
RobustBench: a standardized adversarial robustness benchmark

Francesco Croce*
Univ. of Tübingen

Maksym Andriushchenko*
EPFL

Vikash Sehwal*
Princeton Univ.

Edoardo Debenedetti*
EPFL

Nicolas Flammarion
EPFL

Mung Chiang
Purdue Univ.

Prateek Mittal
Princeton Univ.

Matthias Hein
Univ. of Tübingen

Abstract

As a research community, we are still lacking a systematic understanding of the progress on adversarial robustness which often makes it hard to identify the most promising ideas in training robust models. A key challenge in benchmarking robustness is that its evaluation is often error-prone leading to *robustness overestimation*. Our goal is to establish a *standardized benchmark* of adversarial robustness, which as accurately as possible reflects the robustness of the considered models within a reasonable computational budget. To this end, we start by considering the image classification task and introduce restrictions (possibly loosened in the future) on the allowed models. We evaluate adversarial robustness with *AutoAttack* [28], an ensemble of white- and black-box attacks, which was recently shown in a large-scale study to improve almost all robustness evaluations compared to the original publications. To prevent overadaptation of new defenses to AutoAttack, we welcome external evaluations based on adaptive attacks [142], especially where AutoAttack flags a potential overestimation of robustness. Our leaderboard, hosted at <https://robustbench.github.io/>, contains evaluations of 120+ models and aims at reflecting the current state of the art in image classification on a set of well-defined tasks in ℓ_∞ - and ℓ_2 -threat models and on common corruptions, with possible extensions in the future. Additionally, we open-source the library <https://github.com/RobustBench/robustbench> that provides unified access to 80+ robust models to facilitate their downstream applications. Finally, based on the collected models, we analyze the impact of robustness on the performance on distribution shifts, calibration, out-of-distribution detection, fairness, privacy leakage, smoothness, and transferability.

1 Introduction

Since the finding that state-of-the-art deep learning models are vulnerable to small input perturbations called *adversarial examples* [135], achieving adversarially robust models has become one of the most studied topics in the machine learning community. The main difficulty of robustness evaluation is that it is a computationally hard problem even for simple ℓ_p -bounded perturbations [70] and exact approaches [138] do not scale to large enough models. There are already more than 3000 papers on this topic [16], but it is often unclear which defenses against adversarial examples indeed improve robustness and which only make the typically used attacks overestimate the actual robustness. There is an important line of work on recommendations for how to perform adaptive attacks that are selected specifically for a particular defense [5, 17, 142] which have in turn shown that several

*Equal contribution.

Rank	Method	Standard accuracy	AutoAttack robust accuracy	Best known robust accuracy	AA eval. potentially unreliable	Extra data	Architecture	Venue
1	Fixing Data Augmentation to Improve Adversarial Robustness <small>66.56% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)</small>	92.23%	66.58%	66.56%	×	☑	WideResNet-70-16	arXiv, Mar 2021
2	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples <small>65.87% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)</small>	91.10%	65.88%	65.87%	×	☑	WideResNet-70-16	arXiv, Oct 2020
3	Fixing Data Augmentation to Improve Adversarial Robustness <small>It uses additional 1M synthetic images in training. 64.58% robust accuracy is due to the original evaluation (AutoAttack + MultiTargeted)</small>	88.50%	64.64%	64.58%	×	×	WideResNet-106-16	arXiv, Mar 2021

Figure 1: The top-3 entries of our CIFAR-10 leaderboard hosted at <https://robustbench.github.io/> for the ℓ_∞ -perturbations of radius $\varepsilon_\infty = 8/255$.

seemingly robust defenses fail to be robust. However, recently Tramèr et al. [142] observe that although several recently published defenses have tried to perform adaptive evaluations, many of them could still be broken by new adaptive attacks. We observe that there are repeating patterns in many of these defenses that prevent standard attacks from succeeding. This motivates us to impose restrictions on the defenses we consider in our proposed benchmark, RobustBench, which aims at *standardized* adversarial robustness evaluation. Specifically, we rule out (1) classifiers which have zero gradients with respect to the input [13, 53], (2) randomized classifiers [161, 100], and (3) classifiers that use an optimization loop at inference time [118, 84]. Often, non-certified defenses that violate these three restrictions only make gradient-based attacks harder but do not substantially improve robustness [17]. However, we will lift (some of) these constraints if a standardized reliable evaluation method for those defenses becomes available. We start from benchmarking robustness with respect to the ℓ_∞ - and ℓ_2 -threat models, since they are the most studied settings in the literature. We use the recent AutoAttack [28] as our current standard evaluation which is an ensemble of diverse parameter-free attacks (white- and black-box) that has shown reliable performance over a large set of models that satisfy our restrictions. Moreover, we accept and encourage external evaluations, e.g. with adaptive attacks, to improve our standardized evaluation, especially for the leaderboard entries whose evaluation may be unreliable according to the *flag* that we propose. Additionally, we collect models robust against common image corruptions [58] as these represent another important type of perturbations which should not change the decision of a classifier.

Contributions. We make following key contributions with our RobustBench benchmark:

- **Leaderboard** (<https://robustbench.github.io/>): a website with the leaderboard (see Fig. 1) based on *more than 120* evaluations where it is possible to track the progress and the current state of the art in adversarial robustness based on a standardized evaluation using AutoAttack *complemented* by (external) adaptive evaluations. The goal is to clearly identify the most successful ideas in training robust models to accelerate the progress in the field.
- **Model Zoo** (<https://github.com/RobustBench/robustbench>): a collection of the most robust models that are easy to use for any downstream applications. As an example, we expect that this will foster the development of better adversarial attacks by making it easier to perform evaluations on a large set of *more than 80* models.
- **Analysis:** based on the collected models from the Model Zoo, we provide an analysis of how robustness affects the performance on distribution shifts, calibration, out-of-distribution detection, fairness, privacy leakage, smoothness, and transferability. In particular, we find that robust models are significantly *underconfident* that leads to worse calibration, and that not all robust models have higher privacy leakage than standard models.

2 Background and related work

Adversarial perturbations. Let $x \in \mathbb{R}^d$ be an input point and $y \in \{1, \dots, C\}$ be its correct label. For a classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}^C$, we define a *successful adversarial perturbation* with respect to the

perturbation set $\Delta \subseteq \mathbb{R}^d$ as a vector $\delta \in \mathbb{R}^d$ such that

$$\arg \max_{c \in \{1, \dots, C\}} f(x + \delta)_c \neq y \quad \text{and} \quad \delta \in \Delta, \quad (1)$$

where typically the perturbation set Δ is chosen such that *all* points in $x + \delta$ have y as their true label. This motivates a typical robustness measure called *robust accuracy*, which is the fraction of datapoints on which the classifier f predicts the correct class for all possible perturbations from the set Δ . Computing the exact robust accuracy is in general intractable and, when considering ℓ_p -balls as Δ , NP-hard even for single-layer neural networks [70, 149]. In practice, an *upper bound* on the robust accuracy is computed via some *adversarial attacks* which are mostly based on optimizing some differentiable loss (e.g., cross entropy) using local search algorithms like projected gradient descent (PGD) in order to find a successful adversarial perturbation. The tightness of the upper bound depends on the effectiveness of the attack: unsuitable techniques or suboptimal parameters (e.g., the step size and the number of iterations) can make the models appear more robust than they actually are [36, 95], especially in the presence of phenomena like gradient obfuscation [5]. Certified methods [151, 48] instead provide *lower bounds* on robust accuracy which often underestimate robustness significantly, especially if the certification was not part of the training process. Thus, in our benchmark, we do not measure lower bounds and focus only on upper bounds which are typically much tighter [138].

