RobustBench: a standardized adversarial robustness benchmark

Francesco Croce* Univ. of Tübingen
Maksym Andriushchenko* EPFL
Vikash Sehwag* 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 overestimation of the true robustness of models. While adaptive attacks designed for a particular defense are a potential solution, they have to be highly customized for particular models, which makes it difficult to compare different methods. Our goal is to instead 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 evaluate robustness of models for our benchmark, we consider AutoAttack, 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. We also impose some restrictions on the admitted models to rule out defenses that only make gradient-based attacks ineffective without improving actual robustness. Our leaderboard, hosted at `http://robustbench.github.io/`, contains evaluations of 90+ models and aims at reflecting the current state of the art 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 `http://github.com/RobustBench/robustbench` that provides unified access to 60+ 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* [123], 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 [64] and exact approaches [126] do not scale to large enough models. There are already more than 3000 papers on this topic [14], 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 [4, 16, 129] which have in turn shown that several seemingly

*Equal contribution.

Submitted to the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. Do not distribute.

Rank	Method	Standard accuracy	Robust accuracy	Extra data	Architecture	Venue
1	Fixing Data Augmentation to Improve Adversarial Robustness	92.23%	66.56%	☑	WideResNet-70-16	arXiv, Mar 2021
2	Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples	91.10%	65.87%	☑	WideResNet-70-16	arXiv, Oct 2020
3	Fixing Data Augmentation to Improve Adversarial Robustness	88.50%	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 $\varepsilon_\infty = 8/255$.

robust defenses fail to be robust. However, recently Tramèr et al. [129] 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 [12, 48], (2) randomized classifiers [147, 91], and (3) classifiers that contain an optimization loop in their predictions [108, 76]. Often, non-certified defenses that violate these three principles only make gradient-based attacks harder but do not substantially improve adversarial robustness [16]. 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 [26] as our current standard evaluation which is an ensemble of diverse parameter-free attacks (white- and black-box) that has shown for various datasets reliable performance over a large set of models that satisfy our restrictions. Moreover, we accept evaluations based on adaptive attacks whenever they can improve our standard evaluation. Additionally, we collect models robust against common image corruptions [53] as these represent another type of perturbations which should not modify the decision of a classifier although they are not produced in an adversarial way.

Contributions. We make the following contributions with our `RobustBench` benchmark:

- **Leaderboard** https://robustbench.github.io/: a website with the leaderboard (see Fig. 1) based on *more than 90* models 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 (potentially complemented by adaptive attacks). 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. For 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 60* 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.

We believe that our standardized benchmark and accompanied collection of models will accelerate progress on multiple fronts in the area of adversarial robustness.

2 Background and related work

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

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

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 [64, 136]. 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 [34, 86], especially in the presence of phenomena like gradient obfuscation [4]. Certified methods [138, 44] instead provide *lower bounds* on robust accuracy but often underestimate robustness significantly, in particular if the certification was not part of the training process. Thus, we do not consider lower bounds in our benchmark, and focus only on upper bounds which are typically much tighter [126].

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. $\Delta_p = \{\boldsymbol{\delta} \in \mathbb{R}^d, \|\boldsymbol{\delta}\|_p \leq \varepsilon\}$, particularly for $p = \infty$ [123, 42, 79]. 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 [41]. It is an active area of research to develop threat models which are more aligned with the human perception such as spatial perturbations [39, 37], Wasserstein-bounded perturbations [139, 57], perturbations of the image colors [72] or ℓ_p-perturbations in the latent space of a neural network [73, 137]. However, despite the simplicity of the ℓ_p-perturbation model, it has numerous interesting applications that go beyond security considerations [128, 106] and span transfer learning [107, 132], interpretability [130, 65, 36], generalization [144, 158, 8], robustness to unseen perturbations [62, 144, 73, 67], stabilization of GAN training [157]. Thus, improvements in ℓ_p-robustness have the potential to improve many of these downstream applications.

Common corruptions. Unlike adversarial perturbations, common corruptions [53] try to mimic modifications of the input images which can occur naturally: 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 types and severities of corruptions.

