(\mathbf{x}+\delta^*), \mathbf{y}'\right), \tag{21}$$

Table 5: Local Lipschitz Bound with varying LS degree on SSAT

λ_0	λ_1	Local Lipschitz Lower Bound	Local Lipschitz Upper Bound
0.9	0	**15.1433**	**80.4687**
0.8	0	15.8073	84.1924
0.7	0	16.3493	88.3815
0.6	0	17.2021	92.6672
0.5	0	17.8826	97.0716
0.3	0	20.4229	111.0398
0	0	36.5391	205.1484

where

$$\delta^* = \arg \max_{\|\delta\|_p \leq \epsilon} (1+\beta) * L\left(f_\theta(\mathbf{x}+\delta), \mathbf{y}\right) - \beta * L\left(f_\theta(\mathbf{x}+\delta), \mathbf{y}'\right). \tag{22}$$

$L(\cdot,\cdot)$ is the cross-entropy (CE) loss and $\mathbf{y}'$ denotes the targeted label.

Similar to the training proposed in Sec. 4.1, we generate adversarial examples with Eq. 22 and train the model with Eq. 21. From the perspective of label smoothing, in our original implementation, label weights are uniformly distributed among all labels except the true one. By contrast, this regularization term explicitly assigns a penalty weight on the targeted label and hence can be viewed as an adaptive ununiform allocation strategy of label weights. In implementation, the targeted label is adaptively selected as the second most convincing label except the true one. Experiment results (see the 'SSAT+Reg' line in Fig. 6) show that our regularization strategy exactly increases robustness of both targeted and untargeted attacks.

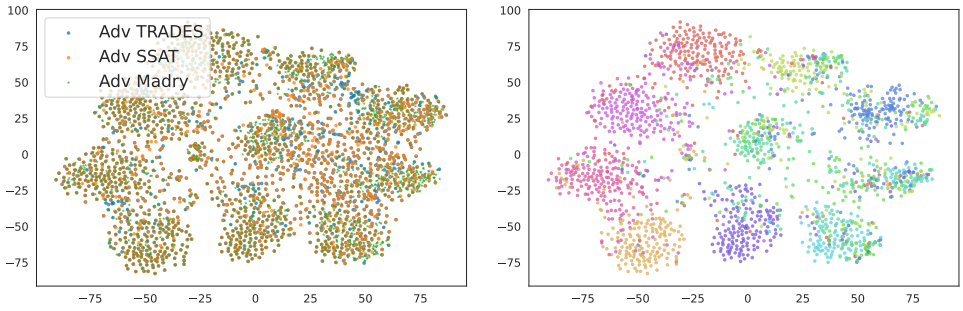
D Comparisons with related AT methods.

Adversarial training with label smoothing Some existing AT methods utilize label smoothing. Wang (2019) perturbs both the image and label during training without considering the geometric property of images. CAT (Cheng et al., 2020) proposes to adjust the perturbation level adaptively to better handle the trade-off between robustness and clean accuracy. It argues that for images near the decision boundary, hard label may harm generalization so it adopts a smaller LS degree with a larger perturbation bound. However, our motivation of adaptive LS is intrinsically different from CAT. We aim to build a smoother loss surface. As such, a larger LS degree is adopted near the clean images while for images near(or across) the decision boundary, we adopt a weaker soft(or hard) label to strengthen the discriminative power of our model: in our statement, LS would harm clean accuracy by lowering classifier's confidence especially for those images near the decision boundary. Further, in implementation, the LS of CAT is deterministic with the given perturbation bound while given a ℓ_p norm bound, our LS varies adaptively in a bi-level along the optimization trajectory.

Adversarial training with instance dependent reweighting Recent studies treat adversarial data differently based on the fact that every data point has different intrinsic robustness. For example, MMA (Ding et al., 2020) proposes to directly maximize the margins to generate instance-dependent perturbation bounds. MART (Wang et al., 2020) revisits the misclassified examples outside of the decision boundary to improve the robustness. GAIRAT (Zhang et al., 2021) explicitly assigns different weights on the objective loss of each adversarial data. Since all of them emphasize the decision boundary to distinguish the adversarial data, our SSAT can be naturally combined with any of them to further boost performance. For example, GAIRAT assigns adaptive weights on the original CE loss while SSAT can adaptively apply LS on the true label vector in the original CE loss: both GAIRAT and our SSAT define and make use of the geometric properties of adversarial data but the object on which they operate is different.