Threat models. We focus on the fully white-box setting, i.e. the model f is assumed to be fully known to the attacker. The threat model is defined by the set Δ of the allowed perturbations: the most widely studied ones are the ℓ_p -perturbations, i.e. $\Delta_p = \{\delta \in \mathbb{R}^d, \|\delta\|_p \leq \varepsilon\}$, particularly for $p = \infty$ [135, 46, 88]. We rely on thresholds ε established in the literature which are chosen such that the true label should stay the same for each in-distribution input within the perturbation set. We note that robustness towards small ℓ_p -perturbations is a necessary but not sufficient notion of robustness which has been criticized in the literature [45]. It is an active area of research to develop threat models which are more aligned with the human perception such as spatial perturbations [42, 39], Wasserstein-bounded perturbations [152, 63], perturbations of the image colors [80] or ℓ_p -perturbations in the latent space of a neural network [81, 150]. However, despite the simplicity of the ℓ_p -perturbation model, it has numerous interesting applications that go beyond security considerations [141, 116] and span transfer learning [117, 145], interpretability [143, 71, 38], generalization [158, 174, 9], robustness to unseen perturbations [68, 158, 81, 73], stabilization of GAN training [173]. Thus, improvements in ℓ_p -robustness have the potential to improve many of these downstream applications.

Additionally, we provide leaderboards for *common image corruptions* [58] that try to mimic modifications of the input images which can occur naturally. Unlike ℓ_p adversarial perturbations, they are not imperceptible and evaluation on them is done in the average-case fashion, i.e. there is no attacker who aims at changing the classifier’s decision. In this case, the robustness of a model is evaluated as classification accuracy on the corrupted images, averaged over corruption types and severities.

Related libraries and benchmarks. There are many libraries that focus primarily on implementations of popular adversarial attacks such as FoolBox [110], Cleverhans [104], AdverTorch [33], AdvBox [47], ART [98], SecML [92], DeepRobust [85]. Some of them also provide implementations of several basic defenses, but they do not include up-to-date state-of-the-art models. The two challenges [79, 10] hosted at NeurIPS 2017 and 2018 aimed at finding the most robust models for specific attacks, but they had a predefined deadline, so they could capture the best defenses only at the time of the competition. Ling et al. [86] proposed DEEPSEC, a benchmark that tests many combinations of attacks and defenses, but suffers from a few shortcomings as suggested by Carlini [15]: (1) reporting average-case instead of worst-case performance over multiple attacks, (2) evaluating robustness in threat models different from the ones used for training, (3) using excessively large perturbations. Chen and Gu [21] proposed a new *hard-label* black-box attack, RayS, and evaluated it on a range of models which led to a leaderboard (<https://github.com/uclaml/RayS>). Despite being a state-of-the-art hard-label black-box attack, the robust accuracy in the leaderboard given by RayS still tends to be overestimated even compared to the original evaluations.

Recently, Dong et al. [35] have provided an evaluation of a few defenses (in particular, 3 for ℓ_∞ - and 2 for ℓ_2 -norm on CIFAR-10) against multiple commonly used attacks. However, they did not include some of the best performing defenses [60, 19, 50, 111] and attacks [49, 27], and in a few cases, their evaluation suggests robustness higher than what was reported in the original papers. Moreover, they do not impose any restrictions on the models they accept to the benchmark. RobustML (<https://www.robust-ml.org/>) aims at collecting robustness claims for defenses together with external evaluations. Their format does not assume running any baseline attack, so it relies entirely

on evaluations submitted by the community, which however do not occur often enough. Thus even though RobustML has been a valuable contribution to the community, now it does not provide a comprehensive overview of the recent state of the art in adversarial robustness.

Finally, it has become common practice to test new attacks wrt ℓ_∞ on the publicly available models from Madry et al. [88] and Zhang et al. [168], since those represent widely accepted defenses which have stood many thorough evaluations. However, having only two models per dataset (MNIST and CIFAR-10) does not constitute a sufficiently large testbed, and, because of the repetitive evaluations, some attacks may already overfit to those defenses.

What is different in RobustBench. Learning from these previous attempts, RobustBench presents a few different features compared to the aforementioned benchmarks: (1) a baseline worst-case evaluation with an ensemble of *strong, standardized* attacks [28] which includes both white- and black-box attacks, unlike RobustML which is *solely* based on adaptive evaluations, integrated by external evaluations, (2) we add a *flag* in AutoAttack raised when the evaluation might be unreliable, in which case we do additional adaptive evaluations ourselves and encourage the community to contribute, (3) clearly defined threat models that correspond to the ones used during training of submitted models, (4) evaluation of not only standard defenses [88, 168] but also of more recent improvements such as [19, 50, 111]. Moreover, RobustBench is designed as an *open-ended* benchmark that keeps an up-to-date leaderboard, and we welcome contributions of new defenses and evaluations using adaptive attacks. Finally, we open source the Model Zoo for convenient access to the 80+ most robust models from the literature which can be used for downstream tasks and facilitate the development of new standardized attacks.

3 Description of RobustBench

We start by providing a detailed layout of our proposed leaderboards for ℓ_∞ , ℓ_2 , and common corruption threat models. Next, we present the Model Zoo, which provides unified access to most networks from our leaderboards.

3.1 Leaderboard

Restrictions. We argue that accurate benchmarking adversarial robustness in a standardized way requires some restrictions on the type of considered models. The goal of these restrictions is to prevent submissions of defenses that cause some standard attacks to fail without truly improving robustness. Specifically, we consider only classifiers $f : \mathbb{R}^d \rightarrow \mathbb{R}^C$ that

- have in general *non-zero gradients* with respect to the inputs. Models with zero gradients, e.g., that rely on quantization of inputs [13, 53], make gradient-based methods ineffective thus requiring zeroth-order attacks, which do not perform as well as gradient-based attacks. Alternatively, specific adaptive evaluations, e.g. with Backward Pass Differentiable Approximation [5], can be used which, however, can hardly be standardized. Moreover, we are not aware of existing defenses solely based on having zero gradients for large parts of the input space which would achieve competitive robustness.
- have a *fully deterministic forward pass*. To evaluate defenses with stochastic components, it is a common practice to combine standard gradient-based attacks with Expectation over Transformations [5]. While often effective it might be not sufficient, as shown by Tramèr et al. [142]. Moreover, the classification decision of randomized models may vary over different runs for the same input, hence even the definition of robust accuracy differs from that of deterministic networks. We note that randomization *can* be useful for improving robustness and deriving robustness certificates [82, 25], but it also introduces variance in the gradient estimators (both white- and black-box) making standard attacks much less effective.
- do not have an *optimization loop* in the forward pass. This makes backpropagation through it very difficult or extremely expensive. Usually, such defenses [118, 84] need to be evaluated adaptively with attacks that rely on a combination of hand-crafted losses.

