Two Coupled Rejection Metrics Can Tell Adversarial Examples Apart

Anonymous CVPR submission

Paper ID 10212

Abstract

Correctly classifying adversarial examples is an essential but challenging requirement for safely deploying machine learning models. As reported in RobustBench, even the state-of-the-art adversarially trained models struggle to exceed 67% robust test accuracy on CIFAR-10, which is far from practical. A complementary way towards robustness is to introduce a rejection option, allowing the model to not return predictions on uncertain inputs, where confidence is a commonly used certainty proxy. Along with this routine, we find that confidence and a rectified confidence (R-Con) can form two coupled rejection metrics, which could provably distinguish wrongly classified inputs from correctly classified ones. This intriguing property sheds light on using coupling strategies to better detect and reject adversarial examples. We evaluate our rectified rejection (RR) module on CIFAR-10, CIFAR-10-C, and CIFAR-100 under several attacks including adaptive ones, and demonstrate that the RR module is compatible with different adversarial training frameworks on improving robustness, with little extra computation.

1. Introduction

The adversarial vulnerability of machine learning models has been widely studied because of its counter-intuitive behavior and the potential effect on safety-critical tasks [2, 16, 42]. Towards this end, many defenses have been proposed, but most of them can be evaded by adaptive attacks [1, 44]. Among the previous defenses, adversarial training (AT) is recognized as an effective defending approach [29,52]. Nonetheless, as reported in RobustBench [9], the state-of-the-art AT methods still struggle to exceed 67% robust test accuracy on CIFAR-10, even after exploiting extra data [17, 34, 38, 46], which is far from practical.

An improvement can be achieved by incorporating a rejection or detection module along with the adversarially trained classifier, which enables the model to refuse to make predictions for abnormal inputs [22, 24, 41]. Although previous rejectors trained via margin-based objectives or confidence calibration can capture some aspects of prediction certainty, they may overestimate the certainty, especially on wrongly classified samples. Furthermore, [43] argues that learning a robust rejector could suffer from a similar accuracy bottleneck as learning robust classifiers, which may be caused by data insufficiency [37] or poor generalization [48].

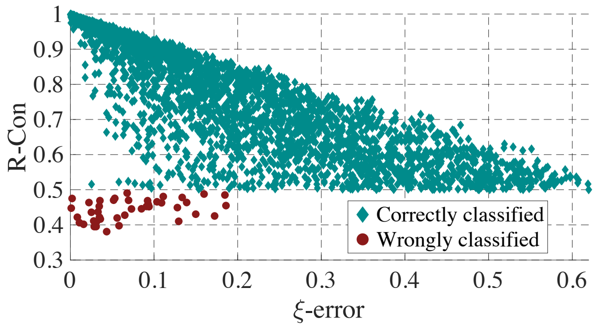


Figure 1. PGD-10 examples crafted against an adversarially trained ResNet-18 on the CIFAR-10 test set. As described in Theorem 1, these adversarial examples are first filtered by the confidence value at $\frac{1}{2-\xi}$ for each ξ. Namely, they pass if the predicted confidence is larger than $\frac{1}{2-\xi}$; otherwise rejected. Then among the remaining examples, the R-Con metric can provably separate correctly and wrongly classified inputs. In Fig. 3 we show that tuning the logits temperature τ can increase the number of remaining examples.

To solve these problems, we first observe that the *true* cross-entropy loss $-\log f_\theta(x)[y]$ reflects how well the classifier $f_\theta(x)$ is generalized on the input x [15], assuming that we can access its true label y. Thus, we propose to treat **true confidence (T-Con)** $f_\theta(x)[y]$, i.e., the predicted probability on the true label as a certainty oracle. Note that T-Con is different from the commonly used **confidence**, which is obtained by taking the maximum as $\max_l f_\theta(x)[l]$. As we shall see in Table 1, executing the rejection based on T-Con can largely increase the test accuracy under a given true positive rate for both standardly and adversarially trained models.

An instructive fact about T-Con is that *if we first threshold confidence by $\frac{1}{2}$, then T-Con can provably distinguish wrongly classified inputs from correctly classified ones*, as stated in Lemma 1. This inspires us to couple two connected metrics like confidence and T-Con to execute rejection options, instead of employing a single metric.

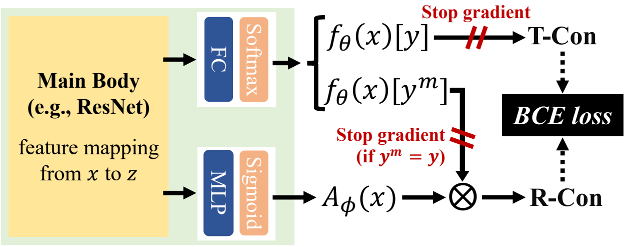


Figure 2. Construction of the objective $\mathcal{L}_{\text{RR}}$ in Eq. (4) for training the RR module, which is the binary cross-entropy (BCE) loss between T-Con and R-Con. The RR module shares a main backbone with the classifier, introducing little extra computation.

The property of T-Con is compelling, but its computation is unfortunately not realizable during inference due to the absence of the true label y. Thus we construct the **rectified confidence (R-Con)** to learn to predict T-Con, by rectifying confidence via an auxiliary function. We prove that if R-Con is trained to align with T-Con within ξ-error where $\xi \in [0, 1)$, then *a ξ-error R-Con rejector and a $\frac{1}{2-\xi}$ confidence rejector can be coupled to distinguish wrongly classified inputs from correctly classified ones*. This property generally holds as long as the learned R-Con rejector performs better than a random guess, as described in Section 4.2.

Technically, as illustrated in Fig. 2, we adopt a two-head structure to model the classifier and our rectified rejection (RR) module, while adversarially training them in an end-to-end manner. Our rejection module is learned by minimizing an extra BCE loss between T-Con and R-Con. The design of a shared main body saves computation and memory costs. Stopping gradients on the confidence $f_\theta(x)[y^m]$ when the predicted label $y^m = y$ can avoid focusing on easy examples and keep the optimal solution of classifier unbiased.

Empirically, we evaluate the performance of our RR module on CIFAR-10, CIFAR-10-C, and CIFAR-100 [21, 23] with extensive experiments. In Section 4, we verify the provable rejection options obtained by coupling confidence and R-Con. To fairly compare with previous baselines, we also use R-Con alone as the rejector, and report both the accuracy for a given true positive rate and the ROC-AUC scores in Section 6. We perform ablation studies on the construction of R-Con, and design adaptive attacks to evade our RR module. Our results demonstrate that the RR module is well compatible with different AT frameworks, and can consistently facilitate the returned predictions to achieve higher robust accuracy under several attacks and threat models, with little computational burden, and is easy to implement.

