Confidence-Aware Training of Smoothed Classifiers for Certified Robustness

Jongheon Jeong*, Seojin Kim*, Jinwoo Shin

Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, 34141 South Korea
{jongheonj, osikjs, jinwoos}@kaist.ac.kr

Abstract

Any classifier can be "smoothed out" under Gaussian noise to build a new classifier that is provably robust to ℓ_2-adversarial perturbations, *viz*., by averaging its predictions over the noise via *randomized smoothing*. Under the *smoothed classifiers*, the fundamental trade-off between accuracy and (adversarial) robustness has been well evidenced in the literature: *i.e*., increasing the robustness of a classifier for an input can be at the expense of decreased accuracy for some other inputs. In this paper, we propose a simple training method leveraging this trade-off to obtain robust smoothed classifiers, in particular, through a *sample-wise* control of robustness over the training samples. We make this control feasible by using "accuracy under Gaussian noise" as an easy-to-compute proxy of adversarial robustness for an input. Specifically, we differentiate the training objective depending on this proxy to filter out samples that are unlikely to benefit from the worst-case (adversarial) objective. Our experiments show that the proposed method, despite its simplicity, consistently exhibits improved certified robustness upon state-of-the-art training methods. Somewhat surprisingly, we find these improvements persist even for other notions of robustness, *e.g*., to various types of common corruptions. Code is available at https://github.com/alinlab/smoothing-catrs.

1 Introduction

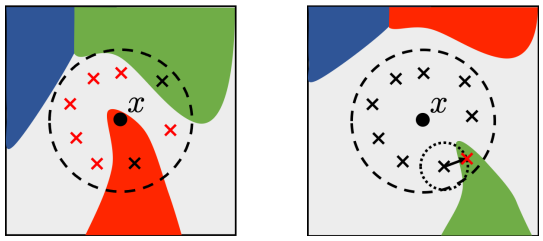
Despite these tremendous advances in *deep neural networks* for a variety of computer vision tasks towards artificial intelligence, the broad existence of *adversarial examples* (Szegedy et al. 2014) is still a significant aspect that reveals the gap between machine learning systems and humans: for a given input x (*e.g*., an image) to a classifier f, say a neural network, f often permits a perturbation δ that completely flips the prediction $f(x+\delta)$, while δ is too small to change the semantic in x. In response to this vulnerability, there have been tremendous efforts in building *robust* neural network based classifiers against adversarial examples, either in forms of *empirical defenses* (Athalye, Carlini, and Wagner 2018; Carlini et al. 2019; Tramer et al. 2020), which are largely based on *adversarial training* (Madry et al. 2018; Zhang et al. 2019; Wang et al. 2020; Zhang et al. 2020c; Wu, Xia, and Wang 2020), or *certified defenses* (Wong and Kolter 2018; Xiao et al. 2019; Cohen, Rosenfeld, and Kolter 2019; Zhang et al. 2020b), depending on whether the robustness claim can be theoretically guaranteed or not.

Randomized smoothing (Lecuyer et al. 2019; Cohen, Rosenfeld, and Kolter 2019), our focus in this paper, is currently a prominent approach in the context of certified defense, thanks to its scalability to arbitrary neural network architectures while previous methods have been mostly limited in network sizes or require strong assumptions, *e.g*., Lipschitz constraint, on their architectures: specifically, for a given classifier f, it constructs a new classifier $\hat{f}$, where $\hat{f}(x)$ is defined to be the class that $f(x+\delta)$ outputs most likely over $\delta \sim \mathcal{N}(0, \sigma^2 I)$, *i.e*., the Gaussian noise. Then, it is shown by Lecuyer et al. (2019) that $\hat{f}$ is certifiably robust in ℓ_2-norm, and Cohen, Rosenfeld, and Kolter (2019) further tightened the ℓ_2-robustness guarantee which is currently considered as the state-of-the-art in certified defense.

However, even with recent methods for adversarial defense, including randomized smoothing, the *trade-off* between robustness and accuracy (Tsipras et al. 2019; Zhang et al. 2019) has been well evidenced, *i.e*., increasing the robustness for a specific input can be at the expense of decreased accuracy for other inputs. For instance, with the current best practices, Salman et al. (2020a) reports that the accuracy of ResNet-50 on ImageNet degrades, *e.g*., 75.8% $\rightarrow$ 63.9%, by an ℓ_∞-adversarial training, *i.e*., optimizing the classifier to ensure robustness at all the given training samples around an ℓ_∞-ball of size $\frac{4}{255}$. In addition, Zhang et al. (2019) has shown that the (empirical) robustness of a classifier can be further boosted in training by paying more expense in accuracy. A similar trend can be also observed with certified defenses, *e.g*., randomized smoothing, as the clean accuracy of smoothed classifiers are usually less than those one can obtain from the standard training on the same architecture (Cohen, Rosenfeld, and Kolter 2019).

Contribution. In this paper, we develop a novel training method for randomized smoothing, coined *Confidence-Aware Training for Randomized Smoothing* (CAT-RS), which incorporates a *sample-wise* control of target robustness on-the-fly motivated by the accuracy-robustness trade-off in smoothed classifiers. Intuitively, a natural approach one can consider in response to the trade-off in robust training is to appropriately lower the robustness requirement for

*These authors contributed equally.



(a) Bottom-K loss (b) Worst-case loss

Figure 1: Illustration of the two proposed losses, *i.e.*, the (a) *bottom-K* and (b) *worst-case* losses. Each $\times$ represents Gaussian noise around x. We aim to minimize the cross-entropy loss only for $\times$'s marked as red for each case.

"hard-to-classify" samples while maintaining those for the remaining ("easier") samples: here, the challenges are (a) which samples should we choose as either "hard-to-classify" (or "easier") for the control in training, and (b) how to control their target robustness. For both (a) and (b), the major difficultly stems from that evaluating adversarial robustness for a given sample is computationally hard in practice.