Related libraries and benchmarks. There are many libraries that focus primarily on implementations of popular adversarial attacks such as FoolBox [100], Cleverhans [95], AdverTorch [31], AdvBox [43], ART [89], SecML [83]. 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 [71, 9] 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. [77] proposed DEEPSEC, a benchmark that tests many combinations of attacks and defenses, but suffers from a few shortcomings as suggested by Carlini [15], in particular: (1) reporting average-case performance over multiple attacks instead of worst-case performance, (2) evaluating robustness in threat models different from the one used for training, (3) using excessively large perturbations.

Recently, Dong et al. [33] 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 [55, 18, 46, 101] and attacks [45, 25], 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. [79] and Zhang et al. [154], 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 [26] which includes both white- and black-box attacks that can be *optionally* extended by adaptive evaluations, (2) clearly defined threat models that correspond to the ones used during training for submitted defenses, (3) evaluation of not only standard defenses [79] but also of more recent improvements such as [18, 46, 101], (4) the Model Zoo that provides convenient access to the 60+ most robust models from the literature which can be used for downstream tasks and facilitate the development of new standardized attacks. 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 of adaptive attacks for particular models.

3 Description of `RobustBench`

In this section, we start by providing a detailed layout of our proposed leaderboard for ℓ_∞, ℓ_2, and the common corruptions 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 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 actually improving robustness. Specifically, we consider only classifiers $f : \mathbb{R}^d \to \mathbb{R}^C$ that

- have in general *non-zero gradients* with respect to the inputs. Models with zero gradients, e.g., that rely on quantization of inputs [12, 48], 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 [4], 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 [4]. While often effective, it might be not sufficient, as shown by Tramèr et al. [129]. 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 also note that randomization *can* be useful for improving robustness and deriving robustness certificates [74, 23], but it also introduces variance in the gradient estimators (both white- and black-box) which can make attacks much less effective.
- do not have an *optimization loop* in the forward pass. This makes backpropagation through the classifier very difficult or extremely expensive. Usually, such defenses [108, 76] need to be evaluated adaptively with attacks considering jointly the loss of the inner loop and the standard classification task.

Some of these restrictions were also discussed by [11] 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.

Overall setup. We set up leaderboards for the ℓ_∞, ℓ_2 and common corruption threat models on CIFAR-10 and CIFAR-100 [69] 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. Most of the models shown there 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 (usually, the dataset introduced by Carmon et al. [18]) is used since it gives a clear advantage for both

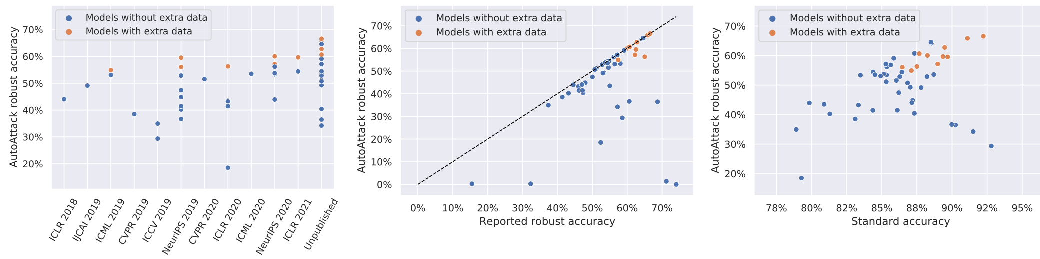

Figure 2: Visualization of the robustness and accuracy of 54 CIFAR-10 models from the `RobustBench` ℓ_∞-leaderboard. Robustness is evaluated using ℓ_∞-perturbations with $\varepsilon_\infty = 8/255$.

clean and robust accuracy. 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 recent conference.

Evaluation of defenses. The evaluation of robust accuracy on common corruptions [53] 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 [26]. It is an ensemble of four attacks: 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 [25], which minimizes the ℓ_p-norm of the perturbations, and (4) the black-box Square Attack [3]. 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. 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 C 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. We also accept evaluations of the individual models on the leaderboard based on adaptive or external attacks to reflect the best available upper bound on the true robust accuracy. For example, Gowal et al. [46] and Rebuffi et al. [101] evaluate their models with a hybrid of AutoAttack and MultiTargeted attack [45], that in some cases report slightly lower robust accuracy than AutoAttack alone. We reflect all such additional evaluations in our leaderboard. The submission of adaptive evaluations is facilitated by a pre-formatted issue template in our repository `https://github.com/RobustBench/robustbench`.