2. Related work

In the literature of standard training, [8] first propose to *jointly* learn the classifier and rejection module, which is later extended to deep networks [12, 13]. Recently, [24] and [22] jointly learn the rejection option during adversarial training (AT) via margin-based objectives, whereas they abandon the ready-made information from the confidence that is shown to be a simple but good solution of rejection for PGD-AT [47]. On the other hand, [41] propose confidence-calibrated AT (CCAT) by adaptive label smoothing, leading to preciser rejection on unseen attacks. However, this calibration acts on the true classes in training, while the confidences obtained by the maximal operation during inference may not follow the calibrated property, especially on the misclassified inputs. In contrast, we exploit true confidence (T-Con) as a certainty oracle, and propose to learn T-Con by rectifying confidence. Our RR module is also compatible with CCAT, where R-Con is trained to be aligned with the calibrated T-Con. [7] used similarly rectified confidence (R-Con) for failure prediction, while we prove that R-Con and confidence can be coupled to provide provable separability in the adversarial setting.

Table 1. Test accuracy (%) on all examples and under given true positive rate of 95% (TPR-95). The model is ResNet-18 that standardly or adversarially trained on CIFAR-10.

	Inputs	All	TPR-95	
			Confidence	T-Con
Standard	Clean	95.36	98.40	**100.0**
	PGD-10	0.22	0.18	**100.0**
Adversarial	Clean	82.67	87.39	**96.55**
	PGD-10	53.58	57.23	**88.75**
Availability			✓	✗

In Appendix B, we introduce more backgrounds on the adversarial training and detection methods, where several representative ones are involved as our baselines.

3. Classification with a rejection option

Consider a data pair (x, y), with $x \in \mathbb{R}^d$ as the input and y as the true label. We refer to $f_\theta(x) : \mathbb{R}^d \to \Delta^L$ as a classifier parameterized by θ, where Δ^L is the probability simplex of L classes. Following [13], a classifier with a rejection module $\mathcal{M}$ can be formulated as

$$(f_\theta, \mathcal{M})(x) \triangleq \begin{cases} f_\theta(x), & \text{if } \mathcal{M}(x) \geq t; \\ \text{don't know}, & \text{if } \mathcal{M}(x) < t, \end{cases} \tag{1}$$

where t is a threshold, and $\mathcal{M}(x)$ is a certainty proxy computed by auxiliary models or statistics.

What to reject? The design of $\mathcal{M}$ is principally decided by what kinds of inputs we intend to reject. In the adversarial setting, most of the previous detection methods aim to reject adversarial examples, which are usually misclassified by standardly trained models (STMs) [6]. In this case, the misclassified and adversarial characters are considered as associated by default. However, for adversarially trained models (ATMs) on CIFAR-10, more than 50% adversarial inputs are correctly classified [10]. Hence, it would be more reasonable to execute rejection depending on whether the input will be misclassified rather than adversarial.

3.1. True confidence (T-Con) as a certainty oracle

To reject misclassified inputs, there are many ready-made choices for computing $\mathcal{M}(x)$. We use $f_\theta(x)[l]$ to represent returned probability on the l-th class, and denote the predicted label as $y^m = \arg\max_l f_\theta(x)[l]$, where $f_\theta(x)[y^m]$ is usually termed as **confidence** [15]. In standard settings, confidence is shown to be one of the best certainty proxies [12], which is often used by practitioners. But the confidence returned by STMs can be adversarially fooled [30].

Different from confidence which is obtained by taking the maximum as $\max_l f_\theta(x)[l]$, we introduce **true confidence (T-Con)** defined as $f_\theta(x)[y]$, i.e., the returned probability on the true label y. When classifiers are trained by minimizing cross-entropy loss $\mathbb{E}[-\log f_\theta(x)[y]]$, the value of $-\log f_\theta(x)[y]$ can better reflect how well the model is generalized on a new input x during inference, compared to its empirical approximation $-\log f_\theta(x)[y^m]$, especially when x is misclassified (i.e., $y^m \neq y$).

As empirically studied in Table 1, we train classifiers on CIFAR-10 and evaluate the effects of confidence and T-Con as the rejection metric $\mathcal{M}$, respectively. We report the accuracy without rejection ('All'), and the accuracy when fixing the rejection threshold at 95% true positive rate ('TPR-95') w.r.t. confidence or T-Con[1], i.e., at most 5% correctly classified examples are rejected. As seen, thresholding on T-Con can vastly improve the accuracy.

To explain the results, note that STMs tend to return high confidences, e.g., 0.95 on both clean and adversarial inputs [31], then if an input x is correctly classified, there is T-Con$(x) = 0.95$; otherwise T-Con$(x) < 1 - 0.95 = 0.05$. Thus it is reasonable to see that thresholding on T-Con for STMs can lead to TPR-95 accuracy of 100% as in Table 1. As a result, we treat T-Con as a certainty oracle, and confidence is actually a proxy of T-Con in inference when we cannot access the true label y. In Section 4, we propose a better proxy R-Con to approximate T-Con.

3.2. Coupling confidence and T-Con

Instead of using a single metric, we observe a fact that properly coupling confidence and T-Con can provably separate wrongly and correctly classified inputs, as stated below:

Lemma 1. *(Separability) Given the classifier f_θ, $\forall x_1, x_2$ with confidences larger than $\frac{1}{2}$, i.e.,*

$$f_\theta(x_1)[y_1^m] > \frac{1}{2}, \text{ and } f_\theta(x_2)[y_2^m] > \frac{1}{2}. \tag{2}$$

If x_1 is correctly classified as $y_1^m = y_1$, while x_2 is wrongly classified as $y_2^m \neq y_2$, then T-Con$(x_1) > \frac{1}{2} >$ T-Con(x_2).