E t-SNE results of Madry

Different from Fig. 2, first Fig. 7(b) shows that the projections of data points with different labels are spread more broadly and different clusters of data points with different labels even intersect with each other, which means Madry exhibits less robustness under adversarial attacks. Fig. 7(a) shows that



(a) PGD robust accuracy (b) CW robust accuracy

Figure 7: **(a)** t-SNE results of PGD-based transfer and white-box attacks on Madry. Yellow and blue dots mean transfer attacks generated by other AT models and green star represents the white-box attacks based on Madry. For each kind of attack, 2000 data points are randomly sampled over the whole dataset CIFAR-10. **(b)** t-SNE results of white-box attacks on Madry with the same data points as **(b)**. Adversarial data points with different labels are colored differently for clarity.

Table 6: prediction confidence of SSAT and Madry over CIFAR-10.

Model	attack	top 1	top 2	top 3	top 4	top 5	top 6	top 7	top 8	top 9	top 10	KL Div
SSAT	clean	**0.1793**	**0.1237**	0.1066	0.0962	0.0896	0.0851	0.0826	0.0809	0.0792	0.0769	**0.0027**
	PGD	**0.1418**	**0.1202**	0.1104	0.1025	0.0957	0.0901	0.0865	0.0844	0.0839	0.0844	**0.0015**
	CW	**0.1452**	**0.1481**	0.1151	0.0993	0.0909	0.0846	0.0809	0.0788	0.0784	0.0788	**0.0030**
Madry	clean	**0.9373**	**0.0425**	0.0108	0.0044	0.0022	0.0012	0.0007	0.0004	0.0003	0.0002	**0.3471**
	PGD	**0.2554**	**0.3477**	0.1392	0.0768	0.0561	0.0399	0.0283	0.0218	0.0196	0.0152	**0.0555**
	CW	**0.1942**	**0.521**	0.1168	0.0521	0.0364	0.0259	0.0201	0.013	0.011	0.0094	**0.0876**

the distribution of the transfer attacks and white-box attacks on Madry is similar to each other with less distinction. Both of Fig. 7(a) and (b) further demonstrate that our SSAT has inherently different defense mechanisms from other AT methods.

F Further Analyses on the whole dataset from Sec. 5

We average the prediction confidence over the whole dataset CIFAR-10 and also calculate the KL divergence between confidence distribution and uniform distribution to show the effects of different types of adversarial attacks on the original confidence distribution. In Table 6, we first list prediction confidence w.r.t. clean input in descending order, predictions under PGD and CW attack are sorted in the same order for comparison. Observations on the whole dataset are consistent with Figure 4(c), specifically:

1. For clean inputs, the distribution of predicted labels of SSAT is much smoother than Madry's.
2. untargeted attacks,i.e. PGD attack, on SSAT behave differently from Madry's by lowering confidence on the top-1 label and increasing that on other labels, which makes distribution closer to uniform. This phenomenon reveals a significant characteristic of the attack.
3. Targeted attacks on SSAT can be evaded by a small margin of 0.003 overall. It tends to increase the confidence of the top-2 label and decrease the confidence of the rest. By contrast, after performing the untargeted attack, the confidence of top2 also shrinks by 0.003 to 0.1202. Such difference motivates us to propose cooperative defense to defend against targeted attack on Sec. 5.2.

Besides, we measure Kullback–Leibler Divergence (KL Div) between each prediction confidence and uniform distribution to uncover different characteristics resulted from targeted and untargeted attacks. For Madry's, both targeted and untargeted attacks result in smaller KL Div values while for SSAT, under PGD attack the value of KL Div is smaller than that with clean inputs, which verifies our observations above.