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 can 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.

[1]A 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`

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

	CIFAR-10		**CIFAR-100**	
Threat model	Model Zoo	Leaderboard	Model Zoo	Leaderboard
ℓ_∞ with $\varepsilon_\infty = {}^{8}/_{255}$	33	55	12	12
ℓ_2 with $\varepsilon_2 = 0.5$	14	14	-	-
Common corruptions [53]	7	12	2	4

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 A 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 [96]. 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='Carmon2019Unlabeled',
                   dataset='cifar10', threat_model='Linf')
```

At the moment, all models (see Table 1 and Appendix E for details) are variations of ResNet [50] and WideResNet architectures [150] of different depth and width. We include the most robust models, e.g. those from Rebuffi et al. [101], but there are also defenses which pursue additional goals alongside adversarial robustness at the fixed threshold we use: e.g., Sehwag et al. [112] consider networks which are robust and compact, Wong et al. [140] focus on computationally efficient adversarial training, Ding et al. [32] 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.

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. [79] and Zhang et al. [154], 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. [79] 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. In the following we analyze various aspects of robust classifiers, reporting results mostly for ℓ_∞-robust models on CIFAR-10 while the results for other threat models and datasets can be found in Appendix D.

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 [98]. 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 [79, 154] but with various enhancements (extra data, early stopping, weight averaging).

Performance across various distribution shifts. Here we test the performance of the models from the Model Zoo on different distribution shifts ranging from common image corruptions (CIFAR-10-C, [53]) to dataset resampling bias (CIFAR-10.1, [102]) and image source shift (CINIC-10, [29]). For each of these datasets, we measure standard accuracy, and Fig. 3 shows that improvement in the robust accuracy (which often comes with an improvement in standard accuracy) on CIFAR-10 also correlates with an improvement in standard accuracy across distributional shifts. On CIFAR-10-C, we observe that robust models (particularly with respect to the ℓ_2-norm) tend to give a significant improvement which agrees with the findings from the previous literature [40]. Concurrently with our work, Taori et al. [125] also 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 [26] (see Fig. 10 in Appendix D), and notice how ℓ_p adversarial robustness generalizes across different datasets, and a clear positive correlation between robust accuracy on CIFAR-10 and its variations.

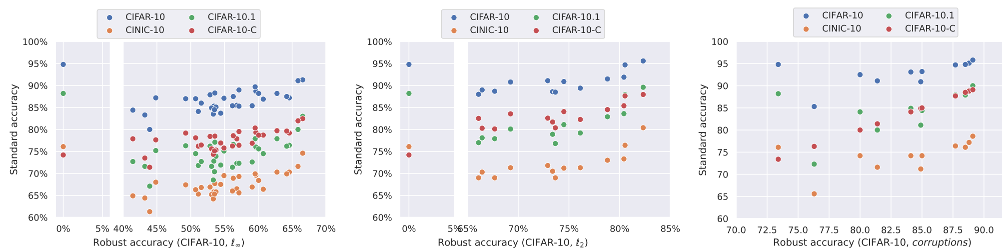

Figure 3: Standard accuracy of classifiers trained against ℓ_∞ (**left**), ℓ_2 (**middle**), and common corruption (**right**) threat model respectively, from our Model Zoo on various distribution shifts.

Calibration. A classifier is *calibrated* if its predicted probabilities correctly reflect the actual accuracy [47]. In the context of adversarial training, calibration was considered in Hendrycks et al. [56] who focus on improving accuracy on common corruptions and in Augustin et al. [6] who focus mostly on 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 [49] to minimize the ECE (which is a simple but effective post-hoc calibration method, see Appendix D 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. [32] 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. [92] (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 D).

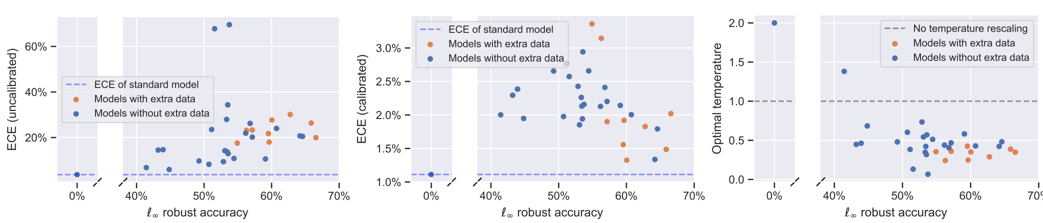

Figure 4: Expected calibration error (ECE) before (**left**) and after (**middle**) temperature rescaling, and the optimal rescaling temperature (**right**) for the ℓ_∞-robust models.

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 [54]. An emerging line of research aims at developing OOD detection methods in conjunction with adversarial robustness [52, 110, 6]. In particular, Song et al. [122] demonstrated that adversarial training [79] 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 [69], SVHN [88], and Describable Textures Dataset [22]. 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. [6] achieves even better performance since their approach explicitly optimizes both robust accuracy and worst-case OOD detection performance.

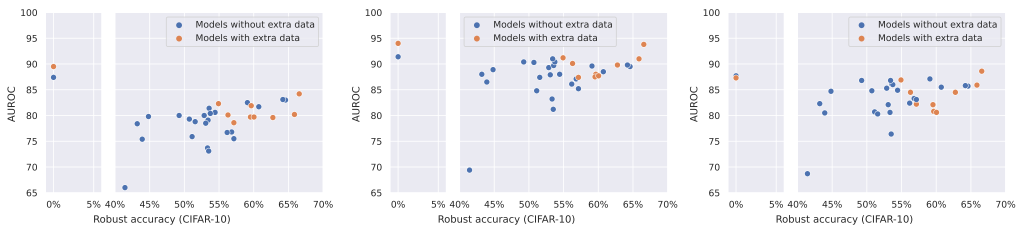