Some of these restrictions were also discussed by [12] for the warm-up phase of their challenge. We refer the reader to Appendix E therein for an illustrative example of a trivial defense that bypasses gradient-based and some of the black-box attacks they consider. We believe that such constraints

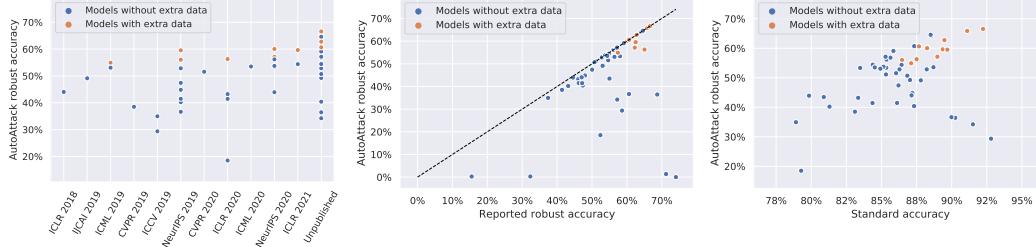


Figure 2: Visualization of the robustness and accuracy of 54 CIFAR-10 models from the RobustBench ℓ_∞ -leaderboard. Robustness is evaluated using ℓ_∞ -perturbations with $\varepsilon_\infty = 8/255$.

are necessary at the moment since they allow an accurate standardized evaluation which makes the leaderboard meaningful and sustainable. In fact, for defenses not fulfilling the restrictions there is no standard evaluation which is shown to generalize and perform well across techniques, thus one has to resort to time-consuming adaptive attacks specifically tailored for each case. In the design of our benchmark, we thought that it is more important that the robustness evaluation is reliable, rather than being open to all possible defenses with the risk that the robustness is drastically overestimated. As this can lead to a potential bias in our leaderboard, we will lift the restrictions if reliable standardized evaluation methods for these modalities become available in the literature.

Overall setup. We set up leaderboards for the ℓ_∞ , ℓ_2 and common corruption threat models on CIFAR-10, CIFAR-100 [75], and ImageNet [32] datasets (see Table 1 for details). We use the fixed budgets of $\varepsilon_\infty = 8/255$ and $\varepsilon_2 = 0.5$ for the ℓ_∞ and ℓ_2 leaderboards for CIFAR-10 and CIFAR-100. For ImageNet, we use $\varepsilon_\infty = 4/255$ and in App. D, we discuss how we handle that different models use different image resolutions for ImageNet. Most of the models shown in the leaderboards are taken from papers published at top-tier machine learning and computer vision conferences as shown in Fig. 2 (left). For each entry we report the reference to the original paper, standard and robust accuracy under the specific threat model (see the next paragraph for details), network architecture, venue where the paper appeared and possibly notes regarding the model. We also highlight when extra data (often, the dataset introduced by Carmon et al. [19]) is used since it gives a clear advantage for both clean and robust accuracy. If any other attack achieves lower robust accuracy than AutoAttack then we also report it. Moreover, the leaderboard allows to search the entries by their metadata (such as title, architecture, venue) which can be useful to compare different methods that use the same architecture or to search for papers published at some conference.

Evaluation of defenses. The evaluation of robust accuracy on common corruptions [58] involves simply computing the average accuracy on corrupted images over different corruption types and severity levels.¹ To evaluate robustness of ℓ_∞ and ℓ_2 defenses, we currently use AutoAttack [28]. It is an ensemble of four attacks that are run sequentially: a variation of PGD attack with automatically adjusted step sizes, with (1) the cross entropy loss and (2) the difference of logits ratio loss, which is a rescaling-invariant margin-based loss function, (3) the targeted version of the FAB attack [27], which minimizes the ℓ_p -norm of the perturbations, and (4) the black-box Square Attack [4]. Each subsequent attack is run on the points for which an adversarial example has not been found by the preceding attacks. We choose AutoAttack as it includes both black-box and white-box attacks, does not require hyperparameter tuning (in particular, the step size), and consistently improves the results reported in the original papers for almost all the models (see Fig. 2 (middle)). If in the future some new standardized and parameter-free attack is shown to consistently outperform AutoAttack on a wide set of models given a similar computational cost, we will adopt it as standard evaluation. In order to verify the reproducibility of the results, we perform the standardized evaluation independently of the authors of the submitted models. We encourage evaluations of the models in the leaderboard with adaptive or external attacks to reflect the best available upper bound on the true robust accuracy (see a pre-formatted issue template in our repository²), in particular in the case where AutoAttack flags that it might not be reliable (see paragraph below). For example, Goyal et al. [50] and Rebuffi et al. [111] evaluate their models with a hybrid of AutoAttack and MultiTargeted attack [49], that in some cases reports slightly lower robust accuracy than AutoAttack alone. We reflect the additional

¹A breakdown over corruptions and severities is also available, e.g. for CIFAR-10 models see: https://github.com/RobustBench/robustbench/blob/master/model_info/cifar10/corruptions/unaggregated_results.csv

²<https://github.com/RobustBench/robustbench/issues/new/choose>

evaluations in our leaderboard by reporting in a separate column the robust accuracy for the worst case of AutoAttack and all other evaluations. Below we show an example of how one can use our library to easily benchmark a model (either external one or taken from the Model Zoo):

```
from robustbench.eval import benchmark
clean_acc, robust_acc = benchmark(model, dataset='cifar10', threat_model='Linf')
```

Moreover, in Appendix E we also show the variability of the robust accuracy given by AutoAttack over random seeds and report its runtime for a few models from different threat models.

Identifying potential need for adaptive attacks. Although AutoAttack provides an accurate estimation of robustness for most models that satisfy the restrictions mentioned above, there might still be corner cases when AutoAttack overestimates robustness of a model that satisfies the restrictions. Carlini et al. [17] suggest that one indicator of possible overestimation of robustness is when black-box attacks are more effective than white-box ones. We noticed that this is the case for the model from Xiao et al. [156] where the black-box Square Attack [4] improves by *more than* 10% the robust accuracy given by the previous white-box attacks in AutoAttack. We run a simple adaptive attack: Square Attack with multiple random restarts (30 instead of 1) decreases the robust accuracy from the 18.50% of AutoAttack to 7.40%. We note that AutoAttack did not fail completely for this model and correctly revealed a lower level of robustness than reported (52.4%), although the exact robust accuracy was overestimated. Based on this case, we integrate a *flag* in AutoAttack: a warning is output whenever Square Attack reduces of more than 0.2% the robust accuracy compared to the white-box gradient-based attacks in AutoAttack. In this case, AutoAttack evaluation might be not fully reliable and adaptive attacks might be necessary, so we flag the corresponding entries in the leaderboard (currently, only the model of Xiao et al. [156]). Moreover, for the sake of convenience, we also integrate in AutoAttack flags that automatically inform the user if the restrictions are violated.³

Adding new defenses. We believe that the leaderboard is only useful if it reflects the latest advances in the field, so it needs to be constantly updated with new defenses. We intend to include evaluations of new techniques and we welcome contributions from the community which help to keep the benchmark up-to-date. We require new entries to (1) satisfy the three restrictions stated above, (2) to be accompanied by a publicly available paper (e.g., an arXiv preprint) describing the technique used to achieve the reported results, and (3) share the model checkpoints (not necessarily publicly). We also allow *temporarily* adding entries without providing checkpoints given that the authors evaluate their models with AutoAttack. However, we will mark such evaluations as *unverified*, and to encourage reproducibility, we reserve the right to remove an entry later on if the corresponding model checkpoint is not provided. It is possible to add a new defense to the leaderboard and (optionally) the Model Zoo by opening an issue with a predefined template in our repository <https://github.com/RobustBench/robustbench>, where more details about new additions can be found.