Proof. Since x_1 is correctly classified, i.e., $y_1^m = y_1$, we have $f_\theta(x_1)[y_1] = f_\theta(x_1)[y_1^m] > \frac{1}{2}$. On the other hand, since x_2 is wrongly classified, i.e., $y_2^m \neq y_2$, we have $f_\theta(x_2)[y_2] \leq 1 - f_\theta(x_2)[y_2^m] < \frac{1}{2}$. Thus we have T-Con$(x_1) > \frac{1}{2} >$ T-Con(x_2). □

Intuitively, Lemma 1 indicates that if we first threshold confidence to be larger than $\frac{1}{2}$, then for any x that pass the confidence rejector, there is T-Con$(x) < \frac{1}{2}$ if x is misclassified; otherwise T-Con$(x) > \frac{1}{2}$. Note that there is no constraint on how the misclassification is caused, i.e., wrongly classified inputs can be adversarial examples, generally corrupted ones, or just the clean samples.

4. Learning T-Con via rectifying confidence

In this section, we describe learning T-Con via rectifying confidence, and formally present the provable separability and the learning difficulty of rectified confidence. Proofs are provided in Appendix A.

4.1. Construction of rectified confidence (R-Con)

When the input x is correctly classified by f_θ, i.e., $y^m = y$, the values of confidence and T-Con become aligned. This inspires us to learn T-Con by rectifying confidence, instead of modeling T-Con from scratch, which facilitates optimization and is conducive to preventing the classifier and the rejector from competing for model capacity. Namely, we introduce an auxiliary function $A_\phi(x) \in [0, 1]$, parameterized by ϕ, and construct the **rectified confidence (R-Con)** as[2]

$$\text{R-Con}(x) = f_\theta(x)[y^m] \cdot A_\phi(x). \tag{3}$$

In training, we encourage R-Con to be aligned with T-Con. This can be achieved by minimizing the binary cross-entropy (BCE) loss (detailed implementation seen in Appendix C.1). Other alternatives like margin-based objectives [22] or mean square error can also be applied. The training objective of our rectified rejection (RR) module can be written as

$$\mathcal{L}_{\text{RR}}(x, y; \theta, \phi) = \text{BCE}\left(f_\theta(x)[y^m] \cdot A_\phi(x) \parallel f_\theta(x)[y]\right), \tag{4}$$

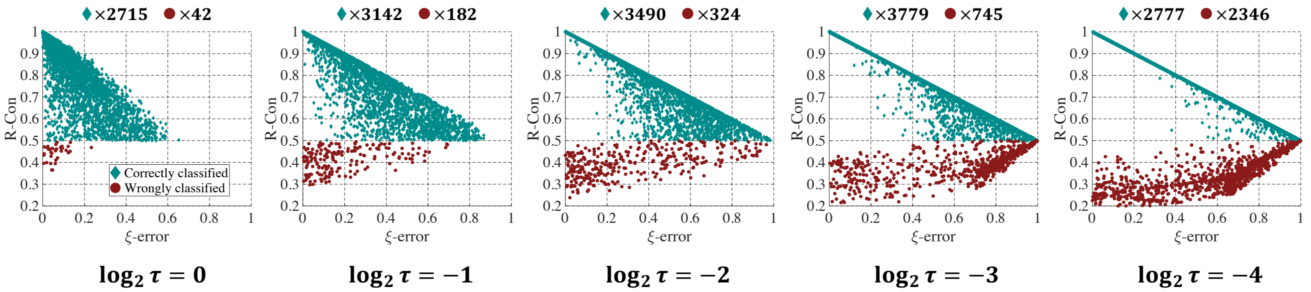
where the optimal solution of minimizing $\mathcal{L}_{\text{RR}}$ w.r.t. ϕ is $A^*_\phi(x) = \frac{f_\theta(x)[y]}{f_\theta(x)[y^m]}$. The auxiliary function $A_\phi(x)$ can be jointly learned with the classifier $f_\theta(x)$ by optimizing

$$\min_{\theta,\phi} \mathbb{E}_{p(x,y)}\Big[\underbrace{\mathcal{L}_{\text{T}}(x^*, y; \theta)}_{\text{classification}} + \lambda \cdot \underbrace{\mathcal{L}_{\text{RR}}(x^*, y; \theta, \phi)}_{\text{rectified rejection}}\Big],$$
$$\text{where } x^* = \mathop{\arg\max}_{x' \in B(x)} \mathcal{L}_{\text{A}}(x', y; \theta). \tag{5}$$

Here λ is a hyperparameter, $B(x)$ is a set of allowed points around x (e.g., a ball of $\|x' - x\|_p \leq \epsilon$), $\mathcal{L}_{\text{T}}$ and $\mathcal{L}_{\text{A}}$ are the training and adversarial objectives for a certain AT method,

[1] Here we assume that the true labels are known when computing T-Con.

[2] It is also feasible to use R-Con$(x) = f_\theta(x)[y^m] - A_\phi(x)$.



$\log_2 \tau = 0$ $\log_2 \tau = -1$ $\log_2 \tau = -2$ $\log_2 \tau = -3$ $\log_2 \tau = -4$

Figure 3. PGD-10 examples crafted on $10,000$ test samples of CIFAR-10, and filtered by $\frac{1}{2-\xi}$ confidence threshold for each ξ. Here $\log_2 \tau = 0$ (i.e., $\tau = 1$) is the case shown in Fig. 1. Simply tuning the temperature τ enables more samples to pass the confidence rejector.

respectively, where $\mathcal{L}_{\mathrm{T}}$ and $\mathcal{L}_{\mathrm{A}}$ can be either the same or chosen differently [33]. We can generalize Eq. (5) to involve clean inputs x in the outer minimization objective, which is compatible with the AT methods like TRADES [52]. The inner maximization problem can also include ϕ.

Architecture of A_ϕ. We consider the classifier with a softmax layer as $f_\theta(x) = \mathbb{S}(Wz+b)$, where z is the mapped feature, W and b are the weight matrix and bias vector, respectively. We apply an extra shallow network to construct $A_\phi(x) = \mathrm{MLP}_\phi(z)$, as detailed in Appendix D.1. This two-head structure incurs little computational burden. Other more flexible architectures for A_ϕ can also be used, e.g., RBF networks [39, 49] or concatenating multi-block features that taking path information into account. Note that we stop gradients on the flows of $f_\theta(x)[y] \rightarrow$ BCE loss, and $f_\theta(x)[y^m] \rightarrow$ R-Con when $y^m = y$. These operations prevent the models from concentrating on correctly classified inputs, while facilitating $f_\theta(x)[y]$ to be aligned with $p_{\text{data}}(y|x)$, as explained in Appendix C.1.

