Calibrated Adversarial Training

Tianjin Huang[1] T.HUANG@TUE.NL
Vlado Menkovski[1] V.MENKOVSKI@TUE.NL
Yulong Pei[1] Y.PEI.1@TUE.NL
Mykola Pechenizkiy[1,2] M.PECHENIZKIY@TUE.NL
[1] *Department of Mathematics and Computer Science, Eindhoven University of Technology, 5600 MB Eindhoven, the Netherland*
[2] *Faculty of Information Technology, University of Jyväskylä, 40100 Jyväskylä, Finland*

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Adversarial training is an approach of increasing the robustness of models to adversarial attacks by including adversarial examples in the training set. One major challenge of producing adversarial examples is to contain sufficient perturbation in the example to flip the model's output while not making severe changes in the example's semantical content. Exuberant change in the semantical content could also change the true label of the example. Adding such examples to the training set results in adverse effects. In this paper, we present the Calibrated Adversarial Training, a method that reduces the adverse effects of semantic perturbations in adversarial training. The method produces pixel-level adaptations to the perturbations based on novel calibrated robust error. We provide theoretical analysis on the calibrated robust error and derive an upper bound for it. Our empirical results show a superior performance of the Calibrated Adversarial Training over a number of public datasets.

Keywords: Adversarial training; Adversarial examples; Generalization

1. Introduction

Despite the impressive success in multiple tasks, e.g. image classification Krizhevsky and Hinton (2012); He et al. (2016), object detection Girshick et al. (2014), semantic segmentation Long et al. (2015), deep neural networks (DNNs) are vulnerable to adversarial examples. In other words, carefully constructed small perturbations of the input can change the prediction of the model drastically Szegedy et al. (2013); Goodfellow et al. (2014). Furthermore, these adversarial examples have been shown high transferability, which greatly threat the security of DNN models Xie et al. (2019); Huang et al. (2021). This vulnerability of DNNs prohibits their adoption in applications with high risk such as autonomous driving, face recognition, medical image diagnosis.

In response to the vulnerability of DNNs, various defense methods have been proposed. These methods can be roughly separated into two categories: 1) certified defense, and 2) empirical defense. Certified defense tries to learn provable robustness against ϵ-ball bounded perturbations Cohen et al. (2019); Wong and Kolter (2018). Empirical defense refers to heuristic methods, including augmenting training data Madry et al. (2017) (e.g. adversarial training), regularization Moosavi-Dezfooli et al. (2018); Jakubovitz and Giryes

(2018), and inspirations from biology Tadros et al. (2019). Among all these defense methods, adversarial training has been the most commonly used defense against adversarial perturbations because of its simplicity and effectiveness Madry et al. (2017); Athalye et al. (2018). Standard adversarial training takes model training as a *minmax* optimization problem (Section 3.2) Madry et al. (2017). It trains a model based on on-the-fly generated adversarial examples X' bounded by uniformly ϵ-ball of input X (i.e. $\|X' - X\| \leq \epsilon$).

Although adversarial training is effective in achieving robustness, it suffers from two problems. Firstly, it achieves robustness with a severe sacrifice on natural accuracy, i.e. accuracy on natural images. Furthermore, the sacrifice will be enlarged rapidly when training with larger ϵ. Secondly, there is an underlying assumption that the on-the-fly generated adversarial examples within ϵ-ball are semantic unchanged. However, recently, Guo et al. (2018) and Sharma et al. (2019) show that adversarial examples bounded by ϵ-ball could be perceptible in some instances. Tramèr et al. (2020) and Jacobsen et al. (2019) find that there are “invariance adversarial examples” for some instances, where “invariance adversarial examples” refer to those adversarial examples that model’s prediction does not change while the true label changes. All these findings indicate that this assumption does not consistently hold, which hurts the performance of the model.

In this paper, we first analyze the limitation for adversarial training and point out that some on-the-fly generated adversarial examples may be harmful for training models. For instance, in Figure 1, the adversarial examples for x_1 may be harmful since it crosses the oracle classifier’s decision boundary. To address the limitation, we propose a calibrated adversarial training, which is derived on the upper bound of a new definition of robust error (Calibrated robust error). Calibrated adversarial training is composed of weighted cross-entropy loss for natural input and **KL** divergence for calibrated adversarial examples where calibrated adversarial examples are pixel-level adapted adversarial examples in order to reduce the adverse effect of adversarial examples with underlying semantic changes.

Specifically, our contributions are summarized as follows:

- Theoretically, we analyze the limitation for adversarial training, and propose a new definition of robust error: Calibrated robust error. Furthermore, we derive an upper bound for the calibrated robust error.
- We propose the calibrated adversarial training based on the upper bound of calibrated robust error, which can reduce the adverse effect of adversarial examples.
- Extensive experiments demonstrate that our method achieves the best performance on both natural and robust accuracy among baselines and provides a good trade-off between natural accuracy and robust accuracy. Furthermore, it enables training with larger perturbations, which yields higher adversarial robustness.

