LTD: Low Temperature Distillation for Gradient Masking-free Adversarial Training

Anonymous authors
Paper under double-blind review

Abstract

Adversarial training has been widely used to enhance the robustness of neural network models against adversarial attacks. However, there is still a notable gap between nature accuracy and robust accuracy. We found one of the reasons is the commonly used labels, one-hot vectors, hinder the learning process for image recognition. Representing an ambiguous image with the one-hot vector is imprecise and the model may fall into a suboptimal solution. In this paper, we propose a method, called Low Temperature Distillation (LTD), which is based on the knowledge distillation framework to generate the desired soft labels. Unlike the previous work, LTD uses a relatively low temperature in the teacher model, and employs different, but fixed, temperatures for the teacher and student models. This modification boosts the robustness without defensive distillation. Moreover, we have investigated the methods to synergize the use of nature data and adversarial ones in LTD. Experimental results show that without extra unlabeled data, the proposed method combined with the previous works achieve 58.19%; 31.13% and 42.08% robust accuracy on CIFAR-10; CIFAR-100 and ImageNet data sets respectively.

1 Introduction

Deep neural networks (DNN) have been widely used in many challenging tasks, such as image classification (Krizhevsky et al., 2012), object detection (Wang et al., 2022), image captioning (Herdade et al., 2019), and semantic analysis (Zhang et al., 2018). Those techniques are the building blocks for many advanced applications, like self-driving car (Grigorescu et al., 2020), or machine translation (Devlin et al., 2018). As the population of DNN applications is growing, researchers are investigating more practical issues other than accuracy, such as robustness, model compression, unsupervised learning, and more.

Among those research directions, the robustness is a critical issue since DNNs are vulnerable to adversarial attacks. The goal of adversarial attacks is to confuse networks with high confidence. For example, the attacks modify the input images with small perturbations, indistinguishable by human's eyes, so the modified images can easily fool the most accurate models (Kurakin et al., 2017; Szegedy et al., 2014). Such kinds of attacks can also exist for audio space (Carlini & Wagner, 2018) and text classification (Miyato et al., 2017; Kwon & Lee, 2022) in natural language processing. Those attacks exist not only in the digital world, but can also happen in the physical world, such as the cell phone camera attack (Kurakin et al., 2018) or road sign attack (Eykholt et al., 2018; Zolfi et al., 2021). Besides the inference time attacks, the backdoor attacks (Yao et al., 2019; Chen et al., 2017b) that weaken model accuracy during the training time have also been investigated.

The adversarial attacks can be refined as the white-box attacks and the black-box attacks, based on how much information is leaked. For white-box attacks, the complete information of target networks is accessible. Examples include FGSM (Goodfellow et al., 2014), CW attack (Carlini & Wagner, 2017) and AutoAttack (Croce & Hein, 2020). For black-box attacks, like SPSA (Uesato et al., 2018), ZOO (Chen et al., 2017a) and Square Attack (Andriushchenko et al., 2019), attackers only have the information about the output probability. Also, the number of queries is also limited. Furthermore, adversarial examples generated by another model can fool the target model if those models have similar architectures (Papernot et al., 2017). Another taxonomy of attacks is based on the constraints of the adversarial examples: restricted attacks

(Szegedy et al., 2014) and unrestricted attacks (Brown et al., 2018). For restricted attacks, the size of perturbation is in the given ϵ-ball. On the contrast, unrestricted attacks allow any type of changes on the input data.

The most widely used defensive strategy is the adversarial training (Madry et al., 2018), which is usually formulated as a min-max optimization problem. The inner maximization is to search for the strongest adversarial examples, and the outer minimization is to reduce the objective loss caused by those adversarial examples. In Madry et al. (2018), PGD training employs repeated PGD to generate adversarial examples. Such method and its varieties (Zhang et al., 2019b; Pang et al., 2021; Gowal et al., 2020) have shown good robustness. But the computation cost of performing PGD is relatively high. Faster training methods, such as AdvForFree (Shafahi et al., 2019), YOPOp (Zhang et al., 2019a), FastFGSM (Wong et al., 2020), and enhanced FGSMp (Chen & Lee, 2020), were proposed to improve the training efficiency, but they often have a lower robust accuracy since they replace strong adversarial examples with lightweight attacks, and potentially causes catastrophic over-fitting.

The use of natural data or auxiliary information from an external data in adversarial training can further improve robust accuracy. In TRADES (Zhang et al., 2019b), authors suggested that solving the loss caused by adversarial examples directly ignores the importance of the natural data. Instead, the surrogate loss term is introduced to mitigate the conflict between the natural data and the adversarial examples. Defensive distillation (Papernot et al., 2016) is another technique, which uses a teacher model trained by clean data to calibrate the target model's parameters polluted by unusual examples during the adversarial training. However, in Carlini & Wagner (2017), authors showed that the defensive distillation causes the gradient masking problem, which cannot be used to defend gradient-free attacks or black-box attacks. In UAT (Alayrac et al., 2019), authors showed that TRADES on CIFAR10 with 200,000 additional images improved the robustness significantly. However, selecting extra data is another task-specific problem that cannot be generalized.

In this paper, we focused on the problem of defending the restricted white-box attacks in the image space. We found that one of the sources making CNNs vulnerable is from the ambiguous examples which have features from multiple classes. Using a one-hot vector to represent ambiguous examples is imprecise, which may lead to learning bias in the training stage. Although the best label representation of ambiguous examples is unknown, we showed that the soft labels generated from a well-trained model are better than one-hot labels. The soft labels can be applied to existing works and improve the robustness. Based on this concept, we proposed a new algorithm, called Low Temperature Distillation (LTD). Unlike defensive distillation (Papernot et al., 2016), we have shown that LTD does not have the gradient masking problem. In addition, we have studied the inconsistent problem of the parameters in batch normalization, which is caused by using both natural data and adversarial examples.

We have conducted experiments to evaluate the effectiveness of the proposed methods. We evaluated LTD using CIFAR10; CFAR100 and ImageNet datasets, trained the models using Wide Residual Network (WRN) family or ResNet-50, and compared LTD with competitors from leaderboard Robustbench (Croce et al., 2021). The experimental results showed that LTD using WRN-34-10 architecture achieves 55.09% robust accuracy without additional data. Combining with Adversarial Weight Perturbation (AWP) (Wu et al., 2020), LTD can achieve 58.19% robust accuracy, which is the current best result. For CIFAR100 dataset, LTD achieves 31.13% robust accuracy, which is also the best result without additional data. For ImageNet, LTD obtains 42.08% robust accuracy, which is about 4% improvement under same network architecture.

The rest of this paper is organized as follows. Section 2 reviews the related works. Section 3 gives a mathematical explanation about why the soft labels give more robust training results than one-hot vectors. Section 4 presents the LTD algorithm and its implementation. Section 5 shows the experimental results. Conclusion and future work are given in the last section.

2 Related Works

In this section, we give an introduction to the related works. Three topics will be covered. First is the adversarial attacks and defenses. Second is the quality of gradient, used to evaluate the robustness of defense strategy. Last is the framework of knowledge distillation.

2.1 Adversarial Attack

The original definition of adversarial examples is the modified input which is indistinguishable from human's eyes but can fool the classifiers. To satisfy the constraints, the distance between the modified image and the original image is required close to each other.

Those attacks are considered as restrict attacks (Szegedy et al., 2014). By contrast, unrestricted attacks (Brown et al., 2018) allow any type of changes on the input data, such as rotations and deformations. Those data in the semantic space are still close to each other, but their distances in the image space could be far away. For example, DNNs may not recognize a photographic road sign from different angles, even though identifying road sign is a simple task for humans. Corrupted images are another type of image which similar to unrestricted attacks. Those corrupted images are computed by selected operations without model architecture's information. From the viewpoint of robustness, DNNs should categorize both types of images correctly. ImageNet-A (Hendrycks et al., 2021) and ImageNet-C (Hendrycks & Dietterich, 2019) are two public data-sets designed for evaluating model's robustness and generalization.