How well is A_ϕ learned? In practice, the auxiliary function $A_\phi(x)$ is usually trained to achieve the optimal solution $A_\phi^*(x)$ within a certain error. We introduce a definition on the *point-wise* error between $A_\phi(x)$ and $A_\phi^*(x)$, which admits two ways of measuring, either geometric or arithmetic:

Definition 1. *(Point-wisely ξ-error) If at least one of the bounds holds at a point x:*

$$\begin{aligned}&\text{Bound (i):}\quad \left|\log\left(\frac{A_\phi(x)}{A_\phi^*(x)}\right)\right| \leq \log\left(\frac{2}{2-\xi}\right);\\&\text{Bound (ii):}\quad \left|A_\phi(x) - A_\phi^*(x)\right| \leq \frac{\xi}{2}.\end{aligned} \tag{6}$$

where $\xi \in [0, 1)$, then $A_\phi(x)$ is called ξ-error at input x.

Remark. We can show that given any A_ϕ trained to be better than a random guess at x, we can always find $\xi \in [0, 1)$ satisfying Definition 1. Specifically, assuming that A_ϕ simply performs random guess on x, i.e., $A_\phi(x) = \frac{1}{2}$. Since $A_\phi^*(x) \in [0, 1]$, there is $\left|A_\phi(x) - A_\phi^*(x)\right| = \left|\frac{1}{2} - A_\phi^*(x)\right| \leq \frac{1}{2}$, which means even a random-guess A_ϕ can satisfy Bound (ii) in Definition 1 with $\xi = 1$.

4.2. Coupling confidence and R-Con

Recall that in Lemma 1 we present how to provably distinguish wrongly and correctly classified inputs, via referring to the values of confidence and T-Con. However, in practice we cannot compute T-Con without knowing the true label y. To this end, we substitute T-Con with R-Con during inference, and demonstrate that a $\frac{1}{2-\xi}$ confidence rejector and a R-Con rejector with ξ-error A_ϕ can be coupled to achieve separability, similar as the property shown in Lemma 1.

Theorem 1. *(Separability) Given the classifier f_θ, for any input pair of x_1, x_2 with confidences larger than $\frac{1}{2-\xi}$, i.e.,*

$$f_\theta(x_1)[y_1^m] > \frac{1}{2-\xi}, \text{ and } f_\theta(x_2)[y_2^m] > \frac{1}{2-\xi}, \tag{7}$$

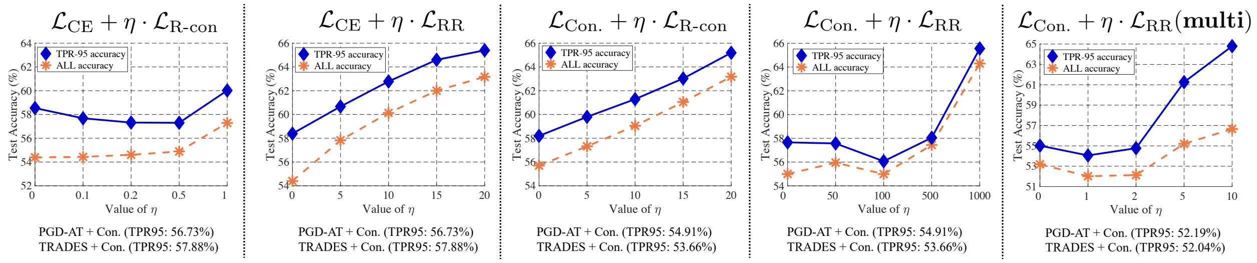
where $\xi \in [0, 1)$. If x_1 is correctly classified as $y_1^m = y_1$, while x_2 is wrongly classified as $y_2^m \neq y_2$, and A_ϕ is ξ-error at x_1, x_2, then there must be R-Con$(x_1) > \frac{1}{2} >$ R-Con(x_2).

Namely, after we first thresholding confidence by $\frac{1}{2-\xi}$ and obtain the remaining samples, any misclassified input will obtain a R-Con value lower than any correctly classified one, as long as A_ϕ is trained to be ξ-error at these points. This property prevents adversaries from simultaneously fooling the predicted labels and R-Con values. As argued in Section 4.3, training A_ϕ to ξ-error could be easier than learning a robust classifier, which justifies the existence of wrongly classified but ξ-error points like x_2. In Fig. 1, we empirically verify Theorem 1 on a ResNet-18 [20] trained with the RR module on CIFAR-10. The test examples are perturbed by PGD-10 and filtered by a $\frac{1}{2-\xi}$ confidence rejector for each ξ. The remaining correctly and wrongly classified samples are separable w.r.t. the R-Con metric, even if we cannot compute ξ-error in practice without knowing true label y.

The effects of temperature tuning. It is known that for a softmax layer $f_\theta(x) = \mathbb{S}(\frac{Wz+b}{\tau})$ with a temperature scalar $\tau > 0$, the true label y and the predicted label y^m are invariant to τ, but the values of confidence and T-Con are not guaranteed to be order-preserving with respect to τ among different inputs. For instance, if there is $f_\theta(x_1)[y_1] < f_\theta(x_2)[y_2]$ under $\tau = 1$, it is possible that for other values of

Table 2. TPR-95 accuracy (%) and ROC-AUC scores evaluated by PGD-100 attacks (10 restarts) on CIFAR-10. The model architecture is ResNet-18, trained by different AT methods and applying different rejectors. GDA* indicates using class-conditional covariance matrices.

AT	Rejector	Clean		ℓ_∞, 8/255		ℓ_∞, 16/255		ℓ_2, 128/255	
		TPR-95	AUC	TPR-95	AUC	TPR-95	AUC	TPR-95	AUC
PGD-AT	KD	82.59	0.618	53.12	0.588	31.97	0.535	64.60	0.599
	LID	84.02	0.712	54.92	0.660	32.75	0.621	66.07	0.663
	GDA	82.35	0.453	52.67	0.461	31.89	0.454	64.13	0.459
	GDA*	84.51	0.664	53.88	0.589	31.94	0.527	65.71	0.605
	GMM	85.44	0.703	54.35	0.607	31.96	0.532	66.54	0.635
CARL	Margin	85.54	0.682	51.67	0.539	30.41	0.516	65.98	0.645
ATRO	Margin	73.42	0.669	36.04	0.654	21.37	0.644	41.52	0.655
TRADES	Con.	86.07	0.837	57.62	0.774	37.55	0.739	67.88	0.781
CCAT	Con.	92.44	0.806	51.68	0.637	45.12	0.683	67.07	0.772
PGD-AT	Con.	86.52	0.857	57.30	0.768	34.77	0.685	69.10	0.783
PGD-AT	SNet	84.19	0.796	56.41	0.730	35.25	0.692	67.49	0.741
PGD-AT	EBD	85.34	0.832	57.04	0.763	34.96	0.690	67.82	0.774
TRADES	**RR**	86.47	0.849	**58.52**	**0.786**	38.06	**0.748**	68.97	0.793
CCAT	**RR**	**94.12**	**0.909**	53.89	0.662	**48.02**	0.688	67.98	0.785
PGD-AT	**RR**	86.91	0.861	58.21	0.776	35.32	0.705	**70.24**	**0.796**