3.2 Model Zoo

We collect the checkpoints of many networks from the leaderboard in a single repository hosted at <https://github.com/RobustBench/robustbench> after obtaining the permission of the authors (see Appendix B for the information on the licenses). The goal of this repository, the Model Zoo, is to make the usage of robust models as simple as possible to facilitate various downstream applications and analyses of general trends in the field. In fact, even when the checkpoints of the proposed method are made available by the authors, it is often time-consuming and not straightforward to integrate them in the same framework because of many factors such as small variations in the architectures, custom input normalizations, etc. For simplicity of implementation, at the moment we include only models implemented in PyTorch [105]. Below we illustrate how a model can be automatically downloaded and loaded via its identifier and threat model within two lines of code:

```
from robustbench.utils import load_model
model = load_model(model_name='Ding2020MMA', dataset='cifar10', threat_model='L2')
```

At the moment, all models (see Table 1 and Appendix G for details) are variations of ResNet [55] and WideResNet architectures [164] of different depth and width. However, we note that the benchmark and Model Zoo are not restricted only to residual or convolutional networks, and we are ready to

³See https://github.com/fra31/auto-attack/blob/master/flags_doc.md for details

Table 1: The total number of models in the Model Zoo and leaderboards per dataset and threat model.

Threat model	CIFAR-10		CIFAR-100		ImageNet	
	Model Zoo	Leaderboard	Model Zoo	Leaderboard	Model Zoo	Leaderboard
ℓ_∞	39	63	14	14	5	6
ℓ_2	17	18	-	-	-	-
Common corruptions [58]	7	15	2	4	5	7

add any other architecture. We include the most robust models, e.g. those from Rebuffi et al. [111], but there are also defenses which pursue additional goals alongside adversarial robustness at the fixed threshold we use: e.g., Sehwag et al. [122] consider networks which are robust and compact, Wong et al. [153] focus on computationally efficient adversarial training, Ding et al. [34] aim at input-adaptive robustness as opposed to robustness within a single ℓ_p -radius. All these factors have to be taken into account when comparing different techniques, as they have a strong influence on the final performance. Thus, we highlight these factors in the footnotes below each paper’s title.

A testbed for new attacks. Another important use case of the Model Zoo is to simplify comparisons between different adversarial attacks on a wide range of models. First, the leaderboard already serves as a strong baseline for new attacks. Second, as mentioned above, new attacks are often evaluated on the models from Madry et al. [88] and Zhang et al. [168], but this may not provide a representative picture of their effectiveness. For example, currently the difference in robust accuracy between the first and second-best attacks in the CIFAR-10 leaderboard of Madry et al. [88] is only 0.03%, and between the second and third is 0.04%. Thus, we believe that a more thorough comparison should involve multiple models to prevent overfitting of the attack to one or two standard robust defenses.

4 Analysis

With unified access to multiple models from the Model Zoo, one can easily compute various performance metrics to see general trends. We analyze various aspects of robust classifiers, mostly for ℓ_∞ -robust models on CIFAR-10. Results for other threat models and datasets can be found in App. F.

Progress on adversarial defenses. In Fig. 2, we plot a breakdown over conferences, the amount of robustness overestimation reported in the original papers, and we also visualize the robustness-accuracy trade-off for the ℓ_∞ -models from the Model Zoo. First, we observe that for multiple published defenses, the reported robust accuracy is highly overestimated. We also find that the use of extra data is able to alleviate the robustness-accuracy trade-off as suggested in previous works [108]. However, so far all models with high robustness to perturbations of ℓ_∞ -norm up to $\varepsilon = 8/255$ still suffer from noticeable degradation in clean accuracy compared to standardly trained models. Finally, it is interesting to note that the best entries of the ℓ_p -leaderboards are still variants of PGD adversarial training [88, 168] but with various enhancements (extra data, early stopping, weight averaging).

Performance across various distribution shifts. We test the performance of the models from the Model Zoo on different distribution shifts ranging from common image corruptions (CIFAR-10-C, [58]) to dataset resampling bias (CIFAR-10.1, [112]) and image source shift (CINIC-10, [31]). For each of these datasets, we measure standard accuracy, and Fig. 3 shows that improvement in robust accuracy (which often comes with an improvement in standard accuracy) on CIFAR-10 correlates with an improvement in standard accuracy across distributional shifts. On CIFAR-10-C, robust models (particularly with respect to the ℓ_2 -norm) tend to give a significant improvement which agrees with the findings in [43]. Concurrently with our work, Taori et al. [137] study the robustness to different distribution shifts of many models trained on ImageNet, including some ℓ_p -robust models. Our conclusions qualitatively agree with theirs, and we hope that our collected set of models will help to provide a more complete picture. Moreover, we measure robust accuracy, in the same threat model used on CIFAR-10, using AutoAttack [28] (see Fig. 10 in Appendix F), in order to see how ℓ_p adversarial robustness generalizes across the datasets representing different distributions shifts, and observe a clear positive correlation between robust accuracy on CIFAR-10 and its variations.

Calibration. A classifier is *calibrated* if its predicted probabilities correctly reflect the actual accuracy [52]. In the context of adversarial training, calibration was considered in Hendrycks et al. [61] who focus on improving accuracy on common corruptions and in Augustin et al. [7] who focus mostly on

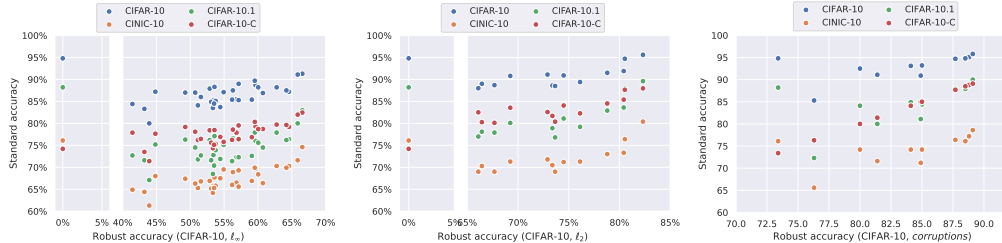


Figure 3: Standard accuracy of classifiers trained against ℓ_∞ (left), ℓ_2 (middle), and common corruption (right) threat model respectively, from our Model Zoo on various distribution shifts.

preventing overconfident predictions on out-of-distribution inputs. We instead focus on *in-distribution* calibration, and in Fig. 4 plot the expected calibration error (ECE) without and with temperature rescaling [54] to minimize the ECE (which is a simple but effective post-hoc calibration method, see Appendix F for details) together with the optimal temperature for a large set of ℓ_∞ models. We observe that most of the ℓ_∞ robust models are significantly *underconfident* since the optimal calibration temperature is less than one for most models. The only two models in Fig. 4 which are *overconfident* are the standard model and the model of Ding et al. [34] that aims to maximize the margin. We see that temperature rescaling is even more important for robust models since without any rescaling the ECE is as high as 70% for the model of Pang et al. [101] (and 21% on average) compared to 4% for the standard model. Temperature rescaling significantly reduces the ECE gap between robust and standard models but it does not fix the problem completely which suggests that it is worth incorporating calibration techniques also during training of robust models. For ℓ_2 robust models, the models can be on the contrary *more calibrated* by default, although the improvement vanishes if temperature rescaling is applied (see Appendix F).