2. Related Work

Many papers have proposed their variants of adversarial training for achieving either more effective adversarial robustness or a better trade-off between adversarial robustness and natural accuracy. Generally, they can be categorized into two groups. The first group is to adapt a loss function for outer minimization or inner maximization. For instance, Kannan

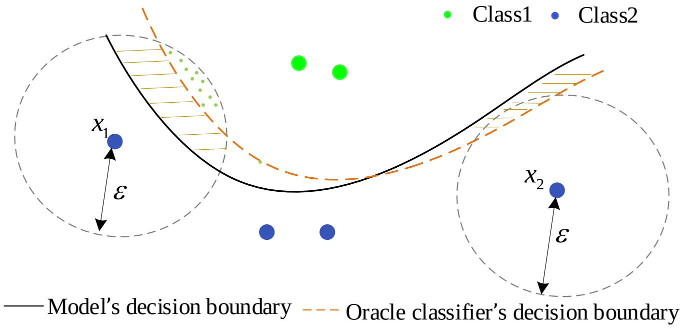

● Class1 ● Class2
x_1
ε
x_2
ε
Model's decision boundary
Oracle classifier's decision boundary

Figure 1: Illustration for neighborhoods of inputs and the decision boundaries.

et al. (2018) introduces a regularization term to enclose the distance between adversarial example and corresponding natural example. Zhang et al. (2019) proposes a theoretically principled trade-off method (Trades). Ding et al. (2019) proposes Max-Margin adversarial (MMA) training by maximizing the margin of a classifier. Wang et al. (2020) proposes MART by introducing an explicit regularization for misclassified examples. Wu et al. (2020) proposes Adversarial Weight Perturbation (AWP) for regularizing the weight loss landscape of adversarial training. Andriushchenko and Flammarion (2020); Huang et al. (2020) propose FGSM adversarial training + gradient-based regularization for achieving more effective adversarial robustness. The other group is to generate adversarial examples with adapted perturbation strength. Our work belongs to this group. Several recent works including Customized adversarial training Cheng et al. (2020), Currium adversarial training Cai et al. (2018), Dynamic adversarial training Wang et al. (2019), Instance adapted adversarial training Balaji et al. (2019), Adversarial training with early stopping (ATES) Sitawarin et al. (2020), Friendly adversarial training (FAT) Zhang et al. (2020), heuristically propose to adapt ϵ in instance-level for adversarial examples.

3. Preliminary

3.1. Notations

We denote capital letters such as X and Y to represent random variables and lower-case letters such as x and y to represent realization of random variables. We denote by $x \in \mathcal{X}$ the sample instance, and by $y \in \mathcal{Y}$ the label, where $\mathcal{X} \in \mathbb{R}^{m\times n}$ indicates the instance space. We use $\mathcal{B}(x,\epsilon)$ to represent the neighborhood of instance x: $\{x' : \|x'-x\|_p \le \epsilon\}$. We denote a neural network classifier as $f_\theta(x)$, the cross-entropy loss as $L(\cdot)$ and Kullback-Leibler divergence as $\mathbf{KL}(\cdot||\cdot)$. We denote $P(Y|X)$ as probability output after softmax and $P(Y=y|X)$ as the probability of $Y=y$. $sgn(\cdot)$ denotes the sign function and f_{oracle} denotes the oracle classifier that maps any inputs to correct labels.

3.2. Standard Adversarial Training

Given a set of instance $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We assume the data are sampled from an unknown distribution $(X,Y) \sim \mathcal{D}$. The standard adversarial training can be formally expressed as follows Madry et al. (2017):

$$\min_{\theta} \rho(\theta), \rho(\theta) = \mathbb{E}_{(X,Y)\sim D}[\max_{X' \in \mathcal{B}(X,\epsilon)} L(f_\theta(X'), Y)]. \tag{1}$$

3.3. Projected Gradient Descent (PGD)

Madry et al. (2017) utilizes projected gradient to generate perturbations. Formally, with the initialization $x^0 = x$, the perturbed data in t-th step x^t can be expressed as follows:

$$x^t = \Pi_{\mathcal{B}(x,\epsilon)}(x^{t-1} + \alpha \cdot sgn(\nabla_x L(f_\theta(x^{t-1}), y))), \tag{2}$$

where $\Pi_{\mathcal{B}(x,\epsilon)}$ denotes projecting perturbations into the set $\mathcal{B}(x,\epsilon)$, α is the step size and $t \in \{1, 2, ..., T\}$. We denote PGD attack with $T = 20$ as PGD-20 and $T = 100$ as PGD-100.

3.4. C&W attack

Given x, C&W attack Carlini and Wagner (2017) searches adversarial examples $\tilde{x}$ by optimizing the following objective function:

$$\|\tilde{x} - x\|_p + c \cdot h(\tilde{x}), \tag{3}$$

with

$$h(\tilde{x}) = \max(\max_{i \neq t} f_\theta(\tilde{x})_i - f_\theta(\tilde{x})_t, -k),$$

where $c > 0$ balances the two loss terms and k encourages adversarial examples to be classified as target t with larger confidence. This paper adopts C&W$_\infty$ attack and follows the implementation in Zhang et al. (2019); Cai et al. (2018) where they replace cross-entropy loss with $h(\tilde{x})$ in PGD attack.

3.5. Robust Error

We introduce the definition of robust error given by Zhang et al. (2019); Schmidt et al. (2018).

