Robust Models are less Over-Confident

Julia Grabinski
Fraunhofer ITWM, Kaiserslautern
Visual Computing, University of Siegen
julia.grabinski@itwm.fraunhofer

Paul Gavrikov
IMLA, Offenburg University

Janis Keuper
Fraunhofer ITWM, Kaiserslautern
IMLA, Offenburg University

Margret Keuper
University of Siegen
Max Planck Institute for Informatics
Saarland Informatics Campus Saarbrücken

Abstract

Despite the success of convolutional neural networks (CNNs) in many academic benchmarks for computer vision tasks, their application in the real-world is still facing fundamental challenges. One of these open problems is the inherent lack of robustness, unveiled by the striking effectiveness of adversarial attacks. Current attack methods are able to manipulate the network's prediction by adding specific but small amounts of noise to the input. In turn, adversarial training (AT) aims to achieve robustness against such attacks and ideally a better model generalization ability by including adversarial samples in the trainingset. However, an in-depth analysis of the resulting robust models beyond adversarial robustness is still pending. In this paper, we empirically analyze a variety of adversarially trained models that achieve high robust accuracies when facing state-of-the-art attacks and we show that AT has an interesting side-effect: it leads to models that are significantly less overconfident with their decisions, even on clean data than non-robust models. Further, our analysis of robust models shows that not only AT but also the model's building blocks (like activation functions and pooling) have a strong influence on the models' prediction confidences.
Data & Project website: https://github.com/GeJulia/robustness_confidences_evaluation

1 Introduction

Convolutional Neural Networks (CNNs) have been shown to successfully solve problems across various tasks and domains. However, distribution shifts in the input data can have a severe impact on the prediction performance. In real-world applications, these shifts may be caused by a multitude of reasons including corruption due to weather conditions, camera settings, noise, and maliciously crafted perturbations to the input data intended to fool the network (adversarial attacks). In recent years, a vast line of research (e.g. [25, 36, 44]) has been devoted to solving robustness issues, highlighting a multitude of causes for the limited generalization ability of networks and potential solutions to facilitate the training of better models.

A second, yet equally important issue that hampers the deployment of deep learning based models in practical applications is the lack of calibration concerning prediction confidences. In fact, most models are overly confident in their predictions, even if they are wrong [31, 45, 57]. Specifically, most conventionally trained models are unaware of their own lack of expertise, i.e. they are trained to make confident predictions in any scenario, even if the test data is sampled from a previously unseen domain. Adversarial examples seem to leverage this weakness, as they are known to not only fool the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

network but also to cause very confident wrong predictions [46]. In turn, adversarial training (AT) has shown to improve the prediction accuracy under adversarial attacks [22, 25, 65, 87]. However, only few works so far have been investigating the links between calibration and robustness [45, 60], leaving a systematic synopsis of adversarial robustness and prediction confidence still pending.

In this work, we provide an extensive empirical analysis of diverse adversarially robust models with regard to their prediction confidences. Therefore, we evaluate more than 70 adversarially robust models and their conventionally trained counterparts, which show low robustness when exposed to adversarial examples. By measuring their output distributions on benign and adversarial examples for correct and erroneous predictions, we show that adversarially trained models have benefits beyond adversarial robustness and are less over-confident.

To cope with the lack of calibration in conventionally trained models, Corbière et al. [13] propose to rather use the true class probability than the standard confidence obtained after the Softmax layer, such as to circumvent the overlapping confidence values for wrong and correct predictions. However, we observe that exactly these overlaps are an indicator for insufficiently calibrated models and can be mitigated by the improvement of CNNs building blocks, namely downsampling and activation functions, that have been proposed in the context of adversarial robustness [17, 28].

Our work analyzes the relationship between robust models and model confidences. Our experiments for 71 robust and non-robust model pairs on the datasets CIFAR10 [43], CIFAR100 and ImageNet [19] confirm that non-robust models are overconfident with their false predictions. This highlights the challenges for usage in real-world applications. In contrast, we show that robust models are generally less confident in their predictions, and, especially CNNs which include improved building blocks (downsampling and activation) turn out to be better calibrated manifesting low confidence in wrong predictions and high confidence in their correct predictions. Further, we can show that the prediction confidence of robust models can be used as an indicator for erroneous decisions. However, we also see that adversarially trained networks (robust models) overfit adversaries similar to the ones seen during training and show similar performance on unseen attacks as non-robust models. Our contributions can be summarized as follows:

- We provide an extensive analysis of the prediction confidence of 71 adversarially trained models (**robust models**), and their conventionally trained counterparts (**non-robust models**). We observe that most non-robust models are exceedingly over-confident while robust models exhibit less confidence and are better calibrated for slight domain shifts.
- We observe that specific layers, that are considered to improve model robustness, also impact the models' confidences. In detail, improved downsampling layers and activation functions can lead to an even better calibration of the learned model.
- We investigate the detection of erroneous decisions by using the prediction confidence. We observe that robust models are able to detect wrong predictions based on their confidences. However, when faced with unseen adversaries they exhibit a similarly weak performance as non-robust models.

Our analysis provides a first synopsis of adversarial robustness and model calibration and aims to foster research that addresses both challenges jointly rather than considering them as two separate research fields. To further promote this research, we released our modelzoo[1].

2 Related Work

In the following, we first briefly review the related work on model calibration which motivates our empirical analysis. Then, we revise the related work on adversarial attacks and model hardening.

Confidence Calibration. For many models that perform well with respect to standard benchmarks, it has been argued that the robust or regular model accuracy may be an insufficient metric [2, 13, 18, 79], in particular when real-world applications with potentially open-world scenarios are considered. In these settings, reliability must be established which can be quantified by the prediction confidence [58]. Ideally, a reliable model would provide high confidence predictions on correct classifications, and low confidence predictions on false ones [13, 57]. However, most networks are not able to

[1] `https://github.com/GeJulia/robustness_confidences_evaluation`

instantly provide a sufficient calibration. Hence, confidence calibration is a vivid field of research and proposed methods are based on additional loss functions [32, 35, 45, 48, 52], on adaptions of the training input by label smoothing [54, 60, 63, 75] or on data augmentation [20, 45, 76, 88]. Further, [58] present a benchmark on classification models regarding model accuracy and confidence under dataset shift. Various evaluation methods have been provided to distinguish between correct and incorrect predictions [13, 56]. Naeini et al. [56] defined the networks *expected calibration error* (ECE) for a model f by with $0 \leq p \leq \infty$

$$\mathrm{ECE}_p = \mathbb{E}[|\hat{z} - \mathbb{E}[1_{\hat{y}=y}|\hat{z}]|^p]^{\frac{1}{p}} \tag{1}$$

where the model f predicts $\hat{y} = y$ with the confidence $\hat{z}$. This can be directly related to the over-confidence $o(f)$ and under-confidence $u(f)$ of a network as follows [81]:

$$|o(f)\mathbb{P}(\hat{y} \neq y) - u(f)\mathbb{P}(\hat{y} = y)| \leq \mathrm{ECE}_p, \tag{2}$$

where [55]

$$o(f) = \mathbb{E}[\hat{z}|\hat{y} \neq y] \quad u(f) = \mathbb{E}[1 - \hat{z}|\hat{y} = y], \tag{3}$$

i.e. the over-confidence measures the expectation of $\hat{z}$ on wrong predictions, under-confidence measures the expectation of $1 - \hat{z}$ on correct predictions and ideally both should be zero. The ECE provides an upper bound for the difference between the probability of the prediction being wrong weighted by the networks over-confidence and the probability of the prediction being correctly weighted by the networks under-confidence and converges to this value for the parameter $p \to 0$ (in eq. 1]). We also recur to this metric as an aggregate measure to evaluate model confidence. Yet, it should be noted that the ECE metric is based on the assumption that networks make correct as well as incorrect predictions. A model that always makes incorrect predictions and is less confident in its few correct decisions than it is in its many erroneous decisions can end up with a comparably low ECE. Therefore, ECE values for models with an accuracy below 50% are hard to interpret.

Most common CNNs are over-confident [31, 45, 57]. Moreover, the most dominantly used activation in modern CNNs [34, 39, 69, 73] remains the ReLU function, while is has been pointed out by Hein et al. [35] that ReLUs cause a general increase in the models' prediction confidences, regardless of the prediction validity. This is also the case for the vast majority of the adversarially trained models we consider, except for the model by [17] to which we devote particular attention.

Adversarial Attacks. Adversarial attacks intentionally add perturbations to the input samples, that are almost imperceptible to the human eye, yet lead to (high-confidence) false predictions of the attacked model [25, 53, 74]. These attacks can be classified into two categories: white-box and black-box attacks. In black-box attacks, the adversary has no knowledge of the model intrinsics [4], and can only query its output. These attacks are often developed on surrogate models [10, 42, 78] to reduce interaction with the attacked model in order to prevent threat detection. In general, though, these attacks are less powerful due to their limited access to the target networks. In contrast, in white-box attacks, the adversary has access to the full model, namely the architecture, weights, and gradient information [25, 44]. This enables the attacker to perform extremely powerful attacks customized to the model. One of the earliest approaches, the *Fast Gradient Sign Method* (FGSM) by [25] uses the sign of the prediction gradient to perturb input samples into the direction of the gradient, thereby increasing the loss and causing false predictions. This method was further adapted and improved by *Projected Gradient Descent* (PGD) [44], *DeepFool* (DF) [53], *Carlini and Wagner* (CW) [5] or *Decoupling Direction and Norm* (DDN) [65]. While FGSM is a single-step attack, meaning that the perturbation is computed in one single gradient ascent step limited by some ϵ bound, multi-step attacks such as PGD iteratively search perturbations within the ϵ-bound to change the models' prediction. These attacks generally perform better but come at an increased cost of the attack. *AutoAttack* [14] is an ensemble of different attacks including an adaptive version of PGD, and has been proposed as a baseline for adversarial robustness. In particular, it is used in robustness benchmarks such as RobustBench [15].