PGD-AT + Con. (TPR95: 56.73%)
TRADES + Con. (TPR95: 57.88%)

PGD-AT + Con. (TPR95: 56.73%)
TRADES + Con. (TPR95: 57.88%)

PGD-AT + Con. (TPR95: 54.91%)
TRADES + Con. (TPR95: 53.66%)

PGD-AT + Con. (TPR95: 54.91%)
TRADES + Con. (TPR95: 53.66%)

PGD-AT + Con. (TPR95: 52.19%)
TRADES + Con. (TPR95: 52.04%)

Figure 4. Performances under *adaptive attacks* on CIFAR-10. We design five adaptive objectives to evade both classifier and rejector. Each attack runs for 500 steps (10 restarts). Our model is ResNet-18 trained by PGD-AT+**RR**. The performances of baselines are on the bottom.

τ the inequality is reversed (detailed in Appendix C.2). As seen in Fig. 3, after we lower down the temperature τ during inference, more PGD-10 examples can satisfy the conditions in Theorem 1, on which R-Con can provably distinguish correctly and wrongly classified inputs.

4.3. The task of learning a ξ-error $A_\phi(x)$

[43] advocates that learning a rejector is nearly as hard as learning a classifier against adversarial examples. So it would be informative to estimate the difficulty of training a ξ-error R-Con rejector. As $A_\phi(x)$ is bounded in $[0, 1]$ by model design, we can convert the regression task of learning ξ-error $A_\phi(x)$ to a substituted classification task as:

Theorem 2. *(Substituted learning task of $A_\phi(x)$) The task of learning a ξ-error $A_\phi(x)$ can be reconstructed into a classification task with number of classes as N_{sub}, where*

$$N_1 = \frac{\log \rho^{-1}}{\log\left(\frac{2}{2-\xi}\right)}+1,\ N_2 = \frac{2}{\xi},\ \textit{and}\ N_{sub} = \lceil \min(N_1, N_2) \rceil.$$

Here $\lceil \cdot \rceil$ is the ceil rounding function, and ρ is a preset rounding error for small values of $A^*_\phi(x)$.

Intuitively, Theorem 2 provides a way to approximate how many test samples are expected to satisfy ξ-error conditions. Under the similar data distribution, the classification problems with a larger number of classes are usually (not necessarily) more challenging to learn [36]. For example, the same model that achieves 90% test accuracy on CIFAR-10 may only achieve 70% test accuracy on CIFAR-100. According to Theorem 2, if we want to obtain a 0.1-error A_ϕ on the CIFAR datasets, then this task can be regarded as a 20-classes classification problem, whose learning difficulty is expected to be between 10-classes one (e.g., CIFAR-10 task) and 100-classes one (e.g., CIFAR-100 task) [51]. Thus, the test accuracy of a 20-classes task is expected to be between 90% and 70% on the CIFAR datasets, i.e., about 70%∼90% test samples may satisfy ξ-error conditions with $\xi = 0.1$.

Similarly, Theorem 2 can also approximate the difficulty of learning a *robust* ξ-error A_ϕ, e.g., for any x' in ℓ_∞-ball around x, we have x' satisfy ξ-error conditions. This task can be converted into training a *certified* classifier [45], and the ratio of test samples that achieve robust ξ-error A_ϕ can be approximated by the performance of certified defenses.

Table 3. TPR-95 accuracy (%) under common corruptions in CIFAR-10-C. The model architecture is ResNet-18, trained by different AT methods and applying different rejectors. The reported accuracy under each corruption is averaged across five severity.

AT	Rejector	CIFAR-10-C									
		Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contra	Elastic	JPEG
PGD-AT	SNet	77.74	75.52	78.72	79.77	75.81	61.32	81.75	42.97	78.59	82.08
PGD-AT	EBD	78.47	77.92	80.47	81.17	79.14	61.16	83.98	42.10	80.86	83.34
CARL	Margin	77.45	74.94	78.00	79.86	74.16	56.09	81.28	40.33	78.17	82.64
ATRO	Margin	55.36	53.74	54.59	50.84	41.12	42.82	50.13	33.54	54.48	56.82
CCAT	Con.	83.04	85.47	89.33	**89.38**	88.21	76.32	**92.71**	55.99	89.34	91.94
TRADES	Con.	79.89	78.48	80.92	78.75	71.61	63.53	80.97	45.22	80.53	84.50
PGD-AT	**RR**	80.87	79.42	81.90	81.89	76.95	63.49	84.02	44.03	82.18	85.12
CCAT	**RR**	**85.03**	**86.26**	**89.83**	89.22	**88.41**	**77.45**	92.62	**58.95**	**89.59**	**92.06**
TRADES	**RR**	80.03	79.15	81.00	80.16	74.18	63.55	82.13	45.99	80.98	84.64

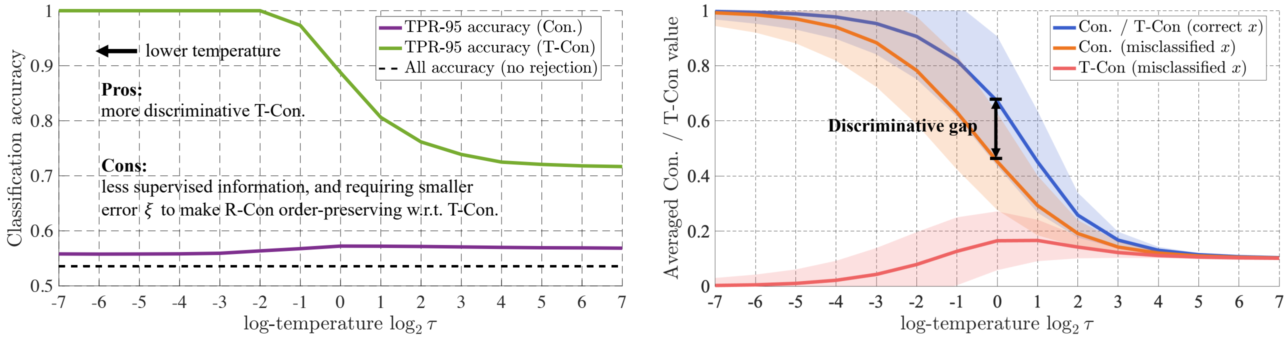


Figure 5. The effects of temperature τ. The model is adversarially trained on CIFAR-10 (no RR module used) and evaded by PGD-10. *Left*: TPR-95 accuracy w.r.t. confidence and T-Con. *Right*: Averaged confidence / T-Con value on correct / misclassified PGD-10 inputs.

