ROBUSTBench: A STANDARDIZED ADVERSARIAL ROBUSTNESS BENCHMARK

Francesco Croce*  
University of Tübingen

Maksym Andriushchenko*  
EPFL

Vikash Sehwag*  
Princeton University

Edoardo Debenedetti  
EPFL

Nicolas Flammarion  
EPFL

Mung Chiang  
Purdue University

Prateek Mittal  
Princeton University

Matthias Hein  
University of Tübingen

ABSTRACT

Evaluation of adversarial robustness is often error-prone leading to overestimation of the true robustness of models. 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. This requires to impose some restrictions on the admitted models to rule out defenses that only make gradient-based attacks ineffective without improving actual robustness. We evaluate robustness of models for our benchmark with AutoAttack, an ensemble of white- and black-box attacks which was recently shown to improve almost all robustness evaluations compared to the original publications. Our leaderboard aims at reflecting the current state of the art in the $\ell_\infty$- and $\ell_2$-threat models and on common image corruptions, with possible extensions in the future. Additionally, we open-source a library that provides unified access to state-of-the-art robust models to facilitate their downstream applications. Finally, we analyze general trends in $\ell_p$-robustness and its impact on other tasks such as robustness to various distribution shifts and out-of-distribution detection.

1 INTRODUCTION

Since the finding that state-of-the-art deep learning models are vulnerable to small input perturbations called adversarial examples (Szegedy et al., 2013), achieving adversarially robust models has become one of the most studied topics in the machine learning community: more than 3000 papers on this topic have appeared, but it is often unclear which defenses are effective. Adaptive attacks, specifically designed to test a particular defense (Athalye et al., 2018; Carlini et al., 2019; Tramèr et al., 2020), have shown that several seemingly effective defenses fail to be robust, but Tramèr et al. (2020) observe that even some published defenses which have tried to perform adaptive evaluations can still be broken by new adaptive attacks. We observe repeating patterns in defenses that prevent standard attacks from succeeding without improving robustness. This motivates us to impose restrictions on the defenses we consider in our proposed benchmark, RobustBench, which aims at standardized adversarial robustness evaluation. We start from benchmarking robustness with respect to the $\ell_\infty$- and $\ell_2$-threat models since they are the most studied settings in the literature. We use the recent AutoAttack (Croce & Hein, 2020b) as our current standard evaluation method, 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 also accept evaluations based on adaptive attacks whenever they improve our evaluation. Additionally, we collect models robust against common image corruptions (Hendrycks & Dietterich, 2019) 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.

We make the following contributions: 1) Leaderboard (https://robustbench.github.io/), a website with the leaderboards based on more than 30 recent papers where it is possible to
track the progress and the current state of the art in adversarial robustness, 2) Model Zoo (https://github.com/RobustBench/robustbench), a collection of the most robust models that are easy to use for any downstream application and as a testbed for new attacks, 3) Analysis of how robust models perform on other tasks, like common corruptions or detection of out-of-distribution inputs. 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 \( x \in \mathbb{R}^d \) be an input point and \( y \in \{1, \ldots, C\} \) 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 \( \arg \max_{c \in \{1, \ldots, C\}} f(x + \delta_c) \neq y \) and \( \delta \in \Delta \), where typically the perturbation set \( \Delta \) is chosen such that all points in \( x + \delta \) have \( y \) as their true label. Then robust accuracy is the fraction of datapoints on which \( f \) predicts the correct class for all possible perturbations from \( \Delta \). Since computing the exact robust accuracy is in general intractable (Katz et al., 2017), an upper bound on it is obtained via some adversarial attacks. We here focus on the white-box setting, i.e. the model \( f \) is assumed to be fully known to the attacker. We focus on \( \ell_p \)-perturbations, i.e. \( \Delta_p = \{ \delta \in \mathbb{R}^d : \|\delta\|_p \leq \varepsilon \} \), since those are the most well-studied perturbations, particularly for \( p \in \{\infty, 2\} \). Szegedy et al. 2013, Goodfellow et al. 2015, Madry et al. 2018, and use thresholds \( \varepsilon \) established in the literature. Also, despite the simplicity of the \( \ell_p \)-perturbation model, it has numerous interesting applications that go beyond security considerations (see App. [A]).