Definition 1 (Robust Error Zhang et al. (2019); Schmidt et al. (2018)) *Given a set of instance $x_1, ..., x_n \in \mathcal{X}$ and labels $y_1, ..., y_n \in \{-1, +1\}$. We assume that the data are sampled from an unknown distribution $(X, Y) \sim D$. The robust error of a classifer $f_\theta : \mathcal{X} \to \mathbf{R}$ is defined as: $\mathcal{R}_{rob}(f) := \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X' \in \mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')Y \leq 0\}$.*

4. Method

4.1. Analysis For Adversarial Training

Current adversarial training including its variants trains a model by minimizing robust error directly, which may hurt the performance of the model. Taking standard adversarial training as an example, it firstly approximates robust error by the inner maximization and then minimizes the approximated robust error. However, the on-the-fly adversarial examples generated by the inner maximization could be semantically damaged for some instances, e.g., in Figure 1, the semantical content of the adversarial examples for x_1 could be damaged since it crosses the decision boundary of f_θ. Therefore, the objective function (Eq. 1) can be decomposed into two terms according to the oracle classifier's decision boundary:

$$\min_\theta \rho(\theta),\ \rho(\theta) = \mathbb{E}_{(X,Y)\sim D}[\overbrace{\max_{\delta\in\mathcal{B}(X,\epsilon)} L(f_\theta(X+\delta),Y)\mathbf{1}\{f_{oracle}(X+\delta)=Y\}}^{(a)} + \overbrace{\max_{\delta\in\mathcal{B}(X,\epsilon)} L(f_\theta(X+\delta),Y)\mathbf{1}\{f_{oracle}(X+\delta)\neq Y\}}^{(b)}]. \tag{4}$$

The term (b) contributes to negative effects since the cross-entropy loss takes Y as the label of adversarial examples $X+\delta$ while the true label of $X+\delta$ is not Y. This term is equivalent to bringing noisy labels in training data, which also explains why a large perturbation magnitude in adversarial training setting will lead to a severe drop in natural accuracy of model.

To address this drawback, we propose calibrated robust error and build our defense method based on it.

4.2. Calibrated Robust Error

Definition 2 (Calibrated Robust Error (Ours)) *Given a set of instances $x_1,...,x_n \in \mathcal{X}$ and labels $y_1,...,y_n \in \{-1,+1\}$. We assume that the data are sampled from an unknown distribution $(X,Y)\sim D$. Assume there is an oracle classifier f_{oracle} that maps any input $x\in\mathbf{R}^d$ into its true label. The calibrated robust error of a classifier $f_\theta:\mathcal{X}\to\mathbf{R}$ is defined as: $\mathcal{R}_{cali}(f) := \mathbf{E}_{(X,Y)\sim D}\mathbf{1}\{\exists X'\in\mathcal{B}(X,\epsilon)\ s.t.\ f_\theta(X')f_{oracle}(X')\le 0\}$.*

Theorem 1 *Given a set of instance $x_1,...,x_n\in\mathcal{X}$, a classifier $f_\theta:\mathcal{X}\to\mathbf{R}$ and an oracle classifier f_{oracle} that maps any input $x\in\mathbf{R}^d$ into its true label and assumed the decision boundaries of f_θ and f_{oracle} are not overlapped [1], we have:*

$$\mathcal{R}_{rob}(f) \le \mathcal{R}_{cali}(f). \tag{5}$$

The proof can be found in Appendix A.1. From Theorem 1, it can be observed that minimizing robust error can be obtained by minimizing calibrated robust error.

4.3. Upper Bound on Calibrated Robust Error

In this section, we derive an upper bound on calibrated robust error.

Theorem 2 (Upper Bound) *Let ψ be a nondecreasing, continuous and convex function:$[0,1]\to[0,\infty]$. Let $\mathcal{R}_\phi(f) := \mathbf{E}\phi(f_\theta(X)Y)$ and $\mathcal{R}^*_\phi := \min_f \mathcal{R}_\phi(f)$, $\mathcal{R}(f) := \mathbf{E}(f_\theta(X)Y)$ and $\mathcal{R}^* = \min_f \mathcal{R}(f)$. For any non-negative loss function ϕ such that $\phi(0)\ge 1$, any measurable $f_\theta:\mathcal{X}\to\mathbf{R}$ and any probability distribution on $\mathcal{X}\times\{+1,-1\}$, we have:*

$$\mathcal{R}_{cali}(f) - \mathcal{R}^* \le \psi^{-1}(\mathcal{R}_\phi(f)-\mathcal{R}^*_\phi) + \mathbf{E}\Big[\max_{\substack{X'\in\mathbf{B}(X,\epsilon)\\ f_{oracle}(X')=Y}} \phi(f_\theta(X')Y)\Big]. \tag{6}$$

[1] Not overlapped denotes f_θ and f_{oracle} are not exactly the same.

The proof can be found in Appendix A.2. From the upper bound, it can be observed:

- If the oracle classifier's decision boundary crosses ϵ-ball, the upper bound is decided by the adversarial examples that close to the oracle classifier's decision boundary. If the oracle classifier's decision boundary does not cross ϵ-ball, the upper bound is decided by the adversarial examples that close to the boundary of ϵ-ball.
- Minimizing $\mathcal{R}_\phi(f) + \mathbf{E}\Big[\max_{\substack{X'\in\mathbf{B}(X,\epsilon)\\ f_{oracle}(X')=Y}} \phi(f(X')Y)\Big]$ can reduce the calibrated robust error. From Theorem 1, we can know that calibrated robust error is the upper bound of robust error. Therefore, it also reduces the robust error of the model.

4.4. Method for Defense

From the upper bound, we define the general objective function as follows:

$$\min_\theta \mathbf{E}\Big[\phi(f_\theta(X)Y) + \max_{\substack{X'\in\mathbf{B}(X,\epsilon)\\ f_{oracle}(X')=Y}} \phi(f_\theta(X')Y)\Big]. \tag{7}$$

The analysis for the difference between Eq. 7 and the general objective function Zhang et al. (2019) derived on the upper bound of robust error can be found in Appendix G.