5. Further discussion

The value of ξ is unknown in inference. Note that explicitly computing the value of ξ-error requires access to T-Con, which is not available in inference. This may raise confusion on how the provable separability helps to promote robustness in practice? The answer is that even though we cannot point-wisely know the value of ξ, the mechanism in Theorem 1 still implicitly works in population. To be specific, if we preset a confidence threshold γ as the first rejector, the input points with $\xi < 2 - \frac{1}{\gamma}$ (i.e., $\gamma > \frac{1}{2-\xi}$) will implicitly obtain provable predictions after using R-Con as the second rejection metric.

Rectified rejection vs. binary rejection. In the limiting case of $\tau \rightarrow 0$, the returned probability vector will tend to one-hot, i.e., $f_\theta(x)[y^m]$ always equals to one, and the optimal solution A_ϕ^* becomes binary as $A_\phi^*(x) = 1$ if x is correctly classified; otherwise $A_\phi^*(x) = 0$. In this case, learning A_ϕ degenerates to a binary classification task, which has been widely studied and applied in previous work [12–14, 22]. However, directly learning a binary rejector abandons the returned confidence that can be informative about the prediction certainty [12, 47]. Besides, since a trained binary rejector $\mathcal{M}$ usually outputs continuous values in $[0, 1]$, e.g., after a sigmoid activation, its returned values could be overwhelmed by the optimization procedure under binary supervision [26]. For example, two wrongly classified inputs x_1, x_2 may have $\mathcal{M}(x_1) < \mathcal{M}(x_2)$ only because $\mathcal{M}$ is easier to optimize on x_1 during training. This trend deviates $\mathcal{M}$ from properly reflecting the prediction certainty of $f_\theta(x)$, and induces suboptimal reject decisions during inference. In contrast, our RR module learns T-Con by rectifying confidence, where T-Con provides more distinctive supervised signals. A ξ-error R-Con metric is approximately order-preserving concerning the T-Con values, enabling R-Con to stick to the certainty measure induced by T-Con and make reasonable reject decisions.

Rectified confidence vs. calibrated confidence. Another concept related with T-Con and R-Con is confidence calibration [19]. Typically, a classifier f_θ with calibrated confidence satisfies that $\forall c \in [0, 1]$, there is $p\big(y^m = y \big| f_\theta(x)[y^m] = c\big) = c$, where the probability is taken over the data distribution. For notation compactness, we let $q_\theta(c) \triangleq p\left(f_\theta(x)[y^m] = c\right)$ be the probability that the returned confidence equals to c. Then if we execute rejection option based on the calibrated confidence, the accuracy on returned predictions can be calculated by $\int_t^1 c \cdot q_\theta(c) \mathrm{d}c / \int_t^1 q_\theta(c) \mathrm{d}c$, where t is the preset threshold. On the positive side, calibrated confidence certifies that the accuracy after rejection is no worse than t. However, since

Table 4. TPR-95 accuracy (%) on CIFAR-10, under multi-target attack and GAMA attacks. The model architecture is ResNet-18, and the threat model is $(\ell_\infty, 8/255)$.

AT	Rejector	Multi-target	GAMA (PGD)	GAMA (FW)
PGD-AT	SNet	55.02	55.79	51.37
PGD-AT	EBD	55.40	56.15	53.24
CARL	Margin	46.17	48.49	44.78
ATRO	Margin	32.53	31.74	28.31
CCAT	Con.	34.21	49.78	38.01
TRADES	Con.	53.69	56.89	50.88
PGD-AT	**RR**	**56.18**	57.57	**54.08**
CCAT	**RR**	36.48	51.30	40.72
TRADES	**RR**	54.83	**57.93**	51.48

there is no explicit supervision on the distribution $q_\theta(c)$, the final accuracy still relies on the difficulty of learning task. In contrast, rejecting via T-Con with a 0.5 threshold will always lead to 100% accuracy, whatever the learning difficulty, which makes T-Con a more ideal supervisor for a generally well-behaved rejection module, as also discussed in [7].

6. Experiments

Our experiments are done on the datasets CIFAR-10, CIFAR-100, and CIFAR-10-C [21]. We choose two commonly used model architectures: ResNet-18 [20] and WRN-34-10 [50]. Following the suggestions in [32], for all the defenses, the default training settings include batch size 128; SGD momentum optimizer with the initial learning rate of 0.1; weight decay 5×10^{-4}. The training runs for 110 epochs with the learning rate decaying by a factor of 0.1 at 100 and 105 epochs. We report the results on the checkpoint with the best 10-steps PGD attack (PGD-10) accuracy [35].

AT frameworks used in our methods. We mainly apply three popular AT frameworks to combine with our RR module, involving PGD-AT [29], TRADES [52], and CCAT [41]. For PGD-AT and TRADES, we use PGD-10 during training, under ℓ_∞-constraint of $8/255$ with step size $2/255$. The trade-off parameter for TRADES is 6 [52], and the implementation of CCAT follows its official code. In the reported results, '**RR**' refers to the model adversarially trained by Eq. (5) with different AT frameworks, and using R-Con as the rejection metric; We set $\lambda = 1$ in Eq. (5) without tuning.

Baselines. We choose two kinds of commonly compared baselines [3]. The first kind constructs statistics upon the learned features after training the classifier, including kernel density (KD) [11], local intrinsic dimensionality (LID) [28], Gaussian discriminant analysis (GDA) [25], and Gaussian mixture model (GMM) [53]. The second kind jointly learns the rejector with the classifier, which involves SelectiveNet (SNet) [13], energy-based detection (EBD) [27], CARL [24], ATRO [22], and CCAT [41]. We emphasize that most of these baselines are originally applied to STMs, while *we adopt them to ATMs as stronger baselines by re-tuning their hyperparameters*, as detailed in Appendix D.2.