Adversarial Training and Robustness. To improve robustness, adversarial training (AT) has proven to be quite successful on common robustness benchmarks. Some attacks can be simply defended by using their adversarial examples in the training set [25, 65] through an additional loss [22, 87]. Furthermore, the addition of more training data, by using external data, or data augmentation techniques such as the generation of synthetic data, has been shown to be promising for more robust models [6, 26, 27, 62, 68, 80]. RobustBench [15] provides a leaderboard to study the improvements made by the aforementioned approaches in a comparable manner in terms of their robust accuracy.

Madry et al. [50] observed that the performance of adversarial training depends on the models' capacity. High-capacity models are able to fit the (adversarial) training data better, leading to increased robust accuracy. Later research investigated the influence on increased model width and depth [26, 85], and quality of convolution filters [24]. Consequently, the best-performing entries on RobustBench [15] are often using Wide-ResNet-70-16's or even larger architectures. Besides this trend, concurrent works also started to additionally modify specific building blocks of CNNs [17, 29]. Grabinski et al. [28] showed that weaknesses in simple AT, like FGSM, can be overcome by improving the network's downsampling operation.

Adversarial Training and Calibration. Only a few but notable prior works such as [45, 60] have investigated adversarial training with respect to model calibration. Without providing a systematic overview, [45] show that AT can help to smoothen the prediction distributions of CNN models. Qin et al. [60] investigate adversarial data points generated using [5] with respect to non-robust models and find that easily attackable data points are badly calibrated while adversarial models have better calibration properties. In contrast, we analyze the robustness and calibration of pairs of robust and non-robust versions of the same models rather than investigating individual data points. [77] introduce an adversarial calibration loss to reduce the calibration error. Further, [72] propose confidence calibrated adversarial training to force adversarial samples to show uniform confidence, while clean samples should be one hot encoded. Complementary to [15], we provide an analysis of the predictive confidences of adversarially trained, robust models and release conventionally trained counterparts of the models from [15] to facilitate future research on the analysis of the impact of training schemes versus architectural choices. Importantly, our proposed large-scale study allows a differentiated view on the relationship between adversarial training and model calibration, as discussed in Section 3. In particular, we find that adversarially trained models are not always better calibrated than vanilla models especially on clean data, while they are consistently less over-confident.

Adversarial Attack Detection. A practical defense besides adversarial training, can also be established by the detection and rejection of malicious input. Most detection methods are based on input sample statistics [23, 30, 33, 37, 47, 49], while others attempt to detect adversarial samples via inference on surrogate models, yet these models themselves might be vulnerable to attacks [12, 51]. While all of these approaches perform additional operations on top of the models' prediction, we show that simply taking the models' prediction confidence can be used as a heuristic to reject erroneous samples.

3 Analysis

In the following, we first describe our experimental setting in which we then conduct an extensive analysis on the two CIFAR datasets with respect to robust and non-robust model[2] confidence on clean and perturbed samples as well as their ECE. Further, we observe by computing the ROC curves of these models that robust models are best suited to distinguish between correct and incorrect predictions based on their confidence. In addition we point out that the improvement of pooling operations or activation functions within the network can enhance the models' calibration further. Last, we also investigate ImageNet as a high resolution dataset and observe that the model with the highest capacity and AT can achieve the best performance results and calibration.

3.1 Experimental Setup

We have collected 71 checkpoints of robust models [1, 3, 7–9, 11, 16, 17, 21, 22, 26, 27, 38, 40, 41, 59, 61, 62, 64, 67, 68, 70, 71, 80, 83, 84, 86, 87, 89, 90] listed on the ℓ_∞-RobustBench leaderboard [15]. Additionally, we compare each appearing architecture to a second model *trained without AT or any specific robustness regularization, and without any external data* (even if the robust counterpart relied on it). Training details can be found in appendix A.

Then we collect the predictions alongside their respective confidences of robust and non-robust models on clean validation samples, as well as on samples attacked by a white-box attack (PGD), and a black-box attack (Squares). PGD (and its adaptive variant APGD [14]) is the most widely used white-box attack and adversarial training schemes explicitly (when using PGD samples for

[2]The classification into robust and non-robust models is based on the models' robustness against adversarial attacks. We consider a model to be robust when it achieves considerably high accuracy under AutoAttack [14].

training) or implicitly (when using the faster but strongly related FGSM attack samples for training) optimize for PGD robustness. In contrast, the *Squares* attack alters the data at random with an allowed budget until the label flips. Such samples are rather to be considered out-of-domain samples even for adversarially trained models and provide a proxy for a model's generalization ability. Thus, *Squares* can be seen as unseen attack for all models while PGD might be not for some adversarially trained, robust models.

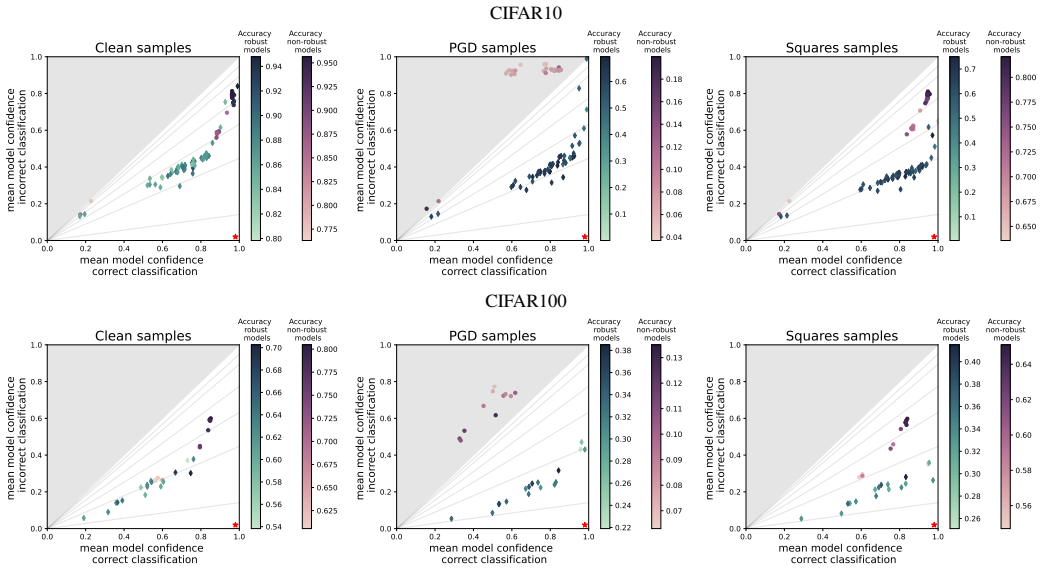

Figure 1: Mean model confidences on their correct (x-axis) and incorrect (y-axis) predictions over the full CIFAR10 dataset (top) and CIFAR100 dataset (bottom), clean (left) and perturbed with the attacks PGD (middle) and Squares (right). Each point represents a model. Circular points (purple color-map) represent non-robust models and diamond-shaped points (green color-map) represent robust models. The color of each point represents the models accuracy, darker signifies higher accuracy (better) on the given data samples. The star in the bottom right corner indicates the optimal model calibration and the gray area marks the area were the confidence distribution of the network is worse than random, i.e. more confident in incorrect predictions than in correct ones.

3.2 CIFAR Models

CIFAR10 [43] is a simple ten class dataset consisting of 50,000 training and 10,000 validation images with a resolution of 32×32. Since it is significantly cheaper to train on CIFAR10 in comparison to e. g. ImageNet, and its low resolution allows to discount additional costs of adversarial training, most entries on RobustBench [15] focus on CIFAR10.

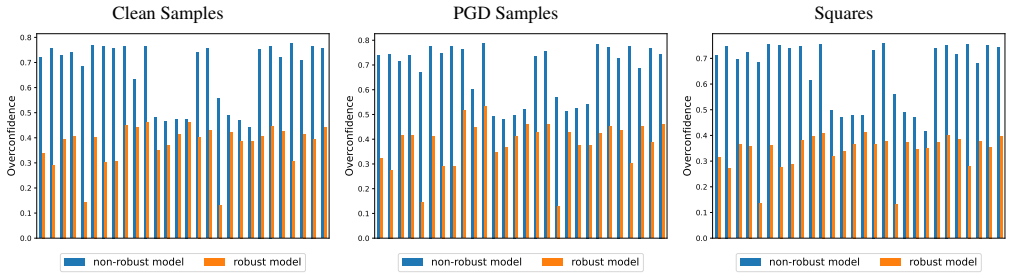

Figure 2: Overconfidence (lower is better) bar plots of robust models and their non-robust counterparts trained on CIFAR10. Non-robust models are highly overconfident, in contrast, their robust counterparts are less over-confident.

Figure 1 shows an overview of all robust and non-robust models trained on CIFAR10 in terms of their accuracy as well as their confidence in their correct and incorrect predictions. Along the isolines, the ratio between confidence in correct and incorrect predictions is constant. The gray

area indicates scenarios where models are even more confident in their incorrect predictions than in their correct predictions. Concentrating on the models' confidence, we can see that robust models (marked by a diamond) are in general less confident in their predictions, while non-robust models (marked by a circle) exhibit high confidence in all their predictions, both correct and incorrect. This indicates that non-robust models are not only more susceptible to (adversarial) distribution shifts but are also highly over-confident in their false predictions. Practically, such behaviour can lead to catastrophic consequences in safety-related, real-world applications. Robust models tend to have lower average confidence and a favorable confidence trade-off even on clean data (Figure 1, top left). When adversarial samples using PGD are considered (Figure 1, top middle), the non-robust models even fall into the gray area of the plot where more confident decisions are likely incorrect. As expected, adversarially trained models not only make fewer mistakes in this case but are also better adjusted in terms of their confidence. Black-box attacks (Figure 1,top right) provide non-targeted out of domain samples. Adversarially trained models are overall better calibrated to this case, i.e. their mean confidences are hardly affected whereas non-robust models' confidences fluctuate heavily.