The first term in Eq. 7 is the surrogate loss of misclassification on natural data, and we design it as cross-entropy weighted by $(1 - \mathit{predicted\ probability})$. Formally, it is expressed as:

$$\phi(f_\theta(X)Y) = L(f_\theta(X), Y)\cdot(1 - P(Y = y|X)). \tag{8}$$

The second term in Eq. 7 is the surrogate loss on adversarial examples. However, it can not be solved directly since f_{oracle} is unknown. Therefore, we propose a approximate solution with two steps. Firstly, we generate adversarial examples based on $\max_{X'\in\mathbf{B}(X,\epsilon)} \phi(f_\theta(X')Y)$. Secondly, we adapt the adversarial examples in pixel-level such that it approximately satisfies the constraint $f_{oracle}(X') = Y$ and we name the pixel-level adapted adversarial examples as **calibrated adversarial examples**. We rewrite $\max_{\substack{X'\in\mathbf{B}(X,\epsilon)\\ f_{oracle}(X')=Y}} \phi(f_\theta(X'), Y)$ as follows:

$$X' = X + \delta = argmax_{X'\in\mathbf{B}(X,\epsilon)}\phi(f_\theta(X')Y) \tag{9}$$

$$X'_{cali} = X + M\odot\delta,\ M\in\mathbb{R}^{m\times n}, M[i,j]\in(0,1), \tag{10}$$

where the $\odot$ denotes Hadamard product. From Eq. 9 and Eq. 10, we can see that calibrated adversarial examples X'_{cali} are obtained by adapting adversarial perturbations with soft mask M. Please refer to Section 5.2.1 and Appendix F for better understanding how does the mask M adapt the adversarial perturbations. δ can be solved by various adversarial attacks, e.g., PGD attack. Therefore, the problem of the inner maximization in Eq. 7 is transformed to find a proper soft mask M. Considering that soft mask M relies on input X and perturbation δ, we propose to learn it by a neural network g_φ, which is defined as follows:

$$M = g_\varphi(X, \delta). \tag{11}$$

Therefore, by replacing $\phi(f_\theta(X)Y)$ with Eq. 8 and X' with X'_{cali}, the objective function (Eq. 7) is transformed to follows:

$$\min_{\theta} \mathbf{E}_{(X,Y)\sim D}[L(f_\theta(X), Y) \cdot (1 - P(Y = y|X)) + \beta \cdot \phi(f_\theta(X'_{cali})Y)], \tag{12}$$

where X'_{cali} is solved by Eq. 10, and β is a hyper-parameter for balancing two terms. In practice, we follow Zhang et al. (2019); Wang et al. (2020) to use **KL** divergence for the surrogate loss $\phi(\cdot)$ in the outer minimization step. Thus, Eq. 12 can be reformulated as follows:

$$\min_{\theta} \mathbf{E}_{(X,Y)\sim D}[L(f_\theta(X), Y) \cdot (1 - P(Y = y|X)) + \beta \cdot \mathbf{KL}(P(Y|X'_{cali})||P(Y|X))]. \tag{13}$$

From Eq. 13, it can be observed that there are two main differences with other variants of adversarial training, e.g., AT, Trades, MART, etc. (See Appendix H for the detailed descriptions of their loss functions.):

- We use weighted cross-entropy loss instead of cross-entropy loss in order to make the loss function pay more attention to misclassified samples.
- The **KL** divergence is based on calibrated adversarial examples that reduce the adverse of some adversarial examples because calibrated adversarial examples are expected to be satisfied with $f_{oracle}(X'_{cali}) = Y$.

Finally we design the objective function for $g_\varphi(X, \delta)$ based on the two constraints: (1) X'_{cali} should be close to X' as far as possible in order to keep the inner maximization constraint in Eq. 7. (2) X'_{cali} is expected to be satisfied with $f_{oracle}(X'_{cali}) = Y$. Therefore, the objective function for $g_\varphi(X, \delta)$ is designed as follows:

$$\min_{\varphi} \mathbf{E}_{(X,Y)\sim D}[\mathbf{KL}(P(Y|X'_{cali})||P(Y|X')) + \beta_1 \cdot L(f_\theta(X'_{cali}), Y)], \tag{14}$$

where **KL** divergence term corresponds to the constraint (1) and cross-entropy loss $L(\cdot)$ corresponds to the constrain (2). β_1 is the hyper-parameter that controls the strength of the constraint (2).

We denote our method as calibrated adversarial training with PGD attack (CAT$_{cent}$) if X' is solved by PGD attack, calibrated adversarial training with C&W$_\infty$ attack (CAT$_{cw}$) if X' is solved by C&W$_\infty$ attack.

5. Experiments

In this section, we first conduct extensive experiments to assess the effectiveness of our approach in achieving natural accuracy and adversarial robustness, then we conduct experiments for understanding the proposed method.

5.1. Evaluation on Robustness and Natural Accuracy

5.1.1. Experimental settings

Two datasets are used in our experiments: MNIST LeCun (1998), and CIFAR-10 Krizhevsky et al. (2010). For MNIST, all defense models are built on four convolution layers and two linear layers. For CIFAR-10, we use PreAct ResNet-18 He et al. (2016) and WideResNet-34-10 Zagoruyko and Komodakis (2016) models. The architectures of auxiliary neural network g_φ for MNIST and CIFAR-10 can be found in Appendix C. Following previous researches Zhang et al. (2019); Wu et al. (2020), Robustness is measured by robust accuracy against white-box and black-box attacks. For white-box attack, we adopt PGD-20/100 attack Madry et al. (2017), FGSM attack Goodfellow et al. (2014) and C&W$_\infty$ Carlini and Wagner (2017). For black-box attack, we adopt a query-based attack: Square attack Andriushchenko et al. (2020).