G IMPLEMENTATION DETAILS OF CO-SSAT FROM SEC. 5.2

G.1 REALIZATION OF OUR DETECTOR

The detector is treated as a two-class linear classifier and the threshold, i.e. the decision boundary is determined by referring to the ROC curve. We would like to choose a threshold that has both good recall rate and accuracy, i.e., to recognize as many as targeted adversarial examples as possible while reducing the number of misclassified untargeted adversarial examples since we expect our SSAT to defend against untargeted attacks.

To be specific, first, we perform untargeted and targeted attacks on the training set, where examples under untargeted attacks are labeled as negative with those under targeted attacks being labeled as positive. Then, we calculate the DLR of each sample and draw the ROC curve. Our goal is to select as many positive samples as possible and send them to the surrogate model for defense, so we expect to obtain a relatively high True Positive Rate (TPR), i.e. recall. We also would like to select most of negative samples, i.e. adversarial examples under untargeted attack, to our SSAT to defend against. Finally, by grid search, we find a threshold with a relatively high recall and accuracy. So it is model-dependent.

As for classification accuracy, the table 7 shows the trade-off between recall and accuracy. Results are based on adversaries generated by AutoAttack. We finally selected the threshold with which the detector achieves TPR of 0.8606 and Accuracy of 0.8819 as the criterion.

Table 7: Performance of our threshold based detector

TPR(Recall)	FPR	Precision	Accuracy
0.9822	0.6228	0.7973	0.8091
0.9282	0.2959	0.8867	0.8641
0.8606	**0.0651**	**0.9706**	**0.8819**
0.8037	0.0163	0.9919	0.8552
0.7544	0.0074	0.9961	0.8226
0.71	0.0044	0.9975	0.7917
0.6251	0.0015	0.9991	0.732
0.5878	0.0015	0.999	0.7053
0.4312	0	1	0.594

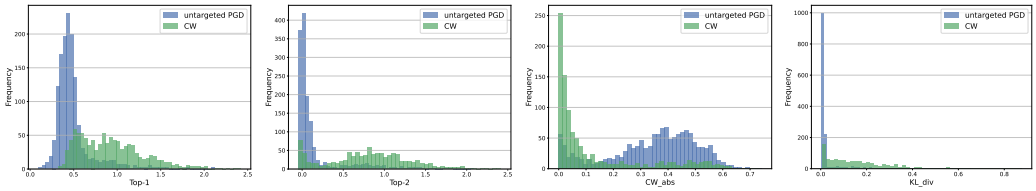
G.2 CONFIDENCE BOUNDED VOTING STRATEGY

Apart from Difference of Logit Ratio (DLR) introduced in Sec. 5.2, to gain a better estimation of whether an adversarial example is untargeted or targeted, we use a simple yet effective voting mechanism to make the final decision. Including DLR, we define five metrics in total: Top-1, Top-2, KL-Div, and CW-abs. For each sample, only when the number of votes exceeds half of the committee members (> 3 in our setting), we assert it as targeted attack.

1. Top-1(x) = z_{π_1}
2. Top-2(x) = z_{π_2}
3. KL-Div(x) = $KL(P(x), \vec{u})$
4. CW-abs(x) = $z_{\pi_1} - z_{\pi_2}$

where $P(\mathbf{x})$ is the probability distribution of $\mathbf{z}$ w.r.t $\mathbf{x}$, $\vec{u}$ is a unit vector scaled by $\frac{1}{K}$, and $KL(\cdot, \cdot)$ is the Kullback–Leibler divergence of the two distributions.

The intuition that we choose the metrics above is similar: the feedbacks of robust neural networks trained by SSAT over the two types of adversarial attacks: untargeted and targeted, as shown in Figure 4. We analyze such difference from two levels: label-level and distribution-level. Top-1 and Top-2 belong to the label-level, which vary with the attack direction. Furthermore, KL-Div, CW-abs, and DLR all belong to the distribution-level, which also shows the distinction of targeted and untargeted attacks.



(a) Top-1 (b) Top-2 (c) CW-abs (d) KL-Div

Figure 8: Frequency histogram of untargeted and targeted attack over different metrics.Note all the above methods can roughly distinguish targeted attacks and untargeteded attacks. We adopt voting strategy to improve the distiguishment.

