Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks

David Stutz [1] **Matthias Hein** [2] **Bernt Schiele** [1]

Abstract

Adversarial training yields robust models against a specific threat model, e.g., L_∞ adversarial examples. Typically robustness does *not* generalize to previously unseen threat models, e.g., other L_p norms, or larger perturbations. Our **confidence-calibrated adversarial training (CCAT)** tackles this problem by biasing the model towards low confidence predictions on adversarial examples. By allowing to reject examples with low confidence, robustness generalizes beyond the threat model employed during training. CCAT, trained *only* on L_∞ adversarial examples, increases robustness against larger L_∞, L_2, L_1 and L_0 attacks, adversarial frames, distal adversarial examples and corrupted examples and yields better clean accuracy compared to adversarial training. For thorough evaluation we developed novel white- and black-box attacks directly attacking CCAT by maximizing confidence. For each threat model, we use 7 attacks with up to 50 restarts and 5000 iterations and report worst-case robust test error, extended to our confidence-thresholded setting, across *all* attacks.

1. Introduction

Deep networks were shown to be susceptible to adversarial examples (Szegedy et al., 2014): adversarially perturbed examples that cause mis-classification while being nearly "imperceptible", i.e., close to the original example. Here, "closeness" is commonly enforced by constraining the L_p norm of the perturbation, referred to as threat model. Since then, numerous defenses against adversarial examples have been proposed. However, many were unable to keep up with more advanced attacks (Athalye et al., 2018; Athalye & Carlini, 2018). Moreover, most defenses are tailored to only one specific threat model.

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken [2]University of Tübingen, Tübingen. Correspondence to: David Stutz <david.stutz@mpi-inf.mpg.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

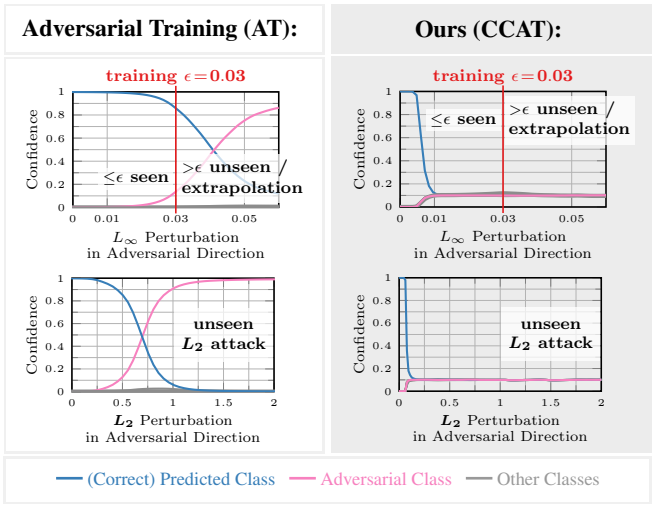

Figure 1: Adversarial Training (AT) versus our CCAT. We plot the confidence in the direction of an adversarial example. AT enforces high confidence predictions for the correct class on the L_∞-ball of radius ϵ (*"seen"* attack during training, top left). As AT enforces no particular bias beyond the ϵ-ball, adversarial examples can be found right beyond this ball. In contrast CCAT enforces a decaying confidence in the correct class up to uniform confidence within the ϵ-ball (top right). Thus, CCAT biases the model to extrapolate uniform confidence beyond the ϵ-ball. This behavior also extends to *"unseen"* attacks during training, e.g., L_2 attacks (bottom), such that adversarial examples can be rejected via confidence-thresholding.

Adversarial training (Goodfellow et al., 2015; Madry et al., 2018), i.e., training on adversarial examples, can be regarded as state-of-the-art. However, following Fig. 1, adversarial training is known to "overfit" to the threat model *"seen"* during training, e.g., L_∞ adversarial examples. Thus, robustness does not extrapolate to larger L_∞ perturbations, cf. Fig. 1 (top left), or generalize to *"unseen"* attacks, cf. Fig. 1 (bottom left), e.g., other L_p threat models (Sharma & Chen, 2018; Tramèr & Boneh, 2019; Li et al., 2019; Kang et al., 2019; Maini et al., 2020). We hypothesize this to be a result of enforcing high-confidence predictions on adversarial examples. However, high-confidence predictions are difficult to extrapolate beyond the adversarial examples seen during training. Moreover, it is not meaningful to extrapolate high-confidence predictions to arbitrary regions.

Finally, adversarial training often hurts accuracy, resulting in a robustness-accuracy trade-off (Tsipras et al., 2019; Stutz et al., 2019; Raghunathan et al., 2019; Zhang et al., 2019).

Contributions: We propose **confidence-calibrated adversarial training (CCAT)** which trains the network to predict a convex combination of uniform and (correct) one-hot distribution on adversarial examples that becomes more uniform as the distance to the attacked example increases. This is illustrated in Fig. 1. Thus, CCAT implicitly biases the network to predict a uniform distribution beyond the threat model *seen* during training, cf. Fig. 1 (top right). Robustness is obtained by rejecting low-confidence (adversarial) examples through confidence-thresholding. As a result, having seen *only* L_∞ adversarial examples during training, CCAT improves robustness against previously *unseen* attacks, cf. Fig. 1 (bottom right), e.g., L_2, L_1 and L_0 adversarial examples or larger L_∞ perturbations. Furthermore, robustness extends to adversarial frames (Zajac et al., 2019), distal adversarial examples (Hein et al., 2019), corrupted examples (e.g., noise, blur, transforms etc.) and accuracy of normal training is preserved better than with adversarial training.

For thorough evaluation, following best practices (Carlini et al., 2019), we adapt several state-of-the-art white- and black-box attacks (Madry et al., 2018; Ilyas et al., 2018; Andriushchenko et al., 2019; Narodytska & Kasiviswanathan, 2017; Khoury & Hadfield-Menell, 2018) to CCAT by explicitly maximizing confidence and improving optimization through a backtracking scheme. In total, we consider 7 different attacks for each threat model (i.e., L_p for $p \in \{\infty, 2, 1, 0\}$), allowing up to 50 random restarts and 5000 iterations each. We report worst-case robust test error, extended to our confidence-thresholded setting, across *all* attacks and restarts, on a *per test example* basis. We demonstrate improved robustness against unseen attacks compared to standard adversarial training (Madry et al., 2018), TRADES (Zhang et al., 2019), adversarial training using multiple threat models (Maini et al., 2020) and two detection methods (Ma et al., 2018; Lee et al., 2018), while training *only* on L_∞ adversarial examples.

We make our code (training and evaluation) and pre-trained models publicly available at davidstutz.de/ccat.