To implement this idea, we focus on a peculiar correspondence from *prediction confidence* to adversarial robustness that smoothed classifiers offer: due to its local-Lipschitzness (Salman et al. 2019), achieving a high confidence at x from a smoothed classifier also implies a high (certified) robustness at x. Inspired by this, we propose to use the sample-wise confidence of smoothed classifiers as an efficient proxy of the certified robustness, and defines two new losses, namely the *bottom-K* and *worst-case* Gaussian training, each of those targets different levels of confidence so that the overall training can prevent low-confidence samples from being enforced to increase their robustness.

We verify the effectiveness of our proposed method through an extensive comparison with existing robust training methods for smoothed classifiers, including the state-of-the-arts, on a wide range of benchmarks on MNIST, Fashion-MNIST, CIFAR-10/100, and ImageNet. Our experimental results constantly show that the proposed method can significantly improve the previous state-of-the-art results on certified robustness achievable from a given neural network architecture, by (a) maximizing the robust radii of high-confidence samples while (b) reducing the risk of deteriorating the accuracy at low-confidence samples. More intriguingly, we also observe that such a training scheme also helps smoothed classifiers to generalize beyond adversarial robustness, as evidenced by significant improvements in robustness against common corruptions compared to other robust training methods. Our extensive ablation study further confirms that each of both proposed components has an individual effect on improving certified robustness, and can effectively control the accuracy-robustness trade-off with the hyperparameter between the two proposed losses.

Related work. There have been continual attempts to provide a certificate on robustness of deep neural networks against adversarial attacks (Gehr et al. 2018; Wong and Kolter 2018; Mirman, Gehr, and Vechev 2018; Xiao et al. 2019; Gowal et al. 2019; Zhang et al. 2020b), and correspondingly to further improve the robustness with respect to those certification protocols (Croce, Andriushchenko, and Hein 2019; Croce and Hein 2020; Balunovic and Vechev 2020).[1] *Randomized smoothing* (Cohen, Rosenfeld, and Kolter 2019) has attracted a particular attention among them, due to its scalability to large datasets and its flexibility to various applications (Rosenfeld et al. 2020; Salman et al. 2020b; Wang et al. 2021; Fischer, Baader, and Vechev 2021; Wu et al. 2022) or other threat models (Li et al. 2021b; Yang et al. 2020; Lee et al. 2019; Jia et al. 2020; Zhang et al. 2020a; Salman et al. 2022).

This work aims to improve adversarial robustness of randomized smoothing, along a line of research on designing training schemes specialized for smoothed classifiers (Salman et al. 2019; Zhai et al. 2020; Jeong and Shin 2020; Jeong et al. 2021). Specifically, we focus on the relationship between confidence and robustness of smoothed classifiers, a property rarely investigated previously but few (Kumar et al. 2020; Jeong et al. 2021). We leverage the property to overcome challenges in estimating sample-wise robustness, and to develop a data-dependent adversarial training which has been also challenging even for empirical robustness (Wang et al. 2020; Zhang et al. 2021).

2 Preliminaries

Adversarial robustness. Consider a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ sampled from P, where $x \in \mathbb{R}^d$ and $y \in \mathcal{Y} := \{1, \cdots, K\}$, and let $f : \mathbb{R}^d \to \mathcal{Y}$ be a classifier. Given that f is discrete, one can consider a differentiable $F : \mathbb{R}^d \to \Delta^{K-1}$ to allow a gradient-based optimization assuming $f(x) := \arg\max_{k \in \mathcal{Y}} F_k(x)$, where Δ^{K-1} is probability simplex in $\mathbb{R}^K$. The standard framework of *empirical risk minimization* to optimize f assumes that the samples in $\mathcal{D}$ are *i.i.d.* from P and expect f to perform well given that the future samples also follow the *i.i.d.* assumption.

However, in the context of *adversarial robustness* (and for other notions of robustness as well), the *i.i.d.* assumption on the future samples does not hold anymore: instead, it assumes that the samples can be *arbitrarily* perturbed up to a certain restriction, *e.g.*, a bounded ℓ_2-ball, and focuses on the *worst-case* performance over the perturbed samples. One way to quantify this is the *average minimum-distance* of adversarial perturbation (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini et al. 2019):

$$R(f; P) := \mathbb{E}_{(x,y)\sim P}\left[\min_{f(x')\neq y} ||x' - x||_2\right]. \quad (1)$$

Randomized smoothing. The essential challenge in achieving adversarial robustness in neural networks, however, stems from that directly evaluating (1) (and further optimizing it) is usually computationally infeasible, *e.g.*, under the standard practice that F is modeled by a complex, high-dimensional neural network. *Randomized smoothing*

[1] A more extensive survey on certified robustness can be found in Li et al. (2021a).

(Lecuyer et al. 2019; Cohen, Rosenfeld, and Kolter 2019) bypasses this difficulty by constructing a new classifier $\hat{f}$ from f instead of letting f to directly model the robustness: specifically, it transforms the base classifier f with a certain *smoothing measure*, where in this paper we focus on the case of Gaussian distributions $\mathcal{N}(0, \sigma^2 I)$:

$$\hat{f}(x) := \arg\max_{c \in \mathcal{Y}} \mathbb{P}_{\delta \sim \mathcal{N}(0,\sigma^2 I)} \left(f(x+\delta) = c \right). \quad (2)$$

Then, the robustness of $\hat{f}$ at (x, y), namely $R(\hat{f}; x, y)$, can be lower-bounded in terms of the *certified radius* $\underline{R}(\hat{f}, x, y)$, *e.g.*, Cohen, Rosenfeld, and Kolter (2019) showed that the following bound holds which is tight for ℓ_2-adversary:

$$R(\hat{f}; x, y) \geq \sigma \cdot \Phi^{-1}(p_f(x, y)) =: \underline{R}(\hat{f}, x, y) \quad (3)$$

$$\text{where} \quad p_f(x, y) := \mathbb{P}_\delta(f(x+\delta) = y), \quad (4)$$

provided that $\hat{f}(x) = y$, otherwise $R(\hat{f}; x, y) := 0$.[2] Here, we remark that the formula for certified radius (3) is essentially a function of p_f (4), which represents the *prediction confidence* of $\hat{f}$ at x, or equivalently, the *accuracy* of $f(x+\delta)$ over $\delta \sim \mathcal{N}(0, \sigma^2 I)$. In other words, unlike standard neural networks, smoothed classifiers can guarantee a correspondence from prediction confidence to adversarial robustness - which is the key motivation of our method.