Common corruptions. Unlike adversarial perturbations, common corruptions (Hendrycks & Dietterich, 2019) 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. Many libraries focus primarily on implementations of popular adversarial attacks: FoolBox (Rauber et al., 2017), Cleverhans (Papernot et al., 2018), Adversarial Perturbations (Ding et al., 2019), Adversarial box (Goodman et al., 2020), ART (Nicolae et al., 2018), SecML (Melis et al., 2019). Some of them also provide implementations of several basic defenses, but they do not include up-to-date state-of-the-art models. Moreover, many benchmarks aiming at tracking progress of adversarial defenses have appeared: the challenges (Kurakin et al., 2018, Brendel et al., 2018) hosted at NeurIPS 2017 and 2018, DEEPSEC (Ling et al., 2019), the work of Dong et al. (2020), RobustML (https://www.robust-ml.org/). However, these prior works present some weaknesses which we aim to overcome (see a detailed discussion in App. [B]).

3 DESCRIPTION OF RobustBench

RobustBench presents a few different features compared to the existing benchmarks: (1) a baseline worst-case evaluation with an ensemble of strong, standardized attacks optionally extended by adaptive ones, (2) clearly defined threat models, (3) inclusion of the most recent improvements such as (Gowal et al., 2020), (4) Model Zoo providing convenient access to models robust with respect to different threat models. Next, we describe the main components of RobustBench.

3.1 LEADERBOARD

Restrictions. We argue that benchmarking adversarial robustness in a standardized way requires some restrictions on the considered models to prevent submissions of defenses that cause some standard attacks to fail without actually improving robustness. Specifically, we consider only classifiers that 1) have in general non-zero gradients with respect to the inputs, 2) have a fully deterministic forward pass, 3) do not have an optimization loop in the forward pass. All these cases would require specific techniques and adaptive evaluations, which can be hardly standardized. Moreover, we are not aware of existing defenses solely based on such techniques which would achieve competitive robustness (see App. [C.1] for further discussion). Some of these restrictions were also discussed by Brown et al. (2018) (App. E therein) for the warm-up phase of their challenge.
Figure 1: Visualization of the robustness and accuracy of 46 CIFAR-10 models from the RobustBench $\ell_\infty$-leaderboard. Robustness is evaluated using $\ell_\infty$-perturbations with $\varepsilon_\infty = 8/255$.

**Initial setup.** We set up leaderboards for the $\ell_\infty$ (46 models), $\ell_2$ (9 models), and common corruption (7 models) threat models. For the first two, we use fixed budgets of $\varepsilon_\infty = 8/255$ and $\varepsilon_2 = 0.5$ on CIFAR-10 [Krizhevsky & Hinton, 2009] respectively. For common corruptions, we evaluate the accuracy on CIFAR-10-C [Hendrycks & Dietterich, 2019]. We highlight the models using data beyond the training set since it gives a clear advantage for both clean and robust accuracy.

**Evaluation of defenses.** Currently, we perform the standardized evaluation of the reported defenses using AutoAttack [Croce & Hein, 2020b], which is an ensemble of four adversarial attacks. We choose it since 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 as shown in Fig. 1 (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 a standard evaluation. For reproducibility, we perform the standardized evaluation independently of the authors of the submitted models.

**Adding new defenses and adaptive evaluations.** The leaderboard is useful if it reflects the latest advances in the field, so it needs to be kept up-to-date. We require new entries to 1) satisfy the three restrictions, 2) be accompanied by a paper (e.g. an arXiv preprint) describing the technique used to achieve the reported results, and 3) make checkpoints available. We also allow temporarily adding entries without providing checkpoints given that the evaluation is done with AutoAttack, but mark such evaluations as unverified and reserve the right to remove them later on if the corresponding checkpoints are not provided. To achieve the most accurate approximation of the true robustness we accept the submission of adaptive evaluations to complement the standardized one.

**Adding new threat models.** Our intention is to add leaderboards for other threat models which are becoming widely accepted in the community, like sparse perturbations, e.g. bounded by $\ell_0$, $\ell_1$-norm or adversarial patches [Brown et al., 2017] [Croce & Hein, 2019] [Modas et al., 2019] [Croce et al., 2020], multiple $\ell_p$-norms perturbations [Tramèr & Boneh, 2019] [Maini et al., 2020], adversarially optimized common corruptions [Kang et al., 2019a,b]. The long term goal, and the direction towards which many recent works are moving, is achieving general robustness [Brown et al., 2018].