Most attacks compute adversarial examples by the gradient's ascending direction to increase the objective loss. FGSM attack (Goodfellow et al., 2014) generates adversarial examples as

$$x^{\text{FGSM}} = \mathcal{P}(x + \alpha \text{sign}(\nabla L(x, y))), \tag{1}$$

where L is the objective loss, α is the step size, and $\mathcal{P}$ is the projector which ensures that x^{FGSM} is still in the feasible set. Similarly, PGD attack generates adversarial examples by iteratively running m times FGSM in (1) with smaller step-size α/m to get stronger adversarial images.

Currently, AutoAttack (AA) (Croce & Hein, 2020), is the strongest attacking algorithm, which includes APGD-CE, APGD-DIR, FAB and Square attack. APGD-CE, APGD-DIR and FAB are PGD-based attacks with different objective functions or updating rules. Since most defenses compute adversarial examples by a specific attacking algorithm, the model may over-fit to the given attacking algorithm. This type of model may be defeated by others algorithms and the robust accuracy has a significant dropping eventually. Therefore, the white-box evaluation requires multiple attacks to ensure generalization. Moreover, AA includes targeted version of attacks to ensure the imbalanced gradient does not exist. The square attack is a query-efficient black-box attack that can detect the gradient masking effect. To speed up computing efficiency, AA filters out misclassified data by current attack quickly and the remaining candidates will be tested with the following attacks.

2.2 Adversarial Defense

We categorize adversarial defenses into three types. The first type of adversarial training focuses on optimization methods. The second type employs extra data sets in adversarial training. The last category puts model weights into consideration. They are introduced as follows.

2.2.1 Adversarial Training

The standard adversarial training proposed by Madry et al. (2018) is designed to defend against adversarial examples, which can be formulated as

$$\min_{\theta} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \max_{\tilde{x}: D(\tilde{x}, x) < \epsilon} L(Z(\tilde{x}; \theta), y). \tag{2}$$

where $D(\tilde{x}, x)$ is a distance function for $\tilde{x}$ and x. The goal of (2) is to minimize the losses caused by the strongest adversarial examples defined in a constraint set. The strongest adversarial examples are indeterminable, so the inner maximization is usually replaced by some optimization-based attacks practically.

TRADES (Zhang et al., 2019b) proposed that the worst-case loss in (2) cannot be optimized effectively. Instead, TRADES decomposed the adversarial loss into the natural loss and the boundary loss:

$$L_{\text{TRADES}}(x, x', y) = L(Z(x;\theta), y) + \lambda \Delta L(x, x', y; \theta), \tag{3}$$

where $L(Z(x;\theta), y))$ is original loss in (2) and $\Delta L(x, x', y; \theta)$ is a regularization term. The first term in (3) maximizes model output distribution between natural data and its corresponding label. The second term encourages the output distribution to be smooth and pushes the decision boundary away from given examples. ALP (Kannan et al., 2018) also firstly suggested adding $L2$ penalty on output probability as ΔL and the follow-up works (Zhang et al., 2019b; Wang et al., 2019) have shown the robustness can be improved significantly if KL divergence (KLD) is used for the regularization loss. The major benefit of KLD loss is label-free. Recent works showed KLD can be used to improve the robustness with extra unlabeled data (Alayrac et al., 2019; Carmon et al., 2019; Gowal et al., 2020).

2.2.2 Additional Data Sets

Traditional supervised learning almost ignores low-frequency data so DNNs cannot recognize the low-distribution data correctly. The adversarial training is increasing the frequency of adversarial examples which do not appear in natural data. Another strategy is adding extra data which may cover the data in the regime of low frequency into the original training set. UAT (Alayrac et al., 2019) showed TRADES on CIFAR10 with 200,000 additional images improved the robustness significantly. One may criticize that the number of additional images is much larger than that of CIFAR10's training set (50,000 images) and processing those data requires more computational cost. In UAT, the authors also concluded that selecting subset from the additional images properly has better robustness than that of using entire additional images. The follow-up work RST (Carmon et al., 2019) designed a special classifier to select relevant images, which joined into the adversarial training set, from another dataset. Those similar works, which rely on huge additional images, cannot be extended to ImageNet or other large-scale data sets. Besides, selecting data under the same distribution is an opening problem.

2.2.3 Adversarial Weight Perturbation

The original adversarial training computes adversarial perturbation in input space and minimizes the loss caused by those perturbations. However, the vulnerability may come from the weights in the hidden layers. In AWP (Wu et al., 2020), the authors suggested that the descent direction should be composed by gradients from adversarial perturbation in the input space and in the weight space. This approach can be integrated with existing works, and the empirical results showed using composed gradients to update the weights in the model has higher robustness.

2.3 Quality of Gradients

The gradient of input images with respect to the loss function plays an important role in adversarial training. A common way to compute adversarial examples is by using the gradient direction of input images. Based on which, one type of defense strategies, called gradient masking methods, are designed to make the gradient indeterminable, so the robust accuracy is overestimated by gradient-based attacks. But the gradient masking methods are vulnerable to gradient-free or black-box attacks (Athalye et al., 2018). Therefore, a model is called robust if high-quality gradients are kept and attackers cannot defeat the defense with those gradients.

In Athalye et al. (2018), authors listed five rules to evaluate the quality of gradients. They are based on three concepts. First, a model will be defeated eventually if the strength of adversarial attacks is increased. Second, gradient-based attacks should be more efficient than gradient-free attacks. Third, multi-step attacks should be stronger than one-step attacks. If a model violates one of the properties, it has a high probability to have the gradient masking problem. In Carlini et al. (2019), authors summarized what pitfalls should be avoided and useful guidelines for designing a robust training procedure.

Adversarial robustness may still be overestimated although the gradient masking does not exist (Jiang et al., 2020). In Jiang et al. (2020), authors claimed that the direction of the gradient may be towards suboptimal direction if the gradient of one loss dominates that of other losses. The phenomenon is called an imbalanced gradient, which cannot be detected by the rules enumerated by Athalye et al. (2018); Carlini et al. (2019). A probable strategy to escape the suboptimal point is applying distinct losses as ascending directions.

2.4 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) originally is designed for training a smaller model which can be deployed on edge devices. Nowadays, it has been a stepping stone for numerous algorithms (Gou et al., 2020). The core concept of knowledge distillation is to provide a meaningful label representation from a pre-trained model, called the teacher model. The information distilled from the given model is called the soft labels p for a given temperature $T = \tau$,

$$p_i^{T=\tau} = \frac{\exp(q/\tau)_i}{\sum_{j=1}^{k} \exp(q/\tau)_j}, \tag{4}$$

where q is the logit computed by the given model. The target model is trained by the following loss function:

$$\begin{aligned} L = &- \sum_{i=1}^{k} y_i \log p_i^{s,T=1} \\ &+ \lambda \sum_{i=1}^{k} p_i^{t,T=\tau} \log \left(\frac{p_i^{t,T=\tau}}{p_i^{s,T=\tau}} \right), \end{aligned} \tag{5}$$

where y is the given one-hot labels; $p^{s,T=\tau}$ and $p^{t,T=\tau}$ are labels given from the target model and the teacher model using the temperature τ respectively, and λ is a scaled factor. The first term in (5) is ordinary categorical loss and the second term is KL divergence (KLD). In Hinton et al. (2015), the authors claimed that the magnitudes of the gradient of KLD loss in temperature τ is scaled by $1/\tau^2$ and the proper λ is τ^2 to balance two losses.

Defensive distillation (Papernot et al., 2016) used knowledge distillation for adversarial training. However, CW attack (Carlini & Wagner, 2017) proved that the principle of defensive distillation has the gradient masking problem (Athalye et al., 2018). There are two major factors to cause this problem. First, the gradient of images can be formulated as a function of the output probability:

$$\nabla_x L_{\text{SCE}} = (p_t - 1)\nabla_x q_t + \sum_{i \neq t} p_i \nabla_x q_i,$$

where p_i is the probability of class i and q_i is the logit of class i. This equation shows the gradient almost vanishes when p_t is close to 1.