3 Confidence-aware Randomized Smoothing

We aim to develop a new training method to maximize the certified robustness of a smoothed classifier $\hat{f}$, considering the trade-off relationship between robustness and accuracy (Zhang et al. 2019): even though randomized smoothing can be applied for any classifier f, the actual robustness of $\hat{f}$ depends on how much f classifies well under presence of Gaussian noise, *i.e.*, by $p_f(x, y)$ defined in (4). A simple way to train f for a robust $\hat{f}$, therefore, is to minimize the cross-entropy loss (denoted by $\mathbb{CE}$ below) with Gaussian augmentation as in Cohen, Rosenfeld, and Kolter (2019):

$$\min_F \mathbb{E}_{\substack{(x,y)\sim P \\ \delta \sim \mathcal{N}(0,\sigma^2 I)}} \left[\mathbb{CE}(F(x+\delta), y) \right]. \quad (5)$$

In this paper, we extend this basic form of training to incorporate a *confidence-aware* strategy to decide which noise samples $\delta_i \sim \mathcal{N}(0, \sigma^2 I)$ should be used sample-wise for training f. Ideally, one may wish to obtain a classifier f that achieves $p_f(x, y) \approx 1$ for every $(x, y) \sim P$ to maximize its certified robustness. In practice, however, such a case is highly unlikely, and there usually exists a sample x that $p_f(x, y)$ should be quite lower than 1 to maintain the discriminativity with other samples: in other words, these samples can be actually "beneficial" to be misclassified at some (hard) Gaussian noises, otherwise the classifier has to memorize the noises to correctly classify them. On the other hand, for the samples which can indeed achieve $p_f(x, y) \approx 1$, the current Gaussian training (5) may not be able to provide enough samples of δ_i for x throughout the training, as $p_f(x, y) \approx 1$ implies that $f(x+\delta)$ must be correctly classified "almost surely" for $\delta_i \sim \mathcal{N}(0, \sigma^2 I)$.

In these respects, we propose two different variants of Gaussian training (5) that address each of the possible cases, *i.e.*, whether (a) $p_f(x, y) < 1$ or (b) $p_f(x, y) \approx 1$, namely with (a) *bottom-K* and (b) *worst-case* Gaussian training, respectively. During training, the method first estimates $p_f(x, y)$ for each sample by computing their accuracy over M random samples of $\delta \sim \mathcal{N}(0, \sigma^2 I)$, and applies different forms of loss depending on the value. In the following two sections, Section 3.1 and 3.2, we provide the details on each loss, and Section 3.3 describes how to combine the two losses and defines the overall training scheme.

3.1 Bottom-K Loss for Low-confidence Samples

Consider a base classifier f and a training sample $(x, y) \in \mathcal{D}$, and suppose that $p_f(x, y) \ll 1$, *e.g.*, $\hat{f}$ has a low-confidence at x. Figure 1(a) visualizes this scenario: in this case, by definition of $p_f(x, y)$ in (4), $f(x+\delta)$ would be correctly classified to y only with probability p over $\delta \sim \mathcal{N}(0, \sigma^2 I)$, and this implies either (a) $x+\delta$ has not yet been adequately exposed to f during the training, or (b) $x+\delta$ may be indeed hard to be correctly classified for some δ, so that minimizing the loss at these noises could harm the generalization of $\hat{f}$. The design goal of our proposed *bottom-K Gaussian loss* is to modify the standard Gaussian training (5) to reduce the optimization burden from (b) while minimally retaining its ability to cover enough noise samples during training for (a).

We first assume M random *i.i.d.* samples of δ, say $\delta_1, \delta_2, \cdots, \delta_M \sim \mathcal{N}(0, \sigma^2 I)$. One can notice that the random variables $\mathbb{1}[f(x+\delta_i) = y]$'s are also *i.i.d.* each, which follows the Bernoulli distribution of probability $p_f(x, y)$. This means that, if the current $p_f(x, y)$ is the value one attempts to keep instead of further increasing it, the number of "correct" noise samples, namely $\sum_i \mathbb{1}[f(x+\delta_i) = y]$, would follow the *binomial distribution* $K \sim \mathrm{Bin}(M, p)$ - this motivates us to consider the following loss that only minimizes the K*-smallest* cross-entropy losses out of from M Gaussian samples around x:

$$L^{\texttt{low}} := \frac{1}{M} \sum_{i=1}^{K} \mathbb{CE}(F(x+\delta_{\pi(i)}), y), \quad (6)$$

where $K \sim \mathrm{Bin}(M, p_f(x, y))$. Here, $\pi(i)$ denotes the index with the i-th smallest loss value in the M samples.

Yet, the loss defined in (6) may not handle the *cold-start* problem on $p_f(x, y)$, *e.g.*, at the early stage of the training where $x+\delta$ has not been adequately exposed to f, so that it is uncertain whether the current $p_f(x, y)$ is optimal: in this case, $L^{\texttt{low}}$ can be minimized with an under-estimated $p_f \approx 0$, potentially with samples those never optimize the cross-entropy losses during training. Nevertheless, we found that a simple workaround of *clamping* K can effectively handle the issue, *i.e.*, by using $K^+ \leftarrow \max(K, 1)$ instead of K: in other words, we always allow the "easiest" noise among the M samples to be fed into f throughout the training.

[2] Φ denotes the cumulative distribution function of $\mathcal{N}(0, 1^2)$.

3.2 Worst-case Loss for High-confidence Samples

Next, we focus on the case when $p_f(x, y) \approx 1$, *i.e.*, $\hat{f}$ has a high confidence at x, as illustrated in Figure 1(b). In contrast to the previous scenario in Section 3.1 (and Figure 1(a)), now the major drawback of Gaussian training (5) does not come from the *abundance* of hard noises in training, but from the *rareness* of such noises: considering that one can only present a limited number of noise samples to f throughout its training, naïvely minimizing (5) may not cover some "potentially hard" noise samples, and this would result in a significant harm in the final certified radius of the smoothed classifier $\hat{f}$. The purpose of *worst-case* Gaussian training is to overcome this lack of samples via an *adversarial* search around each of the noise samples.