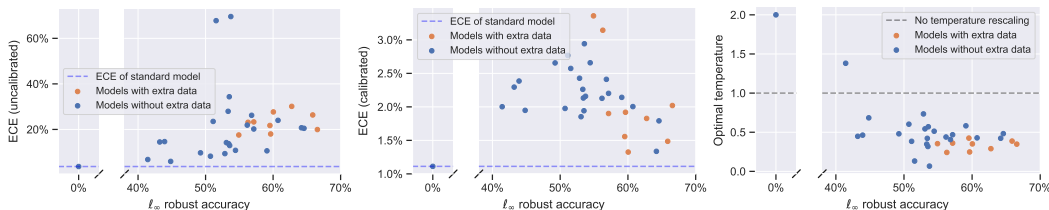


Figure 4: Expected calibration error (ECE) before (left) and after (middle) temperature rescaling, and the optimal rescaling temperature (right) for the ℓ_∞ -robust models.

Out-of-distribution detection. Ideally, a classifier should exhibit high uncertainty in its predictions when evaluated on *out-of-distribution* (OOD) inputs. One of the most straightforward ways to extract this uncertainty information is to use some threshold on the predicted confidence where OOD inputs are expected to have low confidence from the model [59]. An emerging line of research aims at developing OOD detection methods in conjunction with adversarial robustness [57, 120, 7]. In particular, Song et al. [132] demonstrated that adversarial training [88] leads to degradation in the robustness against OOD data. We further test this observation on all ℓ_∞ -models trained on CIFAR-10 from the Model Zoo on three OOD datasets: CIFAR-100 [75], SVHN [97], and Describable Textures Dataset [24]. We use the area under the ROC curve (AUROC) to measure the success in the detection of OOD data, and show the results in Fig. 5. With ℓ_∞ robust models, we find that compared to standard training, various robust training methods indeed lead to degradation of the OOD detection quality. While extra data in standard training can improve robustness against OOD inputs, it fails to provide similar improvements with robust training. We further find that ℓ_2 robust models have in general comparable OOD detection performance to standard models (see Fig. 12 in Appendix), while the model of Augustin et al. [7] achieves even better performance since their approach explicitly optimizes both robust accuracy and worst-case OOD detection performance.

Fairness in robustness. Recent works [8, 160] have noticed that robust training [88, 168] can lead to models whose performance varies significantly across subgroups, e.g. defined by classes. We will refer to this performance difference as *fairness*, and here we study the influence of robust training

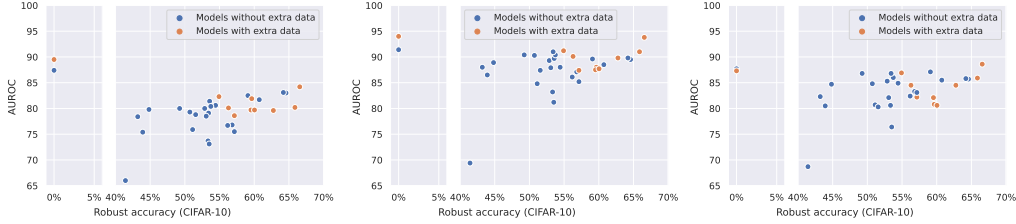


Figure 5: Visualization of the OOD detection quality (higher AUROC is better) for the ℓ_∞ -robust models trained on CIFAR-10 on three OOD datasets: CIFAR-100 (**left**), SVHN (**middle**), Describable Textures (**right**). We detect OOD inputs based on the maximum predicted confidence [59].

methods on fairness. In Fig. 6 we show the breakdown of standard and robust accuracy for the ℓ_∞ robust models, where one can see how the achieved robustness largely varies over classes. While in general the classwise standard and robust accuracy correlate well, the class “deer” in ℓ_∞ -threat model suffers a significant degradation, unlike what happens for ℓ_2 (see Appendix F), which might indicate that the features of such class are particularly sensitive to ℓ_∞ -bounded attacks. Moreover, we measure fairness with the relative standard deviation (RSD), defined as the standard deviation divided over the average, of robust accuracy over classes for which lower values mean more uniform distribution and higher robustness. We observe that better robust accuracy generally leads to lower RSD values which implies that the disparity among classes is reduced. (with a strong linear trend): improving the robustness of the models has then the effect of reducing the disparities among classes. However, some training techniques like MART [148] can noticeably increase the RSD and thus *increase the disparity* compared to other methods which achieve similar robustness (around 57%).

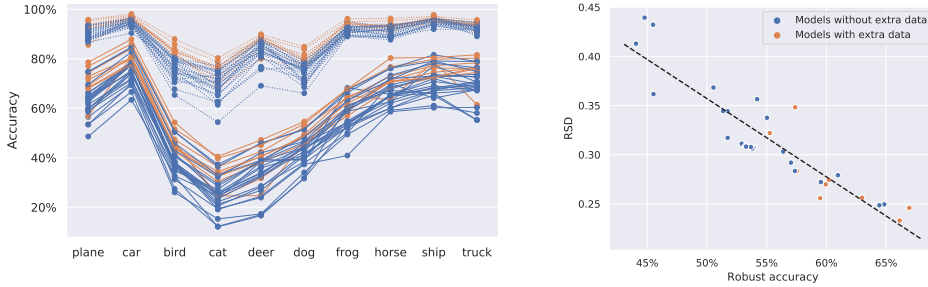


Figure 6: Fairness of ℓ_∞ -robust models. **Left**: classwise standard (dotted lines) and robust (solid) accuracy. **Right**: relative standard deviation (RSD) of robust accuracy over classes vs its average.

Privacy leakage. Deep neural networks are prone to memorizing training data [127, 18]. Recent work has highlighted that robust training exacerbates this problem [131]. We benchmark privacy leakage of training data across robust networks (Fig. 7). We calculate membership inference accuracy using output confidence of adversarial images from the training and test sets (see Appendix F for more details). It measures how accurately we can infer whether a sample was present in the training dataset. Our analysis reveals mixed trends. First, our results show that not all robust models have a significantly higher privacy leakage than a standard model. We find that the inference accuracy across robust models has a large variation, where some models even have lower privacy leakage than a standard model, and there is no strong correlation with robust accuracy. In contrast, it is largely determined by the generalization gap, as using the classifier confidence does not lead to a much higher inference accuracy than the baseline determined by the generalization gap (as shown in Fig. 7 (right)). Thus one can expect lower privacy leakage in robust networks as previous work explicitly aimed to reduce the generalization gap in robust training e.g. via early stopping [113, 168, 50].