Figure 5: Visualization of the OOD detection quality (higher AUROC is better) for the ℓ_∞-robust models trained on CIFAR-10 on three OOD datasets: CIFAR-100 (**left**), SVHN (**middle**), Describable Textures (**right**). We detect OOD inputs based on the maximum predicted confidence [54].

Fairness in robustness. Recent works [7, 146] have noticed that robust training [79, 154] 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 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 D), 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. However, some training techniques like MART [135] can noticeably increase the RSD and thus *increase the disparity* compared to other methods which achieve similar robustness (around 57%).

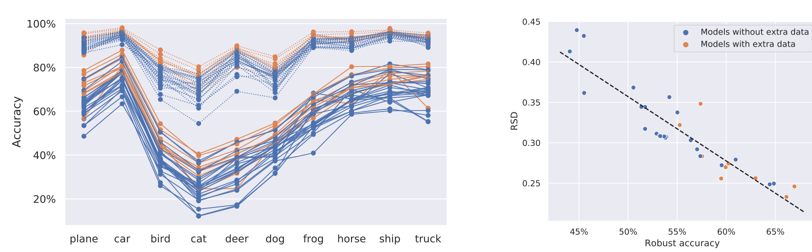

Figure 6: Fairness of ℓ_∞-robust models. **Left:** classwise standard (dotted lines) and robust (solid) accuracy. **Right:** relative standard deviation (RSD) of robust accuracy over classes vs its average.

Privacy leakage. Deep neural networks are prone to memorizing training data [117, 17]. Recent works have highlighted that robust training exacerbates this problem [121]. Here 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 D 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. It also does not have a strong correlation with the robust accuracy. In contrast, it is largely determined by the generalization gap, as using classification confidence information 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 multiple previous works have explicitly aimed to reduce the generalization gap in robust training using techniques such as early stopping [103, 154, 46]. It further suggests that reducing the generalization gap in robust networks can further reduce privacy leakage.

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**).

Extra experiments. In Appendix D, 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. Moreover, since prior works [51, 148] connected higher smoothness with better robustness, we analyze the smoothness of the 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 illustrates that one can use the collected models to study the *internal* properties of robust networks as well.

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 can be useful to show which ideas in training robust models are the most successful. Recent works have already referred to our leaderboards [68, 149, 80, 124, 145], in particular as reflecting the current state of the art [101, 75, 94], and used the networks of our Model Zoo to test new adversarial attacks [83, 105, 38, 109], to evaluate test-time defenses [133] or to evaluate perceptual distances derived from them [61]. 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.

Broader impact. In our work, we do not only perform a standardized benchmarking of adversarial robustness but also analyze multiple other properties of robust models such as calibration, privacy leakage, fairness, etc. Such analyses are important, in our opinion, since they allow us to assess the broader impact of improving robustness on other crucial performance metrics of neural networks. Additionally, in motivating higher robustness against adversarial examples, our work leaves an unwanted side effect on tasks where adversarial attacks can actually be used for beneficial purposes [115, 99]. 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 necessarily a realistic threat model (although it is a insightful problem to work on) and the real-world robustness is likely to require more domain-specific threat models.

Future plans. Our intention in the future is to keep the current leaderboards up-to-date (see the maintenance plan in Appendix B) and add new leaderboards for other datasets (in particular, for ImageNet [30]) 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 [10, 24, 84, 27], (2) multiple ℓ_p-norm perturbations [127, 81], (3) adversarially optimized common corruptions [62, 63], (4) a broad set of perturbations unseen during training [73]. Another possible direction of development of the benchmark is including defenses based on some form of test-time adaptation [116, 133], which do not fulfill the third restriction (no optimization loop). However, since those are showing promising results and drawing attention from the community, one can introduce a separate leaderboard with specific rules and evaluation protocol for them.

References

[1] M. Alfarra, J. C. Perez, A. Bibi, A. Thabet, P. Arbelaez, and B. Ghanem. Clustr: Clustering training for robustness. *arXiv*, 2020.

[2] M. Andriushchenko and N. Flammarion. Understanding and improving fast adversarial training. *NeurIPS*, 2020.

[3] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

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