Samples \ Robustness	Clean	PGD	Squares
non-robust models	0.6736 ± 0.1208	0.6809 ± 0.1061	0.6635 ± 0.1156
robust models	0.1894 ± 0.1531	0.2688 ± 0.1733	0.2126 ± 0.1431

Table 1: Mean ECE (lower is better) and standard deviation over all non-robust model versus all their robust counterparts trained on CIFAR10. Robust model exhibit a significantly lower ECE on all samples.

Four models stand out in Figure 1 (top left): two robust and two non-robust models which are much less confident in their true and false predictions than others. These less confident models are indeed trained from two different model architectures, with and without adversarial training. [59] uses a hypersphere embedding which normalizes the features in the intermediate layers and weights in the softmax layer, the other model [11] uses an ensemble of three different pretrained models (ResNet-50) to boost robustness. These architectural changes have a significant impact on the absolute model confidence, yet, do not necessarily lead to a better calibration. These models are under-confident in their correct predictions and tend to be comparably confident in wrong predictions.

Table 1 reports the mean ECE over all robust models and their non-robust counterparts. Robust models are better calibrated which results in a significantly lower ECE [3]. Figure 13 further visualizes the significant decrease in over-confidence of robust models w.r.t. their non-robust counterparts.

CIFAR100, although otherwise similar to CIFAR10, includes 100 classes and can be seen as a more challenging classification task. This is reflected in the reduced model accuracy on the clean and adversarial samples (Figure 1 , bottom). On this data, robust models are again less over-confident. They are slightly closer to the optimal calibration point in the lower right corner even on clean data and perform significantly better on PGD samples where the confidences of non-robust models are again reversed (middle). The Squares attack again illustrates the stable behavior of robust models' confidences[4]. We also report the ECE values for CIFAR100 in the Appendix. Please note that the accuracy of the CIFAR100 models is not very high (ranging between 56.87% and 70.25% even for clean samples), resulting in an unreliable calibration metric. Especially under PGD attacks, non-robust networks make mostly incorrect predictions such that the ECE collapses to being the expected confidence value of incorrect predictions (see eq. [1]), regardless of the confidences of the few correct predictions. In this case, ECE is not meaningful.

Another interesting observation is that non-robust models can achieve higher accuracy on the clean data and, quite surprisingly, on the applied black-box attacks (Figure 1, right). This indicates that most robust models overfit white-box attacks used during training and are not generalizing very well to other attacks. While making more mistakes, robust models still have a favorable distribution of confidence over non-robust models in this case.

Model confidences can predict erroneous decisions. Next, we evaluate the prediction confidences in terms of their ability to predict whether a network prediction is correct or incorrect. We visualize the ROC curves for all models and compare the averages of robust and non-robust models in Figure 3 (top row for CIFAR10, bottom row for CIFAR100), which allows us to draw conclusions about the confidence behavior. While robust and non-robust models perform on average very similarly on clean

[3]The models' full empirical confidence distributions are given in Figure 10 in the Appendix.

[4]The models' full empirical confidence distributions are given in Figure 11 in the Appendix

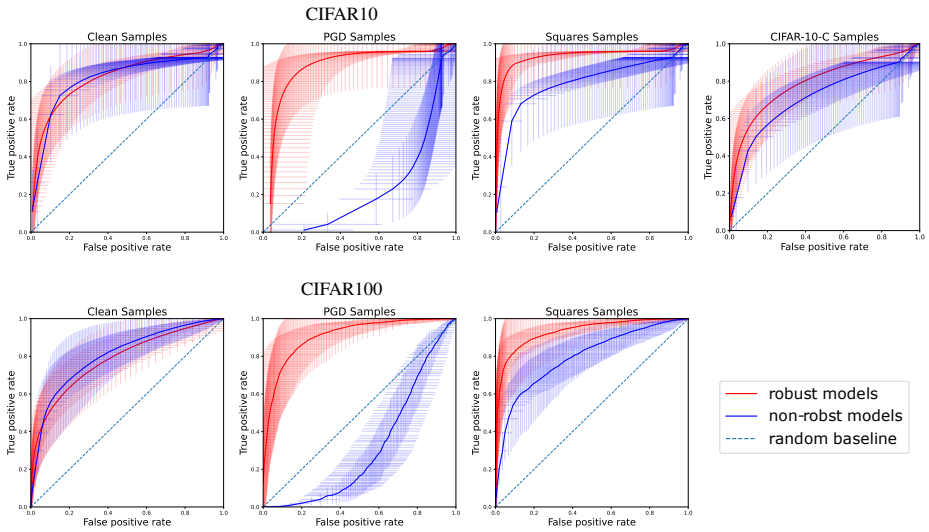

Figure 3: Average ROC curve for all robust and all non-robust models trained on CIFAR10 (top) and CIFAR100 (bottom). Standard deviation is marked by the error bars. The dashed line would mark a model which has the same confidence for each prediction. We observe that the models confidences can be an indicator for the correctness of the prediction. However, on PGD samples the non-robust models fail while the robust models can distinguish correct from incorrect predictions based on the prediction confidence.

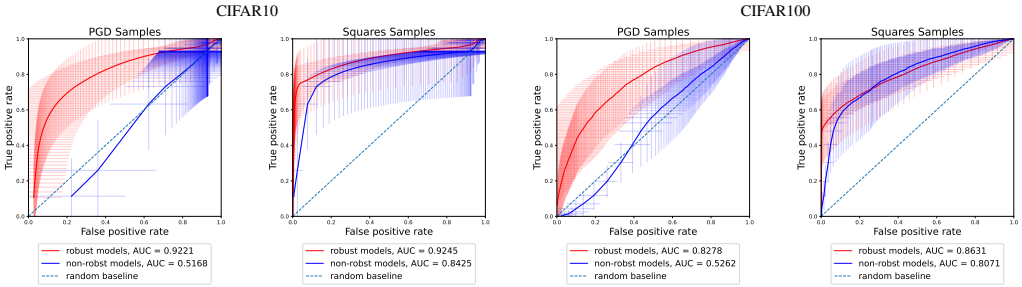

Figure 4: Average ROC curve over all robust and non-robust models of confidence on clean correctly classified samples and perturbated wrongly classified samples. The robust model confidences can be used as threshold for detection of white-box adversarial attacks (PGD). For black-box adversarial attacks (Squares) the robust as well as non-robust models can partially detect the erroneous samples.

data, robust model confidences can reliably predict erroneous classification results on adversarial examples where non-robust models fail. Also, for out-of-domain samples from the black-box attack *Squares* (middle right) and common corruptions [36] (right), robust models can reliably assess their prediction quality and can better predict whether their classification result is correct.

Robust model confidences can detect adversarial samples. Further, we evaluate the adversarial detection rate of the robust models based on their ROC curves (averaged over all robust models) in Figure 4, comparing the confidence of correct predictions on clean samples and incorrect predictions caused by adversarial attacks. We observe different behavior for gradient-based, white-box attacks and black-box attacks. While non-robust models fail completely against gradient based attacks they are almost as good as robust models for the detection of black-box attacks. Similarly, when taking the left two plots from Figure 3 into account, one might get the impression that non-robust models perform similar or even better on detecting erroneous samples compared to robust ones. Thus, we hypothesize that robust models indeed overfit the adversaries seen during training, as those are mostly gradient-based adversaries. Therefore we assume that adversarially trained models are not better calibrated in general, however, when strictly looking at overconfidence robust models are consistently less overconfident and therefore better applicable for safety critical applications.

Downsampling techniques. Most common CNNs apply downsampling to compress featuremaps with the intent to increase spatial invariance and overall higher sparsity. However, Grabinski et al. [29] showed that aliasing during the downsampling operation highly correlates with the lack of adversarial robustness, and provided a new downsampling operation, called *frequency low cut pooling* [28], which enables improved downsampling of the featuremaps. Figure 6 compares the confidence distribution of three different networks. The top row shows a PRN-18 baseline without adversarial training, the second row the approach by Grabinski et al. [28] applied to the same architecture (additional models are evaluated in the appendix D), and the third row shows a robust model trained by Rebuffi et al. [62]. The baseline model is highly susceptible to adversarial attacks, especially under white-box attacks, while the two robust counter-parts remain low-confident in false predictions, and show higher confidence in correct predictions. However, while the model of Rebuffi et al. [62] shows a high variance amongst the predicted confidences, the approach by Grabinski et al. [28] significantly improves this by disentangling the confidences. Their model provides low-variance and high-confidence on correct predictions and reduced confidence on false predictions across all evaluated samples.

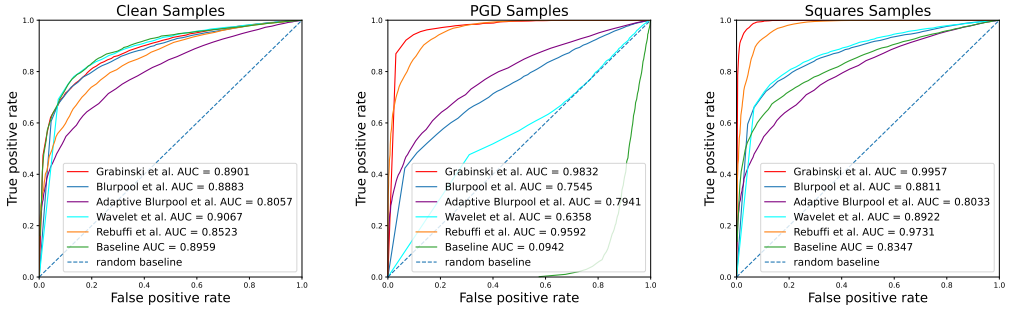

Figure 5: ROC curves and AUC values for different pooling variation in combination with adversarial training. FLC Pooling [28] outperforms all other pooling methods as well as the baseline.