Specifically, for given M samples of Gaussian noise δ_i as considered in (6), namely $\delta_1, \delta_2, \cdots, \delta_M \sim \mathcal{N}(0, \sigma^2 I)$, we propose to modify (5) to find the *worst-case* noise δ^* (a) around an ℓ_2-ball for each noise as well as (b) among the M samples, and minimize the loss at δ^* instead of the average-case loss. To find such worst-case noise, our proposed loss optimizes a given δ_i to maximize the *consistency* of its prediction from a certain label assignment $\hat{y} \in \Delta^{K-1}$ per x:

$$L^{\texttt{high}} := \max_i \max_{\|\delta_i^* - \delta_i\|_2 \le \varepsilon} \mathrm{KL}(F(x + \delta_i^*), \hat{y}), \qquad (7)$$

where $\mathrm{KL}(\cdot, \cdot)$ denotes the Kullback-Leibler divergence. This objective is motivated by (Jeong and Shin 2020) that the consistency of prediction across different Gaussian noise controls the trade-off between accuracy and robustness of smoothed classifiers. Notice from (7) that the objective is equivalent to the cross-entropy loss if $\hat{y}$ is assigned as (hard-labeled) y, while we observe having a soft-labeled $\hat{y}$ is beneficial in practice: its log-probability, where the consistency targets, can now be bounded so $F(x + \delta_i^*)$'s can also minimize their variance in the logit space.

There can be various ways to assign $\hat{y}$ for a given x. One reasonable strategy, which we use in this paper by default, is to assign $\hat{y}$ by the *smoothed prediction* of another classifier $\bar{f}$, pre-trained on $\mathcal{D}$ via Gaussian training (5) with some σ_0. This approach is (a) easy to compute, and (b) naturally reflects sample-wise difficulties under Gaussian noise, while (c) maintaining the label information from y. Nevertheless, we also confirm in Appendix G.1 that $L^{\texttt{high}}$ is still effective even when $\hat{y}$ is defined in a simpler way, namely by the average of $F(x + \delta_i)$'s without the Gaussian pre-training.

In practice, we use the *projected gradient descent* (PGD) (Madry et al. 2018) to solve the inner maximization in (7): namely, we perform a T-step gradient ascent from each δ_i with step size $2 \cdot \varepsilon / T$ while projecting the perturbations to be in the ℓ_2-ball of size ε. This procedure would find a noise δ^* that maximizes the loss around x, while maintaining the Gaussian-like noise appearance due to the projected search in a small ε-ball. In order to further make sure that the Gaussian likelihood of δ^* is maintained from the original δ, we additionally apply a simple trick of *normalizing* the mean and standard deviation of δ^* to follow those of δ.

Comparison to SmoothAdv. The idea of incorporating an adversarial search for the robustness of smoothed classifiers has been also considered in previous works (Salman et al. 2019; Jeong et al. 2021): *e.g.*, Salman et al. (2019) have proposed *SmoothAdv* that applies adversarial training (Madry et al. 2018) to a "soft" approximation of $\hat{f}$ given f and M noise samples:

$$x^* = \operatorname*{arg\,max}_{||x' - x||_2 \le \epsilon} \left(-\log \left(\frac{1}{M} \sum_i F_y(x' + \delta_i) \right) \right). \qquad (8)$$

Our method is different from the previous approaches in which part of the inputs is adversarially optimized: *i.e.*, we directly optimize the noise samples δ_i's instead of x, with no need to assume a soft relaxation of $\hat{f}$. This is due to our unique motivation of finding the worst-case Gaussian noise, and our experimental results in Section 4 further support the effectiveness of this approach.

3.3 Overall Training Scheme

Given the two losses $L^{\texttt{low}}$ and $L^{\texttt{high}}$ defined in Section 3.1 and 3.2, respectively, we now define the full objective of our proposed *Confidence-Aware Training for Randomized Smoothing* (CAT-RS). Overall, in order to differentiate how to combine the two losses per sample basis, we use the smoothed confidence $p_f(x, y)$ (4) as the guiding proxy: specifically, we aim to apply the worst-case loss of $L^{\texttt{high}}$ only for the samples where $p_f(x, y)$ is already high enough. In practice, however, one does not have a direct access to the value of $p_f(x, y)$ during training, and we estimate this with the M noise samples[3] as done for $L^{\texttt{low}}$ and $L^{\texttt{high}}$, *i.e.*, by $\hat{p}_f(x, y) := \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}[f(x + \delta_i) = y]$. Then, we consider a simple and intuitive masking condition of "$K = M$" to activate $L^{\texttt{high}}$, where $K \sim \mathrm{Bin}(M, \hat{p}_f(x, y))$ is the random variable defined in (6) for $L^{\texttt{low}}$. The final loss becomes:

$$L^{\texttt{CAT-RS}} := L^{\texttt{low}} + \lambda \cdot \mathbb{1}[K = M] \cdot L^{\texttt{high}}, \qquad (9)$$

where $\mathbb{1}[\cdot]$ is the indicator random variable, and $\lambda > 0$. In other words, the training minimizes $L^{\texttt{high}}$ only when $L^{\texttt{low}}$ (6) minimizes the "full" cross-entropy losses for all the M noise samples given around (x, y). The hyperparameter λ in (9) controls the trade-off between accuracy and robustness (Zhang et al. 2019) of CAT-RS: given that $L^{\texttt{high}}$ targets samples that achieves high confidence (*i.e.*, they are already robust), having larger weights on $L^{\texttt{high}}$ results in higher certified robustness at large radii. In terms of computational complexity, the proposed CAT-RS takes a similar training cost with recent methods those also perform adversarial searches with smoothed classifiers, *e.g.*, SmoothAdv (Salman et al. 2019) and SmoothMix (Jeong et al. 2021).[4] The complete procedure of computing our proposed CAT-RS loss can be found in Algorithm 1 of Appendix A.

4 Experiments

We evaluate the effectiveness of our proposed training scheme based on various well-established image classification benchmarks to measure robustness, including MNIST

[3] We use $M = 4$ for our method unless otherwise noted.

[4] A comparison of actual training costs is given in Appendix E.