[5] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman. Controlling neural level sets. *NeurIPS*, 2019.

[6] M. Augustin, A. Meinke, and M. Hein. Adversarial robustness on in- and out-distribution improves explainability. *ECCV*, 2020.

[7] 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.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[9] 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.

[10] 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.

[11] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow. Unrestricted adversarial examples. *arXiv preprint arXiv:1809.08352*, 2018.

[12] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

[13] D. A. Calian, F. Stimberg, O. Wiles, S.-A. Rebuffi, A. Gyorgy, T. Mann, and S. Gowal. Defending against image corruptions through adversarial augmentations. *arXiv*, 2021.

[14] N. Carlini. A complete list of all (arxiv) adversarial example papers. `https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html`. Accessed: 2021-06-08.

[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. 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.

[17] 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.

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

[19] A. Chan, Y. Tay, Y. S. Ong, and J. Fu. Jacobian adversarially regularized networks for robustness. *ICLR*, 2020.

[20] J. Chen, Y. Cheng, Z. Gan, Q. Gu, and J. Liu. Efficient robust training via backward smoothing. *arXiv*, 2020.

[21] 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.

[22] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[23] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

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

[25] F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020.

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

[27] 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.

[28] J. Cui, S. Liu, L. Wang, and J. Jia. Learnable boundary guided adversarial training. *arXiv*, 2020.

[29] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[30] 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.

[31] G. W. Ding, L. Wang, and X. Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

[32] 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.

[33] 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.

[34] 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.

[35] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

[36] 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.

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

[38] 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.

[39] A. Fawzi and P. Frossard. Manitest: Are classifiers really invariant? In *BMVC*, 2015.

[40] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019.

[41] 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.

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

[43] 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.

[44] 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.

[45] S. Gowal, 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.

[46] S. Gowal, C. Qin, J. Uesato, T. Mann, and P. Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv*, 2020.

[47] 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.

[48] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018.

[49] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger. Simple black-box adversarial attacks. In *ICML*, 2019.

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

[51] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017.

[52] 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.

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

[54] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

[55] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *ICML*, 2019.

[56] 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.

[57] 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.

[58] L. Huang, C. Zhang, and H. Zhang. Self-adaptive training: beyond empirical risk minimization. *NeurIPS*, 2020.

[59] Y. Jang, T. Zhao, S. Hong, and H. Lee. Adversarial defense via learning to generate diverse attacks. *ICCV*, 2019.

[60] C. Jin and M. Rinard. Manifold regularization for adversarial robustness. *arXiv*, 2020.

[61] A. Ju. Generative models as a robust alternative for image classification: Progress and challenges. *PhD thesis, UC Berkeley*, 2021.

[62] D. Kang, Y. Sun, T. Brown, D. Hendrycks, and J. Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.

[63] D. Kang, Y. Sun, D. Hendrycks, T. Brown, and J. Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

[64] 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.

[65] 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.