Table 8: Robustness (%) with PGD-100 for white box seen and unseen attacks under WideResNet-34-10. Best in bold.

Defense	seen $\epsilon = 8/255$	unseen $\epsilon = 12/255$	unseen $\epsilon = 16/255$	L2 $\epsilon = 2$	L1 $\epsilon = 24$
Free AT	45.6	24.29	10.11	1.5	55.74
Madry	47.77	31.12	17.07	3.63	51.06
MMA	51.42	41.43	33.76	23.73	64.75
MART	62.8	44.31	26.98	4.48	53.72
TRADES	55.37	35.34	19.83	2.46	39.39
SSAT(ours)	**81.67**	**71.81**	**58.72**	**41.17**	**86.05**

Table 9: Robustness (%) with PGD-100 for black box seen and unseen attacks under WideResNet-34-10.

Defense	seen $\epsilon = 8/255$	unseen $\epsilon = 12/255$	unseen $\epsilon = 16/255$	L2 $\epsilon = 2$	L1 $\epsilon = 24$
Free AT	86.18	85.33	**84.52**	**84.63**	86.75
Madry	83.36	82.66	81.96	81.95	83.53
MMA	82.73	81.41	80.04	80.99	83.37
MART	86.01	85.28	83.87	84.32	86.62
TRADES	83.71	83.04	82.14	81.9	84.3
SSAT(ours)	**86.44**	**85.37**	84.01	84.46	**87.01**

As the Figure 8 shows, the difference of untargeted and targeted examples is distinct under each metric and we calculate the ROC score for each metric: 0.86 (DLR), 0.89 (Top-1), 0.85 (Top-2), 0.80 (CW-abs), and 0.89 (KL-Div), both of which confirm the effectiveness of the chosen metrics. So by ensembling the five metrics together, we obtain a good detector of telling the features of the attack.

H Evaluation Results under Stronger Attacks

H.1 Black-box and White-box PGD-100 Attack

The table 8 and 9 show results of the white and black box PGD-100 attack respectively. Detailed experiment settings are given in Sec. 6. Results boldfaced denote the best among all defense methods, which show that SSAT outperforms with the besy

H.2 Benchmarking the State-of-the-art Robustness against AutoAttack (AA)

AutoAttack (AA) (Croce & Hein, 2020a) is an ensemble of three white-box attacks (APGD-CE (Croce & Hein, 2020a), APGD-CLR (Croce & Hein, 2020a), and FAB) and one black-box attack (Square Attack).

We test our Co-SSAT against AA to gain a reliable and thorough estimation of robustness. For Co-SSAT, we apply AA to our robustly trained model M_{SSAT} and M_{base} and generate twice the size of adversarial samples that it has on one model. Then we use our detector to predict and dispatch those adversarial examples to one of our two models to make the final decision. Here for fair comparison with baseline methods, we report the two results with combined two baselines (see experiment details in Sec. 6). The results in Table 11 demonstrate that our method outperforms other baselines even when faced with the ensemble of strong attacks.

Table 10: Co-SSAT by AA under the setting of white-box and half-white-box attack.

attack / models	white-box attack AA(%)	half-white-box attack AA(%)
SSAT	23.67	23.67
Madry	43.36	43.36
Co-SSAT (Madry)	**46.65**	**65.74**
TRADES	53.36	53.36
Co-SSAT (TRADES)	51.37	**64.8**
MART	57.11	57.11
Co-SSAT (MART)	55.74	**69.77**

Table 11: The white-box robust accuracy against AA of a vanilla cooperative strategy based on baseline methods. The cooperative strategy works by choosing which prediction with higher confidence of any of the two models.

models	Roubst Accuracy(%) based on AA
Madry	43.36
TRADES	53.36
MART	57.11
Madry + TRADES	44.48
Madry + MART	45.23
TRADES + MART	56.95

I Ablation Study on hyper-parameters λ_0 and λ_1

The corresponding values of the SSAT curve in Fig. 6 are given in the table 12. With stronger LS, i.e. larger λ_1, clean accuracy and robust accuracy under targeted attack (i.e. CW-10) get lower while robust accuracy under untargeted attack gets higher, which shows a trade-off between robustness and accuracy. It is worth noting that very strong LS (i.e. $\lambda_1 = 0.9$) may harm training, degrading general performance on accuracy and robustness.