σ	Methods	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
0.25	Gaussian	0.424	76.6	61.2	42.2	25.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Stability	0.420	73.0	58.9	42.9	26.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothAdv	0.544	73.4	65.6	57.0	47.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MACER	0.531	79.5	69.0	55.8	40.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Consistency	0.552	75.8	67.6	58.1	46.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothMix	0.553	77.1	67.9	57.9	46.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CAT-RS (Ours)	**0.562**	76.3	**68.1**	**58.8**	**48.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.50	Gaussian	0.525	65.7	54.9	42.8	32.5	22.0	14.1	8.3	3.9	0.0	0.0	0.0
	Stability	0.531	62.1	52.6	42.7	33.3	23.8	16.1	9.8	4.7	0.0	0.0	0.0
	SmoothAdv	0.684	65.3	57.8	49.9	41.7	33.7	26.0	19.5	12.9	0.0	0.0	0.0
	MACER	0.691	64.2	57.5	49.9	42.3	34.8	27.6	20.2	12.6	0.0	0.0	0.0
	Consistency	0.720	64.3	57.5	50.6	43.2	36.2	29.5	22.8	16.1	0.0	0.0	0.0
	SmoothMix	0.737	61.8	55.9	49.5	43.3	37.2	31.7	25.7	19.8	0.0	0.0	0.0
	CAT-RS (Ours)	**0.757**	62.3	**56.8**	**50.5**	**44.6**	**38.5**	**32.7**	**27.1**	**20.6**	0.0	0.0	0.0
1.00	Gaussian	0.511	47.1	40.9	33.8	27.7	22.1	17.2	13.3	9.7	6.6	4.3	2.7
	Stability	0.514	43.0	37.8	32.5	27.5	23.1	18.8	14.7	11.0	7.7	5.2	3.1
	SmoothAdv	0.790	43.7	40.3	36.9	33.8	30.5	27.0	24.0	21.4	18.4	15.9	13.4
	MACER	0.744	41.4	38.5	35.2	32.3	29.3	26.4	23.4	20.2	17.4	14.5	12.1
	Consistency	0.756	46.3	42.2	38.1	34.3	30.0	26.3	22.9	19.7	16.6	13.8	11.3
	SmoothMix	0.773	45.1	41.5	37.5	33.8	30.2	26.7	23.4	20.2	17.2	14.7	12.1
	CAT-RS (Ours)	**0.815**	43.2	40.2	**37.2**	**34.3**	**31.0**	**28.1**	**24.9**	**22.0**	**19.3**	**16.8**	**14.2**

Table 1: Comparison of ACR and approximate certified test accuracy (%) on CIFAR-10. For each column, we set our result bold-faced if it improves the Gaussian baseline. We set the result underlined if it achieves the highest among the baselines.

Methods	ACR	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
Gaussian	0.875	44	38	33	26	19	15	12	9
Consistency	0.982	41	37	32	28	24	21	17	14
SmoothAdv	1.040	40	37	34	30	27	25	20	15
SmoothMix	1.047	40	37	34	30	26	24	20	17
CAT-RS (Ours)	**1.071**	**44**	**38**	**35**	**31**	**27**	**24**	**20**	**17**

Table 2: Comparison of ACR and approximate certified accuracy (%) on ImageNet. For each column, we set our result bold-faced whenever it improves the Gaussian baseline. We set the result underlined if it achieves the highest among the baselines.

(LeCun et al. 1998), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR-10/100 (Krizhevsky 2009), and ImageNet (Russakovsky et al. 2015) (for certified robustness)[5], as well as MNIST-C (Mu and Gilmer 2019)[6] and CIFAR-10-C (Hendrycks and Dietterich 2019) (for corruption robustness). For a fair comparison, we follow the standard protocol and training setup of the previous works (Cohen, Rosenfeld, and Kolter 2019; Zhai et al. 2020; Jeong and Shin 2020).[7]

Overall, the results show that our method can consistently outperform the previous best efforts to improve the average certified radius by (a) maximizing the robust radii of high-confidence samples while (b) better maintaining the accuracy at low-confidence samples.[8] Moreover, the results on CIFAR-10-C, a corrupted version of CIFAR-10, show that our training scheme also helps smoothed classifiers to generalize on out-of-distribution inputs beyond adversarial examples, as shown by a significant improvement in corruption robustness compared to other robust training methods. We also perform an ablation study, showing that, *e.g.*, the hyperparameter λ in (9) between $L^{\texttt{low}}$ and $L^{\texttt{high}}$ can balance the trade-off between robustness and accuracy well.

Baselines. We compare our method with an extensive list of baseline methods in the literature of training smoothed classifiers: (a) *Gaussian training* (Cohen, Rosenfeld, and Kolter 2019) simply trains a classifier with Gaussian augmentation (5); (b) *Stability training* (Li et al. 2019) adds a

[5]Results on MNIST, Fashion-MNIST, and CIFAR-100 can be found in Appendix C.

[6]Results on MNIST-C can be found in Appendix I.

[7]More details, *e.g.*, training setups, datasets, and hyperparameters, can be found in Appendix B.

[8]Although our experiments are mainly based on ℓ_2, we also provide results for ℓ_∞ adversary on CIFAR-10 in Appendix C.3.

Type	*Gaussian*	*Stability*	*SmoothAdv*	*MACER*	*Consistency*	*SmoothMix*	***CAT-RS* (Ours)**
Clean	76.6	73.0	73.4	**79.5**	75.8	77.1	76.3
Gaussian	70.8	64.6	70.2	72.6	69.8	<u>73.4</u>	**76.8**
Shot	70.0	65.6	68.4	<u>72.8</u>	69.6	72.6	**76.6**
Impulse	70.2	61.6	69.0	<u>74.0</u>	70.4	73.6	**75.6**
Defocus	64.8	65.4	68.4	<u>71.2</u>	69.2	70.6	**74.2**
Glass	65.2	62.0	68.6	71.6	69.0	<u>72.0</u>	**72.8**
Motion	66.2	62.4	67.2	**72.2**	70.8	69.6	<u>71.6</u>
Zoom	65.2	64.2	65.6	70.6	68.4	<u>71.4</u>	**75.4**
Snow	67.0	64.6	64.0	<u>70.8</u>	67.0	69.2	**71.4**
Frost	65.6	63.0	64.0	<u>69.0</u>	66.8	**70.2**	67.8
Fog	52.4	38.8	45.4	**53.8**	49.2	50.4	<u>51.4</u>
Bright	71.0	70.6	67.6	<u>73.8</u>	73.2	<u>73.8</u>	**76.4**
Constrast	39.4	30.0	34.8	**42.8**	35.6	36.4	<u>37.8</u>
Elastic	64.4	63.4	64.6	<u>71.0</u>	66.4	69.8	**71.4**
Pixel	66.4	67.6	68.6	<u>74.4</u>	69.8	69.8	**76.2**
JPEG	67.8	66.8	68.6	<u>70.8</u>	68.4	<u>70.8</u>	**76.2**
mAcc	64.4	60.7	63.7	<u>68.8</u>	65.6	67.7	**70.1**