[66] J. Kim and X. Wang. Sensible adversarial learning. *OpenReview*, 2019.

[67] K. Kireev, M. Andriushchenko, and N. Flammarion. On the effectiveness of adversarial training against common corruptions. *arXiv*, 2021.

[68] 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.

[69] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[70] S. Kundu, M. Nazemi, P. A. Beerel, and M. Pedram. A tunable robust pruning framework through dynamic network rewiring of dnns. *ASP-DAC*, 2021.

[71] 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.

[72] C. Laidlaw and S. Feizi. Functional adversarial attacks. In *NeurIPS*, 2019.

[73] C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

[74] 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.

[75] 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.

[76] Y. Li, J. Bradshaw, and Y. Sharma. Are generative classifiers more robust to adversarial attacks? In *ICML*, 2019.

[77] 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.

[78] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

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

[80] T. Maho, B. Bonnet, T. Furon, and E. L. Merrer. Robic: A benchmark suite for assessing classifiers robustness. *arXiv preprint arXiv:2102.05368*, 2021.

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

[82] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray. Metric learning for adversarial robustness. *NeurIPS*, 2019.

[83] 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.

[84] A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard. Sparsefool: a few pixels make a big difference. In *CVPR*, 2019.

[85] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard. Robustness via curvature regularization, and vice versa. *CVPR*, 2019.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *ICLR*, 2020.

[91] T. Pang, K. Xu, and J. Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020.

[92] T. Pang, X. Yang, Y. Dong, K. Xu, H. Su, and J. Zhu. Boosting adversarial training with hypersphere embedding. *NeurIPS*, 2020.

[93] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu. Bag of tricks for adversarial training. *ICLR*, 2021.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *ICML*, 2020.

[99] 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.

[100] 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.

[101] 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. URL https://arxiv.org/pdf/2103.01946.

[102] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

[103] L. Rice, E. Wong, and J. Z. Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

[104] 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.

[105] J. Rony, E. Granger, M. Pedersoli, and I. Ben Ayed. Augmented lagrangian adversarial attacks. *arXiv preprint arXiv:2011.11857*, 2020.

[106] P. Saadatpanah, A. Shafahi, and T. Goldstein. Adversarial attacks on copyright detection systems. In *ICML*, 2020.

[107] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? *NeurIPS*, 2020.

[108] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *ICLR*, 2018.

[109] 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.

[110] V. Sehwag, 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.

[111] V. Sehwag, S. Wang, P. Mittal, and S. Jana. Hydra: Pruning adversarially robust neural networks. *NeurIPS*, 2020.

[112] V. Sehwag, S. Wang, P. Mittal, and S. Jana. On pruning adversarially robust neural networks. *NeurIPS*, 2020.

[113] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal. Improving adversarial robustness using proxy distributions. *arXiv*, 2021.

[114] 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.

[115] 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.

[116] 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=_i3ASPp12WS.

[117] 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.

[118] 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.

[119] C. Sitawarin, S. Chakraborty, and D. Wagner. Improving adversarial robustness through progressive hardening. *arXiv*, 2020.

[120] L. Song and P. Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[121] 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.

[122] L. Song, V. Sehwag, A. N. Bhagoji, and P. Mittal. A critical evaluation of open-world machine learning. *arXiv preprint arXiv:2007.04391*, 2020.

[123] C. Szegedy, W. Zaremba, I. Sutskever, D. E. Joan Bruna, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2013.

[124] L. Tao, L. Feng, J. Yi, S.-J. Huang, and S. Chen. Provable defense against delusive poisoning. *arXiv preprint arXiv:2102.04716*, 2021.

[125] 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.

[126] V. Tjeng, K. Xiao, and R. Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *ICLR*, 2019.