3.2 **Model Zoo**

We collect checkpoints of many networks from the leaderboard in a single repository and, with the permission of the authors, make them easily accessible (see examples in App. C.2), as it is often time-consuming and not straightforward to integrate models from different papers in the same framework due to small variations in the architectures, custom input normalizations, etc. Currently, we include only models implemented in PyTorch [Paszke et al., 2017]. 20 models trained for $\ell_\infty$-robustness, 8 for $\ell_2$-robustness, 2 for common corruptions, and a standardly trained one as a baseline.

**A testbed for new attacks.** An important use case of the Model Zoo is to simplify comparisons between different adversarial attacks on a wide range of models. First, the current leaderboard can already serve as a strong baseline for new attacks. Second, new attacks are often evaluated on the publicly available models from [Madry et al., 2018] and [Zhang et al., 2019b], with the risk of overfitting to these classifiers, which can be prevented if many models are used.
4 Analysis

With unified access to the Model Zoo, one can easily compute various performance metrics. In Fig. 1 we plot some statistics for the $\ell_\infty$-robust classifiers: we observe that 1) for multiple published defenses the reported robust accuracy is highly overestimated, 2) using extra data can alleviate the robustness-accuracy trade-off as suggested in previous work (Raghunathan et al., 2020), 3) the most robust classifiers rely on adversarial training (Madry et al., 2018) or TRADES (Zhang et al., 2019b).

**Performance across various distribution shifts.** We test the performance of the $\ell_\infty$-robust classifiers from the Model Zoo on different distribution shifts, ranging from common image corruptions (CIFAR-10-C), dataset resampling bias (CIFAR-10.1, Recht et al. (2019)), and image source shift (CINIC-10, Darlow et al. (2018)). We plot the results in Fig. 2 which in particular shows that $\ell_p$-robust models have lower standard accuracy on distribution shifts than a standard model except for CIFAR-10-C where there is a clear improvement. Moreover, $\ell_p$ adversarial robustness generalizes across different distribution shifts (see App. D).

**Out-of-distribution (OOD) detection.** Ideally, a classifier should exhibit uncertainty in its predictions on OOD inputs. Hendrycks & Gimpel (2017) extract this uncertainty information using some threshold on the predicted confidence, assuming that OOD inputs receive low confidence from the model. Song et al. (2020) demonstrated that $\ell_\infty$ adversarial training can degrade the ability to recognize OOD inputs, measured as area under the ROC curve (AUROC), which is confirmed by our evaluation in Fig. 3 on three OOD datasets, CIFAR-100 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), Describable Textures Dataset (Cimpoi et al., 2014). We notice that, unlike for $\ell_p$-robustness, using extra data for training is not beneficial (see App. D for more results).

5 Outlook

We expect that a clearly defined, up-to-date leaderboard of the state-of-the-art robust models, easily accessible via a Model Zoo, will help to discover new insights and improve the current algorithms. We plan to expand our benchmark to other datasets such as CIFAR-100 and ImageNet.
REFERENCES


Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In ICML, 2019c.


Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? In ICML, 2019.


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


A BACKGROUND ON $\ell_p$-ROBUSTNESS

As mentioned in Sec. 2, computing the exact robust accuracy is in general intractable and NP-hard even for single-layer neural networks (Katz et al., 2017; Weng et al., 2018). Upper bounds are given by 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 (in particular, the step size and the number of iterations) can make the models appear more robust than they actually are (Engstrom et al., 2018; Mosbach et al., 2018), especially in the presence of phenomena like gradient obfuscation (Athalye et al., 2018). At the same time, certified methods such as Wong & Kolter (2018) and Gowal et al. (2019a) 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 (Tjeng et al., 2019).

We note that robustness towards small $\ell_p$-bounded perturbations is a necessary but not sufficient notion of robustness which has been criticized in the literature (Gilmer et al., 2018). It is an active area of research to develop threat models which are more aligned with the human perception such as spatial perturbations (Fawzi & Frossard, 2015; Engstrom et al., 2019b), Wasserstein-bounded perturbations (Wong et al., 2019; Hu et al., 2020), perturbations of the image colors (Laidlaw & Feizi, 2019) or $\ell_p$-perturbations in the latent space of a neural network (Laidlaw et al., 2020; Wong & Kolter, 2020). However, despite the simplicity of the $\ell_p$-perturbation model, it has numerous interesting applications that go beyond security considerations (Tramèr et al., 2019; Saadatpanah et al., 2020) and span transfer learning (Salman et al., 2020; Utrera et al., 2020), interpretability (Tsipras et al., 2019; Kaur et al., 2019; Engstrom et al., 2019b), generalization (Xie et al., 2020; Zhu et al., 2019; Bochkovskiy et al., 2020), robustness to unseen perturbations (Kang et al., 2019a; Xie et al., 2020; Laidlaw et al., 2020), stabilization of GAN training (Zhong et al., 2020). Thus, improvements in $\ell_p$-robustness have the potential to improve many of these downstream applications.