In Figure 5, we compare different pooling methods combined with AT to standard pooling with AT as well as standard pooling without AT. The results show that the pooling method by Grabinski et al. [28] outperforms all other pooling methods. They consistently achieve the highest AUC under adversarial samples (white- and black-box attack) and are similar to the baseline on clean samples.

Activation functions. Next, we analyze the influence of activation functions. Only one RobustBench model utilizes an activation other than ReLU. Dai et al. [17] introduce learnable activation functions with the intent to improve robustness. Figure 7 shows at the top row a WRN-28-10 baseline model without AT, the model by Dai et al. [17] in the middle and a model with the same architecture adversarially trained by Carmon et al. [6].
Although this is an arguably sparse basis for a thorough investigation, we observe that the model by [17] can retain high confidence in correct predictions for both clean and perturbed samples. Furthermore, the model is much less confident in its wrong predictions for the clean as well as the adversarial samples. Similar to the used pooling variation, also the activation function seems to influence the model's calibration.

Summary of low resolution datasets. On CIFAR10 and CIFAR100 non-robust models can achieve higher standard accuracy and at least match or even exceed the performance of robust models under black-box attacks like *Squares*. Only under the white-box attack PGD, the robust models show higher accuracy. However non-robust models are highly over-confident in all their predictions and are hence limited in their applicability for real-world tasks. In contrast, the correctness of a robust models' prediction can be estimated by the prediction confidence. and is additionally serving as a defence against adversarial attacks. Further, we observe that the confidence of non-robust models decreases with increasing task complexity. In contrast, robust models are less affected by the increased task complexity and exhibit similar confidence characteristics on both datasets.

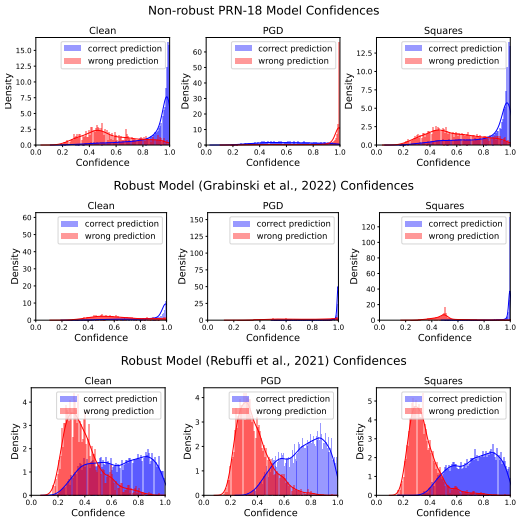

Figure 6: Confidence distribution on three different PRN-18. The first row shows a model without adversarial training and standard pooling, the second row the model by Grabinski et al. [28] which uses flc pooling instead of standard pooling and the third row shows the model by Rebuffi et al. [62] adversarially trained and with standard pooling.

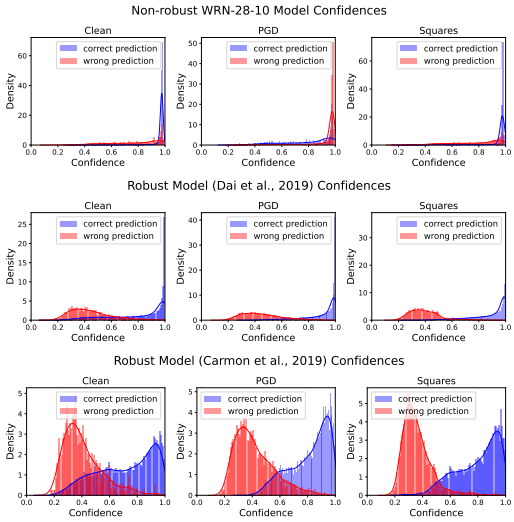

Figure 7: Confidence distribution on three different WRN-28-10. The first row shows a model without adversarial training and standard activation (ReLU), the second row the model by Dai et al. [17] which uses learnable activation functions instead of fixed ones and the third row shows the model by Carmon et al. [6] adversarially trained and with the standard activation (ReLU).

3.3 ImageNet

We rely on the models provided by RobustBench [15] for our ImageNet evaluation. We report the clean and robust accuracy against *PGD* and *Squares* in Table 4 in the appendix. The non-robust model, trained without AT, achieves the highest performance on clean samples but collapses under white- and black-box attacks. Further, the models trained with multistep adversaries by Engstrom et al. [22] and Salman et al. [66] achieve higher robust and clean accuracy than the model trained by Wong et al. [83] which is trained with single-step adversaries. Moreover, the largest model, a WRN-50-2, yields the best robust performance. Still, the amount of robust networks on ImageNet is quite small, thus we can not make any generalized assumptions. Figure 9 shows the precision-recall curve for our evaluated models. Under evaluation with clean samples, the non-robust model without AT performs best. Under both attacks the largest model (a WRN-50-2 by Salman et al. [66]) performs best and the worst performer is the smallest model (RN-18). This may be suggesting that bigger models can not only achieve the better trade-off in clean and robust accuracy but also more successfully disentangle confidences between correct and incorrect predictions. Figure 8 confirms that the over-confidence is decreased in robust models and the ECE is lower than in the non-robust models.

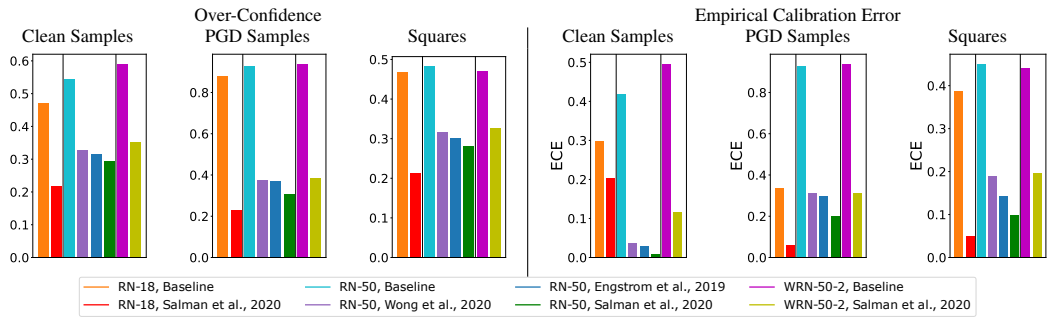

Figure 8: Overconfidence (left) and ECE (right) (lower is better) bar plots of the models trained on ImageNet provided by RobustBench [15] and their non-robust counterparts. The non-robust baselines exhibits the highest overconfidence and ECE. In contrast, the robust models are better calibrated.

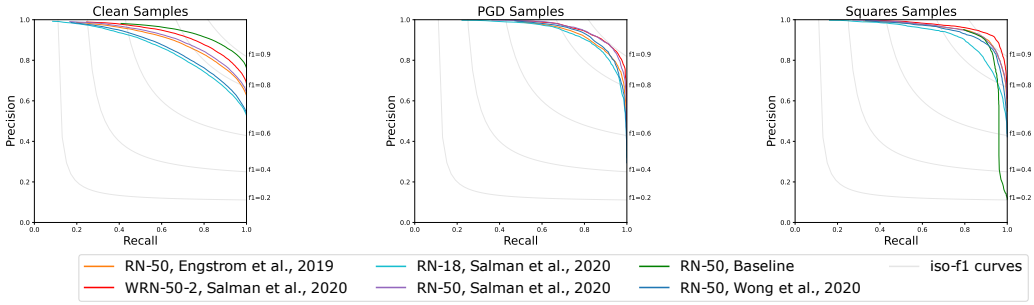

Figure 9: Precision Recall curves for the classification of correct versus erroneous predictions based on the confidence on ImageNet, evaluated over 10,000 samples. Robust and non-robust models are taken from RobustBench [15]. For clean samples (left) the non robust baseline performs best, while its confidences are less reliable under attack (middle and right). The robust WRN-50-2 by Salman et al. [66] performs best on the PGD and Squares samples.

3.4 Discussion

Our experiments confirm that the prediction confidences of non-robust models are highly over-confident, especially under gradient based, white-box attacks. However, when confronted with clean samples, common corruptions or unseen black-box attacks like Squares [4] non-robust and robust models are equally able to detect wrongly classified samples based on their prediction confidence. Indicating that adversarially trained networks overfit the kind of adversaries seen during training.

Further, our results indicate that the selection of the activation functions as well as the downsampling are important factors for the models' performance and confidence. The method by Grabinski et al. [28], which improves the downsampling, as well as the method by Dai et al. [17], which improves the activation function, exhibit the best calibration for the networks prediction; High confidence on correct predictions and low confidence on the incorrect ones. While further optimizing deep neural networks' architectures and training schemes, we should consider the synopsis of model robustness and calibration instead of optimizing each of these aspects separately.

Limitations. Our evaluation is based on the models provided on RobustBench [15]. Thus the amount of networks on more complex datasets, like ImageNet, is rather small and therefore the evaluation not universally applicable. While the number of models for CIFAR is large, the proposed database can only be understood as a starting point for future research. This is particularly true for the analysis of neural network building blocks - models that are adversarially trained and employ smooth activation functions might be very promising concerning their calibration but a more in-depth analysis of this setting with new, dedicated datasets is desirable. Additionally, we rely simply on the confidence obtained after the Softmax layer, while there are many other metrics for uncertainty measurement.