2. Related Work

Adversarial Examples: Adversarial examples can roughly be divided into white-box attacks, i.e., with access to the model gradients, e.g. (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2017b), and black-box attacks, i.e., only with access to the model's output, e.g. (Ilyas et al., 2018; Narodytska & Kasiviswanathan, 2017; Andriushchenko et al., 2019). Adversarial examples were also found to be transferable between models (Liu et al., 2017; Xie et al., 2019). In addition to *imperceptible* adversarial examples, adversarial transformations, e.g., (Engstrom et al., 2019; Alaifari et al., 2019), or adversarial patches (Brown et al., 2017) have also been studied. Recently, projected gradient ascent to maximize the cross-entropy loss or surrogate objectives, e.g., (Madry et al., 2018; Dong et al., 2018; Carlini & Wagner, 2017b), has become standard. Instead, we directly maximize the confidence in any but the true class, similar to (Hein et al., 2019; Goodfellow et al., 2019), in order to effectively train and attack CCAT.

Adversarial Training: Numerous defenses have been proposed, of which several were shown to be ineffective (Athalye et al., 2018; Athalye & Carlini, 2018). Currently, adversarial training is standard to obtain robust models. While it was proposed in different variants (Szegedy et al., 2014; Miyato et al., 2016; Huang et al., 2015), the formulation by (Madry et al., 2018) received considerable attention and has been extended in various ways: (Shafahi et al., 2020; Pérolat et al., 2018) train on universal adversarial examples, in (Cai et al., 2018), curriculum learning is used, and in (Tramèr et al., 2018; Grefenstette et al., 2018) ensemble adversarial training is proposed. The increased sample complexity (Schmidt et al., 2018) was addressed in (Lamb et al., 2019; Carmon et al., 2019; Alayrac et al., 2019) by training on interpolated or unlabeled examples. Adversarial training on multiple threat models is also possible (Tramèr & Boneh, 2019; Maini et al., 2020). Finally, the observed robustness-accuracy trade-off has been discussed in (Tsipras et al., 2019; Stutz et al., 2019; Zhang et al., 2019; Raghunathan et al., 2019). Adversarial training has also been combined with self-supervised training (Hendrycks et al., 2019). In contrast to adversarial training, CCAT imposes a target distribution which tends towards a uniform distribution for large perturbations, allowing the model to extrapolate beyond the threat model used at training time. Similar to adversarial training with an additional "abstain" class (Laidlaw & Feizi, 2019), robustness is obtained by rejection. In our case, rejection is based on confidence thresholding.

Detection: Instead of correctly classifying adversarial examples, several works (Gong et al., 2017; Grosse et al., 2017; Feinman et al., 2017; Liao et al., 2018; Ma et al., 2018; Amsaleg et al., 2017; Metzen et al., 2017; Bhagoji et al., 2017; Hendrycks & Gimpel, 2017; Li & Li, 2017; Lee et al., 2018) try to detect adversarial examples. However, several detectors have been shown to be ineffective against adaptive attacks aware of the detection mechanism (Carlini & Wagner, 2017a). Recently, the detection of adversarial examples by confidence, similar to our approach with CCAT, has also been discussed (Pang et al., 2018). Instead, Goodfellow et al. (2019) focus on evaluating confidence-based detection methods using adaptive, targeted attacks maximizing confidence. Our attack, although similar in spirit, is untargeted and hence suited for CCAT.

3. Generalizable Robustness by Confidence Calibration of Adversarial Training

To start, we briefly review adversarial training on L_∞ adversarial examples (Madry et al., 2018), which has become standard to train robust models, cf. Sec. 3.1. However, robustness does not generalize to larger perturbations or unseen attacks. We hypothesize this to be the result of enforcing high-confidence predictions on adversarial examples. CCAT addresses this issue with minimal modifications, cf. Sec. 3.2 and Alg. 1, by encouraging low-confidence predictions on adversarial examples. During testing, adversarial examples can be rejected by confidence thresholding.

Notation: We consider a classifier $f : \mathbb{R}^d \to \mathbb{R}^K$ with K classes where f_k denotes the confidence for class k. While we use the cross-entropy loss $\mathcal{L}$ for training, our approach also generalizes to other losses. Given $x \in \mathbb{R}^d$ with class $y \in \{1, \dots, K\}$, we let $f(x) := \operatorname{argmax}_k f_k(x)$ denote the predicted class for notational convenience. For $f(x) = y$, an adversarial example $\tilde{x} = x + \delta$ is defined as a "small" perturbation δ such that $f(\tilde{x}) \neq y$, i.e., the classifier changes its decision. The strength of the change δ is measured by some L_p-norm, $p \in \{0, 1, 2, \infty\}$. Here, $p = \infty$ is a popular choice as it leads to the smallest perturbation per pixel.

3.1. Problems of Adversarial Training

Following (Madry et al., 2018), adversarial training is given as the following min-max problem:

$$\min_w \mathbb{E}\left[\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w), y)\right] \tag{1}$$

with w being the classifier's parameters. During mini-batch training the inner maximization problem,

$$\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w), y), \tag{2}$$

is approximately solved. In addition to the L_∞-constraint, a box constraint is enforced for images, i.e., $\tilde{x}_i = (x + \delta)_i \in [0, 1]$. Note that maximizing the cross-entropy loss is equivalent to finding the adversarial example with *minimal* confidence in the true class. For neural networks, this is generally a non-convex optimization problem. In (Madry et al., 2018) the problem is tackled using projected gradient descent (PGD), initialized using a random δ with $\|\delta\|_\infty \leq \epsilon$.

In contrast to adversarial training as proposed in (Madry et al., 2018), which computes adversarial examples for the *full* batch in each iteration, others compute adversarial examples only for *half* the examples of each batch (Szegedy et al., 2014). Instead of training *only* on adversarial examples, each batch is divided into 50% clean and 50% adversarial examples. Compared to Eq. (1), 50%/50% adversarial

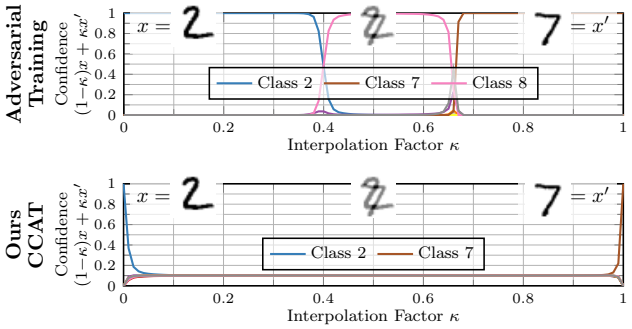

Figure 2: Extrapolation of Uniform Predictions. We plot the confidence in each class along an interpolation between two test examples x and x', "2" and "7", on MNIST (LeCun et al., 1998): $(1-\kappa)x + \kappa x'$ where κ is the interpolation factor. CCAT quickly yields low-confidence, uniform predictions in between both examples, extrapolating the behavior enforced within the ϵ-ball during training. Regular adversarial training, in contrast, consistently produces high-confidence predictions, even on unreasonable inputs.

training effectively minimizes

$$\underbrace{\mathbb{E}\left[\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w), y)\right]}_{\text{50\% adversarial training}} + \underbrace{\mathbb{E}\left[\mathcal{L}(f(x; w), y)\right]}_{\text{50\% "clean" training}}. \tag{3}$$

This improves test accuracy on clean examples compared to 100% adversarial training but typically leads to worse robustness. Intuitively, by balancing both terms in Eq. (3), the trade-off between accuracy and robustness can already be optimized to some extent (Stutz et al., 2019).