Extra experiments. In Appendix F, we show extra experiments related to the points analyzed above and describe some of the implementation details. Also, we study how adversarial perturbations transfer between different models. We find that adversarial examples strongly transfer from robust to robust, non-robust to robust, and non-robust to non-robust networks. However, we observe poor transferability of adversarial examples from robust to non-robust networks. Finally, since prior works [56, 162] connected higher smoothness with better robustness, we analyze the smoothness of the

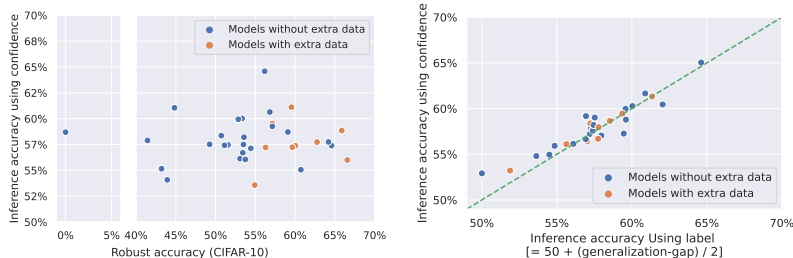


Figure 7: Privacy leakage of ℓ_∞ -robust models. We measure privacy leakage of training data in robust networks and compare it with robust accuracy (**left**) and generalization gap (**right**).

models both at intermediate and output layers. This confirms that, for a fixed architecture, standard training yields classifiers that are significantly less smooth than robust ones. This study of properties of networks illustrates another useful aspect of our Model Zoo.

5 Outlook

Conclusions. We believe that a *standardized* benchmark with clearly defined threat models, restrictions on submitted models, and tight upper bounds on robust accuracy is useful to show which ideas in training robust models are the most successful. While AutoAttack is for most models very reliable and accurate and allows a standardized comparison, we ensure by flagging potentially unreliable evaluations and doing additional adaptive attacks that the benchmark reflects the best possible robustness assessment with limited resources as the exact robustness evaluation is computationally infeasible. We remark that recent works have already referred to our leaderboards [74, 163, 89, 136, 159], in particular as reflecting the current state of the art [111, 83, 103], and used the networks of our Model Zoo to test new adversarial attacks [92, 115, 41, 119], evaluate test-time defenses [146] or perceptual distances derived from them [67], explore further properties of robust models [134, 172]. Additionally, we have shown that unified access to a *large* and *up-to-date* set of robust models can be useful to analyze multiple aspects related to robustness. First, one can easily analyze the progress of adversarial defenses over time including the amount of robustness overestimation and the robustness-accuracy tradeoff. Second, one can conveniently study the impact of robustness on other performance metrics such as accuracy under distribution shifts, calibration, out-of-distribution detection, fairness, privacy leakage, smoothness, and transferability. Overall, we think that the community has to develop a better understanding of how different types of robustness affect other aspects of the model performance and RobustBench can help to achieve this goal. Finally, we note that a good performance on our benchmark does not guarantee the safety of the benchmarked model in a real-world deployment since ℓ_p - and corruption robustness may not be sufficiently representative of all realistic threat models.

Future plans. Our intention in the future is to keep the current leaderboards up-to-date (see the maintenance plan in Appendix C) and add new leaderboards for other datasets and other threat models which become widely accepted in the community. We see as potential candidates (1) sparse perturbations, e.g. bounded by ℓ_0 , ℓ_1 -norm or adversarial patches [11, 26, 93, 29], (2) multiple ℓ_p -norm perturbations [140, 90], (3) adversarially optimized common corruptions [68, 69], (4) a broad set of perturbations unseen during training [81]. Another possible direction is the development of a standardized evaluation of recent defenses based on some form of test-time adaptation [126, 146], which do not fulfill the third restriction (no optimization loop). Finally, although the benchmark currently focuses on image classification, we think that its structure and principles should apply to other tasks (e.g., image segmentation [157], image retrieval [139]) and domains (e.g., natural language processing [2], malware detection [51]) where adversarial robustness can be of interest. Since this direction requires more domain-specific expertise, we welcome contributions from different communities to expand RobustBench.

Acknowledgements

We thank the authors who granted permission to use their models in our library. We also thank Chong Xiang for the helpful feedback on the benchmark, Eric Wong for the advice regarding the name of

the benchmark, and Evan Shelhamer for the helpful discussion on test-time defenses. Moreover, we thank the reviewers of both rounds of the NeurIPS 2021 Datasets and Benchmarks Track for their very useful suggestion that helped to improve the paper and make the discussion on standardized and adaptive attacks more balanced.

F.C. and M.H. acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A), the DFG Cluster of Excellence “Machine Learning – New Perspectives for Science”, EXC 2064/1, project number 390727645, and by DFG grant 389792660 as part of TRR 248. V.S. and P.M. acknowledge the support of the National Science Foundation (grants CNS-1553437 and CNS-1704105), the Army Research Office Young Investigator Prize, Army Research Laboratory (ARL) Army Artificial Intelligence Institute (A2I2), Office of Naval Research (ONR) Young Investigator Award, Schmidt DataX Fund, and Princeton E-affiliates Partnership.

References

- [1] M. Alfarra, J. C. Perez, A. Bibi, A. Thabet, P. Arbelaez, and B. Ghanem. Clustr: Clustering training for robustness. *arXiv preprint arXiv:2006.07682*, 2020.
- [2] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *EMNLP*, 2018.
- [3] M. Andriushchenko and N. Flammarion. Understanding and improving fast adversarial training. *NeurIPS*, 2020.
- [4] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.
- [5] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.
- [6] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman. Controlling neural level sets. *NeurIPS*, 2019.
- [7] M. Augustin, A. Meinke, and M. Hein. Adversarial robustness on in- and out-distribution improves explainability. *ECCV*, 2020.
- [8] P. Benz, C. Zhang, A. Karjauv, and I. S. Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. *arXiv preprint arXiv:2010.13365*, 2020.
- [9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [10] W. Brendel, J. Rauber, A. Kurakin, N. Papernot, B. Veliqi, M. Salathé, S. P. Mohanty, and M. Bethge. Adversarial vision challenge. In *NeurIPS Competition Track*, 2018.
- [11] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. In *NeurIPS 2017 Workshop on Machine Learning and Computer Security*, 2017.
- [12] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.
- [13] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.
- [14] D. A. Calian, F. Stimberg, O. Wiles, S.-A. Rebuffi, A. Gyorgy, T. Mann, and S. Goyal. Defending against image corruptions through adversarial augmentations. *arXiv*, 2021.
- [15] N. Carlini. A critique of the deepsec platform for security analysis of deep learning models. *arXiv preprint arXiv:1905.07112*, 2019.
- [16] N. Carlini. A complete list of all (arxiv) adversarial example papers. <https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html>, 2021. Accessed: 2021-06-08.
- [17] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.
- [18] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019.
- [19] Y. Carmon, A. Raghunathan, L. Schmidt, P. Liang, and J. C. Duchi. Unlabeled data improves adversarial robustness. *NeurIPS*, 2019.
- [20] A. Chan, Y. Tay, Y. S. Ong, and J. Fu. Jacobian adversarially regularized networks for robustness. *ICLR*, 2020.