Therefore, since $\ell_p$-bounded perturbations are widely studied, well-defined, and have many reasons of interest, they constitute the threat models on which we decided to focus on initially, in addition to common image corruptions which cover a complementary aspect of robustness.

B RELATED WORK

Alongside with the libraries for adversarial attacks (see Sec. 2), many benchmarks aiming at tracking progress of adversarial defenses have appeared, with different forms. The two challenges (Kurakin et al., 2018; Brendel et al., 2018) 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. (2019) proposed DEEPSEC, a benchmark that tests many combinations of attacks and defenses, but suffers from a few shortcomings as suggested by Carlini (2019), 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. (2020) have provided an evaluation of a few defenses (in particular, 3 for $\ell_\infty$- and 2 for $\ell_2$-norm on CIFAR-10) against multiple commonly used attacks. However, they did not include some of the best performing defenses (Hendrycks et al., 2019; Carmon et al., 2019) and attacks (Gowal et al., 2019b; Croce & Hein, 2020a), 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. However, external evaluations are not submitted often enough, and 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.

Furthermore, it has become common practice to test new attacks wrt $\ell_\infty$ on the publicly available models from Madry et al. (2018) and Zhang et al. (2019b), since those represent widely accepted de-
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. For example, currently the difference in robust accuracy between the first and second-best attacks in the CIFAR-10 leaderboard of Madry et al. (2018) is only 0.03%, and between the second and third is 0.04%.

Finally, for common corruptions, there exists a benchmark at https://github.com/hendrycks/robustness which, however, only considers ImageNet and limits the classifiers to use only the ResNet-50 architecture. Thus, we think it is useful to track the progress also in this task for other datasets, and we start with CIFAR-10.

C DETAILS OF ROBUSTBENCH

C.1 LEADERBOARD

Restrictions. As mentioned in Sec. 3, 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 (Buckman et al. 2018; Guo et al. 2018), 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 (Athalye et al. 2018), 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 (Athalye et al. 2018). While often effective, it might be not sufficient, as shown by Tramer et al. (2020). 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 (Lecuyer et al. 2019; Cohen et al. 2019), 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 (Samangouei et al. 2018; Li et al. 2019) need to be evaluated adaptively with attacks considering jointly the loss of the inner loop and the standard classification task.

As described, the evaluation of classifiers not fulfilling these requirements can hardly be standardized. Moreover, we are not aware of models falling in one of these categories and not relying on standard methods such as adversarial training which are shown to achieve competitive robustness.

Evaluation of defenses. As described in Sec. 3, we perform the standardized evaluation of the $\ell_p$-robustness of the reported defenses using AutoAttack (Croce & Hein 2020b), an ensemble of four attacks (see appendix), 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 FAB attack (Croce & Hein 2020a), which minimizes the $\ell_p$-norm of the perturbations, and (4) the black-box Square Attack (Andriushchenko et al. 2020). Moreover, we keep open the option of submitting new evaluations of adversarial robustness by adaptive attacks. The goal is to achieve the most accurate approximation of the true robustness that can complement the standardized evaluation in some exceptional cases. Thus, we will report in the leaderboard both the results of the standardized attack and the best adaptive evaluation if it outperforms the standard one.
Figure 4: Visualization of the robustness and accuracy of 9 CIFAR-10 models from the RobustBench $\ell_2$-leaderboard. Robustness is evaluated using $\ell_2$-perturbations with $\varepsilon_2 = 0.5$.

C.2 Model Zoo

As mentioned in Sec. 3.2, we gather the checkpoints of many networks from the leaderboard. Below we illustrate how a model can be automatically downloaded and loaded via its identifier, the dataset on which it has been trained, and threat model to which it is robust within two lines of code:

```python
from robustbench.utils import load_model
model = load_model(model_name='Carmon2019Unlabeled',
                   dataset='cifar10', threat_model='Linf')
```

For the moment, the only available dataset is 'cifar10' and the available threat models are 'Linf' (for $\ell_\infty$ robustness), 'L2' (for $\ell_2$ robustness), and 'corruptions' (for common corruption robustness).