Problems: Trained on L_∞ adversarial examples, the robustness of adversarial training does not generalize to previously unseen adversarial examples, including larger perturbations or other L_p adversarial examples. We hypothesize that this is because adversarial training explicitly enforces high-confidence predictions on L_∞ adversarial examples within the ϵ-ball seen during training ("seen" in Fig. 1). However, this behavior is difficult to extrapolate to arbitrary regions in a meaningful way. Thus, it is not surprising that adversarial examples can often be found right beyond the ϵ-ball used during training, cf. Fig. 1 (top left). This can be described as "overfitting" to the L_∞ adversarial examples used during training. Also, larger ϵ-balls around training examples might include (clean) examples from other classes. Then, Eq. (2) will focus on these regions and reduce accuracy as considered in our theoretical toy example, see Proposition 1, and related work (Jacobsen et al., 2019b;a).

As suggested in Fig. 1, both problems can be addressed by enforcing low-confidence predictions on adversarial examples in the ϵ-ball. In practice, we found that the low-confidence predictions on adversarial examples within the ϵ-ball are extrapolated beyond the ϵ-ball, i.e., to larger pertur-

Algorithm 1 Confidence-Calibrated Adversarial Training (CCAT). The only changes compared to standard adversarial training are the attack (line 4) and the probability distribution over the classes (lines 6 and 7), which becomes more uniform as distance $\|\delta\|_\infty$ increases. During testing, low-confidence (adversarial) examples are rejected.

1: **while** true **do**
2: choose random batch $(x_1, y_1), \ldots, (x_B, y_B)$.
3: **for** $b = 1, \ldots, {}^{B}/_{2}$ **do**
4: $\delta_b := \underset{\|\delta\|_\infty \leq \epsilon}{\operatorname{argmax}} \max_{k \neq y_b} f_k(x_b + \delta)$ (Eq. (4))
5: $\tilde{x}_b := x_b + \delta_b$
6: $\lambda(\delta_b) := (1 - \min(1, \|\delta_b\|_\infty / \epsilon))^\rho$ (Eq. (6))
7: $\tilde{y_b} := \lambda(\delta_b)\, \text{one\_hot}(y_b) + (1 - \lambda(\delta_b))\, \frac{1}{K}$ (Eq. (5))
8: **end for**
9: update parameters using Eq. (3):
10: $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$
11: **end while**

bations, unseen attacks or distal adversarial examples. This allows to reject adversarial examples based on their low confidence. We further enforce this behavior by explicitly encouraging a "steep" transition from high-confidence predictions (on clean examples) to low-confidence predictions (on adversarial examples). As result, the (low-confidence) prediction is almost flat close to the boundary of the ϵ-ball. Additionally, there is no incentive to deviate from the uniform distribution outside of the ϵ-ball. For example, as illustrated in Fig. 2, the confidence stays low in between examples from different classes and only increases if necessary, i.e., close to the examples.

3.2. Confidence-Calibrated Adversarial Training

Confidence-calibrated adversarial training (CCAT) addresses these problems with minimal modifications, as outlined in Alg. 1. During training, we train the network to predict a convex combination of (correct) one-hot distribution on clean examples and uniform distribution on adversarial examples as target distribution within the cross-entropy loss. During testing, adversarial examples can be rejected by confidence thresholding: adversarial examples receive near-uniform confidence while test examples receive high-confidence. By extrapolating the uniform distribution beyond the ϵ-ball used during training, previously unseen adversarial examples such as larger L_∞ perturbations can be rejected, as well. In the following, we first introduce an alternative objective for generating adversarial examples. Then, we specifically define the target distribution, which becomes more uniform with larger perturbations $\|\delta\|_\infty$. In Alg. 1, these changes correspond to lines 4, 6 and 7, requiring only few lines of code in practice.

Given an example x with label y, our adaptive attack during training maximizes the confidence in any other label $k \neq y$. This results in effective attacks against CCAT, as CCAT will reject low-confidence adversarial examples:

$$\max_{\|\delta\|_\infty \leq \epsilon} \max_{k \neq y} f_k(x + \delta; w) \tag{4}$$

Note that Eq. (2), in contrast, minimizes the confidence in the true label y. Similarly, (Goodfellow et al., 2019) uses targeted attacks in order to maximize confidence, whereas ours is untargeted and, thus, our objective is the maximal confidence over all other classes.

Then, given an adversarial example from Eq. (4) during training, CCAT uses the following combination of uniform and one-hot distribution as target for the cross-entropy loss:

$$\tilde{y} = \lambda(\delta)\ \text{one\_hot}(y) + \big(1 - \lambda(\delta)\big)\frac{1}{K}, \tag{5}$$

with $\lambda(\delta) \in [0, 1]$ and $\text{one\_hot}(y) \in \{0, 1\}^K$ denoting the one-hot vector corresponding to class y. Thus, we enforce a convex combination of the original label distribution and the uniform distribution which is controlled by the parameter $\lambda = \lambda(\delta)$, computed given the perturbation δ. We choose λ to decrease with the distance $\|\delta\|_\infty$ of the adversarial example to the attacked example x with the intention to enforce uniform predictions when $\|\delta\|_\infty = \epsilon$. Then, the network is encouraged to extrapolate this uniform distribution beyond the used ϵ-ball. Even if extrapolation does not work perfectly, the uniform distribution is much more meaningful for extrapolation to arbitrary regions as well as regions between classes compared to high-confidence predictions as encouraged in standard adversarial training, as demonstrated in Fig. 2. For controlling the trade-off λ between one-hot and uniform distribution, we consider the following "power transition":

$$\lambda(\delta) := \Big(1 - \min\Big(1, \frac{\|\delta\|_\infty}{\epsilon}\Big)\Big)^\rho \tag{6}$$

This ensures that for $\delta = 0$ we impose the original (one-hot) label. For growing δ, however, the influence of the original label decays proportional to $\|\delta\|_\infty$. The speed of decay is controlled by the parameter ρ. For $\rho = 10$, Fig. 1 (top right) shows the transition as approximated by the network. The power transition ensures that for $\|\delta\|_\infty \geq \epsilon$, i.e., perturbations larger than encountered during training, a uniform distribution is enforced as λ is 0. We train on 50% clean and 50% adversarial examples in each batch, as in Eq. (3), such that the network has an incentive to predict correct labels.