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

[128] F. Tramèr, P. Dupré, G. Rusak, G. Pellegrino, and D. Boneh. Adversarial: Perceptual ad blocking meets adversarial machine learning. In *ACM SIGSAC CCS*, 2019.

[129] F. Tramèr, N. Carlini, W. Brendel, and A. Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020.

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

[131] J. Uesato, J.-B. Alayrac, P.-S. Huang, R. Stanforth, A. Fawzi, and P. Kohli. Are labels required for improving adversarial robustness? *NeurIPS*, 2019.

[132] 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.

[133] 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.

[134] J. Wang and H. Zhang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. *ICCV*, 2019.

[135] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu. Improving adversarial robustness requires revisiting misclassified examples. *ICLR*, 2020.

[136] 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.

[137] E. Wong and J. Z. Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.

[138] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ICML*, 2018.

[139] E. Wong, F. R. Schmidt, and J. Z. Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *ICML*, 2019.

[140] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *ICLR*, 2020.

[141] B. Wu, J. Chen, D. Cai, X. He, and Q. Gu. Do wider neural networks really help adversarial robustness? *arXiv*, 2020.

[142] D. Wu, S. tao Xia, and Y. Wang. Adversarial weight perturbation helps robust generalization. *NeurIPS*, 2020.

[143] C. Xiao, P. Zhong, and C. Zheng. Enhancing adversarial defense by k-winners-take-all. *ICLR*, 2020.

[144] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. Adversarial examples improve image recognition. In *CVPR*, 2020.

[145] C. Xu, X. Li, and M. Yang. An orthogonal classifier for improving the adversarial robustness of neural networks. *arXiv preprint arXiv:2105.09109*, 2021.

[146] 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.

[147] Y. Yang, G. Zhang, D. Katabi, and Z. Xu. Me-net: Towards effective adversarial robustness with matrix estimation. In *ICML*, 2019.

[148] 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.

[149] 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.

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

[151] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong. You only propagate once: Accelerating adversarial training via maximal principle. *NeurIPS*, 2019.

[152] H. Zhang and J. Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *NeurIPS*, 2019.

[153] H. Zhang and W. Xu. Adversarial interpolation training: A simple approach for improving model robustness. *OpenReview*, 2019.

[154] 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.

[155] 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.

[156] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, and M. Kankanhalli. Geometry-aware instance-reweighted adversarial training. *ICLR*, 2021.

[157] J. Zhong, X. Liu, and C.-J. Hsieh. Improving the speed and quality of gan by adversarial training. *arXiv preprint arXiv:2008.03364*, 2020.

[158] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu. Freelb: Enhanced adversarial training for natural language understanding. In *ICLR*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We clearly describe all aspects of the benchmark in Sec. 3, Sec. 4 and in the Appendix.
   (b) Did you describe the limitations of your work? [Yes] We discuss that we only perform a standardized evaluation and do not evaluate models against adaptive attacks, although we accept third-party evaluations based on adaptive attacks as mentioned in Sec. 3.1. Also, our current set of leaderboards can be seen as a limitation, so in Sec. 5 we describe how we plan to expand the benchmark to new threat models and datasets.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Sec. 3.2 and the Model Zoo page.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Sec. C.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We give in Sec. C an example of the computational time and infrastructure used to run AutoAttack on several models. However, we could not provide such details for all models in the benchmark since those where collected over time with different resources.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Sec. E, and in the text we cite the authors of the datasets and algorithms we use.
   (b) Did you mention the license of the assets? [Yes] See Sec. A.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide code and data in the Model Zoo page.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Sec. 3.2 and Sec. A.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]