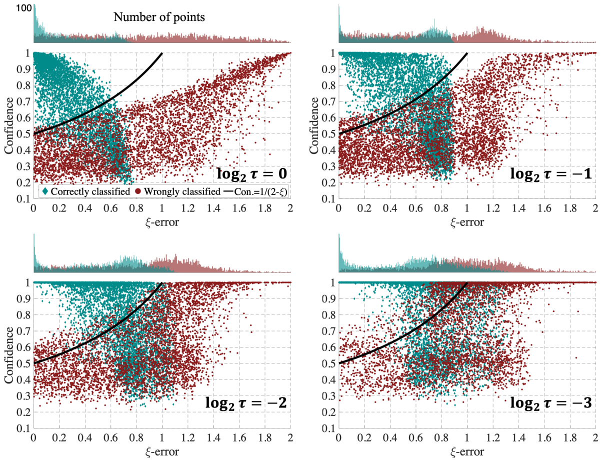


Figure 6. Confidence values w.r.t. ξ-error values of ResNet-18 trained by PGD-AT+**RR** on CIFAR-10. Here ξ is calculated as the minimum value satisfying Definition 1, black line is Con. $= \frac{1}{2-\xi}$. The settings are the same as in Fig. 3, with different temperatures.

Adversarial attacks. We evaluate PGD [29], C&W [6], AutoAttack [10], multi-target attack [18], GAMA attack [40], and general corruptions in CIFAR-10-C [21]. More details on the attacking hyperparameters are in Appendix D.3.

6.1. Performance against normal attacks

We report the results on defending normal attacks, i.e., those only target at fooling the classifiers.

PGD attacks. The results on CIFAR-10 are in Table 2 (results on CIFAR-100 are in Appendix D.4). 'All' accuracy indicates the case with no rejection. As for 'TPR-95' accuracy, we fix the thresholds to 95% true positive rate, which means at most 5% of correctly classified examples can be rejected. We evaluate under PGD-100 $(\ell_\infty, \epsilon = 8/255)$, and unseen attacks with different perturbation $(\epsilon = 16/255)$, threat model (ℓ_2), or more steps (PGD-1000 in Table 8). We apply untargeted mode with 10 restarts.

More advanced attacks. In Table 4, we evaluate under multi-target attack and GAMA attacks. As to AutoAttack, its algorithm returns crafted adversarial examples for successful evasions, while returns original clean examples otherwise. By using **RR** to train a ResNet-18, the All (TPR-95) accuracy (%) under AutoAttack is 48.62 (84.32) and 25.20 (70.99) on CIFAR-10 and CIFAR-100, respectively.

Common corruptions. We also investigate the performance of our methods against the out-of-distribution corruptions on CIFAR-10-C, as summarized in Table 3.

As seen, our RR module can incorporate different AT frameworks, which outperform previous baselines. Besides, the improvement on CIFAR-100 is more significant than it on CIFAR-10, which verifies our formulation on learning difficulty in Section 4.3.

Table 5. Ablation studies on the effect of temperature τ for **RR**. Note that in the objective Eq. (5), τ is only tuned in the term of $\mathcal{L}_{\mathrm{RR}}$, while the temperature for $\mathcal{L}_{\mathrm{T}}$ is kept to be 1.

$\log_2 \tau$	Clean inputs		PGD-10 inputs	
	TPR-95	AUC	TPR-95	AUC
-1	**86.86**	0.866	59.11	**0.770**
-2	86.62	0.865	60.63	0.762
-3	85.18	**0.868**	**61.12**	0.741
-4	80.22	0.836	55.15	0.740

Table 6. Ablation studies on rectified construction of R-Con in Eq. (3). Here '$f_\theta(x)[y^m]$' and '$A_\phi(x)$' indicate using confidence and auxiliary function to substitute R-Con in $\mathcal{L}_{\mathrm{RR}}$, respectively.

Rejector	Clean inputs		PGD-10 inputs	
	TPR-95	AUC	TPR-95	AUC
$A_\phi(x)$	85.77	0.844	56.97	0.765
RR	**86.91**	**0.861**	**58.39**	**0.776**
$f_\theta(x)[y^m]$	86.76	0.865	57.42	0.768
RR (Con.)	**87.12**	**0.868**	**58.49**	**0.777**

Table 7. Minimal perturbations required by successful evasions, searched by CW attacks. Here 'Normal (Nor.)' refers to fooling the classifier, and 'Adaptive (Ada.)' refers to *adaptively* fooling both the classifier and rejector.

Rejector	CIFAR-10				CIFAR-100			
	CW-ℓ_∞		CW-ℓ_2		CW-ℓ_∞		CW-ℓ_2	
	Nor.	Ada.	Nor.	Ada.	Nor.	Ada.	Nor.	Ada.
SNet	14.30	30.48	0.84	2.70	8.20	23.05	0.56	2.37
EBD	14.70	37.54	0.85	2.42	8.58	25.69	0.60	1.81
RR	14.99	**38.58**	0.87	**3.28**	8.53	**28.67**	0.61	**3.21**

Table 8. Classification accuracy (%) and ROC-AUC scores under PGD-1000 attacks (10 restarts), where the step size is $2/255$ and the perturbation constraint is $8/255$ under ℓ_∞ threat model.

Rejector	CIFAR-10		CIFAR-100	
	TPR-95	AUC	TPR-95	AUC
SNet	55.83	0.725	32.69	0.744
EBD	56.12	0.763	33.35	0.769
RR	**57.57**	**0.773**	**34.48**	**0.776**

6.2. Performance against adaptive attacks

Following the suggestions in [4], we design adaptive attacks to evade the classifier and rejector simultaneously.