Baselines Standard adversarial training and the three latest defense methods are considered: 1)TRADES Zhang et al. (2019), 2)MART Wang et al. (2020), 3)FAT Zhang et al. (2020). The detailed descriptions of baseline methods can be found in Appendix C.

Hyper-parameter settings During training phase, for MNIST, we set $T = 20$, $\epsilon = 0.3$, $\alpha = \epsilon/T$ for the training attack, and set $\beta = 1$, $\beta_1 = 0.3$ by default. For CIFAR-10, we set $T = 10$, $\alpha = 2/255$, $\epsilon = 8/255$ for the training attack and set $\beta = 5$ by default. We train models with $\beta_1 = 0.05, 0.1, 0.3$ respectively. For all baselines, they are trained using the official code that their authors provided and the hyper-parameters for them are set as per their original papers. More training details are introduced in Appendix C.

During test phase, for MNIST, we set $\epsilon = 0.3$ and $\alpha = 0.015$ for PGD attack. For CIFAR-10, we set $\epsilon = 8/255$ and $\alpha = 0.003$ for PGD attack. And we follow the implementation in Zhang et al. (2020) for C&W$_\infty$ attack where $\epsilon = 0.031$, $\alpha = 0.003$, $T = 30$ and $k = 50$.

Note that during the training process, we use the PGD attack with random start, i.e. adding random perturbation of $[-\epsilon, \epsilon]$ to the input before PGD perturbation. But for the test in our experiments, we use PGD attack without random start by default [2].

5.1.2. Evaluation on White-box Robustness

This section shows the evaluation on white-box attacks. All attacks have full access to model parameters. We first conduct an evaluation on a simple benchmark dataset: MNIST and then conduct an evaluation on a complex dataset: CIFAR-10.

MNIST Table 1 reports natural accuracy and robust accuracy under PGD-20 and PGD-100 respectively. For baselines, we do not include results from FAT and MART since they do not provide training code for MNIST. From Table 1, we can see that the proposed method can achieve higher natural accuracy and robust accuracy compared with standard adversarial training. Besides, we notice that with larger $\epsilon = 0.4$, adversarial robustness can be boosted further by our defense method.

CIFAR-10 We evaluate the performance based on two benchmark architectures, i.e., PreAct ResNet-18 and WideResNet-34-10. All defense models are tested under the same attack settings as described in Section 5.1.1 except for *FAT* on WideResNet-34-10 since this evaluation is copied from their paper directly where it is evaluated with $\epsilon = 0.031$ for PGD

[2] We find that PGD attack (restart=1) without random start is stronger than that with random start.

Table 1: Evaluation on MNIST. The value besides model name denotes the max perturbation magnitude used in the training phase. -: denotes the training loss fails in decrease. We report mean with 5 repeated runs and skip the standard deviations since they are small ($< 0.4\%$), which hardly affects the results.

Models	Natural	PGD-20	PGD-100
AT(0.3)	99.2	93.4	92.3
AT(0.4)	-	-	-
TRADES(0.3)*	99.3	94.9	92.9
TRADES(0.4)*	99.1	95.3	91.6
$\text{CAT}_{cent}(0.3)$	**99.3**	95.4	93.2
$\text{CAT}_{cent}(0.4)$	99.2	96.8	95.8
$\text{CAT}_{cw}(0.3)$	99.1	96.2	95.0
$\text{CAT}_{cw}(0.4)$	99.1	**97.1**	**96.2**

* Model is trained with $\beta = 1.0$.

Table 2: Evaluation on CIFAR-10 for PreAct ResNet-18 under white-box setting.

Models	Natural	FGSM	PGD-20	PGD-100	CW_∞	Avg
AT	83.0	57.3	52.9	51.9	50.9	59.2
TRADES($\beta : 6$)	82.8	57.6	52.8	51.7	50.9	59.2
MART ($\lambda : 5$)	83.0	**60.2**	53.9	52.3	49.9	59.9
FAT(β:6)	85.1	58.3	52.1	50.5	50.4	59.3
$\text{CAT}_{cent}(\beta_1 : 0.05)$	84.1 ± 0.3	59.5 ± 0.2	**55.6 ± 0.3**	**54.9±0.3**	50.8±0.2	**61.0**
$\text{CAT}_{cent}(\beta_1 : 0.1)$	85.9 ±0.2	58.5±0.3	54.1 ±0.1	53.4 ±0.06	50.44±0.3	60.4
$\text{CAT}_{cw}(\beta_1 : 0.05)$	84.2 ±0.3	58.9±0.2	55.3 ±0.4	54.5 ±0.5	**51.3±0.3**	60.9
$\text{CAT}_{cw}(\beta_1 : 0.1)$	85.1 ±0.5	58.9±0.3	54.9 ±0.5	54.1 ±0.4	51.2±0.1	60.8
$\text{CAT}_{cent}(\beta_1 : 0.3)$	88.0 ±0.2	57.0±0.4	51.1 ±0.5	49.9 ±0.4	47.8±0.2	58.8
$\text{CAT}_{cw}(\beta_1 : 0.3)$	**88.1** ±0.1	57.4±0.5	51.5 ±0.1	50.1 ±0.2	48.8±0.2	59.2

attack. Table 2 and Table 3 report natural accuracy and robust accuracy on the test set. “Avg” denotes the average of natural accuracy and all robust accuracy, and it indicates the overall performance on both natural accuracy and robust accuracy. For our method, we report mean + standard deviation with 5 repeated runs.