Table 3: Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C. We report the average across five different corruption severities. We set the highest and runner-up values bold-faced and underlined, respectively.

cross-entropy term between the logits from clean and noisy images; (c) *SmoothAdv* (Salman et al. 2019) employs adversarial training for smoothed classifiers (8); (d) *MACER* (Zhai et al. 2020) adds a regularization that aims to maximize a soft approximation of certified radius; (e) *Consistency* (Jeong and Shin 2020) regularizes the variance of confidences over Gaussian noise; (f) *SmoothMix* (Jeong et al. 2021) proposes a mixup-based (Zhang et al. 2018) adversarial training for smoothed classifiers. Whenever possible, we use the pre-trained models publicly released by the authors to reproduce the results.

Evaluation metrics. We follow the standard evaluation protocol for smoothed classifiers (Salman et al. 2019; Zhai et al. 2020; Jeong and Shin 2020; Jeong et al. 2021): specifically, Cohen, Rosenfeld, and Kolter (2019) has proposed a practical Monte-Carlo-based certification procedure, namely CERTIFY, that returns the prediction of $\hat{f}$ and a lower bound of certified radius, $\mathrm{CR}(f, \sigma, x)$, over the randomness of n samples with probability at least $1 - \alpha$, or abstains the certification. Based on CERTIFY, we consider two major evaluation metrics: (a) the *average certified radius* (ACR) (Zhai et al. 2020): the average of certified radii on the test set $\mathcal{D}_{\texttt{test}}$ while assigning incorrect samples as 0:

$$\mathrm{ACR} := \frac{1}{|\mathcal{D}_{\texttt{test}}|} \sum_{(x,y) \in \mathcal{D}_{\texttt{test}}} [\mathrm{CR}(f, \sigma, x) \cdot \mathbb{1}_{\hat{f}(x)=y}], \quad (10)$$

and (b) the *approximate certified test accuracy* at r: the fraction of the test set which CERTIFY classifies correctly with the radius larger than r without abstaining. We use $n = 100,000$, $n_0 = 100$, and $\alpha = 0.001$ for CERTIFY, following previous works (Cohen, Rosenfeld, and Kolter 2019; Salman et al. 2019; Jeong and Shin 2020; Jeong et al. 2021).

4.1 Results on CIFAR-10

Table 1 shows the performance of the baselines and our model on CIFAR-10 for $\sigma \in \{0.25, 0.5, 1.0\}$. We also plot the approximate certified accuracy over r in Figure 5 (of Appendix C.3). For the baselines, we report best-performing configurations for each σ in terms of ACR among reported in previous works, so that the hyperparameters of the same method can vary over σ (the details can be found in Appendix B.2). Overall, CAT-RS achieves a significant improvement of ACR compared to the baselines. In case of $\sigma = 0.25$ and $\sigma = 0.5$, CAT-RS clearly offers a better trade-off between the clean accuracy and robustness compared to other baselines. Especially, CAT-RS achieves higher approximate certified accuracy for all radii compared to SmoothMix in case of $\sigma = 0.5$. For $\sigma = 1.0$, the ACR of our method significantly surpasses the previous best model, SmoothMix, by $0.773 \rightarrow 0.815$. The improvement of CAT-RS is most evident in $\sigma = 1.0$. This means that our proposed CAT-RS can be more effective at challenging tasks, where it is more likely that a given classifier gets a more diverse confidence distribution for the training samples, so that our proposed confidence-aware training can better play its role.

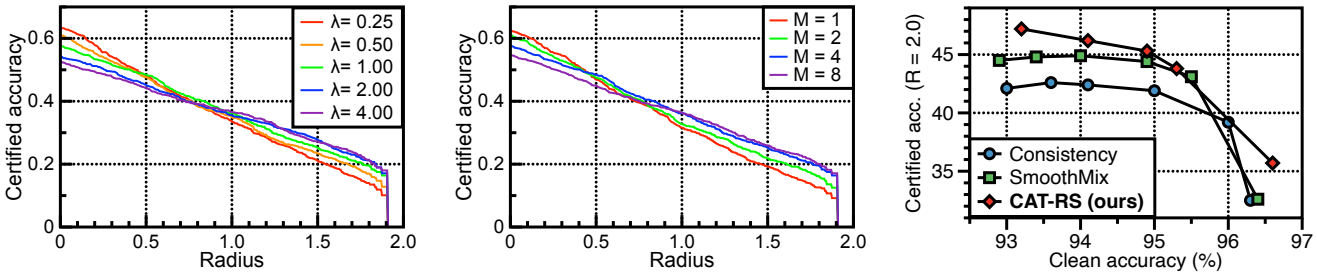
4.2 Results on ImageNet

In this section, we compare the certified robustness of our method on ImageNet (Russakovsky et al. 2015) dataset for $\sigma = 1.0$. We evaluate the performance on the uniformly-subsampled 500 samples in the ImageNet validation dataset following (Cohen, Rosenfeld, and Kolter 2019; Jeong and Shin 2020; Salman et al. 2019; Jeong et al. 2021). The results shown in Table 2 confirm that our method achieves the best results in terms of ACR and certified test accuracy compared to the considered baselines, verifying the effectiveness of CAT-RS even in the large-scale dataset.