The convex combination of uniform and one-hot distribution in Eq. (5) resembles the label smoothing regularizer introduced in (Szegedy et al., 2016). In concurrent work, label smoothing has also been used as regularizer for adversarial training (Cheng et al., 2020). However, in our case,

$\lambda = \lambda(\delta)$ from Eq. (6) is not a fixed hyper-parameter as in (Szegedy et al., 2016; Cheng et al., 2020). Instead, λ depends on the perturbation δ and reaches zero for $\|\delta\|_\infty = \epsilon$ to encourage low-confidence predictions beyond the ϵ-ball used during training. Thereby, λ explicitly models the transition from one-hot to uniform distribution.

3.3. Confidence-Calibrated Adversarial Training Results in Accurate Models

Proposition 1 discusses a problem where standard adversarial training is unable to reconcile robustness and accuracy while CCAT is able to obtain *both* robustness and accuracy:

Proposition 1. *We consider a classification problem with two points $x = 0$ and $x = \epsilon$ in $\mathbb{R}$ with deterministic labels, i.e., $p(y = 2|x = 0) = 1$ and $p(y = 1|x = \epsilon) = 1$, such that the problem is fully determined by the probability $p_0 = p(x = 0)$. The Bayes error of this classification problem is zero. Let the predicted probability distribution over classes be $\tilde{p}(y|x) = \frac{e^{g_y(x)}}{e^{g_1(x)}+e^{g_2(x)}}$, where $g : \mathbb{R}^d \to \mathbb{R}^2$ is the classifier and we assume that the function $\lambda : \mathbb{R}_+ \to [0, 1]$ used in CCAT is monotonically decreasing and $\lambda(0) = 1$. Then, the error of the Bayes optimal classifier (with cross-entropy loss) for*

- *adversarial training on 100% adversarial examples, cf. Eq. (1), is $\min\{p_0, 1 - p_0\}$.*
- *adversarial training on 50%/50% adversarial/clean examples per batch, cf. Eq. (3), is $\min\{p_0, 1 - p_0\}$.*
- *CCAT on 50% clean and 50% adversarial examples, cf. Alg. 1, is* zero *if $\lambda(\epsilon) < \min\{p_0/1-p_0, 1-p_0/p_0\}$.*

Here, 100% and 50%/50% standard adversarial training are unable to obtain *both* robustness and accuracy: The ϵ-ball used during training contains examples of different classes such that adversarial training enforces high-confidence predictions in contradicting classes. CCAT addresses this problem by encouraging low-confidence predictions on adversarial examples within the ϵ-ball. Thus, CCAT is able to improve accuracy while preserving robustness.

4. Detection and Robustness Evaluation with Adaptive Attack

CCAT allows to reject (adversarial) inputs by confidence-thresholding before classifying them. As we will see, this "reject option", is also beneficial for standard adversarial training (AT). Thus, evaluation also requires two stages: First, we fix the confidence threshold at 99% true positive rate (TPR), where correctly classified clean examples are positives such that at most 1% (correctly classified) clean examples are rejected. Second, on the non-rejected examples, we evaluate accuracy and robustness using *confidence-thresholded* (robust) test error.

4.1. Adaptive Attack

As CCAT encourages low confidence on adversarial examples, we use PGD to maximize the confidence of adversarial examples, cf. Eq. (4), as effective adaptive attack against CCAT. In order to effectively optimize our objective, we introduce a simple but crucial improvement: after each iteration, the computed update is only applied if the objective is improved; otherwise the learning rate is reduced. Additionally, we use momentum (Dong et al., 2018) and run the attack for exactly T iterations, choosing the perturbation corresponding to the best objective across all iterations. In addition to random initialization, we found that $\delta = 0$ is an effective initialization against CCAT. We applied the same principles for (Ilyas et al., 2018), i.e., PGD with approximated gradients, Eq. (4) as objective, momentum and backtracking; we also use Eq. (4) as objective for the black-box attacks of (Andriushchenko et al., 2019; Narodytska & Kasiviswanathan, 2017; Khoury & Hadfield-Menell, 2018).

4.2. Detection Evaluation

In the first stage, we consider a detection setting: adversarial example are *negatives* and correctly classified clean examples are *positives*. The confidence threshold τ is chosen extremely conservatively by requiring a **99% true positive rate (TPR)**: at most 1% of correctly classified clean examples can be rejected. As result, the confidence threshold is determined *only* by correctly classified clean examples, independent of adversarial examples. Incorrectly rejecting a significant fraction of correctly classified clean examples is unacceptable. This is also the reason why we do not report the area under the receiver operating characteristic (ROC) curve as related work (Lee et al., 2018; Ma et al., 2018). Instead, we consider the **false positive rate (FPR)**. The supplementary material includes a detailed discussion.

4.3. Robustness Evaluation

In the second stage, after confidence-thresholding, we consider the widely used robust test error (RErr) (Madry et al., 2018). It quantifies the model's test error in the case where all test examples are allowed to be attacked, i.e., modified within the chosen threat model, e.g., for L_p:

$$\text{"Standard" RErr} = \frac{1}{N}\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon} \mathbb{1}_{f(x_n+\delta)\neq y_n} \qquad (7)$$

where $\{(x_n, y_n)\}_{n=1}^N$ are test examples and labels. In practice, RErr is computed empirically using adversarial attacks. Unfortunately, standard RErr does not take into account the option of rejecting (adversarial) examples.

We propose a generalized definition adapted to our confidence-thresholded setting where the model can reject examples. For fixed confidence threshold τ at 99%TPR, the

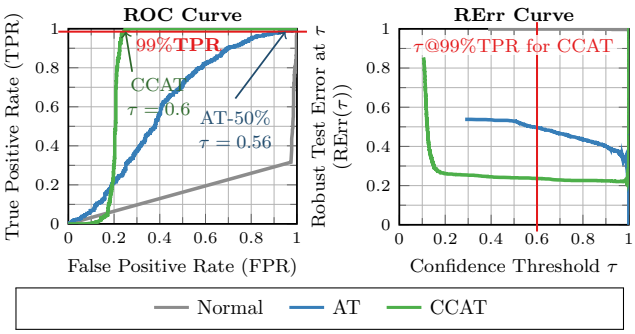

Figure 3: ROC and RErr Curves. On SVHN, we show ROC curves when distinguishing *correctly classified* test examples from adversarial examples by confidence (left) and (confidence-thresholded) RErr against confidence threshold τ (right) for worst-case adversarial examples across L_∞ attacks with $\epsilon = 0.03$. The confidence threshold τ is chosen exclusively on correctly classified clean examples to obtain 99%TPR. For CCAT, this results in $\tau \approx 0.6$. Note that RErr subsumes both Err and FPR.