Second, the temperatures of the target mode in the training stage (T_t) and in the inference stage (T_i) are different, where T_t is high and $T_i = 1$. It means that the output probability at high temperature (T_t) is in the one-hot distribution, and the magnitude of logits in the inference time is $(T_t/T_i) = T_t$ times larger than that of in training time. Consequently, the largest logit dominates others and the output probability converges to a one-hot vector and falls into the area of gradient-vanishing.

The combination of those two factors makes the gradient masking problem more serious. Therefore, attackers cannot generate adversarial examples by the gradient in inference time.

3 Data Labeling

3.1 Problem definition

Here we define notations used in this article. First, the natural data are a set of pairs (x, y), where x is an input image and y is its label, which is a one-hot vector. More specifically, if x belongs to class c_t, label y is defined as

$$y \in \{0,1\}^k : y_i = \begin{cases} 1, & \text{if } i = c_t, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

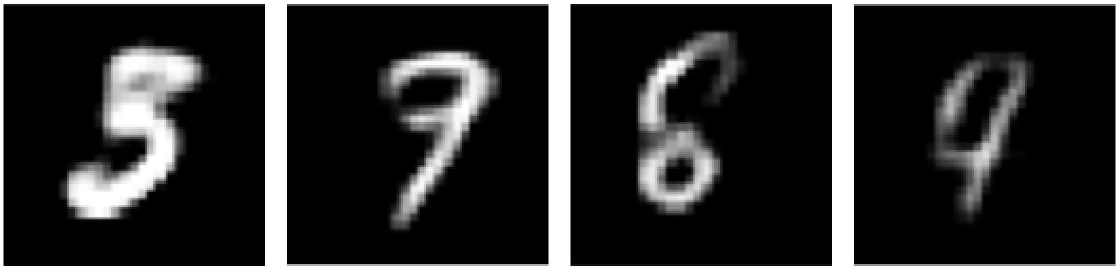


Figure 1: Ambiguous images in MNIST dataset.

where k is the number of classes. A classifier $Z(x;\theta)$ is a function of input image x and model's weights θ, and outputs logits q. The classifier predicts the input x belong to label h by the function

$$h = \arg\max q_i = \arg\max Z(x;\theta)_i,$$

where q_i is the logit for class i.

The normal training, which optimizes the given objective loss L by receiving data pairs (x, y) on the training set and a classifier Z as inputs, can be defined as an optimization problem:

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{train}}} L(Z(x,\theta), y). \tag{7}$$

The performance of the classifier can be evaluated by how many examples on the testing set $\mathcal{D}_{\text{test}}$ are classified correctly. The metric, called 0-1 loss, can be represented as follows

$$\max \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{test}}} \mathbb{1}(h = c_t), \tag{8}$$

or

$$\min \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{test}}} \mathbb{1}(h \neq c_t), \tag{9}$$

where $\mathbb{1}$ is the indicator function. However, 0-1 loss is a non-differentiable function which cannot be used to optimize the weights θ of the classifier in (7). Alternatively, the commonly used L is softmax cross entropy loss (SCE):

$$L = -\sum_{i=1}^{k} y_i \log p_i, \tag{10}$$

where p_i represents the normalized probability of each class using the softmax function,

$$p_i = \frac{\exp(q_i)}{\sum_{j=1}^{k} \exp(q_j)}. \tag{11}$$

From one-hot representation, SCE loss in (10) is equivalent to

$$L = -\sum_{i=1}^{k} y_i \log p_i = -\log p_{c_t}. \tag{12}$$

The above loss maximizes the probability of the target class and implicitly minimizes (9) simultaneously.

3.2 Real World Scenario

For a k-class classification problem, three implicit assumptions are required: closed-world assumption, independent and identically distributed (i.i.d) assumption and clean and big data assumption (Zhang et al., 2020b). The closed-world assumption supposes that the number of the class k is predefined and all examples

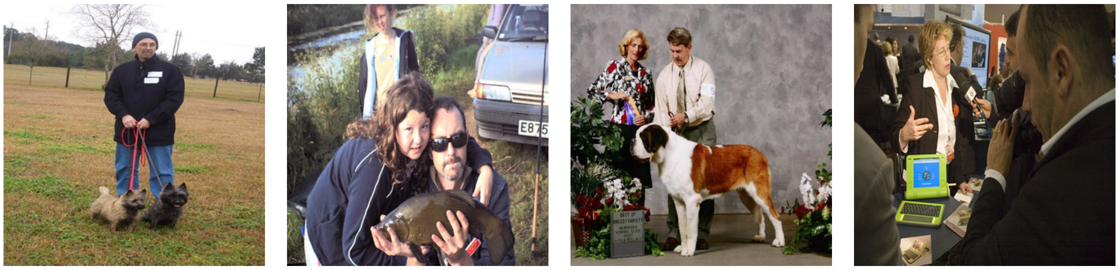


Figure 2: Images with multiple objects in ImageNet dataset.

come from one and the only one predefined class. In practice, 0-1 loss is not measurable because samples on $\mathcal{D}_{\text{test}}$ are unknown. The i.i.d assumption supposes that all samples on $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{test}}$ are drawn from an identical distribution. Under the i.i.d assumption, 0-1 loss can be approximated by (7), called empirical risk, which is estimated with observed samples on $\mathcal{D}_{\text{train}}$. The clean and big data assumption supposes that all collected data should be well-labeled and large enough for covering the population.

In fact, none of the above assumptions are satisfied in a real-world scenario. Figure 1 shows some images in MNIST dataset are located in the boundary of two classes in semantic space. It is acceptable for those images to be labeled as either of two classes or both classes. Figure 2 shows images with multiple objects in ImageNet dataset. Moreover, semantic distances among classes are not uniform. For examples, *automobile* and *truck* are two similar classes in CIFAR10 dataset. For ImageNet dataset, *sunglass* (n04355933) and *sunglasses* (n04356056) are duplicated. *laptop* (n03642806) and *notebook* (n03832673) in ImageNet dataset are identical in semantic space but notebook may refer to a book of plain paper. Previous studies argued that wrong annotation procedure might cause performance degradation (Beyer et al., 2020; Tsipras et al., 2020).

The trained model using SCE loss with a one-hot vector has a low empirical loss, but it makes overconfident predictions on ambiguous samples or poor predictions on samples drawn from different distributions. This phenomenon is called over-fitting or poor generalization which has been witnessed in many applications. Additionally, minimizing SCE loss with a one-hot vector in (12) fades the rest of classes' probabilities except for the selected class. This constraint is stricter than that of 0-1 loss, by which the predicted label is obtained from the relatively important class among all classes. The evidence suggests that the information from multiple classes in those ambiguous images cannot be delivered with one-hot representations properly.

An intuitive strategy to break the closed-world assumption is replacing SCE loss with another widely used function Kullback–Leibler divergence (KLD), which measures the difference between two distributions by a weighted sum:

$$L = \sum_{i=1}^{k} y_i^g \log \frac{y_i^g}{p_i}, \tag{13}$$

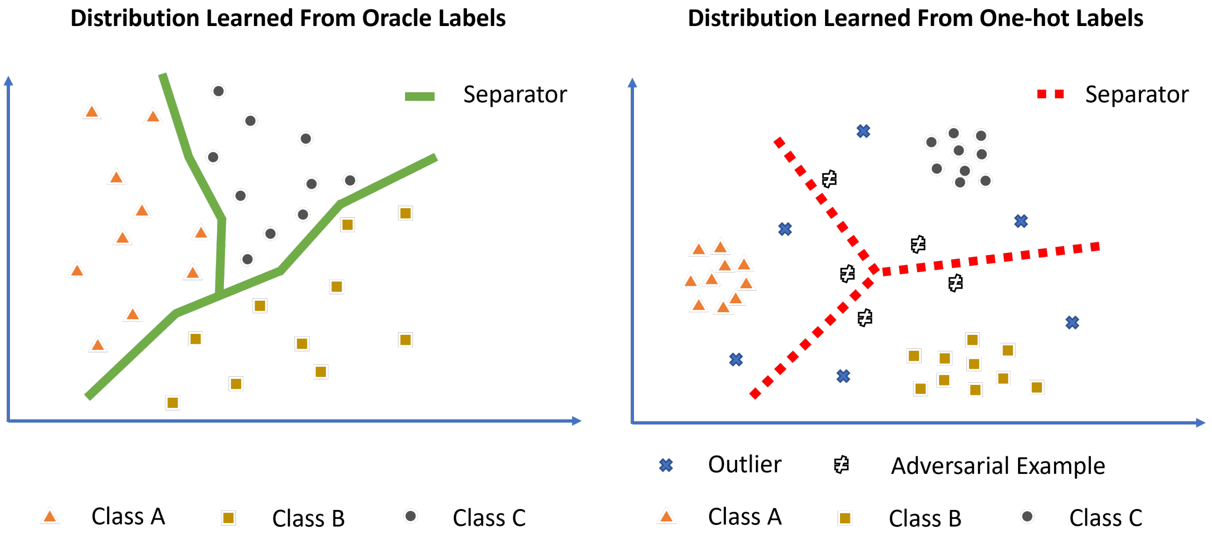
where p is the predicted output probability and y^g is the given probability. The weight of each unit is its given probability y_i^g and the distance is measured by $\log(\cdot)$. Normally, y^g is predefined and cannot be modified, so (13) is simplified as