From Table 2 and Table 3, it can be seen that our method achieves the best performance on both natural accuracy and robust accuracy under all attacks except for FGSM among baselines. Moreover, with $\beta_1 = 0.3$, our method improves natural accuracy with a large margin while keeps comparable performance with baselines on robust accuracy. Besides, our method achieves high “Avg” value, which indicates our method has a good trade-off between natural accuracy and robust accuracy. Finally, we observe that the robustness achieved by our method has smaller accuracy under stronger attacks, i.e. PGD-100 and CW_∞, than weaker attacks, i.e. FGSM and PGD-20. It indicates that the robustness achieved by our method is not caused by “gradient masking” Athalye et al. (2018).

Experiments on CIFAR-100 can be found in Appendix D.

Table 3: Evaluation on CIFAR-10 for WideResNet-34-10 under white-box setting.

Models	Natural	FGSM	PGD-20	PGD-100	CW_∞	Avg
AT	86.1	61.8	56.1	55.8	54.2	62.8
TRADES($\beta : 6$)	84.9	60.9	56.2	55.1	54.5	62.3
MART ($\lambda : 5$)	83.6	61.6	57.2	56.1	53.7	62.5
FAT(β:6)	86.6±0.6	61.9±0.6	55.9±0.2	55.4±0.3	54.3±0.2	62.8
$CAT_{cent}(\beta_1 : 0.05)$	86.6±0.1	60.9 ± 0.1	57.7 ± 0.1	57.2 ±0.2	53.9 ±0.6	63.3
$CAT_{cent}(\beta_1 : 0.1)$	87.5±0.51	61.5 ±0.5	57.2 ±0.3	56.6 ±0.4	54.0±0.4	63.4
$CAT_{cw}(\beta_1 : 0.05)$	86.4±0.1	**62.7 ±0.2**	**59.7 ±0.1**	**58.7 ±0.3**	**56.0±0.1**	**64.7**
$CAT_{cw}(\beta_1 : 0.1)$	87.4±0.1	62.3 ±0.1	58.6 ±0.2	57.3 ±0.19	55.6±0.07	64.2
$CAT_{cent}(\beta_1 : 0.3)$	88.9 ± 0.4	59.8 ±0.6	54.8 ±0.7	53.9 ±0.6	51.6±0.2	61.8
$CAT_{cw}(\beta_1 : 0.3)$	**89.3±0.1**	60.8±0.27	55.1±0.3	53.2±0.5	52.6±0.4	62.2

5.1.3. Evaluation on Black-box Robustness

We conduct evaluation on black-box settings. We choose to use Square attack Andriushchenko et al. (2020) in our experiments. Square attack is a query-efficient black-box attack, which has been shown that it achieves white-box comparable performance and resists "gradient masking" Andriushchenko et al. (2020). In our experiments, we set hyper-parameters $n_{queries} = 5000$ and $eps = 8/255$ for Square attack. The experiments are carried out on CIFAR-10 test set based on PreAct ResNet-18 and WideResNet-34-10 architectures. Results are showed in Table 4. It can be seen that our method achieves the best accuracy among all baselines under square attack. Besides, by comparing Table 4 with Table 2 and Table 3, we can find that accuracy under black-box attack is lower than under white-box attack like PGD and CW_∞ attacks. It demonstrates that adversarial robustness achieved by our method is not due to "gradient masking " Athalye et al. (2018).

5.2. Understanding the Proposed Defense Method

5.2.1. Visualization of Soft Mask M

We visualize the learned soft mask M for further understanding calibrated adversarial examples. As showed in Figure 2, natural images are randomly selected from MNIST, and adversarial examples are generated by PGD-20 attack with $\epsilon = 0.4$. Soft masks and calibrated adversarial examples are generated accordingly. It can be observed that soft masks have high values on the background but have low values on the digit, which indicates that they try to reduce perturbations on the digit. Furthermore, by comparing calibrated adversarial examples with adversarial examples, we find that pixel values on digits for calibrated adversarial examples tend to be homogeneous, which is more consist with them on natural images. In other words, soft masks try to prevent adversarial examples from breaking semantic information that could impact the performance of the model.

5.2.2. Training with Larger Perturbation Bound

Our method adapts adversarial examples for mitigating the adverse effect, which enables a model trained with larger perturbations. To verify the performance, we conduct experiments

Table 4: Evaluation on CIFAR-10 for PreAct ResNet-18 and WideResNet-34-10 under black-box setting. -: Not Available.

Models	ResNet	WRN
AT	55.12	59.19
TRADES	54.85	59.0
MART	54.98	57.7
FAT	55.35	-
CAT$_{cent}$(β_1 : 0.05)	56.4±0.1	59.1±0.5
CAT$_{cent}$(β_1 : 0.1)	56.4±0.1	59.6±0.8
CAT$_{cw}$(β_1 : 0.05)	56.3 ±0.2	60.9±0.1
CAT$_{cw}$(β_1 : 0.1)	**56.5 ±0.1**	**60.9±0.2**