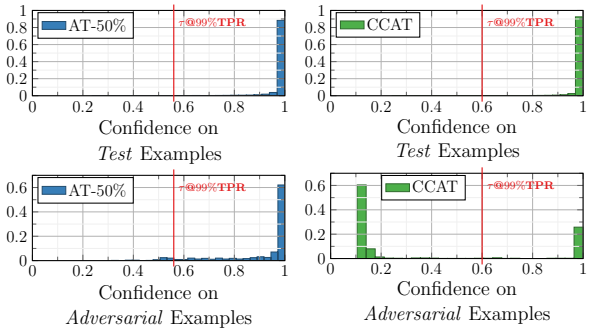

Figure 4: Confidence Histograms. On SVHN, for AT (50%/50% adversarial training, left) and CCAT (right), we show confidence histograms corresponding to *correctly classified* test examples (top) and adversarial examples (bottom). We consider the worst-case adversarial examples across all L_∞ attacks for $\epsilon = 0.03$. While the confidence of adversarial examples is reduced slightly for AT, CCAT is able to distinguish the majority of adversarial examples from (clean) test examples by confidence thresholding (in red).

confidence-thresholded RErr is defined as

$$\text{RErr}(\tau) = \frac{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon, c(x_n+\delta) \geq \tau} \mathbb{1}_{f(x_n+\delta) \neq y_n}}{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon} \mathbb{1}_{c(x_n+\delta) \geq \tau}} \tag{8}$$

with $c(x) = \max_k f_k(x)$ and $f(x)$ being the model's confidence and predicted class on example x, respectively. Essentially, this is the **test error on test examples that can be modified within the chosen threat model *and* pass confidence thresholding**. For $\tau = 0$ (i.e., all examples pass confidence thresholding) this reduces to the standard RErr, comparable to related work. We stress that our adaptive attack in Eq. (4) directly maximizes the numerator of Eq. (8) by maximizing the confidence of classes not equal y. A (clean) **confidence-thresholded test error (Err(τ))** is obtained similarly. In the following, if not stated otherwise, we report *confidence-thresholded* RErr and Err as default and omit the confidence threshold τ for brevity.

FPR and RErr: FPR quantifies how well an adversary can perturb (correctly classified) examples while not being rejected. The confidence-thresholded RErr is more conservative as it measures *any* non-rejected error (adversarial or not). As result, RErr implicitly includes FPR *and* Err. Therefore, we report only RErr and include FPRs for all our experiments in the supplementary material.

Per-Example Worst-Case Evaluation: Instead of reporting average or per-attack results, we use a per-example *worst-case* evaluation scheme: For each individual test example, all adversarial examples from all attacks (and restarts) are accumulated. Subsequently, *per test example*, only the adversarial example with highest confidence is considered, resulting in a significantly stronger robustness evaluation compared to related work.

5. Experiments

We evaluate CCAT in comparison with AT (Madry et al., 2018) and related work (Maini et al., 2020; Zhang et al., 2019) on MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011) and Cifar10 (Krizhevsky, 2009) as well as MNIST-C (Mu & Gilmer, 2019) and Cifar10-C (Hendrycks & Dietterich, 2019) with corrupted examples (e.g., blur, noise, compression, transforms etc.). We report ***confidence-thresholded* test error (Err; ↓ lower is better)** and ***confidence-thresholded* robust test error (RErr; ↓ lower is better)** for a confidence-threshold τ corresponding to 99% true positive rate (TPR); we omit τ for brevity. We note that normal and standard adversarial training (AT) are also allowed to reject examples by confidence thresholding. Err is computed on 9000 test examples. RErr is computed on 1000 test examples. The confidence threshold τ depends *only* on correctly classified clean examples and is fixed at 99%TPR on the held-out *last* 1000 test examples.

Attacks: For thorough evaluation, we consider 7 different L_p attacks for $p \in \{\infty, 2, 1, 0\}$. As white-box attacks, we use PGD to maximize the objectives Eq. (2) and (4), referred to as PGD-CE and PGD-Conf. We use $T = 1000$ iterations and 10 random restarts with random initialization plus one restart with zero initialization for PGD-Conf, and $T = 200$ with 50 random restarts for PGD-CE. For L_∞, L_2, L_1 and L_0 attacks, we set **ϵ to 0.3, 3, 18, 15 (MNIST) or 0.03, 2, 24, 10 (SVHN/Cifar10)**.

	SVHN: RErr @99%TPR, L_∞, $\epsilon = 0.03$					
	worst case	top-5 attacks/restarts out of 7 attacks with 84 restarts				
AT-50%	56.0	52.1	52.0	51.9	51.6	51.4
CCAT	39.1	23.6	13.7	13.6	12.6	12.5

Table 1: Per-Example Worst-Case Evaluation. We compare confidence-thresholded RErr with τ@99%TPR for the per-example worst-case and the top-5 individual attacks/restarts among 7 attacks with 84 restarts in total. Multiple restarts are *necessary* to effectively attack CCAT, while a single attack and restart is nearly sufficient against AT-50%. This demonstrates that CCAT is more difficult to "crack".

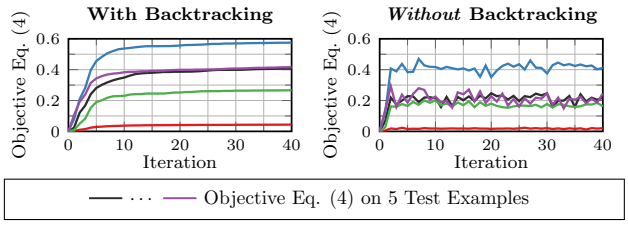

Figure 5: Backtracking. Our L_∞ PGD-Conf attack, i.e., PGD maximizing Eq. (4), using 40 iterations with momentum and our developed backtracking scheme (left) and without both (right) on SVHN. We plot Eq. (4) over iterations for the first 5 test examples corresponding to different colors. Backtracking avoids oscillation and obtains higher overall objective values within the same number of iterations.

As black-box attacks, we additionally use the Query Limited (QL) attack (Ilyas et al., 2018) adapted with momentum and backtracking for $T = 1000$ iterations with 10 restarts, the Simple attack (Narodytska & Kasiviswanathan, 2017) for $T = 1000$ iterations and 10 restarts, and the Square attack (L_∞ and L_2) (Andriushchenko et al., 2019) with $T = 5000$ iterations. In the case of L_0 we also use Corner Search (CS) (Croce & Hein, 2019). For all L_p, $p \in \{\infty, 2, 1, 0\}$, we consider 5000 (uniform) random samples from the L_p-ball and the Geometry attack (Khoury & Hadfield-Menell, 2018). Except for CS, all black-box attacks use Eq. (4) as objective:

Attack	Objective	T	Restarts
PGD-CE	Eq. (2), random init.	200	50
PGD-Conf	Eq. (4), zero + random init.	1000	11
QL†	Eq. (4), zero + random init.	1000	11
Simple†	Eq. (4)	1000	10
Square†	Eq. (4), L_∞, L_2 only	5000	1
CS†	Eq. (2), L_0 only	200	1
Geometry†	Eq. (4)	1000	1
Random†	Eq. (4)	–	5000

† Black-box attacks.