$$L = \sum_{i=1}^{k} y_i^g \log \frac{y_i^g}{p_i} = -\sum_{i=1}^{k} y_i^g \log \frac{p_i}{y_i^g} = -\sum_{i=1}^{k} y_i^g (\log p_i - \log y_i^g) = -\sum_{i=1}^{k} y_i^g \log p_i + C, \tag{14}$$

where C is a constant. Comparing to SCE in (10), one can find that the one-hot label y_i is replaced by y_i^g. Therefore, solving the classification problem using SCE loss is a special case for minimizing KLD distance between predicted probability and one-hot distribution.

3.3 Generalization Problem

We argue that the generalization issue is from the wrong assumption of the distribution on label space. As shown in Figure 3(a), the separator learned from the oracle distribution is a complicated Y shape since data



(a) optimal solution

(b) one-hot vector solution

Figure 3: The separators learned from different distributions. (a) The separator learned from the oracle distribution. (b) The separator learned from the one-hot labels.

drawn from oracle distribution cover the entire space uniformly. Those data near the boundary refine the shape of the separator. On the contrary, Figure 3(b) displays the separator learned from the one-hot labels. Under one-hot representation, samples in each class are wrongly assumed to be far from the boundary. The separator can be an arbitrary shape in the empty zone among classes. Consequently, the separator learned from different distributions may inconsistent.

Although 0-1 loss is a useful metric for evaluating the performance of classifiers, it cannot evaluate the discrepancy between two separators in Figure 3 and it is not clear when images are confusable. For instance, two models have the same 0-1 loss but one of which misclassifies more trivial samples than that of the other. We would say the latter one is the better model although it has high misclassification rate at in the confusing area. We need a precise metric to estimate distribution mismatch. Here, we measure the generalization of a model Z by the closeness between the output probability p and the oracle probability y^g for each class. The closeness is measured by KLD loss.

Definition 1. *Let $\mathcal{D}$ be the set of all data to be classified, x be an instance in $\mathcal{D}$, $y^g(x)$ be the oracle distribution of x, and p be the probability outputs by Z of x. The generalization of Z is defined as the following expression,*

$$G(Z) = \mathop{\mathbb{E}}_{(x,y^g)\sim\mathcal{D}} \left[\sum_{i=1}^{k} -y_i^g(x) \log p_i(x) \right]. \tag{15}$$

The above metric tells if a model Z has a small $G(Z)$, it is more general because its output is closer to the oracle distribution.

3.4 Distribution of Adversarial examples

We use x' to represent the adversarial image if x' is close to the original image x but the classifier $Z(x;\theta)$ predicts that those two images belong to different classes. The description can be represented as follows

$$\mathcal{S} = \left\{ x' \left| \begin{array}{l} \arg\max Z(x;\theta)_i = \arg\max y_i \\ \arg\max Z(x;\theta)_i \neq \arg\max Z(x';\theta)_i \\ ||x' - x||_\infty \leq \epsilon \end{array} \right. \right\}, \tag{16}$$

where ϵ is the allowed distance in L_∞ space. The difference between x' and x is called the adversarial noise, which is the perturbation, smaller than or equal to ϵ, computed by any attacking algorithm for the corresponding image x.

Most of adversarial examples are man-made data near the decision boundary. Although their corresponding losses are much higher, they contribute tiny efforts to empirical loss in 7 because adversarial examples are rare occurrences in the natural distribution. Therefore, the robustness of state-of-the-art classifiers is poor.

To calibrate the distribution shift, the optimization problem is reformulated as follows,

$$\min_{\theta}(1-\lambda) \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_{\text{nat}}} L(Z(x,\theta),y) + \lambda \mathop{\mathbb{E}}_{(x',y)\sim\mathcal{D}_{\text{adv}}} L(Z(x',\theta),y), \tag{17}$$

where $\mathcal{D}_{\text{nat}}$ is the natural distribution, $\mathcal{D}_{\text{adv}}$ is the adversarial distribution and λ is a term used for adjusting the importance of adversarial examples. A pioneering study (Madry et al., 2018) achieved promising improvement on the robustness by choosing λ to be 1.

There are many approaches being studied to recover the generalization. A simple solution is to enlarge the sampling space by augmentation, such as AutoAugment (Cubuk et al., 2019) and RandAugment (Cubuk et al., 2020), or using additional data-sets, which may cover more unseen data in the original data set. However, ambiguous examples, which have multiple interpretations on the objects in the images, cannot be processed well by typical categorical classifiers because those examples are still represented by one-hot labels.

3.5 Batch Normalization Parameters Updating

When solving the multi-objective optimization problem (17), two different distributions are used. In this section, we discuss the impact of different distributions in Batch Normalization (BN) (Ioffe & Szegedy, 2015).

BN updates model weights by the following procedure. Each mini-batch's mean μ and variance σ^2 are updated during the forward phase, and the output $\hat{v}$ of each channel of BN layer's is calculated as follows:

$$\hat{v} = \gamma \frac{v-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta, \tag{18}$$

where γ and β represent scale and bias respectively, and ϵ is a small value to avoid the numerical instability. In most implementations, μ and σ^2 are usually updated by the moving average:

$$\begin{cases} \mu_{s+1} = m\mu_{s-1} + (1-m)\mu_s \\ \sigma^2_{s+1} = m\sigma^2_{s-1} + (1-m)\sigma^2_s, \end{cases} \tag{19}$$

where m is the momentum for the moving average. More detail of BN implementation can be found in Summers & Dinneen (2019).

BN is an essential component of deep neural networks. However, it is an obstacle in adversarial training (Pang et al., 2021; Xie et al., 2020; Xie & Yuille, 2019) because the distributions of adversarial examples and natural data are quite distinct. In Xie et al. (2020), authors argued that maintaining two mixture distributions in one BN is imprecise. To solve the problem, authors proposed an auxiliary BNs to disentangle two distributions. The original BNs contain an estimation from natural data; adversarial data's statistical information are stored in the auxiliary BNs, which are not used in the inference time. In other words, each type of data has its own BNs. Experimental results showed that models enjoy the benefit from adversarial features and natural accuracy is improved. However, the authors did not investigate whether the robustness is improved or not.

We argue that the auxiliary BN cannot be integrated into TRADES or modern defense strategies because the regularization loss receives both x and x', so μ and σ^2 must be updated twice by both type of data. Additionally, the distributions of adversarial examples computed by distinct attacking algorithms are diverged. For example, FGSM attack cannot enumerate amounts of adversarial examples generated by PGD-k attack efficiently although two algorithms search adversarial examples whthin the given ϵ boundary.