- [21] J. Chen and Q. Gu. Rays: A ray searching method for hard-label adversarial attack. In *KDD*, 2020.
- [22] J. Chen, Y. Cheng, Z. Gan, Q. Gu, and J. Liu. Efficient robust training via backward smoothing. *arXiv*, 2020.
- [23] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, and Z. Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020.
- [24] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.
- [25] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.
- [26] F. Croce and M. Hein. Sparse and imperceptible adversarial attacks. In *ICCV*, 2019.
- [27] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.
- [28] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.
- [29] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *ECCV Workshop on Adversarial Robustness in the Real World*, 2020.
- [30] J. Cui, S. Liu, L. Wang, and J. Jia. Learnable boundary guided adversarial training. *ICCV*, 2021.
- [31] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. Cinc-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [33] G. W. Ding, L. Wang, and X. Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.
- [34] G. W. Ding, Y. Sharma, K. Y. C. Lui, and R. Huang. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.
- [35] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In *CVPR*, 2020.
- [36] L. Engstrom, A. Ilyas, and A. Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *NeurIPS 2018 Workshop on Security in Machine Learning*, 2018.
- [37] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras. Robustness (python library), 2019. URL <https://github.com/MadryLab/robustness>.
- [38] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.
- [39] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. Exploring the landscape of spatial robustness. In *ICML*, 2019.
- [40] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song. Robust physical-world attacks on machine learning models. In *CVPR*, 2018.
- [41] F. Faghri, C. Vasconcelos, D. J. Fleet, F. Pedregosa, and N. L. Roux. Bridging the gap between adversarial robustness and optimization bias. *arXiv preprint arXiv:2102.08868*, 2021.
- [42] A. Fawzi and P. Frossard. Manitest: Are classifiers really invariant? In *BMVC*, 2015.
- [43] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019.
- [44] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ICLR*, 2019.
- [45] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.
- [46] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [47] D. Goodman, H. Xin, W. Yang, W. Yuesheng, X. Junfeng, and Z. Huan. Advbox: a toolbox to generate adversarial examples that fool neural networks. *arXiv preprint arXiv:2001.05574*, 2020.
- [48] S. Gowal, K. D. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli. Scalable verified training for provably robust image classification. In *ICCV*, 2019.

- [49] S. Goyal, J. Uesato, C. Qin, P.-S. Huang, T. Mann, and P. Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019.
- [50] S. Goyal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv*, 2020.
- [51] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial examples for malware detection. In *European symposium on research in computer security*, 2017.
- [52] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- [53] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.
- [54] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger. Simple black-box adversarial attacks. In *ICML*, 2019.
- [55] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [56] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017.
- [57] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 2019.
- [58] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.
- [59] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.
- [60] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.
- [61] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *ICLR*, 2020.
- [62] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.
- [63] J. E. Hu, A. Swaminathan, H. Salman, and G. Yang. Improved image wasserstein attacks and defenses. *ICLR Workshop: Towards Trustworthy ML: Rethinking Security and Privacy for ML*, 2020.
- [64] L. Huang, C. Zhang, and H. Zhang. Self-adaptive training: beyond empirical risk minimization. *NeurIPS*, 2020.
- [65] Y. Jang, T. Zhao, S. Hong, and H. Lee. Adversarial defense via learning to generate diverse attacks. *ICCV*, 2019.
- [66] C. Jin and M. Rinard. Manifold regularization for adversarial robustness. *arXiv*, 2020.
- [67] A. Ju. Generative models as a robust alternative for image classification: Progress and challenges. *PhD thesis, UC Berkeley*, 2021.
- [68] D. Kang, Y. Sun, T. Brown, D. Hendrycks, and J. Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.
- [69] D. Kang, Y. Sun, D. Hendrycks, T. Brown, and J. Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.
- [70] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: an efficient smt solver for verifying deep neural networks. In *ICCAV*, 2017.
- [71] S. Kaur, J. Cohen, and Z. C. Lipton. Are perceptually-aligned gradients a general property of robust classifiers? In *NeurIPS Workshop: Science Meets Engineering of Deep Learning*, 2019.
- [72] J. Kim and X. Wang. Sensible adversarial learning. *OpenReview*, 2019.
- [73] K. Kireev, M. Andriushchenko, and N. Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv*, 2021.
- [74] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint arXiv:2012.07421*, 2020.
- [75] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

- [76] B. Kulynych, R. Overdorf, C. Troncoso, and S. Gürses. Pots: protective optimization technologies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 177–188, 2020.
- [77] S. Kundu, M. Nazemi, P. A. Beerel, and M. Pedram. A tunable robust pruning framework through dynamic network rewiring of dnns. *ASP-DAC*, 2021.
- [78] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ICLR Workshop*, 2017. URL <https://openreview.net/forum?id=HJGU3Rod1>.
- [79] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, et al. Adversarial attacks and defences competition. In *NeurIPS Competition Track*, 2018.
- [80] C. Laidlaw and S. Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.
- [81] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.
- [82] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE S&P*, 2019.
- [83] L. Li, M. Weber, X. Xu, L. Rimanic, B. Kailkhura, T. Xie, C. Zhang, and B. Li. Tss: Transformation-specific smoothing for robustness certification. In *ACM CCS*, 2021.
- [84] Y. Li, J. Bradshaw, and Y. Sharma. Are generative classifiers more robust to adversarial attacks? In *ICML*, 2019.
- [85] Y. Li, W. Jin, H. Xu, and J. Tang. Deeprobust: A pytorch library for adversarial attacks and defenses. *arXiv preprint arXiv:2005.06149*, 2020.
- [86] X. Ling, S. Ji, J. Zou, J. Wang, C. Wu, B. Li, and T. Wang. Deepsec: A uniform platform for security analysis of deep learning model. In *IEEE S&P*, 2019.
- [87] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.
- [88] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [89] T. Maho, B. Bonnet, T. Furon, and E. L. Merrer. Robic: A benchmark suite for assessing classifiers robustness. *arXiv preprint arXiv:2102.05368*, 2021.
- [90] P. Maini, E. Wong, and J. Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *ICML*, 2020.
- [91] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray. Metric learning for adversarial robustness. *NeurIPS*, 2019.
- [92] M. Melis, A. Demontis, M. Pintor, A. Sotgiu, and B. Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019.
- [93] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard. Sparsefool: a few pixels make a big difference. In *CVPR*, 2019.
- [94] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard. Robustness via curvature regularization, and vice versa. *CVPR*, 2019.
- [95] M. Mosbach, M. Andriushchenko, T. Trost, M. Hein, and D. Klakow. Logit pairing methods can fool gradient-based attacks. In *NeurIPS 2018 Workshop on Security in Machine Learning*, 2018.
- [96] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao. Adversarial defense by restricting the hidden space of deep neural networks. *ICCV*, 2019.
- [97] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. *Technical Report*, 2011.
- [98] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards. Adversarial robustness toolbox v1.2.0. *arXiv preprint arXiv:1807.01069*, 2018.
- [99] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *ICLR*, 2020.
- [100] T. Pang, K. Xu, and J. Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020.
- [101] T. Pang, X. Yang, Y. Dong, K. Xu, H. Su, and J. Zhu. Boosting adversarial training with hypersphere embedding. *NeurIPS*, 2020.
- [102] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. *ICLR*, 2021.