Additionally, we consider adversarial frames and distal adversarial examples: Adversarial frames (Zajac et al., 2019) allow a 2 (MNIST) or 3 (SVHN/Cifar10) pixel border to be manipulated arbitrarily within $[0, 1]$ to maximize Eq. (4) using PGD. Distal adversarial examples start with a (uniform) random image and use PGD to maximize (4) within a L_∞-ball of size $\epsilon = 0.3$ (MNIST) or $\epsilon = 0.03$ (SVHN/Cifar10).

Training: We train 50%/50% AT (AT-50%) and CCAT as well as 100% AT (AT-100%) with L_∞ attacks using $T = 40$ iterations for PGD-CE and PGD-Conf, respectively, and $\epsilon = 0.3$ (MNIST) or $\epsilon = 0.03$ (SVHN/Cifar10). We use ResNet-20 (He et al., 2016), implemented in PyTorch (Paszke et al., 2017), trained using stochastic gradient descent. For CCAT, we use $\rho = 10$.

Baselines: We compare to multi-steepest descent (MSD) adversarial training (Maini et al., 2020) using the pre-trained LeNet on MNIST and pre-activation ResNet-18 on Cifar10 trained with L_∞, L_2 and L_1 adversarial examples and ϵ set to $0.3, 1.5, 12$ and $0.03, 0.5, 12$, respectively. The L_2 and L_1 attacks in Tab. 2 (larger ϵ) are unseen. For TRADES (Zhang et al., 2019), we use the pre-trained convolutional network (Carlini & Wagner, 2017b) on MNIST and WRN-10-28 (Zagoruyko & Komodakis, 2016) on Cifar10, trained on L_∞ adversarial examples with $\epsilon = 0.3$ and $\epsilon = 0.03$, respectively. On Cifar10, we further consider the pre-trained ResNet-50 of (Madry et al., 2018) (AT-Madry, L_∞ adversarial examples with $\epsilon = 0.03$). We also consider the Mahalanobis (MAHA) (Lee et al., 2018) and local intrinsic dimensionality (LID) detectors (Ma et al., 2018) using the provided pre-trained ResNet-34 on SVHN/Cifar10.

5.1. Ablation Study

Evaluation Metrics: Fig. 3 shows ROC curves, i.e., how well adversarial examples can be rejected by confidence. As marked in red, we are only interested in the FPR for the conservative choice of 99%TPR, yielding the confidence threshold τ. The RErr curves highlight how robustness is influenced by the threshold: AT also benefits from a reject option, however, not as much as CCAT which has been explicitly designed for rejecting adversarial examples.

Worst-Case Evaluation: Tab. 1 illustrates the importance of worst-case evaluation on SVHN, showing that CCAT is significantly "harder" to attack than AT. We show the worst-case RErr over all L_∞ attacks as well as the top-5 individual attacks (each restart treated as separate attack). For AT-50%, a single restart of PGD-Conf with $T = 1000$ iterations is highly successful, with 52.1% RErr close to the overall worst-case of 56%. For CCAT, in contrast, multiple restarts are crucial as the best individual attack, PGD-Conf with $T = 1000$ iterations and zero initialization obtains only 23.6% RErr compared to the overall worst-case of 39.1%.

Backtracking: Fig. 5 illustrates the advantage of backtracking for PGD-Conf with T=40 iterations on 5 test examples of SVHN. Backtracking results in better objective values and avoids oscillation, i.e., a stronger attack for training and

MNIST:	**Err** ↓ in %		*confidence-thresholded* **RErr** ↓ for τ@99%TPR						**FPR** ↓	**Err** ↓
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.3$	L_∞ $\epsilon = 0.4$	L_2 $\epsilon = 3$	L_1 $\epsilon = 18$	L_0 $\epsilon = 15$	adv. frames	distal	corrupted MNIST-C
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	0.4	0.1	100.0	100.0	100.0	100.0	92.3	87.7	100.0	32.8
AT-50%	0.5	**0.0**	**1.7**	100.0	81.5	24.6	23.9	73.7	100.0	12.6
AT-100%	0.5	**0.0**	**1.7**	100.0	84.8	21.3	**13.9**	62.3	100.0	17.6
CCAT	**0.3**	0.1	7.4	**11.9**	**0.3**	**1.8**	14.8	**0.2**	**0.0**	**5.7**
* MSD	1.8	0.9	34.3	98.9	59.2	55.9	66.4	8.8	100.0	6.0
* TRADES	0.5	0.1	4.0	99.9	44.3	9.0	35.5	**0.2**	100.0	7.9

SVHN:	**Err** ↓ in %		*confidence-thresholded* **RErr** ↓ for τ@99%TPR						**FPR** ↓
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$	adv. frames	distal
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	3.6	2.6	99.9	100.0	100.0	100.0	83.7	78.7	87.1
AT-50%	3.4	2.5	56.0	88.4	99.4	99.5	73.6	33.6	86.3
AT-100%	5.9	4.6	48.3	87.1	99.5	99.8	89.4	26.0	81.0
CCAT	**2.9**	**2.1**	**39.1**	**53.1**	**29.0**	**31.7**	**3.5**	**3.7**	**0.0**
* LID	3.3	2.2	91.0	93.1	92.2	90.0	41.6	89.8	8.6
* MAHA	3.3	2.2	73.0	79.5	78.1	67.5	41.5	9.9	**0.0**

CIFAR10:	**Err** ↓ in %		*confidence-thresholded* **RErr** ↓ for τ@99%TPR						**FPR** ↓	**Err** ↓
	(clean) $\tau = 0$	(clean) 99%TPR	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$	adv. frames	distal	corrupted CIFAR10-C
	(seen)	(seen)	seen	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**	**unseen**
Normal	8.3	7.4	100.0	100.0	100.0	100.0	84.7	96.7	83.3	12.3
AT-50%	16.6	15.5	62.7	93.7	98.4	98.4	74.4	78.7	75.0	16.2
AT-100%	19.4	18.3	59.9	90.3	98.3	98.0	72.3	79.6	72.5	19.6
CCAT	10.1	6.7	68.4	92.4	**52.2**	**58.8**	**23.0**	**66.1**	**0.0**	**8.5**
* MSD	18.4	17.6	53.2	89.4	88.5	68.6	39.2	82.6	76.7	19.3
* TRADES	15.2	13.2	**43.5**	**81.0**	70.9	96.9	36.9	72.1	76.2	15.0
* AT-Madry	13.0	11.7	45.1	84.5	98.7	97.8	42.3	73.3	78.5	12.9
* LID	**6.4**	**4.9**	99.0	99.2	70.6	89.4	47.0	**66.1**	0.1	11.59
* MAHA	**6.4**	**4.9**	94.1	95.3	90.6	97.6	49.8	70.0	2.4	12.4

Table 2: Main Results: Generalizing Robustness. For L_∞, L_2, L_1, L_0 attacks and adversarial frames, we report per-example worst-case (confidence-thresholded) Err and RErr at 99%TPR across all attacks; ϵ is reported in the corresponding columns. For distal adversarial examples and corrupted examples, we report FPR and Err, respectively. L_∞ attacks with ϵ=0.3 on MNIST and $\epsilon = 0.03$ on SVHN/Cifar10 were used for training (seen). The remaining attacks were not encountered during training (**unseen**). CCAT outperforms AT and the other baselines regarding robustness against unseen attacks. FPRs included in the supplementary material. * Pre-trained models with different architecture, LID/MAHA use the same model.

testing. In addition, while T=200 iterations are sufficient against AT, we needed up to T=1000 iterations for CCAT.