4 Conclusion

We provide an extensive study on the confidences of robust models and observe an overall trend: robust models tent to be less over-confident than non-robust models. Thus, while achieving a higher robust accuracy, adversarial training generates models that are less overconfident. Further, the prediction confidence of robust models can actually be used to reject wrongly classified samples on clean data and even adversarial examples.
Moreover, we see indications that exchanging simple building blocks like the activation function [17] or the downsampling method [28] alters the properties of robust models with respect to confidence calibration. On the examples we investigate, the models' prediction confidence on their correct predictions can be increased while the confidence on the erroneous predictions remains low. Our findings should nurture future research on jointly considering model calibration and robustness.
However, robust models' overall performance on robustness tasks are highly questionable as they seem to overfit the adversaries seen during training.

References

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

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL `https://arxiv.org/abs/1606.06565`.

[3] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

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

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

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

[7] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness, 2022.

[8] Erh-Chung Chen and Che-Rung Lee. Ltd: Low temperature distillation for robust adversarial training, 2021.

[9] Jinghui Chen, Yu Cheng, Zhe Gan, Quanquan Gu, and Jingjing Liu. Efficient robust training via backward smoothing, 2021.

[10] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[11] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020.

[12] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14453–14462, 2020.

[13] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[15] 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.

[16] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training, 2021.

[17] Sihui Dai, Saeed Mahloujifar, and Prateek Mittal. Parameterizing activation functions for adversarial robustness, 2021.

[18] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[20] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[21] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkeryxBtPB`.

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

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

[24] Paul Gavrikov and Janis Keuper. Adversarial robustness through the lens of convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 139–147, June 2022.

[25] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[26] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021.

[27] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021.

[28] Julia Grabinski, Steffen Jung, Janis Keuper, and Margret Keuper. Frequencylowcut pooling–plug & play against catastrophic overfitting. *arXiv preprint arXiv:2204.00491*, 2022.

[29] Julia Grabinski, Janis Keuper, and Margret Keuper. Aliasing and adversarial robust generalization of cnns. *Machine Learning*, pages 1–27, 2022.

[30] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

[31] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/guo17a.html`.

[32] Corina Gurau, Alex Bewley, and Ingmar Posner. Dropout distillation for efficiently estimating model confidence. *arXiv preprint arXiv:1809.10562*, 2018.

[33] Paula Harder, Franz-Josef Pfreundt, Margret Keuper, and Janis Keuper. Spectraldefense: Detecting adversarial attacks on cnns in the fourier domain. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[35] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem, 2019.

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

[37] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. *arXiv preprint arXiv:1608.00530*, 2016.

[38] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[39] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[40] Hanxun Huang, Yisen Wang, Sarah Monazam Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks, 2022.

[41] Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization, 2020.

[42] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

[43] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[44] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.

[45] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[46] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018. URL `https://arxiv.org/abs/1807.03888`.

[47] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, pages 5764–5772, 2017.

[48] Zhizhong Li and Derek Hoiem. Improving confidence estimates for unfamiliar examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2686–2695, 2020.

[49] Peter Lorenz, Paula Harder, Dominik Straßel, Margret Keuper, and Janis Keuper. Detecting autoattack perturbations in the frequency domain. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL `https://openreview.net/forum?id=8uW0Txbwo-Z`.

[50] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[51] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[52] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *international conference on machine learning*, pages 7034–7044. PMLR, 2020.

[53] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[54] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[55] Dennis Mund, Rudolph Triebel, and Daniel Cremers. Active online confidence boosting for efficient object classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1367–1373, 2015. doi: 10.1109/ICRA.2015.7139368.

[56] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[57] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[58] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

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

[60] Yao Qin, Xuezhi Wang, Alex Beutel, and Ed Chi. Improving calibration through the relationship with adversarial robustness. *Advances in Neural Information Processing Systems*, 34:14358–14369, 2021.

[61] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. URL `https://openreview.net/forum?id=BuD2LmNaU3a`.

[62] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness, 2021.

[63] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

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

[65] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019.

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

[67] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks, 2020.

[68] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness?, 2021.

[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[70] Chawin Sitawarin, Supriyo Chakraborty, and David Wagner. Sat: Improving adversarial training via curriculum-based loss smoothing, 2021.

[71] Kaustubh Sridhar, Oleg Sokolsky, Insup Lee, and James Weimer. Improving neural network robustness via persistency of excitation, 2021.

[72] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning*, pages 9155–9166. PMLR, 2020.

[73] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[74] 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*, 2014. URL `http://arxiv.org/abs/1312.6199`.

[75] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[76] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[77] Christian Tomani and Florian Buettner. Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9886–9896, 2021.

[78] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

[79] Kush R Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big data*, 5(3):246–255, 2017.

[80] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rklOg6EFwS`.

[81] Jonathan Wenger, Hedvig Kjellström, and Rudolph Triebel. Non-parametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR, 2020.

[82] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

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

[84] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

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

[86] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle, 2019.

[87] 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*, 2019.

[88] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[89] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger, 2020.

[90] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=iAX0l6Cz8ub.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 3.4
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide the model weights for the standard trained counterparts to the model architectures reported on RobustBench.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 3.1 and Section A
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We included mean and standard deviation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The training time for each normal training depends on the network architecture provided and was not tracked. The calculation of the model confidence can be simply done by collecting the models output after Softmax and does not require much computational effort or resources.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] RobustBench [15] as well as the papers used on their benchmark (section 3.1)
   (b) Did you mention the license of the assets? [Yes] Appendix section G
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide the model weights for the standard trained counterparts to the model architectures reported on RobustBench.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Non-robust Model Training

For training, *CIFAR-10/100* data was zero-padded by 4 px along each dimension, and then transformed using 32×32 px random crops, and random horizontal flips. Channel-wise normalization was replicated as reported by the original dataset authors. Training hyper parameters have been set to an initial learning rate of 1e-2, a weight decay of 1e-2, a batch-size of 256 and a nesterov momentum of 0.9. We scheduled the SGD optimizer to decrease the learning rate every 30 epochs by a factor of $\gamma = 0.1$ and trained for a total of 125 epochs. The loss is determined using Categorical Cross Entropy and we used the model obtained at the epoch with the highest validation accuracy. Training was executed on a *A+ Server* SYS-2123GQ-NART-2U machine with four *NVIDIA* A100-SXM4-40GB GPUs for approximately 17 GPU hours. Training *ImageNet1k* architectures with our hyperparameters resulted in a rather poor performance and we therefore rely on the baseline model without AT provided by *timm* [82].

B Additional Evaluation CIFAR10/100

In this section we provide an overview over ECE on CIFAR10 and CIFAR100 of all robust models and their non-robust conunterparts.

B.1 Confidence Distribution

The model confidence distributions are shown in Figure 10 and Figure 11. Each row contains the robust and non-robust counterpart and their confidence distributions on the clean samples and the perturbated samples by PGD and Squares.

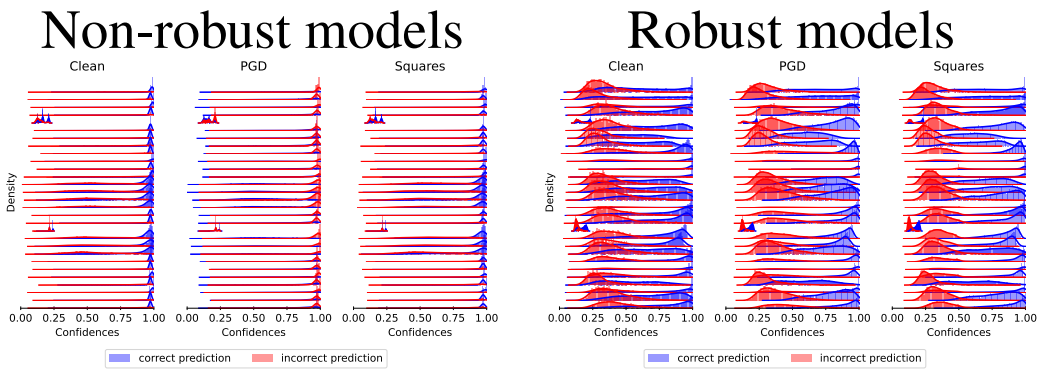

Figure 10: Density plots for robust and non-robust models on CIFAR10 over the models confidence on its correct and incorrect predictions. Each row contains the same model adversarially and standard trained. The non-robust models show high confidence in all of their predictions, however, those might be wrong. Especially in the case of PGD samples, the models are highly confident in their false predictions. In contrast, the robust models are better calibrated. The robust models are confident in their correct predictions and less confident in their false predictions.

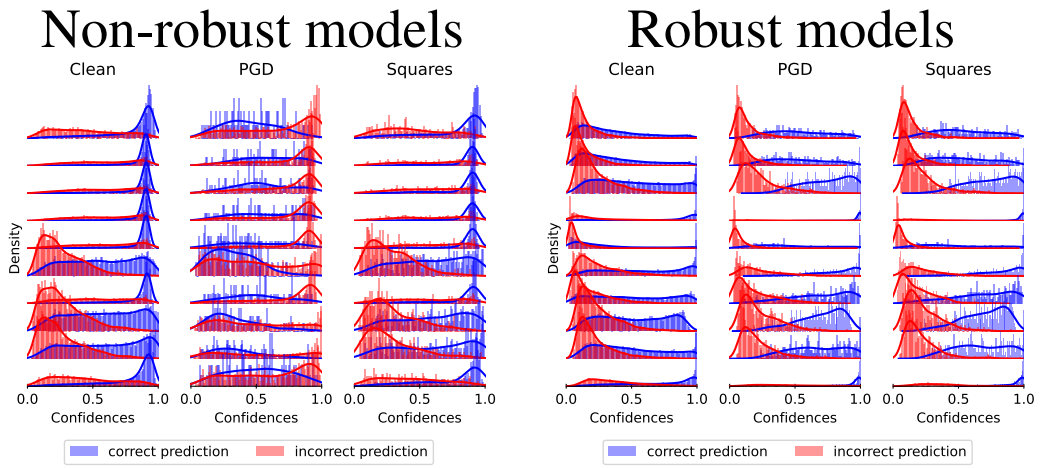

Figure 11: Density plots for robust and non-robust models on CIFAR100 over the models confidence on its correct and incorrect predictions. Each row contains the same model adversarially and standard trained. The non-robust models show high confidence in all of their predictions, however, those might be wrong. Especially in the case of PGD samples, the models are highly confident in their false predictions. In contrast, the robust models are better calibrated. The robust models are confident in their correct predictions and less confident in their false predictions.