Furthermore, we did an ablation study on the effectiveness of λ_0 with $\lambda_1 = 1$. The table 13 shows that with smaller λ_0, we can obtain a little better clean accuracy while robustness under untargeted attack (i.e. PGD-10) degrades. For robustness under targeted attack (i.e. CW-10), the effect of λ_0 can be negligible.

In addition, the two tables above show that adjusting the magnitude of λ_1 is more effective on building local linearization of SSAT with a big value of λ_0 (i.e. λ_0 = 1.0 in our implementation).

J Implementation Details

We adopt the framework of free AT to build the local smoothness of loss surface. To achieve the effectiveness of our method, we just have several simple yet critical rules to follow such that we do not need to tune hyper-parameters.

1. **Clip after Initialization**: For each new minibatch, instead of directly reusing the perturbation from the previous one, we reset the perturbation by uniformly sampling from the ϵ ball and clip it to ensure that the adversary starts attack near the clean example. The clip ratio γ is set as a fixed value 0.3 for all experiments on Resnet-18 and WideResNet-34-10.

Table 12: Performance of SSAT by varying λ_1.

λ_0	λ_1	Clean	FGSM	PGD-10	CW-10
1.0	0.9	0.7251	0.6973	0.5749	0.2146
1.0	0.8	0.7837	0.7609	**0.6996**	0.3466
1.0	0.7	0.7976	0.7699	0.6558	0.4104
1.0	0.6	0.7975	0.7618	0.6038	0.4372
1.0	0.5	0.7999	0.757	0.571	**0.4468**
1.0	0.3	0.8236	**0.7749**	0.5364	0.4293
1.0	0.0	**0.8242**	0.767	0.5059	0.445

Table 13: Performance of SSAT by varying λ_0.

λ_0	λ_1	Clean	FGSM	PGD-10	CW-10
1.0	0	0.8242	**0.767**	**0.5059**	0.445
0.9	0	0.8139	0.7538	0.4834	0.4426
0.8	0	0.8153	0.751	0.4782	**0.4472**
0.7	0	0.8114	0.7514	0.4715	0.4453
0.6	0	0.8168	0.754	0.467	0.4445
0.5	0	0.8168	0.7533	0.4607	0.4456
0.3	0	0.8201	0.7536	0.4563	0.4447
0	0	**0.8271**	0.7579	0.4476	0.4416

2. **Maximum Iterations Bounded Step Size**: To make full use of every intermediate adversarial example on each minibatch, we simply set the relative step size as $\frac{1}{m}$ during m iterations on the same minibatch to ensure that we can fully smoothe the whole region within the ϵ ball for each minibatch.
3. **Strong Label Smoothing Degree**: The key that strong LS works is that adversarial examples are treated differently based on their relationship with the decision boundary. Based on that, for the smoothing bound (λ_0, λ_1) shown in Eq. 6, we simply set λ_0 as 0, and λ_1 as 0.3 for Resnet-18 and 0.4 for WideResNet-34-10 with a larger model capacity.
4. **Small Targeted Regularization Loss Weight**: Experiments have shown that we only need apply a small weight on our targeted regularization loss as shown in Eq. 22 to achieve a better trade-off of robust accuracy between untargeted attack and targeted tar attack. Here we set β as 0.2 for Resnet-18 and 0.1 for WideResNet-34-10.

Furthermore, to speed up AT, we adopt cyclic learning rate and mixed precision arithmetic from the DAWNBench competition adopted by (Wong et al., 2020). We adopt the same learning rate setting as (Wong et al., 2020): the learning rate linearly increases from zero to the maximum value 0.04 and then back down to zero. To achieve a better smoothness of local loss surface, we use longer training epochs. Specifically, for Resnet-18, the number of training epochs N is 20 while the number of minibatch replays m is 8. For WideResNet-34-10, N is set as 24 with m as 4. All the models are trained using SGD with momentum 0.9, weight decay 5×10^{-4}.