5.2. Main Results (Table 2)

Robustness Against seen L_∞ Attacks: Considering Tab. 2 and L_∞ adversarial examples as seen during training, CCAT exhibits comparable robustness to AT. With 7.4%/68.4% RErr on MNIST/Cifar10, CCAT lacks behind AT-50% (1.7%/62.7%) only slightly. On SVHN, in contrast CCAT outperforms AT-50% and AT-100% significantly with 39.1% vs. 56.0% and 48.3%. We note that CCAT and AT-50% are trained on 50% clean / 50% adversarial examples. This is in contrast to AT-100% trained on 100% adversarial examples, which improves robustness slightly, e.g., from 56%/62.7% to 48.3%/59.9% on SVHN/Cifar10.

Robustness Against unseen L_p Attacks: Regarding unseen attacks, AT's robustness deteriorates quickly while CCAT is able to generalize robustness to novel threat models. On SVHN, for example, RErr of AT-50% goes up to 88.4%, 99.4%, 99.5% and 73.6% for larger L_∞, L_2, L_1 and L_0 attacks. In contrast, CCAT's robustness generalizes to these unseen attacks significantly better, with 53.1%, 29%, 31.7% and 3.5%, respectively. The results on MNIST and Cifar10 or for AT-100% tell a similar story. However, AT generalizes better to L_1 and L_0 attacks on MNIST, possibly due to the large L_∞-ball used during training ($\epsilon = 0.3$).

Here, training purely on adversarial examples, i.e., AT-100% is beneficial. On Cifar10, CCAT has more difficulties with large L_∞ attacks ($\epsilon = 0.06$) with 92.4% RErr. As detailed in the supplementary material, these observations are supported by considering FPR. AT benefits from considering FPR as clean Err is not taken into account. On Cifar10, for example, 47.6% FPR compared to 62.7% RErr for AT-50%. This is less pronounced for CCAT due to the improved Err compared to AT. Overall, CCAT improves robustness against arbitrary (unseen) L_p attacks, demonstrating that CCAT indeed extrapolates near-uniform predictions beyond the L_∞ ϵ-ball used during training.

Comparison to MSD and TRADES: TRADES is able to outperform CCAT alongside AT (including AT-Madry) on Cifar10 with respect to the L_∞ adversarial examples seen during training: 43.5% RErr compared to 68.4% for CCAT. This might be a result of training on 100% adversarial examples and using more complex models: TRADES uses a WRN-10-28 with roughly 46.1M weighs, in contrast to our ResNet-18 with 4.3M (and ResNet-20 with 11.1M for MSD). However, regarding unseen L_2, L_1 and L_0 attacks, CCAT outperforms TRADES with 52.2%, 58.8% and 23% compared to 70.9%, 96.9% and 36.9% in terms of RErr. Similarly, CCAT outperforms MSD. This is surprising, as MSD trains on both L_2 and L_1 attacks with smaller ϵ, while CCAT does not. Only against larger L_∞ adversarial examples with $\epsilon = 0.06$, TRADES reduces RErr from 92.4% (CCAT) to 81%. Similar to AT, TRADES also generalizes better to L_2, L_1 or L_0 on MNIST, while MSD is not able to compete. Overall, compared to MSD and TRADES, the robustness obtained by CCAT generalizes better to previously unseen attacks. We also note that, on MNIST, CCAT outperforms the robust Analysis-by-Synthesis (ABS) approach of (Schott et al., 2019) wrt. L_∞, L_2, and L_0 attacks.

Detection Baselines: The detection methods LID and MAHA are outperformed by CCAT across all datasets and threat models. On SVHN, for example, MAHA obtains 73% RErr against the seen L_∞ attacks and 79.5%, 78.1%, 67.5% and 41.5% RErr for the unseen L_∞, L_2, L_1 and L_0 attacks. LID is consistently outperformed by MAHA on SVHN. This is striking, as we *only* used PGD-CE and PGD-Conf to attack these approaches and emphasizes the importance of training *adversarially* against an adaptive attack to successfully reject adversarial examples.

Robustness Against Unconventional Attacks: Against adversarial frames, robustness of AT reduces to 73.7% /62.3% RErr (AT-50%/100%), even on MNIST, while CCAT achieves 0.2%. MSD, in contrast, is able to preserve robustness better with 8.8% RErr, which might be due to the L_2 and L_1 attacks seen during training. CCAT outperforms both approaches with 0.2% RErr, as does TRADES. On SVHN and Cifar10, however, CCAT outperforms all approaches, including TRADES, considering adversarial frames. Against distal adversarial examples, CCAT outperforms all approaches significantly, with 0% FPR, compared to the second-best of 72.5% for AT-100% on Cifar10. Only the detection baselines LID and MAHA are competitive, reaching close to 0% FPR. This means that CCAT is able to extrapolate low-confidence distributions to far-away regions of the input space. Finally, we consider corrupted examples (e.g., blur, noise, transforms etc.) where CCAT also improves results, i.e., mean Err across all corruptions. On Cifar10-C, for example, CCAT achieves 8.5% compared 12.9% for AT-Madry and 12.3% for normal training. On MNIST-C, only MSD yields a comparably low Err: 6% vs. 5.7% for CCAT.

Improved Test Error: CCAT also outperforms AT regarding Err, coming close to that of normal training. On all datasets, *confidence-thresholded* Err for CCAT is better or equal than that of normal training. On Cifar10, only LID/MAHA achieve a better standard and confidence-thresholded Err using a ResNet-34 compared to our ResNet-20 for CCAT (21.2M vs. 4.3M weights). In total the performance of CCAT shows that the robustness-generalization trade-off can be improved significantly.

Our **supplementary material** includes detailed descriptions of our PGD-Conf attack (including pseudo-code), a discussion of our confidence-thresholded RErr, and more details regarding baselines. We also include results for confidence thresholds at 98% and 95%TPR, which improves results only slightly, at the cost of "throwing away" significantly more clean examples. Furthermore, we provide ablation studies, qualitative examples and per-attack results.