4.3 Results on CIFAR-10-C

We also examine the performance of CAT-RS on CIFAR-10-C (Hendrycks and Dietterich 2019), a collection of 75 replicas of the CIFAR-10 test dataset, which consists of 15 different types of common corruptions (*e.g.*, fog, snow, etc.), each of which contains 5 levels of corruption severities. Similarly to (Sun et al. 2021), for a given smoothed classifier trained on CIFAR-10, we report ACR and the certified accuracy at $r = 0.0$ for each corruption type of CIFAR-10-C after averaging over five severity levels, as well as their means over the types, *i.e.*, as the *mean-ACR* (mACR) and *mean-accuracy* (mAcc), respectively. We uniformly subsample each corrupted dataset with size 100, *i.e.*, to have 7,500 samples in total, and use $\sigma = 0.25$ throughout this experiment.

Table 3 summarizes the results on the certified accuracy at $r = 0.0$ (the results on ACR is given in Appendix). Overall, CAT-RS significantly improves mAcc compared to other methods, *i.e.*, for 11 out of 15 corruption types. In other



(a) Effect of λ (b) Effect of M (c) Clean *vs.* robustness trade-off

Figure 2: Comparison of certified accuracy of CAT-RS ablations. For (a) and (b), we train ResNet-20 on CIFAR-10 ($\sigma = 0.5$), while (c) is based on MNIST ($\sigma = 1.0$) for varying control hyperparameters. The detailed results can be found in Appendix G.2.

words, CAT-RS can improve smoothed classifiers to generalize better on unseen corruptions, at the same time maintaining the robustness for such inputs. It is remarkable that the observed gains are not from any prior knowledge about multiple corruption (Hendrycks et al. 2020, 2021) (except for Gaussian noise), but from a better training method. Given the limited gains from other baseline methods on CIFAR-10-C, we attribute that the *sample-dependent calibration* of training objective, a unique aspect of CAT-RS compared to prior arts, is important to explain the effectiveness of CAT-RS on out-of-distribution generalization: *e.g.*, although SmoothAdv also adopts adversarial search in training similarly to CAT-RS, it could not improve mAcc on CIFAR-10-C from Gaussian.

4.4 Ablation Study

In this section, we conduct an ablation study to further analyze individual effectiveness of the design components in our method. Unless otherwise specified, we use ResNet-20 (He et al. 2016) and test it on a uniformly subsampled CIFAR-10 test set of size 1,000. We provide more ablations on the loss design and the detailed results in Appendix G.

Effect of λ. In CAT-RS, λ introduced in (9) controls the relative contribution of $L^{\texttt{high}}$ over $L^{\texttt{low}}$. Here, Figure 2(a) shows the impact of λ to the model on varying $\lambda \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$, assuming $\sigma = 0.5$. The results show that λ successfully balances the trade-off between robustness and clean accuracy (Zhang et al. 2019). In addition, Figure 2(c) further verifies that CAT-RS offers more effective trade-off compared to other baseline training methods, as further discussed later in this section.

Effect of M. We investigate the effect of the number of noise M. Figure 2(b) illustrates the approximate test certified accuracy with varying $M \in \{1, 2, 4, 8\}$. The robustness of the smoothed classifier increases as M increases, sacrificing its clean accuracy. For large M, the classifier can incorporate the information of many Gaussian noises and take advantage of increasing p_f (4). Therefore, the smoothed classifier can provide a more robust prediction.

Accuracy-robustness trade-off. To further validate that our method can exhibit a better trade-off between accuracy and robustness compared to other methods, we additionally compare the performance trends between clean accuracy and certified accuracy at $r = 2.0$ as we vary a hyperparameter to control the trade-off, *e.g.*, λ (9) in case of our method. We use $\sigma = 1.0$ on MNIST dataset for this experiment. We choose Consistency and SmoothMix for this comparison, considering that they also offer a single hyperparameter (namely λ and η, respectively) for the balance between accuracy and robustness similar to our method, while both generally achieve good performances among the baselines considered. The results plotted in Figure 2(c) show that CAT-RS indeed exhibits a higher trade-off frontier compared to both methods, which confirms the effectiveness of our method. More detailed results can be found in Appendix F.

5 Conclusion

This paper explores a close relationship between confidence and robustness, a natural property of smoothed classifiers yet neural networks cannot currently offer. We have successfully leveraged this to relax the hard-to-compute metric of adversarial robustness into an easier concept of prediction confidence. Consequently, we propose a practical training method that enables a sample-level control of adversarial robustness, which has been difficult in a conventional belief. We believe our work could be a useful step for the future research on exploring the interesting connection between adversarial robustness and *confidence calibration* (Guo et al. 2017), and even towards the *out-of-distribution generalization*, through the randomized smoothing framework.

Acknowledgments

This work was conducted by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD190031RD).

References

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*, volume 80, 274–283.

Balunovic, M.; and Vechev, M. 2020. Adversarial Training and Provable Defenses: Bridging the Gap. In *International Conference on Learning Representations*.

Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; and Madry, A. 2019. On evaluating adversarial robustness. arXiv:1902.06705.

Cohen, J.; Rosenfeld, E.; and Kolter, Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *International Conference on Machine Learning*, volume 97, 1310–1320.

Croce, F.; Andriushchenko, M.; and Hein, M. 2019. Provable Robustness of ReLU networks via Maximization of Linear Regions. In *Proceedings of Machine Learning Research*, volume 89, 2057–2066.

Croce, F.; and Hein, M. 2020. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In *International Conference on Learning Representations*.

Fischer, M.; Baader, M.; and Vechev, M. 2021. Scalable Certified Segmentation via Randomized Smoothing. In *International Conference on Machine Learning*, volume 139, 3340–3351.

Gehr, T.; Mirman, M.; Drachsler-Cohen, D.; Tsankov, P.; Chaudhuri, S.; and Vechev, M. 2018. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy*.

Gowal, S.; Dvijotham, K. D.; Stanforth, R.; Bunel, R.; Qin, C.; Uesato, J.; Arandjelovic, R.; Mann, T.; and Kohli, P. 2019. Scalable verified training for provably robust image classification. In *IEEE/CVF International Conference on Computer Vision*, 4842–4851.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, volume 70, 1321–1330.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *IEEE/CVF International Conference on Computer Vision*, 8340–8349.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Method to Improve Robustness and Uncertainty under Data Shift. In *International Conference on Learning Representations*.