- [103] T. Pang, H. Zhang, D. He, Y. Dong, H. Su, W. Chen, J. Zhu, and T.-Y. Liu. Adversarial training with rectified rejection. *arXiv preprint arXiv:2105.14785*, 2021.
- [104] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.
- [105] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *Technical Report*, 2017.
- [106] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli. Adversarial robustness through local linearization. *NeurIPS*, 2019.
- [107] R. Rade and S.-M. Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. *OpenReview*, 2021.
- [108] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020.
- [109] M. S. Rahman, M. Imani, N. Mathews, and M. Wright. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. *IEEE Transactions on Information Forensics and Security*, 16:1594–1609, 2020.
- [110] J. Rauber, W. Brendel, and M. Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *ICML Reliable Machine Learning in the Wild Workshop*, 2017.
- [111] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.
- [112] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.
- [113] L. Rice, E. Wong, and J. Z. Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.
- [114] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. *CVPR*, 2019.
- [115] J. Rony, E. Granger, M. Pedersoli, and I. Ben Ayed. Augmented lagrangian adversarial attacks. *arXiv preprint arXiv:2011.11857*, 2020.
- [116] P. Saadatpanah, A. Shafahi, and T. Goldstein. Adversarial attacks on copyright detection systems. In *ICML*, 2020.
- [117] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? *NeurIPS*, 2020.
- [118] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *ICLR*, 2018.
- [119] L. Schwinn, R. Raab, A. Nguyen, D. Zanca, and B. Eskofier. Exploring robust misclassifications of neural networks to enhance adversarial attacks. *arXiv preprint arXiv:2105.10304*, 2021.
- [120] V. Schwag, A. N. Bhagoji, L. Song, C. Sitawarin, D. Cullina, M. Chiang, and P. Mittal. Analyzing the robustness of open-world machine learning. In *12th ACM Workshop on Artificial Intelligence and Security*, 2019.
- [121] V. Schwag, S. Wang, P. Mittal, and S. Jana. Hydra: Pruning adversarially robust neural networks. *NeurIPS*, 2020.
- [122] V. Schwag, S. Wang, P. Mittal, and S. Jana. On pruning adversarially robust neural networks. *NeurIPS*, 2020.
- [123] V. Schwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal. Improving adversarial robustness using proxy distributions. *arXiv*, 2021.
- [124] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. Adversarial training for free! *NeurIPS*, 2019.
- [125] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1589–1604, 2020.
- [126] C. Shi, C. Holtz, and G. Mishne. Online adversarial purification based on self-supervised learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=_i3ASp12WS.
- [127] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

- [128] M. Singh, A. Sinha, N. Kumari, H. Machiraju, B. Krishnamurthy, and V. N. Balasubramanian. Harnessing the vulnerability of latent layers in adversarially trained models. *IJCAI*, 2019.
- [129] C. Sitawarin, S. Chakraborty, and D. Wagner. Improving adversarial robustness through progressive hardening. *arXiv*, 2020.
- [130] L. Song and P. Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.
- [131] L. Song, R. Shokri, and P. Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257, 2019.
- [132] L. Song, V. Sehwal, A. N. Bhagoji, and P. Mittal. A critical evaluation of open-world machine learning. *arXiv preprint arXiv:2007.04391*, 2020.
- [133] K. Sridhar, O. Sokolsky, I. Lee, and J. Weimer. Robust learning via persistency of excitation. *arXiv*, 2021.
- [134] D. Stutz, M. Hein, and B. Schiele. Relating adversarially robust generalization to flat minima. In *ICCV*, 2021.
- [135] C. Szegedy, W. Zaremba, I. Sutskever, D. E. Joan Bruna, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2013.
- [136] L. Tao, L. Feng, J. Yi, S.-J. Huang, and S. Chen. Provable defense against delusive poisoning. *arXiv preprint arXiv:2102.04716*, 2021.
- [137] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.
- [138] V. Tjeng, K. Xiao, and R. Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *ICLR*, 2019.
- [139] G. Tolias, F. Radenovic, and O. Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *ICCV*, 2019.
- [140] F. Tramèr and D. Boneh. Adversarial training and robustness for multiple perturbations. In *NeurIPS*, 2019.
- [141] F. Tramèr, P. Dupré, G. Rusak, G. Pellegrino, and D. Boneh. Adversarial: Perceptual ad blocking meets adversarial machine learning. In *ACM SIGSAC CCS*, 2019.
- [142] F. Tramèr, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.
- [143] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.
- [144] J. Uesato, J.-B. Alayrac, P.-S. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? *NeurIPS*, 2019.
- [145] F. Utrera, E. Kravitz, N. B. Erichson, R. Khanna, and M. W. Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020.
- [146] D. Wang, A. Ju, E. Shelhamer, D. Wagner, and T. Darrell. Fighting gradients with gradients: Dynamic defenses against adversarial attacks. *arXiv preprint arXiv:2105.08714*, 2021.
- [147] J. Wang and H. Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. *ICCV*, 2019.
- [148] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. *ICLR*, 2020.
- [149] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel. Towards fast computation of certified robustness for relu networks. In *ICML*, 2018.
- [150] E. Wong and J. Z. Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.
- [151] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ICML*, 2018.
- [152] E. Wong, F. R. Schmidt, and J. Z. Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *ICML*, 2019.
- [153] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *ICLR*, 2020.
- [154] B. Wu, J. Chen, D. Cai, X. He, and Q. Gu. Do wider neural networks really help adversarial robustness? *arXiv*, 2020.
- [155] D. Wu, S. tao Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. *NeurIPS*, 2020.

- [156] C. Xiao, P. Zhong, and C. Zheng. Enhancing adversarial defense by k-winners-take-all. *ICLR*, 2020.
- [157] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille. Adversarial examples for semantic segmentation and object detection. In *ICCV*, 2017.
- [158] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. In *CVPR*, 2020.
- [159] C. Xu, X. Li, and M. Yang. An orthogonal classifier for improving the adversarial robustness of neural networks. *arXiv preprint arXiv:2105.09109*, 2021.
- [160] H. Xu, X. Liu, Y. Li, and J. Tang. To be robust or to be fair: Towards fairness in adversarial training. *arXiv preprint arXiv:2010.06121*, 2020.
- [161] Y. Yang, G. Zhang, D. Katabi, and Z. Xu. Me-net: Towards effective adversarial robustness with matrix estimation. In *ICML*, 2019.
- [162] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. Salakhutdinov, and K. Chaudhuri. A closer look at accuracy vs. robustness. *Advances in Neural Information Processing Systems*, 33, 2020.
- [163] Y. Yu, Z. Yang, E. Dobriban, J. Steinhardt, and Y. Ma. Understanding generalization in adversarial training via the bias-variance decomposition. *arXiv preprint arXiv:2103.09947*, 2021.
- [164] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016.
- [165] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Accelerating adversarial training via maximal principle. *NeurIPS*, 2019.
- [166] H. Zhang and J. Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *NeurIPS*, 2019.
- [167] H. Zhang and W. Xu. Adversarial interpolation training: A simple approach for improving model robustness. *OpenReview*, 2019.
- [168] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.
- [169] H. Zhang, H. Chen, C. Xiao, S. Gowal, R. Stanforth, B. Li, D. Boning, and C.-J. Hsieh. Towards stable and efficient training of verifiably robust neural networks. *ICLR*, 2020.
- [170] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. *ICML*, 2020.
- [171] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. *ICLR*, 2021.
- [172] X. Zhang and D. Evans. Incorporating label uncertainty in understanding adversarial robustness. *arXiv preprint arXiv:2107.03250*, 2021.
- [173] J. Zhong, X. Liu, and C.-J. Hsieh. Improving the speed and quality of gan by adversarial training. *arXiv preprint arXiv:2008.03364*, 2020.
- [174] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu. FreeLb: Enhanced adversarial training for natural language understanding. In *ICLR*, 2019.