6. Conclusion

Adversarial training results in robust models against the threat model *seen* during training, e.g., L_∞ adversarial examples. However, generalization to *unseen* attacks such as other L_p adversarial examples or larger L_∞ perturbations is insufficient. We propose **confidence-calibrated adversarial training (CCAT)** which biases the model towards low confidence predictions on adversarial examples and beyond. Then, adversarial examples can easily be rejected based on their confidence. Trained exclusively on L_∞ adversarial examples, CCAT improves robustness against unseen threat models such as larger L_∞, L_2, L_1 and L_0 adversarial examples, adversarial frames, distal adversarial examples and corrupted examples. Additionally, accuracy is improved in comparison to adversarial training. We thoroughly evaluated CCAT using 7 different white-and black-box attacks with up to 50 random restarts and 5000 iterations. These attacks where adapted to CCAT by directly maximizing confidence. We reported worst-case robust test error, extended to our confidence-thresholded setting, across *all* attacks.

References

Alaifari, R., Alberti, G. S., and Gauksson, T. Adef: an iterative algorithm to construct adversarial deformations. In *ICLR*, 2019.

Alayrac, J., Uesato, J., Huang, P., Fawzi, A., Stanforth, R., and Kohli, P. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019.

Amsaleg, L., Bailey, J., Barbe, D., Erfani, S. M., Houle, M. E., Nguyen, V., and Radovanovic, M. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *WIFS*, 2017.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv.org*, abs/1912.00049, 2019.

Athalye, A. and Carlini, N. On the robustness of the CVPR 2018 white-box adversarial example defenses. *arXiv.org*, abs/1804.03286, 2018.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Bhagoji, A. N., Cullina, D., and Mittal, P. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv.org*, abs/1704.02654, 2017.

Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. *arXiv.org*, abs/1712.09665, 2017.

Cai, Q., Liu, C., and Song, D. Curriculum adversarial training. In *IJCAI*, 2018.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec*, 2017a.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *SP*, 2017b.

Carlini, N., Athalye, A., Brendel, N. P. W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness. *arXiv.org*, abs/1902.06705, 2019.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Cheng, M., Lei, Q., Chen, P., Dhillon, I. S., and Hsieh, C. CAT: customized adversarial training for improved robustness. *arXiv.org*, abs/2002.06789, 2020.

Croce, F. and Hein, M. Sparse and imperceivable adversarial attacks. In *ICCV*, 2019.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *ICML*, 2019.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B. Detecting adversarial samples from artifacts. *arXiv.org*, abs/1703.00410, 2017.

Gong, Z., Wang, W., and Ku, W. Adversarial and clean data are not twins. *arXiv.org*, abs/1704.04960, 2017.

Goodfellow, I., Qin, Y., and Berthelot, D. Evaluation methodology for attacks against confidence thresholding models, 2019. URL https://openreview.net/forum?id=H1g0piA9tQ.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Grefenstette, E., Stanforth, R., O'Donoghue, B., Uesato, J., Swirszcz, G., and Kohli, P. Strength in numbers: Trading-off robustness and computation via adversarially-trained ensembles. *arXiv.org*, abs/1811.09300, 2018.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv.org*, abs/1702.06280, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hein, M., Andriushchenko, M., and Bitterwolf, J. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *CVPR*, 2019.

Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Hendrycks, D. and Gimpel, K. Early methods for detecting adversarial images. In *ICLR*, 2017.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

Huang, R., Xu, B., Schuurmans, D., and Szepesvári, C. Learning with a strong adversary. *arXiv.org*, abs/1511.03034, 2015.

Ilyas, A., Engstrom, L., Athalye, A., and Lin, J. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018.

Jacobsen, J., Behrmann, J., Carlini, N., Tramèr, F., and Papernot, N. Exploiting excessive invariance caused by norm-bounded adversarial robustness. *arXiv.org*, abs/1903.10484, 2019a.

Jacobsen, J., Behrmann, J., Zemel, R. S., and Bethge, M. Excessive invariance causes adversarial vulnerability. In *ICLR*, 2019b.

Kang, D., Sun, Y., Hendrycks, D., Brown, T., and Steinhardt, J. Testing robustness against unforeseen adversaries. *arXiv.org*, abs/1908.08016, 2019.

Khoury, M. and Hadfield-Menell, D. On the geometry of adversarial examples. *arXiv.org*, abs/1811.00525, 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Laidlaw, C. and Feizi, S. Playing it safe: Adversarial robustness with an abstain option. *arXiv.org*, abs/1911.11253, 2019.

Lamb, A., Verma, V., Kannala, J., and Bengio, Y. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *AISec*, 2019.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11), 1998.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Li, B., Chen, C., Wang, W., and Carin, L. On norm-agnostic robustness of adversarial training. *arXiv.org*, abs/1905.06455, 2019.

Li, X. and Li, F. Adversarial examples detection in deep networks with convolutional filter statistics. In *ICCV*, 2017.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. *ICLR*, 2017.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S. N. R., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models. *ICML*, 2020.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. In *ICLR*, 2017.

Miyato, T., Maeda, S.-i., Koyama, M., Nakae, K., and Ishii, S. Distributional smoothing with virtual adversarial training. *ICLR*, 2016.

Mu, N. and Gilmer, J. Mnist-c: A robustness benchmark for computer vision. *ICML Workshops*, 2019.

Narodytska, N. and Kasiviswanathan, S. P. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, 2017.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NeurIPS*, 2011.

Pang, T., Du, C., Dong, Y., and Zhu, J. Towards robust detection of adversarial examples. In *NeurIPS*, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. In *NeurIPS Workshops*, 2017.

Pérolat, J., Malinowski, M., Piot, B., and Pietquin, O. Playing the game of universal adversarial perturbations. *CoRR*, abs/1809.07802, 2018.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. Adversarial training can hurt generalization. *arXiv.org*, abs/1906.06032, 2019.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *ICLR*, 2019.

Shafahi, A., Najibi, M., Xu, Z., Dickerson, J. P., Davis, L. S., and Goldstein, T. Universal adversarial training. In *AAAI*, 2020.

Sharma, Y. and Chen, P. Attacking the madry defense model with l_1-based adversarial examples. In *ICLR Workshops*, 2018.

Stutz, D., Hein, M., and Schiele, B. Disentangling adversarial robustness and generalization. *CVPR*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *NeurIPS*, 2019.

Tramèr, F., Kurakin, A., Papernot, N., Boneh, D., and McDaniel, P. D. Ensemble adversarial training: Attacks and defenses. *ICLR*, 2018.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., and Yuille, A. L. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zajac, M., Zolna, K., Rostamzadeh, N., and Pinheiro, P. O. Adversarial framing for image and video classification. In *AAAI Workshops*, 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.