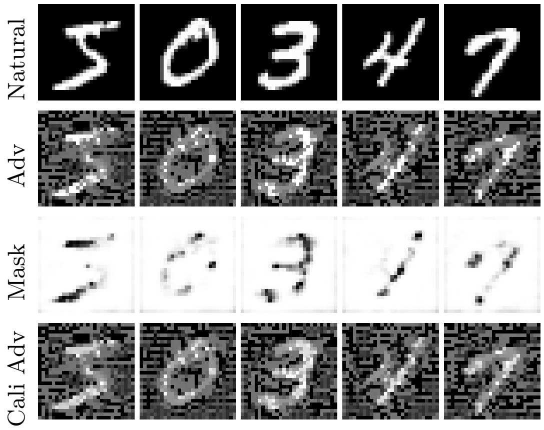
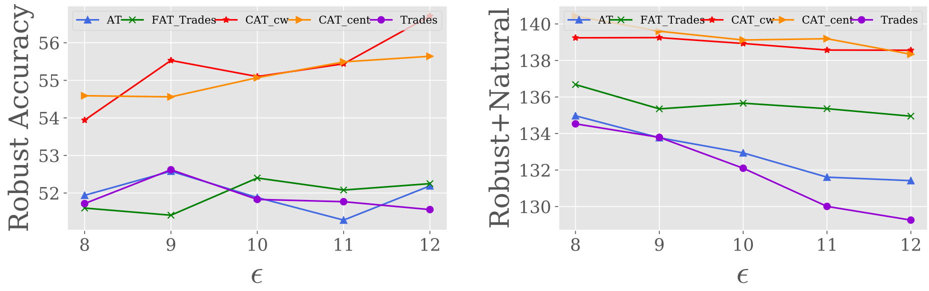


Figure 2: Visualization of soft mask M.

on PreAct ResNet-18 models trained with $\epsilon = 8, 9, 10, 11, 12$ respectively and test them on CIFAR-10 test set. Baselines are trained with their official codes. Results are showed in Figure 3. From Figure 3(a)subfigure, it can be observed that our method has a clearly increasing trend on robust accuracy with the increase of ϵ. From Figure 3(b)subfigure, we can see that the sum of robust accuracy and natural accuracy has a slightly decreasing trend for our method, indicating a trade-off between robust accuracy and natural accuracy. However, our method's descending grade is lower than Trades and AT, which also verifies that our method has a good trade-off between robust accuracy and natural accuracy.



(a) Robust

(b) Robust+Natural

Figure 3: Evaluation on models trained with larger ϵ. Robust accuracy is calculated by PGD-100 attack without random start.β_1 is fixed to 0.1 for CAT_{cw} and CAT_{cent}.

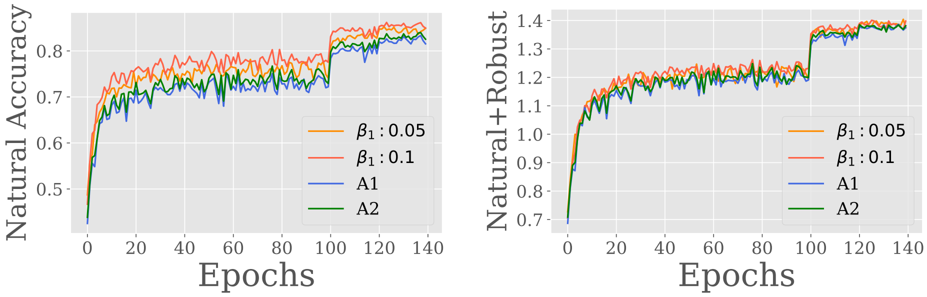
5.2.3. Ablation Study

We empirically verify the effect of weighted cross-entropy loss and soft mask M. Besides, we compare the effect of different loss functions selected in Eq. 9 for generating adversarial examples.

Effect of the weighted cross-entropy loss and mask M We remove M by replacing X'_{cali} with X' and remove $L(f_\theta(X), Y) \cdot (1 - P(Y = y|X))$ by replacing it with $L(f_\theta(X), Y)$. We train PreAct ResNet-18 models based on CAT$_{cent}$ with removing both weighted cross-entropy loss and M (marked as A1 model), and with removing M only (marked as A2 model).

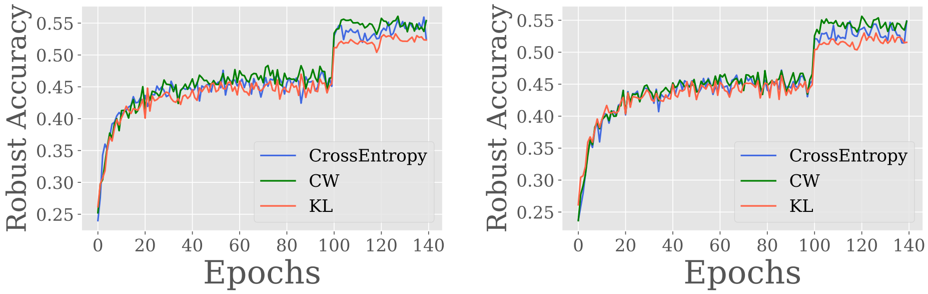
We plot natural accuracy and robust accuracy on CIFAR-10 test set. Robust accuracy is computed by PGD-10 with random start ($\alpha = 2/255, \epsilon = 8/255$). Results are reported in Figure 4. It can be observed that after removing soft mask M, there is a clearly decrease in natural accuracy and overall performance (natural+robust accuracy). Furthermore, after removing weighted cross-entropy loss, there is a slight decrease in natural accuracy.

Comparison of different loss functions There are many choices for the surrogate loss in Eq. 9 used to generate adversarial examples, e.g., cross-entropy loss, KL divergence used in Trades Zhang et al. (2019), CW$_\infty$ loss. Here we evaluate the effect of these three losses in our method. We plot robust accuracy on CIFAR-10 test set for $\beta_1 = 0.1$ and $\beta_1 = 0.05$ respectively, and robust accuracy is calculated by PGD-10 attack with random start ($\alpha = 2/255, \epsilon = 8/255$). The experiments are based on PreAct ResNet-18 model. Results are showed in Figure 5 and it can be seen that **KL** divergence is less effective in achieving robustness than cross-entropy loss and CW$_\infty$ loss for both $\beta_1 = 0.1$ and $\beta_1 = 0.05$ settings.