Jeong, J.; Park, S.; Kim, M.; Lee, H.-C.; Kim, D.-G.; and Shin, J. 2021. SmoothMix: Training confidence-calibrated smoothed classifiers for certified robustness. In *Advances in Neural Information Processing Systems*, volume 34, 30153–30168.

Jeong, J.; and Shin, J. 2020. Consistency Regularization for Certified Robustness of Smoothed Classifiers. In *Advances in Neural Information Processing Systems*, volume 33, 10558–10570.

Jia, J.; Cao, X.; Wang, B.; and Gong, N. Z. 2020. Certified Robustness for Top-k Predictions against Adversarial Perturbations via Randomized Smoothing. In *International Conference on Learning Representations*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto.

Kumar, A.; Levine, A.; Feizi, S.; and Goldstein, T. 2020. Certifying Confidence via Randomized Smoothing. In *Advances in Neural Information Processing Systems*, volume 33, 5165–5177.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy*, 656–672. IEEE.

Lee, G.-H.; Yuan, Y.; Chang, S.; and Jaakkola, T. 2019. Tight Certificates of Adversarial Robustness for Randomly Smoothed Classifiers. In *Advances in Neural Information Processing Systems*, volume 32.

Li, B.; Chen, C.; Wang, W.; and Carin, L. 2019. Certified Adversarial Robustness with Additive Noise. In *Advances in Neural Information Processing Systems*, 9464–9474.

Li, L.; Qi, X.; Xie, T.; and Li, B. 2021a. SoK: Certified Robustness for Deep Neural Networks. arXiv:2009.04131.

Li, L.; Weber, M.; Xu, X.; Rimanic, L.; Kailkhura, B.; Xie, T.; Zhang, C.; and Li, B. 2021b. TSS: Transformation-Specific Smoothing for Robustness Certification. In *ACM SIGSAC Conference on Computer and Communications Security*, 535–557. ISBN 9781450384544.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Mirman, M.; Gehr, T.; and Vechev, M. 2018. Differentiable Abstract Interpretation for Provably Robust Neural Networks. In *International Conference on Machine Learning*, volume 80, 3578–3586.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. DeepFool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

Mu, N.; and Gilmer, J. 2019. MNIST-C: A Robustness Benchmark for Computer Vision. arXiv:1906.02337.

Rosenfeld, E.; Winston, E.; Ravikumar, P.; and Kolter, Z. 2020. Certified Robustness to Label-Flipping Attacks via Randomized Smoothing. In *International Conference on Machine Learning*, volume 119, 8230–8241.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020a. Do Adversarially Robust ImageNet Models Transfer Better? In *Advances in Neural Information Processing Systems*, volume 33, 3533–3545.

Salman, H.; Jain, S.; Wong, E.; and Madry, A. 2022. Certified patch robustness via smoothed vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15137–15147.

Salman, H.; Li, J.; Razenshteyn, I.; Zhang, P.; Zhang, H.; Bubeck, S.; and Yang, G. 2019. Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers. In *Advances in Neural Information Processing Systems*, 11289–11300.

Salman, H.; Sun, M.; Yang, G.; Kapoor, A.; and Kolter, J. Z. 2020b. Denoised Smoothing: A Provable Defense for Pretrained Classifiers. In *Advances in Neural Information Processing Systems*, volume 33, 21945–21957.

Sun, J.; Mehra, A.; Kailkhura, B.; Chen, P.-Y.; Hendrycks, D.; Hamm, J.; and Mao, Z. M. 2021. Certified Adversarial Defenses Meet Out-of-Distribution Corruptions: Benchmarking Robustness and Simple Baselines. arXiv:2112.00659.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On Adaptive Attacks to Adversarial Example Defenses. In *Advances in Neural Information Processing Systems*, volume 33.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*.

Wang, B.; Jia, J.; Cao, X.; and Gong, N. Z. 2021. Certified Robustness of Graph Neural Networks against Adversarial Structural Perturbation. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1645–1653. ISBN 9781450383325.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving Adversarial Robustness Requires Revisiting Misclassified Examples. In *International Conference on Learning Representations*.

Wong, E.; and Kolter, Z. 2018. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *International Conference on Machine Learning*, volume 80, 5286–5295.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial Weight Perturbation Helps Robust Generalization. In *Advances in Neural Information Processing Systems*, volume 33, 2958–2969.

Wu, F.; Li, L.; Huang, Z.; Vorobeychik, Y.; Zhao, D.; and Li, B. 2022. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. In *International Conference on Learning Representations*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747.

Xiao, K. Y.; Tjeng, V.; Shafiullah, N. M. M.; and Madry, A. 2019. Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability. In *International Conference on Learning Representations*.

Yang, G.; Duan, T.; Hu, J. E.; Salman, H.; Razenshteyn, I.; and Li, J. 2020. Randomized Smoothing of All Shapes and Sizes. In *International Conference on Machine Learning*, volume 119, 10693–10705.

Zhai, R.; Dan, C.; He, D.; Zhang, H.; Gong, B.; Ravikumar, P.; Hsieh, C.-J.; and Wang, L. 2020. MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius. In *International Conference on Learning Representations*.

Zhang, D.; Ye, M.; Gong, C.; Zhu, Z.; and Liu, Q. 2020a. Black-Box Certification with Randomized Smoothing: A Functional Optimization Based Framework. In *Advances in Neural Information Processing Systems*, volume 33, 2316–2326.

Zhang, H.; Chen, H.; Xiao, C.; Gowal, S.; Stanforth, R.; Li, B.; Boning, D.; and Hsieh, C.-J. 2020b. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *International Conference on Learning Representations*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; Ghaoui, L. E.; and Jordan, M. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*, volume 97, 7472–7482.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020c. Attacks Which Do Not Kill Training Make Adversarial Learning Stronger. In *International Conference on Machine Learning*, volume 119, 11278–11287.

Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. 2021. Geometry-aware Instance-reweighted Adversarial Training. In *International Conference on Learning Representations*.