The potential implementation choices which may affect the robustness are listed below

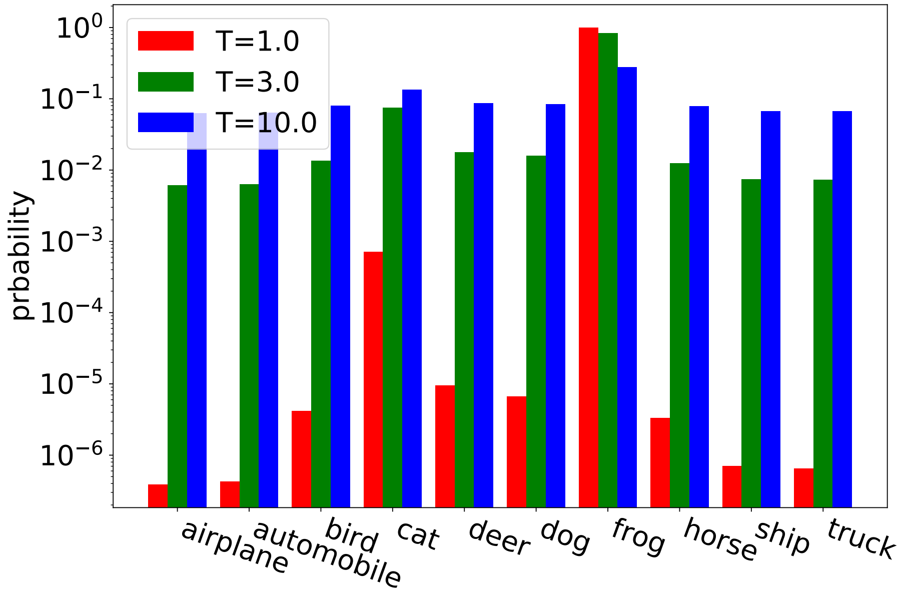


Figure 4: The output probability of the naturally trained model on different temperatures on CIFAR10 dataset.

1. *Should* μ *and* σ^2 *be updated during the computation of adversarial examples?*
2. *Which data (adversarial or natural) should be used to update* μ *and* σ^2 *when computing the objective loss?*
3. *If both data (adversarial or natural) are used, what is the order to use them?*

The first question has been discussed in Pang et al. (2021); Xie & Yuille (2019). Although their empirical results show that learning μ and σ^2 during generating adversarial examples would enhance the robustness, the authors advocated not to do so, because the natural accuracy would drop and the adversarial distribution would be blurred. We argue that learning the arbitrary distribution from the adversarial examples generating by multiple attacking algorithms would improve robustness even though we cannot know what the true adversarial data's distributions are. Including them can enrich the distributions. Therefore, there is no a strong reason that model should only learn μ and σ^2 from the final adversarial examples.

Question 2) and question 3) need be investigated empirically, because as shown in (19), moving average is not a commutative operation and the final result depends on the order of computation. However, we suggest that the statistical information is amalgamated if the model trained with the mixture of distributions. On the contrary, if the statistical information is updated by date from one of distribution only, the model weight cannot converge.

3.6 Soft Labels by Knowledge Distillation

As mentioned in Section 3.4, one-hot labels are not reliable for a real-world scenario. We need an approach to approximate the oracle probability. Label smoothing (Müller et al., 2019) is a particular skill which redistributes partial probability from the target class to others equally and the model is trained by SCE loss with the soft labels. Although label smoothing can reduce the model's output confidence, redistributing probability equally does not solve issues about ambiguous images and give a correct inter-class distance. A better label representation is to replace the one-hot label with the distribution which can reveal the probability of the classes related to the target class.

Here we propose a new method to ease the generalization problem. The idea is to replace the one-hot representations with soft labels. Even though the training data are still biased, the soft-labels, if correctly generated, can reflect the underlying distributions, and push the decision boundary toward the optimal one.

Our method uses the framework of knowledge distillation to generate the soft labels. Although the defensive distillation (Papernot et al., 2016) also uses a similar framework, its goal still wants to fit the one-hot labels. Instead, the purpose of our training method is to learn the soft labels from the teacher model. This difference distinguishes our method from the defensive distillation, no matter in the methodology or in the experimental results.

One of the most crucial factors in our approach is the temperature selection in the teacher model. The naive selection, such as $T = 1$, does not work well. To illustrate this idea and simplify the discussion, we make the following assumptions.

Assumption 1. *Let M be a model trained by using soft-labels y^p and SCE in (10). When the training finishes, the probability p of each input image x output by M equals to the corresponding soft label y^p.*

This is a reasonable assumption because if M is well-trained, its SCE loss should be less than the given threshold. Eventually, the probability of each image of the trained model can approximate to the given soft label. This ensures that the expected KLD distance between the probability of x and the given soft label y^p is less than the given threshold. To simplify the discussion, we just assume the threshold is negligible.

Next, we look at a simple classification problem, in which there are only two classes: class 1 and class 2. For a given image x, its oracle soft label is $(y_1^p,\, y_2^p)$. Without loss generality, we assume that $y_1^p > y_2^p \geq \xi$, where ξ is a small positive number. Furthermore, let $(p_1^{T=t}, p_2^{T=t})$ be the probability of x after (4). Since the model is making the correct classification, $p_1^{T=t} > p_2^{T=t}$ for any temperature t.

Now we consider two models, Model 1 and Model 2, which have the same network architecture but trained by the soft labels that are generated from the teacher model with different temperatures t_1 and t_2 where $t_2 > t_1 \geq 1$. Using the definition in (15), we can measure their difference of the generalization:

$$\begin{aligned}\Delta L^{t_1 \to t_2} &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_{i=1}^{k} -y_i^p \log p_i^{T=t_2} - \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \sum_{i=1}^{k} -y_i^p \log p_i^{T=t_1} \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}} -\sum_{i=1}^{k} y_i^p \log\left(\frac{p_i^{T=t_2}}{p_i^{T=t_1}}.\right).\end{aligned} \tag{20}$$

We say that the model trained by the soft label using temperature t_2 is more general than temperature t_1 if

$$\Delta L^{t_1\to t_2} < 0. \tag{21}$$

The following theorem gives a sufficient condition of $\Delta L^{t_1\to t_2} < 0$.

Theorem 1. *With the above assumptions, if for each image, $p_1^{T=t} > p_2^{T=t}$ for any t, and*

$$\frac{y_1^p}{y_2^p} \leq \left|\frac{\log(p_2^{T=t_2}/p_2^{T=t_1})}{\log(p_1^{T=t_2}/p_1^{T=t_1})}\right| \tag{22}$$

then $\Delta L^{t_1\to t_2} < 0$.

Proof. We start by simplifying $\Delta L^{t_1\to t_2}$:

$$\begin{aligned}\Delta L^{t_1\to t_2} &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}} -\sum_{i=1}^{k} y_i^p \frac{p_i^{T=t_2}}{p_i^{T=t_1}} \\ &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}} -y_1^p \log\left(\frac{p_1^{T=t_2}}{p_1^{T=t_1}}\right) - y_2^p \log\left(\frac{p_2^{T=t_2}}{p_2^{T=t_1}}\right).\end{aligned} \tag{23}$$

Because p_1 is decreasing as T increases, and p_2 is increasing with T, $\log(p_1^{T=t_2}/p_1^{T=t_1})$ is negative and $\log(p_2^{T=t_2}/p_2^{T=t_1})$ is positive. If condition (22) holds, for each image, we have

$$\begin{gathered}\left|y_1^p \log\left(\frac{p_1^{T=t_2}}{p_1^{T=t_1}}\right)\right| \leq \left|y_2^p \log\left(\frac{p_2^{T=t_2}}{p_2^{T=t_1}}\right)\right| \\ -y_1^p \log\left(\frac{p_1^{T=t_2}}{p_1^{T=t_1}}\right) - y_2^p \log\left(\frac{p_2^{T=t_2}}{p_2^{T=t_1}}\right) \leq 0\end{gathered} \tag{24}$$

Aggregating the results in (24) by taking the expectation, we can show that $\Delta L^{t_1 \to t_2} < 0$. ☐

The result of Theorem 1 shows using the soft label $T = 1$ may not be a good idea. Suppose we have two soft labels using $t_1 = 1$ and $t_2 = \tau$, for $\tau > 1$. Generally, the probability distribution of $p^{T=1}$ is closed to one-hot, so $p_1^{T=1}$ is close to 1, and $p_2^{T=1}$ is very small, as shown in Figure 4. Since the probability of p^T follows equation (4), as T increases from 1 to τ, p_1^T decreases slowly and p_2^T increases rapidly. This can also be verified from Figure 4 for $T = 1$ and $T = 3$. If τ is large enough so that $p_1^{T=\tau} \sim p_1^{T=1}$ and $p_2^{T=\tau} \gg p_2^{T=1}$, the right hand side of (22) grows fast. For example, if $p^{T=1} = (0.999, 0.001)$ and $p^{T=\tau} = (0.99, 0.01)$, the right hand side of (22) is about 254. To make (22) hold, we only need $y_2^p > 0.004$, which is not a very strict condition.