(a) Natural (b) Natural+Robust

Figure 4: The ablation Experiments. A1: Model trained by CAT$_{cent}$ with removing both soft mask M and $(1 - P(Y = y|X))$. A2: Model trained by CAT$_{cent}$ with removing soft mask M only. $\beta_1 : 0.1, 0.05$ denote models trained by CAT$_{cent}$ with setting $\beta_1 = 0.1, 0.05$ respectively.



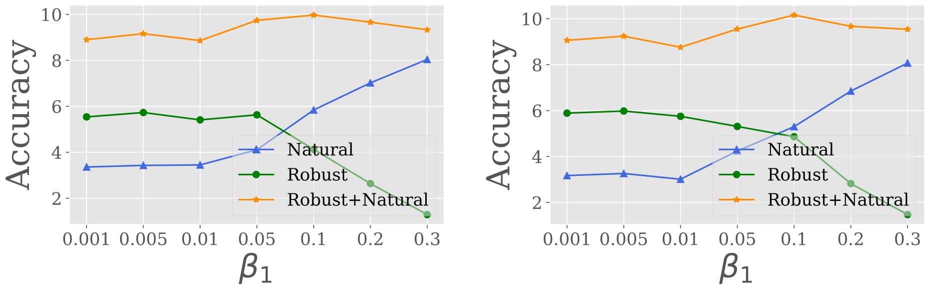
(a) $\beta_1 = 0.05$ (b) $\beta_1 = 0.1$

Figure 5: Comparison of different loss functions on achieving adversarial robustness.

5.2.4. Analysis for Hyper-parameter β_1

There are two hyper-parameters, β and β_1, in our method. β has the same effect as λ in MART Wang et al. (2020) and Trades Zhang et al. (2019). It controls the strength of the regularization for robustness. The analysis for β can be found in Appendix E. β_1 controls the strength that pushes calibrated adversarial examples to be the same class of the input X. In this section, we mainly show the effect of β_1 on robust accuracy and natural accuracy. We train models with β_1 varying from 0.001 to 0.3 based on PreAct ResNet-18 architecture. The robust accuracy is calculated on CIFAR-10 test set by PGD-20 attack without random start.

The trends are showed in Figure 6. The concrete values can be found in Table 8 (Appendix E). From Figure 6, it can be observed that when increasing the value of β_1, natural accuracy has remarkable growth. Meanwhile, PGD+Natural accuracy increases when β_1 is from 0.01 to 0.1, which implies that calibrated adversarial examples release the negative effect of adversarial examples to some degree. With continuously increase β_1, there is a large drop in robust accuracy. It is because a large β_1 will reduce adversarial perturbation strength. However, it can be observed that there is a good trade-off for large β_1 between natural accuracy and robust accuracy. For example, with $\beta_1 = 0.3$, CAT$_{cw}$ achieves 88.08 ± 0.07 for natural accuracy while keeps 51.46 ± 0.11 for robust accuracy, which is much better than the trade-off achieved by Trades Zhang et al. (2019) where natural accuracy is 87.91 and robust accuracy is 41.50 [3].



(a) CAT$_{cent}$ (b) CAT$_{cw}$

Figure 6: Impact of hyper-parameter β_1 on the performance of natural accuracy and robust accuracy. Note: The natural accuracy showed in the figure is ($natural\ accuracy - 80$) and the robust accuracy showed in the figure is ($robust\ accuracy - 50$).

6. Conclusion

In this paper we proposed a new definition of robust error, i.e. calibrated robust error for adversarial training. We derived an upper bound for it, and enabled a more effective way of adversarial training that we call calibrated adversarial training. Our extensive experiments demonstrate that the new method improves natural accuracy with a large margin, and achieves the best performance under both white-box and black-box attacks among all considered state-of-the-art approaches. Our method also has a good trade-off between natural accuracy and robust accuracy.

[3] Results are copied from Zhang et al. (2019)

References

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *NIPS*, 2020.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

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

Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

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

Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit Dhillon, and Cho-Jui Hsieh. Cat: Customized adversarial training for improved robustness. *arXiv preprint arXiv:2002.06789*, 2020.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2019.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014. ISBN 9781479951178. doi: 10.1109/CVPR.2014.81.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. dec 2014.

Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. *arXiv preprint arXiv:1809.08758*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Tianjin Huang, Vlado Menkovski, Yulong Pei, and Mykola Pechenizkiy. Bridging the performance gap between fgsm and pgd adversarial training. *arXiv preprint arXiv:2011.05157*, 2020.

Tianjin Huang, Vlado Menkovski, Yulong Pei, YuHao Wang, and Mykola Pechenizkiy. Direction-aggregated attack for transferable adversarial examples. *arXiv preprint arXiv:2104.09172*, 2021.

Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2019.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 514–529, 2018.

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

Alex Krizhevsky and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012. ISSN 10495258. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 5, 2010.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation ppt. In *CVPR 2015 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. ISBN 9781467369640. doi: 10.1109/CVPR.2015.7298965.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. jun 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. nov 2018.

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

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

Yash Sharma, Gavin Weiguang Ding, and Marcus Brubaker. On the effectiveness of low frequency perturbations. *arXiv preprint arXiv:1903.00073*, 2019.

Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Improving adversarial robustness through progressive hardening. *arXiv preprint arXiv:2003.09347*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. dec 2013.

Timothy Tadros, Giri Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. In *International Conference on Learning Representations*, 2019.

Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. *arXiv preprint arXiv:2002.04599*, 2020.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pages 6586–6595, 2019.

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*, 2020.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

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

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 09–15 Jun 2019.

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 Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11278–11287. PMLR, 13–18 Jul 2020.