B.2 Overconfidence and ECE

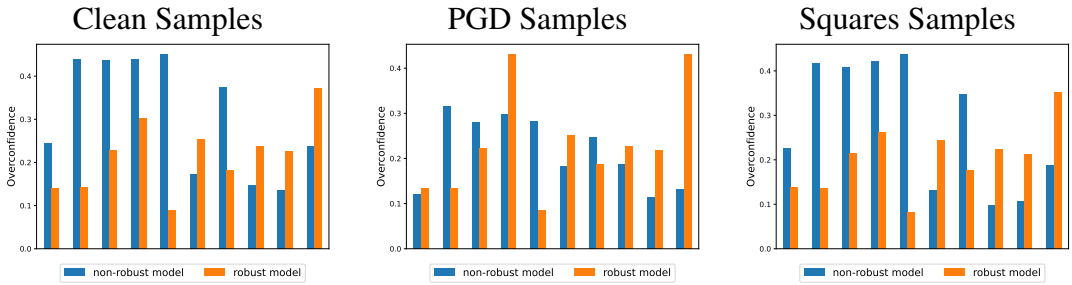

Figure 12: Overconfidence (lower is better) bar plots of robust models and their non-robust counterparts trained on CIFAR100.

Similar, the confidence distributions for the robust and non-robust counterparts on CIFAR100 are depicted in Figure 11.

Samples \ Robustness	Clean	PGD	Squares
non-robust models	0.3077 ± 0.1257	0.2159 ± 0.0738	0.2780 ± 0.1348
robust models	0.2962 ± 0.1722	0.2307 ± 0.1494	0.2076 ± 0.1247

Table 2: Mean ECE (lower is better) and standard deviation over all non-robust model versus all their robust counterparts trained on CIFAR100. Robust model exhibit a significantly lower ECE on all samples.

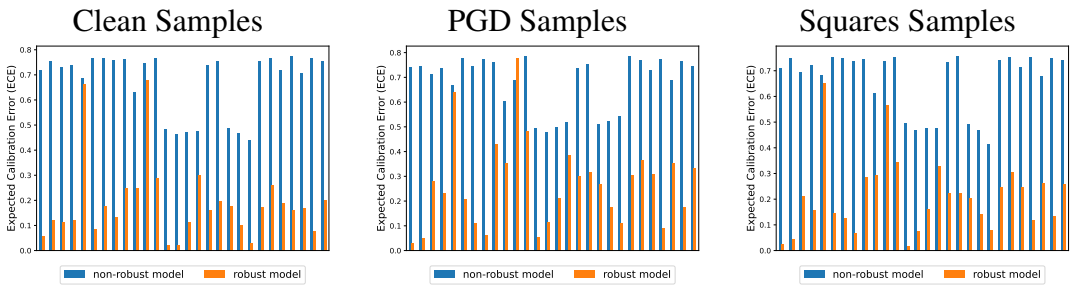

Figure 13: ECE (lower is better) bar plots of robust models and their non-robust counterparts trained on CIFAR10.

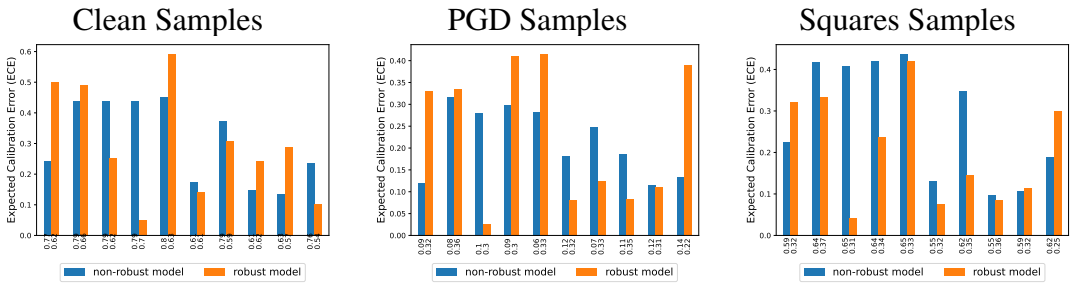

Figure 14: ECE (lower is better) bar plots of robust models and their non-robust counterparts trained on CIFAR100. The models accuracy are marked for the different samples for each bar.

B.3 Precision Recall

For completeness, we included the Precision Recall curves on CIFAR10 and CIFAR100 as mean over all robust and non-robust models with marked standard deviation.

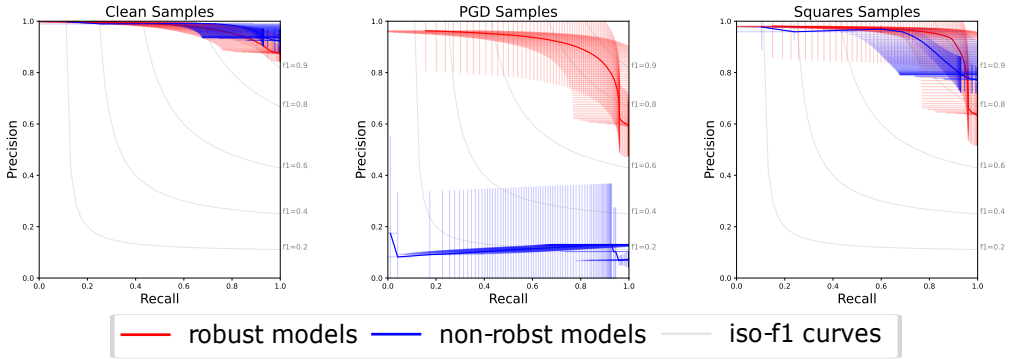

Figure 15: Average precision recall curve for all robust and all non-robust models trained on CIFAR10. Standard deviation is marked by the error bars. For the clean samples, the non-robust models can distinguish slightly better in correct and incorrect predictions based on the confidence of the prediction. The superior of the robust models are visible on the samples created by PGD, the non-robust models are not able to distinguish. However, for the samples created by Squares the classification into correct and incorrect predictions based on the confidence is almost equally possible for robust and non-robust models.

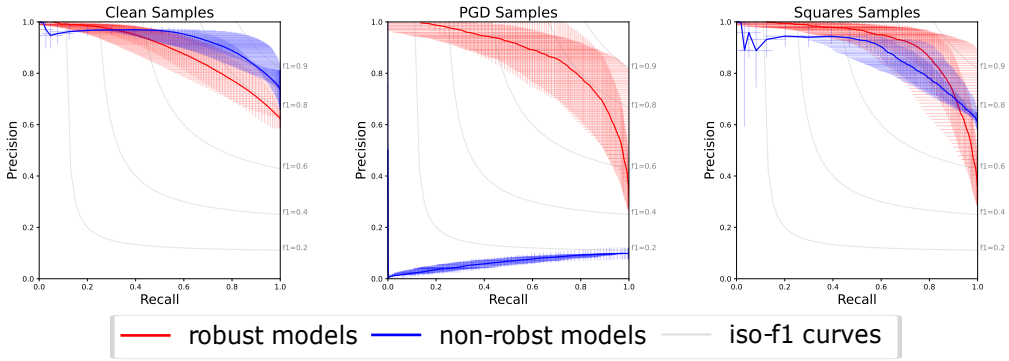

Figure 16: Average precision recall curve for all robust and all non-robust models trained on CIFAR100 for 1000 samples. Standard deviation is marked by the error bars. For the clean samples, the non-robust models can distinguish slightly better in correct and incorrect predictions based on the confidence of the prediction. The superior of the robust models are clearly visible on the samples created by PGD, the non-robust models are not able to distinguish. However, for the samples created by Squares the classification into correct and incorrect predictions based on the confidence is almost equally possible for robust and non-robust models.

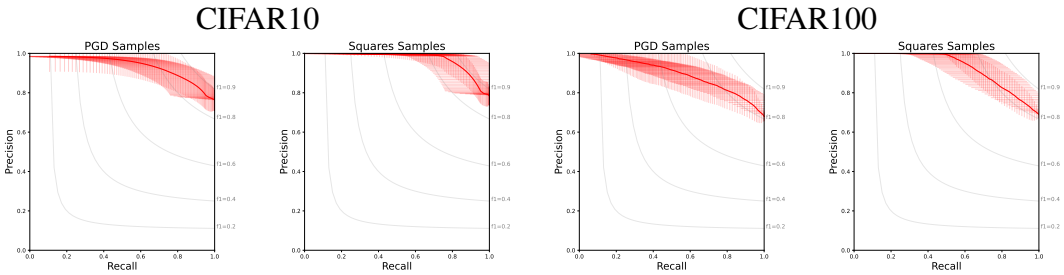

Figure 17: Precision Recall curve between confidence of clean correct samples and perturbated wrong samples on CIFAR10 and CIFAR100. The robust model confidences can be used as threshold for detection of adversarial attacks.

C CIFAR10-C Evaluation

Additionally to the previously studied attacks, we evaluate the confidence of robust versus non-robust CIFAR-10 models on the out-of-distribution dataset CIFAR10-C with severity level 4 (although the results for other severity levels follow the same trajectory and are omitted). There we benchmark models robust to adversarial attacks and their non-robust counterparts and evaluate the prediction confidence.