We can also apply Theorem 1 to explain that high temperature is not a proper choice. Suppose we have two soft labels from temperature t_1 and t_2, $1 \gg t_1 < t_2$. When the temperature increases, the distribution becomes uniform gradually, as can be seen from equation (4). Because the magnitudes of all probabilities are close, $p_1^{T=\tau} \sim p_1^{T=1}$ and $p_2^{T=\tau} \sim p_2^{T=1}$, the right hand side of (22) is small. So to make the condition hold becomes more difficult. For example, for $t_1 = 5$ and $t_2 = 7$, $p^{T=t_1} = (0.80, 0.20)$ and $p^{T=t_2} = (0.75, 0.25)$, right hand side of (22) is about 3.88. To make the condition hold, we need $y_2^p > 0.23$, which is too large to be possible.

Of course, Theorem 1 is a sufficient condition for $\Delta L^{t_1 \to t_2} < 0$. In reality, if there are few images violating the condition (22), the above arguments about the selection of temperature can still hold. One kind of violating examples is the oracle distribution of images is one-hot, in which $y_2^p = 0$ and y_1^p / y_2^p is infinity. If the number of such kind of images is large, the proper choice of temperature T should be close to 1.

For real-world data sets, the model trained by one-hot distribution lacks the generation because oracle probability fit to incorrect distribution. Instead, increasing tiny temperature adjusts the label distribution without additional data and make the target model more general although the modified distribution is imprecise.

4 Algorithm and Implementations

4.1 Formulation of Loss Function

Our algorithm follows the framework of TRADES, as shown in (3). Unlike the formulation of Madry et al. (2018), whose objective loss is $L(Z(x';\theta), y)$, TRADES reformulates it as a multi-objective optimization problem:

$$L(x, x', y; \theta) = L(Z(x;\theta), y) + \Delta L(x', x, y, \theta), \tag{25}$$

where $L(Z(x;\theta), y)$ is the loss of nature data, the regularization loss ΔL minimizes the distance between the natural probability and the adversarial probability. For TRADES, it uses KLD to measure the difference of natural distribution and adversarial distribution,

$$\Delta L(x', x, y; \theta) = -\sum_{i=1}^{k} p_i^{\text{nat}} \log\left(\frac{p_i^{\text{adv}}}{p_i^{\text{nat}}}\right), \tag{26}$$

where p^{nat} and p^{adv} are the probabilities of natural data and adversarial examples. As shown in the previous section, the natural distribution, used as some kinds of soft labels, can work better than the one-hot labels.

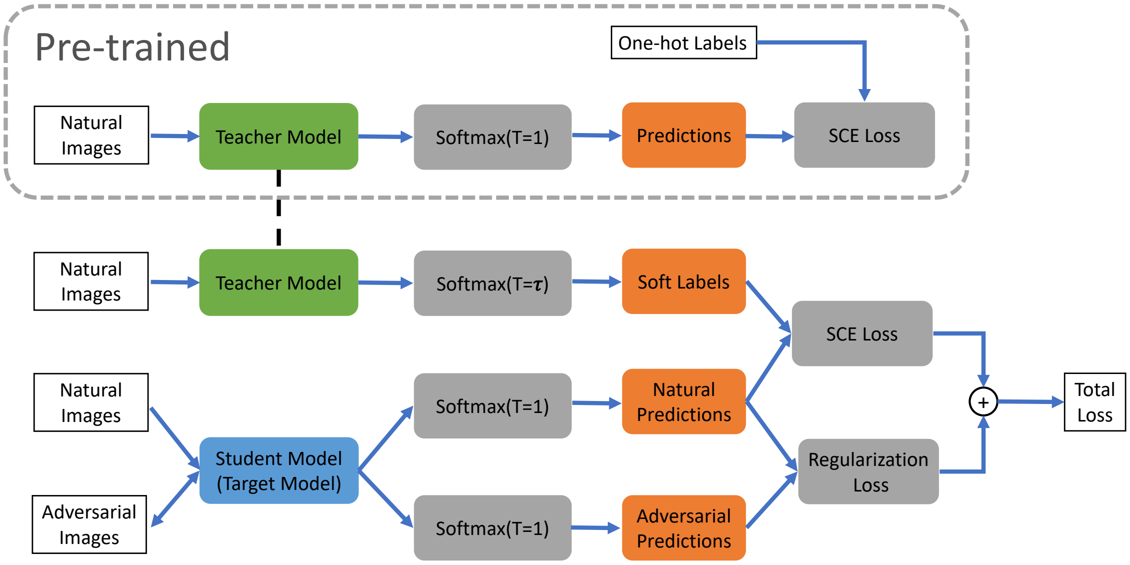


Figure 5: Teacher-target training architecture.

Specifically, the natural distribution shaped by SCE loss $L(Z(x;\theta), y)$ converges to the one-hot distribution. As a result, KLD becomes

$$\sum_{i=1}^{k} p_i^{\text{nat}} \log\left(\frac{p_i^{\text{nat}}}{p_i^{\text{adv}}}\right) = p_t^{\text{nat}} \log\left(\frac{p_t^{\text{nat}}}{p_t^{\text{adv}}}\right) + \eta, \tag{27}$$

where

$$\eta = \sum_{i \neq c_t} p_i^{\text{nat}} \log\left(\frac{p_i^{\text{nat}}}{p_i^{\text{adv}}}\right).$$

By the definition of adversarial attack, there is a class $i \neq c_t$ whose adversarial probability is larger than natural probability. Minimizing the KLD loss means the difference of two probabilities is reduced, so the adversarial probability of class $i \neq c_t$ is decreased or the natural probability of class i is increased. The first case means the strength of adversarial examples is weakened and the second case means class i is related to the target class and its probability is increased.

However, the first term in TRADES formulation still uses one-hot labels, which still create bias in the natural distribution learned from $L(Z(x;\theta), y)$. It may cause the adversarial distribution converges to incorrect distribution. Especially, the importance of ΔL in TRADES is enlarged λ times. In which, natural distribution is a crucial factor for building a robust classifier.

4.2 Training Framework

Based on the analysis, we propose a training framework, called Low Temperature Distillation (LTD), whose architecture is shown in Fig. 5. Followed by the knowledge distillation framework, the teacher model in LTD is obtained from the normal training procedure using SCE loss in (10) with natural images x and one-hot encoded labels y. The trained teacher model has high natural accuracy, but is poor in robustness.

The target model, which is the student model here, is trained with adversarial data and natural data. Its loss function is defined as

$$L_{\text{LTD}} = L(Z(x;\theta), p^t(T = \tau)) + \lambda \Delta L(x', x; \theta), \tag{28}$$

where L is SCE loss; ΔL is KLD loss and λ is to balance two losses. Previous studies (Pang et al., 2021; Wu et al., 2020; Zhang et al., 2019b) have shown λ is about 6.0. During the training stage, adversarial examples are crafted using the target model's information and the original label y. The labels of SCE loss are the soft labels generated by the teacher model, which can be generated in advance or on-fly.

We also compare our approach with defensive distillation (Papernot et al., 2016). The purpose of defensive distillation is to minimize SCE loss with a one-hot label. As mentioned in Section 2.4, applying different temperatures in training and inference time makes output probability closer to one-hot distribution but it causes gradient masking. Instead, the gradient of L_{LTD} with respect to x' is

$$\nabla_{x'} L_{\text{LTD}} = \lambda \sum_i (p_i^{\text{adv}} - p_i^{\text{nat}}) \nabla_{x'} Z(x';\theta)_i. \tag{29}$$

Gradient masking issue occurs if and only if $p^{\text{adv}} \sim p^{\text{nat}}$, which implies that the solution is almost optimal. For defensive distillation, the output probability is computed by (4) and its corresponding gradient is τ times smaller than (29) which worsens the gradient issue. Instead, the temperature of the target model is fixed to 1.0. This configuration ensures that scale of each term in (29) is the same during the training phase and the inference phase. To sum up, our approach has no gradient masking issue.