Evaluate adaptive accuracy. In the first adaptive attack, we consider the mostly commonly used threat model of $(\ell_\infty, 8/255)$, and explore five different adaptive objectives, including $\mathcal{L}_{\mathrm{CE}} + \eta \cdot \mathcal{L}_{\text{R-Con}}$, $\mathcal{L}_{\mathrm{CE}} + \eta \cdot \mathcal{L}_{\mathrm{RR}}$, $\mathcal{L}_{\mathrm{Con.}} + \eta \cdot \mathcal{L}_{\mathrm{RR}}$, $\mathcal{L}_{\mathrm{Con.}} + \eta \cdot \mathcal{L}_{\text{R-Con}}$, and $\mathcal{L}_{\mathrm{Con.}} + \eta \cdot \mathcal{L}_{\mathrm{RR}}$(multi), where $\mathcal{L}_{\mathrm{Con.}}$ is to directly optimize the confidence, $\mathcal{L}_{\text{R-Con}} = \log \text{R-Con}(\cdot)$ and *multi* refers to multi-target version. The results are in Fig. 4, where we also report the TPR-95 accuracy of baselines for reference. As seen, under adaptive attacks, applying our RR module still outperforms the baselines. We also tried using $\mathcal{L}_{\text{R-Con}} = \text{R-Con}(\cdot)$ without log, the conclusions are similar.

Find the minimal distortion. The second one follows [5], where we add the loss term of maximizing R-Con into the original CW objective, and find the minimal distortion for a per-example successful evasion if the classifier is fooled and the rejector value is higher than the median value of the training set. The binary search steps are 9 with 1,000 iteration steps for each search. As in Table 7, adaptive attacks require larger minimal perturbations than normal attacks, and successfully evading our methods is harder than baselines.

6.3. Ablation studies

Empirical effects of temperature τ. In addition to the effects described in Section 4.2, we show the curves of TPR-95 accuracy and averaged confidence / T-Con values in Fig. 5 w.r.t. the temperature scaling, while in Fig. 6 we visualize the sample distributions of ξ-error vs. confidence values. We can observe that the T-Con values become more discriminative for a lower temperature on rejecting misclassified examples, but numerically provide less supervised information and require smaller error ξ to make R-Con order-preserving w.r.t. T-Con. On the other hand, as the temperature τ gets larger above one, the discriminative power of confidence becomes weaker, making R-Con harder to distinguish misclassified inputs from correctly classified ones. In practice, we can trade off between the learning difficulty and the effectiveness of R-Con by tuning τ. In Table 5 we study the effects of tuning temperature values for $f_\theta(x)[y]$ and $f_\theta(x)[y^m]$ in $\mathcal{L}_{\mathrm{RR}}$. We find that moderately lower down τ can benefit model robustness but sacrifice clean accuracy, while overly low temperature degenerates both clean and robust performance.

Formula of R-Con. In Table 6, we investigate the cases if there is no rectified connection (i.e., only use $A_\phi(x)$) or no auxiliary flexibility (i.e., only use $f_\theta(x)[y^m]$) in the constructed rejection module. As shown, our rectifying paradigm indeed promote the effectiveness.

7. Conclusion

We introduce T-Con as a certainty oracle, and train R-Con to mimic T-Con. Intriguingly, a ξ-error R-Con rejector and a $\frac{1}{2-\xi}$ confidence rejector can be coupled to provide provable separability. We also empirically validate the effectiveness of our RR module by using R-Con alone as the rejector, which is well compatible with different AT frameworks.

Limitations. Although provable separability is appealing, only part of the inputs can enjoy this property (e.g., $\sim 50\%$ adversarial points as in Fig. 3). Besides, it is non-trivial to explicitly control the true positive rate when using the coupling rejection strategy. Nevertheless, confidence and R-Con are just one instance of coupled pair, where more advanced and promising coupling rejectors could be developed.

References

[1] 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 (ICML)*, 2018. 1

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013. 1

[3] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous instance detection in deep learning: A survey. *arXiv preprint arXiv:2003.06979*, 2020. 7

[4] 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. 8

[5] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security (AISec)*, 2017. 8

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, 2017. 2, 7

[7] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *International Conference on Neural Information Processing Systems (NeurIPS)*, pages 2902–2913, 2019. 2, 7

[8] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer, 2016. 2

[9] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020. 1

[10] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 3, 7

[11] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 7

[12] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2017. 2, 3, 6

[13] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International Conference on Machine Learning (ICML)*, 2019. 2, 6, 7

[14] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017. 6

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1, 3

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[17] 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. 1

[18] Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019. 7

[19] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017. 6

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision (ECCV)*, pages 630–645. Springer, 2016. 4, 7

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations (ICLR)*, 2019. 2, 7

[22] Masahiro Kato, Zhenghang Cui, and Yoshihiro Fukuhara. Atro: Adversarial training with a rejection option. *arXiv preprint arXiv:2010.12905*, 2020. 1, 2, 3, 6, 7

[23] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2

[24] Cassidy Laidlaw and Soheil Feizi. Playing it safe: Adversarial robustness with an abstain option. *arXiv preprint arXiv:1911.11253*, 2019. 1, 2, 7

[25] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In

Advances in Neural Information Processing Systems (NeurIPS), 2018. 7

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 6

[27] Weitang Liu, Xiaoyun Wang, John Owens, and Sharon Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 7

[28] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018. 7

[29] 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 (ICLR)*, 2018. 1, 7

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016. 3

[31] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. 3

[32] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021. 7

[33] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting adversarial training with hypersphere embedding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[34] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. 1

[35] Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020. 7

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 5

[37] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5019–5031, 2018. 1

[38] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Improving adversarial robustness using proxy distributions. *arXiv preprint arXiv:2104.09425*, 2021. 1

[39] Angelo Sotgiu, Ambra Demontis, Marco Melis, Battista Biggio, Giorgio Fumera, Xiaoyi Feng, and Fabio Roli. Deep neural rejection against adversarial examples. *EURASIP Journal on Information Security*, 2020:1–10, 2020. 4

[40] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 7

[41] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning (ICML)*, 2020. 1, 2, 7

[42] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. 1

[43] Florian Tramer. Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML Workshop on Adversarial Machine Learning*, 2021. 1, 5

[44] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[45] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning (ICML)*, pages 5283–5292, 2018. 5

[46] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 1

[47] Xi Wu, Uyeong Jang, Jiefeng Chen, Lingjiao Chen, and Somesh Jha. Reinforcing adversarial robustness using model confidence induced by adversarial training. In *International Conference on Machine Learning (ICML)*, pages 5334–5342. PMLR, 2018. 2, 6

[48] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020. 1

[49] Pourya Habib Zadeh, Reshad Hosseini, and Suvrit Sra. Deep-rbf networks revisited: Robust classification with rejection. *arXiv preprint arXiv:1812.03190*, 2018. 4

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *The British Machine Vision Conference (BMVC)*, 2016. 7

[51] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[52] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019. 1, 4, 7

[53] Zhihao Zheng and Pengyu Hong. Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 7