C.1 Overconfidence

First, we compare the models overconfidence with respect to each corruption type. In accordance with our findings on adversarial perturbations, robust models are much less overconfident than their non-robust counterparts. Figure 18 shows the overconfidence of each model pair for each corruption type. We can clearly see that robust models are generally much less overconfident.

C.2 ROC-curve

Regarding the mean ROC-curves (Figure 19) our results show that robust models tend to be better calibrated than non-robust models. However, robust models are inferior with respect to their calibration on corruptions changing the color palette of the image, like fog, brightness, contrast, and saturation.

Figure 18: Overconfidence for each robust CIFAR-10 model and the respective normal counterpart evaluated on CIFAR10-C.

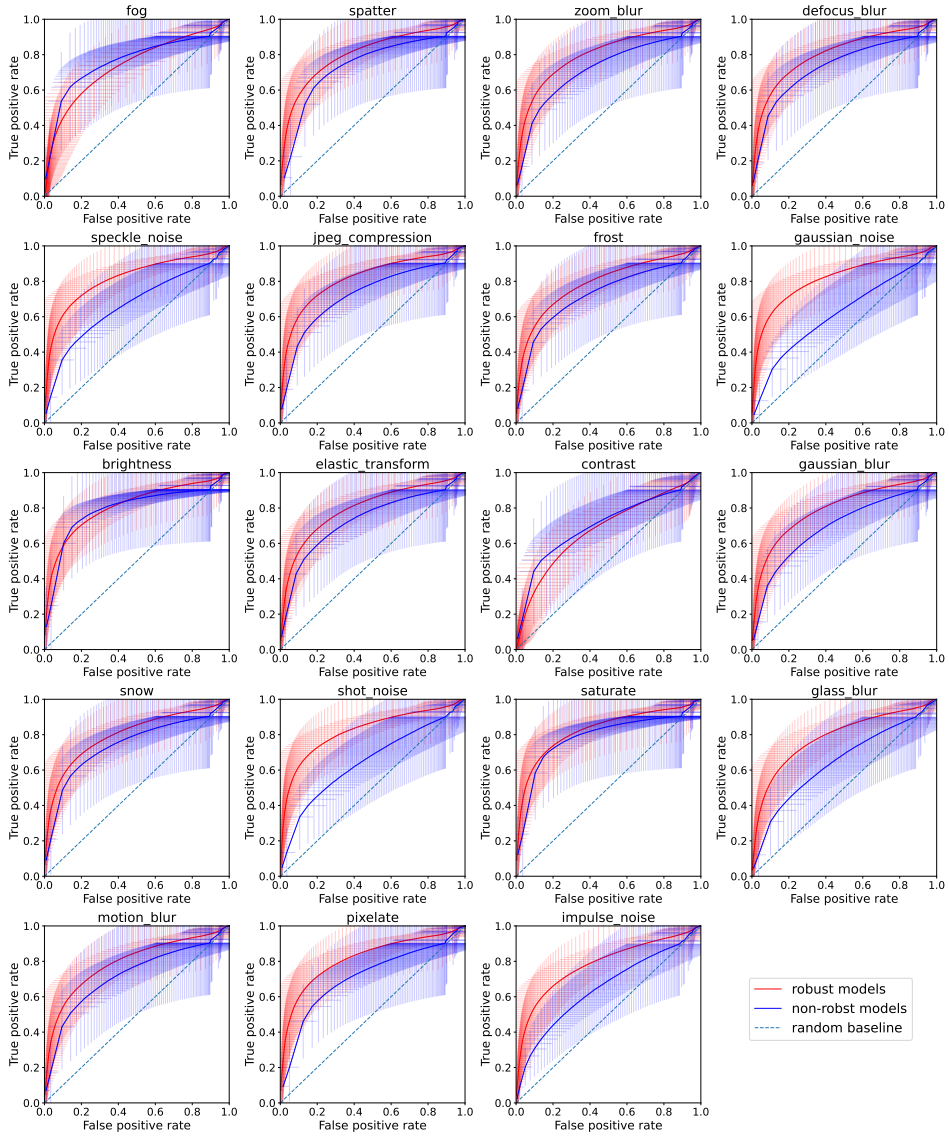

Figure 19: Mean ROC curves for each robust and non-robust CIFAR-10 model pair evaluated on CIFAR10-C.

D FLC Pooling

We evaluate different robust PRN-18 networks trained with flc pooling [28] and FGSM AT in terms of their confidence distribution. For training, we used the training script provided by [83]. We trained with ten different seeds and run for 300 epochs, choosing the batchsize to be 128, a momentum of 0.9, weight decay of 0.0005, a cycling learning rate with minimum value of 0 and maximum value of 0.2, for the adversarial samples we used FGSM with an ϵ of $8/255$ and α of $10/255$. Figure 20 shows the confidence distribution over all ten models and the standard deviation between those models. We can observe that the models with flc pooling are able to disentangle the correct from the incorrect prediction by the prediction confidence. The models provide low-variance and high-confidence in correct predictions and reduced confidence in false predictions across all evaluated samples.

E Downsampling and Activation

E.1 AUC

To show the impact of improved downsampling and activation functions we provide the ROC curves and AUC values of the models with and without those improved building blocks (similar to Figure 20 and Figure 7). Figure 21 shows the ROC curves on the improved building blocks as well as on comparable robust models with the same architecture. One can see that the improved building blocks results in slightly better calibration. The corresponding AUC values are reported in Table 3.

E.2 CIFAR10-C

Next, we compare the confidence impact of improved downsampling operations and activations on out-of-distribution data. Here we summarize our findings by the mean over all corruptions. Figure 22 shows that robust models are on average better calibrated than normal models. The impact of improved downsampling or activation functions is marginal.

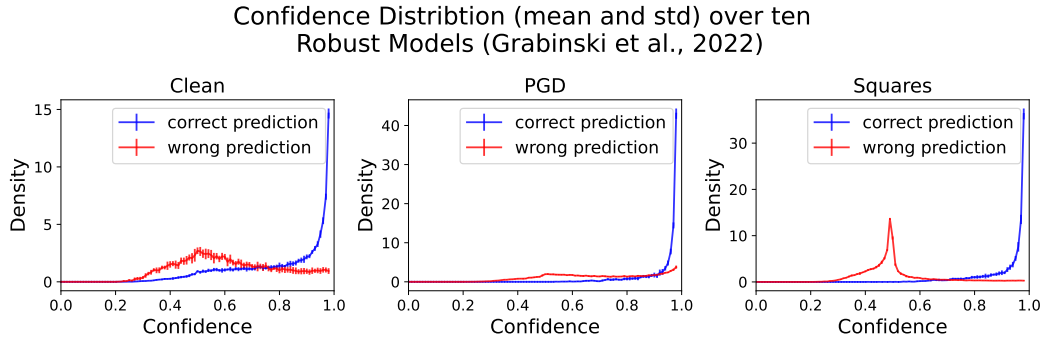

Figure 20: Additional confidence distribution evaluation over ten models (PRN-18) trained on CIFAR10 with flc pooling [28] and AT FGSM [83]. We used 100 bins and present the mean and standard deviation of the ten different models for each bin.

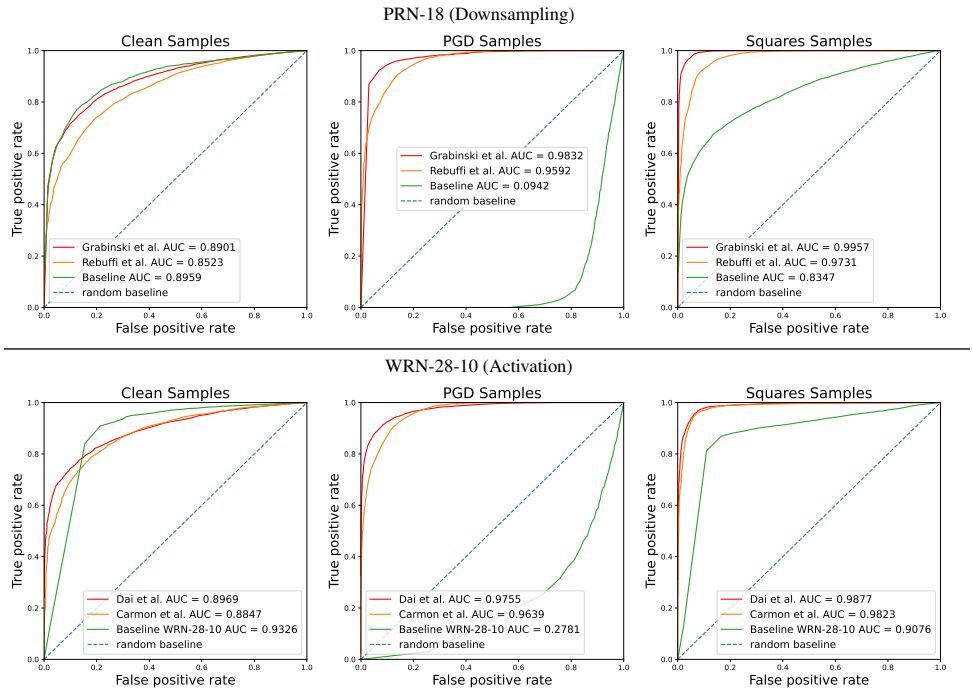

Figure 21: ROC curves for robust models with and without special building block, like downsampling (top) and activation (bottom).

Robust Model	Clean	PGD	Squares
Baseline PRN-18	0.8958	0.0942	0.8347
Grabinski et al. [28]	0.8901	0.9832	0.9923
Rebuffi et al. [62]	0.8523	0.9592	0.9731
Baseline WRN-28-10	0.9326	0.2781	0.9076
Dai et al. [17]	0.8969	0.9755	0.9877
Carmon et al. [6]	0.8847	0.9639	0.9823

Table 3: AUC value for the ROC curves of different robust models provided by [6, 17, 28, 62].