4.3 Temperature Selection

By the property of (4), when the temperature of the teacher model is high enough, the distribution becomes uniform gradually. The inter-class relation is destroyed and the original SCE loss is dominated by irrelevant classes eventually. This implies that the assumption about the distribution is incorrect, so there is no need to consider the robustness. The proper temperature of teacher model dependents on dataset and model architecture. For CIFAR10 dataset, commonly used models have high confident predictions whose distribution is close to one-hot distribution, so the temperature should be increased. For ImageNet or large-scale datasets, the original distribution may have been suitable for LTD already.

We propose that searching for the proper temperature in two steps. The first step determines the highest temperature $\tau_{\max}$ by (28) with natural data only and its natural accuracy should be higher than the given threshold. The feasible temperature must be lower than $\tau_{\max}$ because adversarial examples hurt natural accuracy as well. Second, the best temperature τ_{best} is found by any hyper-parameter optimization algorithm using our proposed loss function in (28) in the range $[1, \tau_{\max}]$.

5 Experiments

This section presents three sets of experiments. First, we evaluated the robustness of LTD against white-box attacks and compare LTD with other methods on CIFAR10; CIFAR100 and ImageNet datasets. Second, we justified our approach can resist black-box attack and showed gradient masking does not occur. Third, we performed hyper-parameter searching and ablation studies. The full experiment configurations are presented in Appendix.

5.1 White-box Robustness

Table 1 shows the experimental results of CIFAR10; Table 2 gives the results of CIFAR100 and Table 3 presents the results of ImageNet where acc_{nat} is the accuracy on natural data and acc_{AA} is the robust accuracy against AA attack. The orders are given by their robustness accuracy acc_{AA}. The numbers in # are the original rankings of other methods in Croce et al. (2021), and the items with $*$ are our results. We also put the results of TRADES in the bottom for comparisons.

As can be seen, when TRADES is combined with LTD, its robustness increases from 53.08% to 55.09%. Furthermore, when AWP is integrated with LTD, its robust accuracy is also improved from 56.17% to 56.90% using WRN-34-10. If WRN-34-20 is used with AWP and LTD, the robustness can reach 58.19%, which is the best result among all. Combing AWP with LTD improves robust accuracy for CIFAR100 from 28.86% to 31.13% using WRN-34-10, which is the best result without extra data, and the model size is smaller than others competitors. Similarly, the robustness is improved from 38.14% to 42.08% for ImageNet. The experimental results shows that LTD can improve robustness efficiently. Comparing to the original

Table 1: Competitors from Robustbench on CIFAR10 (Croce et al., 2021)

#	paper	architecture	acc_{nat}[%]	acc_{AA}[%]
*	Wu et al. (2020) + LTD	WRN-34-20	86.28	58.19
1	Addepalli et al. (2021)	WRN-34-10	85.32	58.04
2	Gowal et al. (2020)	WRN-70-16	85.29	57.20
*	AWP + LTD	WRN-34-10	85.21	56.90
3	Gowal et al. (2020)	WRN-34-20	85.64	56.86
4	AWP (Wu et al., 2020)	WRN-34-10	85.36	56.17
*	TRADES + LTD	WRN-34-10	85.63	55.09
5	Pang et al. (2021)	WRN-34-20	86.43	54.39
6	Pang et al. (2021)	WRN-34-10	85.49	53.94
7	Pang et al. (2020)	WRN-34-20	85.14	53.74
8	Cui et al. (2021)	WRN-34-20	88.70	53.57
9	Zhang et al. (2020a)	WRB-34-10	84.52	53.51
-	TRADES (Zhang et al., 2019b)	WRN-34-10	84.92	53.08

Table 2: Competitors from Robustbench on CIFAR100 (Croce et al., 2021)

#	paper	architecture	acc_{nat}[%]	acc_{AA}[%]
*	Wu et al. (2020) + LTD	WRN-34-10	64.32	31.13
1	Cui et al. (2021)	WRN-34-20	62.55	30.20
2	Gowal et al. (2020)	WRN-70-16	60.86	30.03
3	Wu et al. (2020)	WRN-34-10	60.38	28.86

models, the major difference comes from the used labels. The original models, AWP or TRADES, uses one-hot vectors as their labels, and LTD replaces them with soft labels generated from the teacher model.

LTD improves the robustness significantly for ImageNet data set. The result is consistent with our assumption described in Section 3.2. The data set contain a lot of ambiguous images and images with multiple objects. The Soft label is a better representation for those examples.

5.2 Gradient Masking Verification

Gradient masking issues can be evaluated by the efficiency of gradients. Auto-Attack (AA) (Croce & Hein, 2020) has a sequence of attacks, including three white-box attacks and one black-box attack, to verify if a proposed defensive method has the gradient masking problem. First, AA attack removes misclassified examples and applies an adaptive PGD attack with SCE loss (APGD-CE) on the remaining examples. Because of the imbalance gradient issues, some adversarial examples cannot be found by APGD-CE. AA attack applies the target version of APGD-CE (APGD-T) to find those adversarial examples by all probable loss directions. Next, target version of FAB attack (FAB-T), a variety of PGD attack with different loss, is

Table 3: Competitors from Robustbench on ImageNet (Croce et al., 2021)

#	paper	architecture	acc_{nat}[%]	acc_{AA}[%]
*	LTD	WRN-50-2	68.10	42.08
1	Salman et al. (2020)	WRN-50-2	68.46	38.14
*	LTD	ResNet-50	62.40	36.82
2	Salman et al. (2020)	ResNet-50	64.02	34.96
3	Engstrom et al. (2019)	ResNet-50	62.56	29.22
4	Wong et al. (2020)	ResNet-50	55.62	26.24
5	Salman et al. (2020)	ResNet-18	52.92	25.32

Table 4: Gradient masking evaluation

acc_{nat}[%]	$\text{acc}_{\text{APGD-CE}}$[%]	$\text{acc}_{\text{AGPD-T}}$[%]	$\text{acc}_{\text{FAB-T}}$[%]	$\text{acc}_{\text{square}}$[%]
85.63%	58.37%	55.09%	55.09%	55.09%

Table 5: Ablation study on BatchNorm

	acc_{nat}[%]	acc_{AA}[%]
adv-nat	85.89	54.95
nat-adv	85.63	55.09
nat	10.0	*
adv	10.0	*

applied on the remaining examples to verify that the model can defend against unseen loss. Finally, square attack, a score-based black box attack, is used to detect the gradient masking issue

We applied AA to the model trained by TRADES+LTD on CIFAR10. The results are shown in Table 4. As can be seen, the accuracy under a sequence of white-box attacks ($\text{acc}_{\text{FAB-T}}$) becomes 55.09%, which is the same as the score $\text{acc}_{\text{square}}$, 55.09%. This means the square attack cannot produce any successful adversarial example from the remaining data attacked by APGD-CE, AGPD-T and FAB-T. To sum up, LTD does not have gradient masking problems.

5.3 Ablation Studies

5.3.1 Inconsistent BatchNorm

This experiment investigated how BN updating ordering affects the robustness, as mentioned in Section 3.5. Table 5 shows the results using TRADES+LTD configuration on CIFAR10, in which adv-nat means BN is updated by the adversarial examples first; nat-adv means to update BN with natural data first; nat means using natural data only; and adv means using adversarial data only. As can be seen, if we only used one kind of data to update BN, including nat and adv, the results are poor. However, if we employed both data, the accuracy can be maintained. The updated ordering does not have a significant effect on the robust accuracy. We observed the same results in AWP+LTD configuration as well.