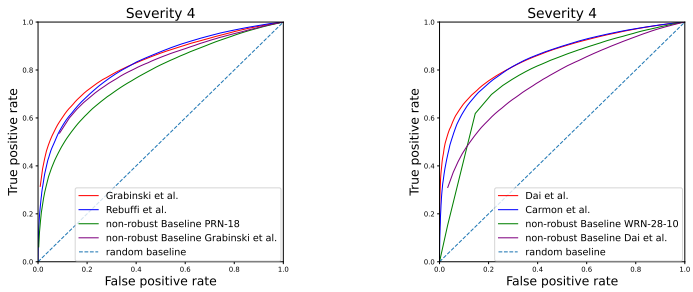

Figure 22: ROC curve for improved downsampling (left) and activation function (right) on CIFAR10-C corruptions. Robust models are superior to the normal models, and, the impact of activation and pooling is marginal.

F Additional Evaluation on ImageNet

Table 4 reports the accuracy evaluation of the robust models as well as the baseline on ImageNet. The accuracy is reported on the clean as well as on the perturbated samples by PGD and Squares with an ϵ of $4/255$.

Method	Architecture	Clean Acc ↑	PGD Acc ↑	Squares Acc ↑
Baseline	RN50	**76.13**	0.00	11.48
Engstrom et al. [22]	RN50	62.41	35.47	54.93
Wong et al. [83]	RN50	53.83	29.43	42.26
Salman et al. [66]	RN50	63.87	42.23	56.58
Salman et al. [66]	WRN50-2	68.41	**44.75**	**61.29**
Salman et al. [66]	RN18	52.50	31.92	43.81

Table 4: Clean and robust accuracy against PGD and Squares (higher is better) over 10000 samples.

For completeness, we included the ROC curve on the clean as well as the perturbated samples for the robust models and the baseline on ImagNet in figure 23.

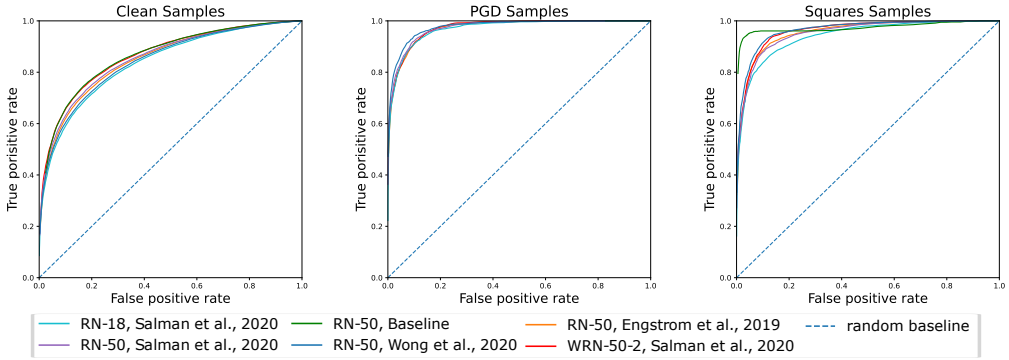

Figure 23: ROC curves for the robust models and the non-robust baseline trained on ImageNet provided on RobustBench [15].

G Model Overview

The robust checkpoints provided by *RobustBench* [15] are licensed under the MIT Licence. The clean models for ImageNet are provided by *timm* [82] under the Apache 2.0 licence.

Paper	Dataset	Architecture	Adv. Trained Clean Acc.	Adv. Trained Robust Acc.	Norm. Trained Clean Acc.	Norm. Trained Robust Acc.
[3]	cifar10	PreActResNet-18	79.84	43.93	94.51	0.0
[6]	cifar10	WideResNet-28-10	89.69	59.53	95.10	0.0
[67]	cifar10	WideResNet-28-10	88.98	57.14	95.10	0.0
[80]	cifar10	WideResNet-28-10	87.50	56.29	95.10	0.0
[38]	cifar10	WideResNet-28-10	87.11	54.92	95.35	0.0
[64]	cifar10	WideResNet-34-20	85.34	53.42	95.46	0.0
[87]	cifar10	WideResNet-34-10	84.92	53.08	95.26	0.0
[22]	cifar10	ResNet-50	87.03	49.25	94.90	0.0
[11]	cifar10	ResNet-50	86.04	51.56	86.50	0.0
[41]	cifar10	WideResNet-34-10	83.48	53.34	95.26	0.0
[59]	cifar10	WideResNet-34-20	85.14	53.74	76.30	0.0
[83]	cifar10	PreActResNet-18	83.34	43.21	94.25	0.0
[21]	cifar10	WideResNet-28-4	84.36	41.44	94.33	0.0
[86]	cifar10	WideResNet-34-10	87.20	44.83	95.26	0.0
[89]	cifar10	WideResNet-34-10	84.52	53.51	95.26	0.0
[84]	cifar10	WideResNet-28-10	88.25	60.04	95.10	0.0
[84]	cifar10	WideResNet-34-10	85.36	56.17	95.64	0.0
[26]	cifar10	WideResNet-70-16	85.29	57.20	87.91	0.0
[26]	cifar10	WideResNet-70-16	91.10	65.88	87.91	0.0
[26]	cifar10	WideResNet-34-20	85.64	56.86	88.33	0.0
[26]	cifar10	WideResNet-28-10	89.48	62.80	88.20	0.0
[68]	cifar10	WideResNet-34-10	85.85	59.09	95.64	0.0
[68]	cifar10	ResNet-18	84.38	54.43	94.87	0.0
[70]	cifar10	WideResNet-34-10	86.84	50.72	95.26	0.0
[9]	cifar10	WideResNet-34-10	85.32	51.12	95.35	0.0
[16]	cifar10	WideResNet-34-20	88.70	53.57	95.44	0.0
[16]	cifar10	WideResNet-34-10	88.22	52.86	95.26	0.0
[90]	cifar10	WideResNet-28-10	89.36	59.64	95.10	0.0
[62]	cifar10	WideResNet-28-10	87.33	60.75	88.20	0.0
[62]	cifar10	WideResNet-106-16	88.50	64.64	86.92	0.0
[62]	cifar10	WideResNet-70-16	88.54	64.25	87.91	0.0
[62]	cifar10	WideResNet-70-16	92.23	66.58	87.91	0.0
[71]	cifar10	WideResNet-28-10	89.46	59.66	95.10	0.0
[71]	cifar10	WideResNet-34-15	86.53	60.41	95.50	0.0
[62]	cifar10	PreActResNet-18	83.53	56.66	89.01	0.0
[61]	cifar10	PreActResNet-18	89.02	57.67	89.01	0.0
[61]	cifar10	PreActResNet-18	86.86	57.09	89.01	0.0
[61]	cifar10	WideResNet-34-10	91.47	62.83	88.67	0.0
[61]	cifar10	WideResNet-28-10	88.16	60.97	88.20	0.0
[40]	cifar10	WideResNet-34-R	90.56	61.56	95.60	0.0
[40]	cifar10	WideResNet-34-R	91.23	62.54	95.60	0.0
[1]	cifar10	ResNet-18	80.24	51.06	94.87	0.0
[1]	cifar10	WideResNet-34-10	85.32	58.04	95.26	0.0
[27]	cifar10	WideResNet-70-16	88.74	66.11	87.91	0.0
[17]	cifar10	WideResNet-28-10-PSSiLU	87.02	61.55	85.53	0.0
[27]	cifar10	WideResNet-28-10	87.50	63.44	88.20	0.0

Continued on next page

Paper	Dataset	Architecture	Adv. Trained Clean Acc.	Adv. Trained Robust Acc.	Norm. Trained Clean Acc.	Norm. Trained Robust Acc.
[27]	cifar10	PreActResNet-18	87.35	58.63	89.01	0.0
[8]	cifar10	WideResNet-34-10	85.21	56.94	95.64	0.0
[8]	cifar10	WideResNet-34-20	86.03	57.71	95.29	0.0
[26]	cifar100	WideResNet-70-16	60.86	30.03	60.56	0.0
[26]	cifar100	WideResNet-70-16	69.15	36.88	60.56	0.0
[16]	cifar100	WideResNet-34-20	62.55	30.20	80.46	0.0
[16]	cifar100	WideResNet-34-10	70.25	27.16	79.11	0.0
[16]	cifar100	WideResNet-34-10	60.64	29.33	79.11	0.0
[9]	cifar100	WideResNet-34-10	62.15	26.94	78.75	0.0
[84]	cifar100	WideResNet-34-10	60.38	28.86	78.79	0.0
[70]	cifar100	WideResNet-34-10	62.82	24.57	79.11	0.0
[38]	cifar100	WideResNet-28-10	59.23	28.42	79.16	0.0
[64]	cifar100	PreActResNet-18	53.83	18.95	76.18	0.0
[62]	cifar100	WideResNet-70-16	63.56	34.64	60.56	0.0
[62]	cifar100	WideResNet-28-10	62.41	32.06	61.46	0.0
[61]	cifar100	PreActResNet-18	56.87	28.50	63.45	0.0
[61]	cifar100	PreActResNet-18	61.50	28.88	63.45	0.0
[1]	cifar100	PreActResNet-18	62.02	27.14	76.66	0.0
[1]	cifar100	WideResNet-34-10	65.73	30.35	79.11	0.0
[8]	cifar100	WideResNet-34-10	64.07	30.59	79.11	0.0
[83]	imagenet	ResNet-50	55.62	26.24	80.37	0.0
[22]	imagenet	ResNet-50	62.56	29.22	80.37	0.0
[66]	imagenet	ResNet-50	64.02	34.96	80.37	0.0
[66]	imagenet	ResNet-18	52.92	25.32	69.74	0.0
[66]	imagenet	WideResNet-50-2	68.46	38.14	81.45	0.0