The failure of convergence for nat and adv trials is the logits of different samples are inconsistent when μ and σ^2 are updated by one of samples. Consequently, it makes the gradient of KLD in (28) unstable. Although we can reduce the importance of the KLD by using smaller λ, the gradient is still fluctuating and we cannot obtain a robust model. This result verifies the hypothesis on the mismatching distribution for natural data and adversarial examples (Xie et al., 2020). Therefore, minimizing the multi-objective loss in TRADE-based methods requires natural data and adversarial examples for updating BN's information.

5.3.2 Temperature

As mentioned in Section 4.3, the best temperature is selected in two steps. We used the model WRN-34-10 and CIFAR10 data set, and trained the model using TRADES+LTD with different temperatures in the teacher model to verify the importance of temperature selection. The results are shown in Table 6. We only searched the temperature τ_{best} in range $[1.0, 50.0]$, because when τ_{max} is 50.0, its natural accuracy is 86.63% which is too low to accept.

The experimental results showed that τ_{best} is 5.0 and the robust accuracy decreases when the temperature increases because the irrelevant classes receive partial probability from the target class which violates our assumption. The gradient masking occurs when the temperature is higher than 15.0, because the temperature increases the probability of each class except of the target class, the adversarial examples can be found by very tiny perturbation or lightweight attacks. This situation is similar to catastrophic overfitting (Madry et al., 2018) whose robust accuracy in the training phase is almost 100% but it cannot defend against adaptive-step attacks or stronger attacks.

Table 6: Ablation study on temperature

	$\mathrm{acc}_{\mathrm{nat}}$[%]	$\mathrm{acc}_{\mathrm{AA}}$[%]
TRADES	84.92	53.08
T=1.0	84.51	54.38
T=2.0	84.96	54.90
T=3.0	85.48	55.03
T=5.0	86.20	55.09
T=8.0	85.23	54.63
T=10.0	84.72	53.56
T=12.0	77.45	43.85
T=15.0	94.72	0.00
T=20.0	94.39	0.00
T=50.0	86.63	*

5.4 Discussion

In spirit, our proposed training framework, replacing the one-hot label with the soft label, is more general than the original training framework. This approach is suitable for the classification problem which violates closed-world assumption, or clean and big data assumption. ImageNet is an obvious example. Estimating the oracle distribution is an crucial problem. LTD is not the only one solution to predict the oracle distribution but it is the first method to fix the gradient masking issue.

On the other hand, some training frameworks which introduced additional examples from an external dataset are excluded from our approach. The major reason that the distribution is shifted entirely. A future work is to design another methods to estimate a mixture of the oracle distributions.

6 Conclusion

One-hot label was a commonly used representation for the classification problem. In this paper, we presented that one-hot label is an imprecise representation and one of the vulnerabilities of DNNs is from the ambiguous examples. We also investigated inconsistent BN issue and showed the distribution of adversarial examples and natural data are different. To get a better label representation, the one-hot label is replaced with soft label using a modified knowledge distillation framework with a properly low temperature. This modified label can be integrated into existing works without any cost. Comparing with knowledge distillation, the gradient masking does not exist in our approach. Experimental results showed that our approach combined with AWP achieves about 58.19% and 31.13% robust accuracy on CIFAR10 and CIFAR100 respectively and 42.08% robust accuracy on ImageNet. We believe that our results provide some insights into how label annotations affect the robustness. Designing a better label representation for real-wold scenarios is an unexplored issue.

References

Sravanti Addepalli, Samyak Jain, Gaurang Sriramanan, Shivangi Khare, and Venkatesh Babu Radhakrishnan. Towards achieving adversarial robustness beyond perceptual limits. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL `https://openreview.net/forum?id=SHB_znlW5G7`.

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Advances in Neural Information Processing Systems*, 32, 2019.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*, 2019.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283. PMLR, 2018.

Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.

Tom B Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

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

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7. IEEE, 2018.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

Erh-Chung Chen and Che-Rung Lee. Towards fast and robust adversarial training for image classification. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

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*, pp. 15–26, 2017a.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017b.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=SSKZPJCt7B`.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15721–15730, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

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

Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*, 2020.

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.

Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pp. 11137–11147, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL `http://arxiv.org/abs/1503.02531`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Linxi Jiang, Xingjun Ma, Zejia Weng, James Bailey, and Yu-Gang Jiang. Imbalanced gradients: A new cause of overestimated adversarial robustness. *arXiv preprint arXiv:2006.13726*, 2020.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

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

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=BJm4T4Kgx`.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

Hyun Kwon and Sanghyun Lee. Ensemble transfer attack targeting text classification systems. *Computers & Security*, 117:102695, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=r1X3g2_xl`.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. *Advances in Neural Information Processing Systems*, 33:7779–7792, 2020.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Xb8xvrtB8Ce`.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pp. 582–597. IEEE, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems*, pp. 3358–3369, 2019.

Cecilia Summers and Michael J Dinneen. Four things everyone should know to improve batch normalization. *arXiv preprint arXiv:1906.03548*, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6199`.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning*, pp. 9625–9635. PMLR, 2020.

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pp. 5025–5034. PMLR, 2018.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BJx040EFvH`.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020.

Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *arXiv preprint arXiv:1906.03787*, 2019.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 819–828, 2020.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2041–2055, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith (eds.), *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87. URL `https://dx.doi.org/10.5244/C.30.87`.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *Advances in Neural Information Processing Systems*, pp. 227–238, 2019a.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019b.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pp. 11278–11287. PMLR, 2020a.

Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y Suen. Towards robust pattern recognition: A review. *Proceedings of the IEEE*, 108(6):894–922, 2020b.

Alon Zolfi, Moshe Kravchik, Yuval Elovici, and Asaf Shabtai. The translucent patch: A physical and universal attack on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15232–15241, 2021.

A Experiment Configurations

We implement our algorithm using CUDA 11.3, cuDNN 8.4 and PyTorch 1.11.0 framework with mixed precision.

A.1 Dataset

CIFAR10 and CIFAR100 contain 50,000 training images and 10,000 testing images. The values of all images value are normalized in range in $[0, 1]$. Each image in training set are cropped randomly with padding 4 pixels on each border and applied random horizontal flip during training procedure.

ImageNet consists of about 1,300,00 training images collected from real world and each image's dimensions are vary. We apply random resized crop with 224x224 during the training phase and center crop with 224x224 in inference phase. Since we assume that adversarial examples and natural images may have different statistical information, the commonly used normalization by subtracting the mean and dividing by standard deviation may not be suitable for this case. Instead, each pixel is normalized in range in $[0, 1]$.

A.2 Implementations

Network architecture is Wide Residual Net (WRN) (Zagoruyko & Komodakis, 2016) with width 10 and depth 34 for CIFAR10 and CIFAR100 and the network architecture is ResNet-50 for ImageNet. Both teacher model and target model use the same architecture and configurations. The teacher models are trained from scratch using normal training with one-hot labels, and image transformation is used as mentioned in A.1. The natural accuracy of teacher's models are 94.5% or above and 77% or above for CIFAR10 and CIFAR100 respectively.

In CFIAR10 and CIFAR100 experiments, the target models are trained for 120 epochs. The initial learning rate is 0.1 and divided by 10 at the end of 80-th and 100th epoch. The optimizer is SGD with a momentum of 0.9; weight decay to 0.0005 and Nesterov is enabled. Other configurations follow default settings. The adversarial examples are generated by PGD-8 for TRADES (Zhang et al., 2019b) and AWP (Wu et al., 2020).

In ImageNet experiment, the initial learning rate is 0.1 and divided by 10 at the end of 50-th and 90th epoch. The optimizer is SGD with a momentum of 0.9; weight decay to 0.0001; Nesterov is enabled and batch size is 320. The training is completed at 90-th epochs. The adversarial examples are generated by PGD-6.

A.3 Metric

We measure the robust accuracy under the white-box environment against the AutoAttack (AA) (Croce & Hein, 2020) with default configuration. For CIFAR10 and CIFAR100, the baseline is $\epsilon = 8/255$ in L_∞ norm. For ImageNet, Robustbench (Croce et al., 2021) evaluates the robustness using fixed 5,000 images from validation set within $\epsilon = 4/255$ in L_∞ norm.