The library also enables to benchmark the robustness of some models even if they are not included in the Model Zoo, on a given dataset and threat model. This can be achieved with the `benchmark` function, in the following way:

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

The function returns the accuracy on the clean dataset, and the robust accuracy in the specified threat model. The `benchmark` function supports the same datasets and threat models as described above.

D Additional Analysis

We here extend the analysis of Sec. 4. As done above for the $\ell_\infty$-robust models, we show some statistics about the defenses for $\ell_2$-robustness in Fig. 4. The most robust model exploits a very large network and additional data for training.

Performance across various distribution shifts. In Fig. 5 we show the robust accuracy of $\ell_\infty$- and $\ell_2$-robust models from the Model Zoo on multiple distribution shifts of CIFAR-10. We report robust accuracy using AutoAttack and the same threat model as used for training on the standard CIFAR-10. We observe a trend that is very close to linear, and thus we conclude that $\ell_p$ robustness successfully transfers even under different distribution shifts. Finally, we note that concurrently with our work, Taori et al. (2020) also study the robustness to different distribution shifts of many models trained on ImageNet, including some $\ell_p$-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.

Out-of-distribution detection. Fig. 6 complements Fig. 3 and shows the ability of $\ell_2$-robust models trained on CIFAR-10 to distinguish inputs from other datasets (CIFAR-100, SVHN, Describable Textures). We find that $\ell_2$ robust models have in general comparable OOD detection performance to standardly trained models, while the model of Augustin et al. (2020) achieves even better perfor-
Figure 5: Robust accuracy of the robust classifiers, trained against $\ell_\infty$ and $\ell_2$ threat model, respectively, from our Model Zoo on various distribution shifts. The data points with 0% robust accuracy correspond to a standardly trained model.

Figure 6: Visualization of the quality of OOD detection (higher AUROC is better) for the $\ell_2$-robust models on three different OOD datasets: CIFAR-100 (left), SVHN (middle), Describable Textures (right).

mance since their approach explicitly optimizes both robust accuracy and worst-case OOD detection performance.

E LEADERBOARDS

We here report the details of all the models included in the various leaderboards, for the $\ell_\infty$, $\ell_2$-threat models and common corruptions. In particular, we show for each model the clean accuracy, robust accuracy (either on adversarial attacks or corrupted images), whether additional data is used for training, the architecture used, and the venue at which it appeared.
Table 1: Leaderboard for the \( \ell_\infty \)-threat model.

<table>
<thead>
<tr>
<th>model</th>
<th>clean</th>
<th>AA</th>
<th>extra</th>
<th>architecture</th>
<th>venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gowal et al. (2020)</td>
<td>91.10</td>
<td>65.87</td>
<td>Y</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Gowal et al. (2020)</td>
<td>89.48</td>
<td>62.76</td>
<td>Y</td>
<td>WideResNet-28-10</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Wu et al. (2020a)</td>
<td>87.67</td>
<td>60.65</td>
<td>Y</td>
<td>WideResNet-34-15</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Wu et al. (2020b)</td>
<td>88.25</td>
<td>60.04</td>
<td>Y</td>
<td>WideResNet-28-10</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Carmon et al. (2019)</td>
<td>89.69</td>
<td>59.53</td>
<td>Y</td>
<td>WideResNet-28-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Gowal et al. (2020)</td>
<td>85.29</td>
<td>57.14</td>
<td>N</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Gowal et al. (2020)</td>
<td>85.64</td>
<td>56.82</td>
<td>N</td>
<td>WideResNet-34-20</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Wu et al. (2020b)</td>
<td>85.36</td>
<td>56.17</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Wu et al. (2020a)</td>
<td>85.64</td>
<td>56.82</td>
<td>N</td>
<td>WideResNet-34-20</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Carmon et al. (2019)</td>
<td>85.36</td>
<td>56.17</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Uesato et al. (2019)</td>
<td>86.46</td>
<td>56.03</td>
<td>Y</td>
<td>WideResNet-28-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Hendrycks et al. (2019)</td>
<td>87.11</td>
<td>54.92</td>
<td>Y</td>
<td>WideResNet-28-10</td>
<td>ICML 2019</td>
</tr>
<tr>
<td>Pang et al. (2020)</td>
<td>86.43</td>
<td>54.39</td>
<td>N</td>
<td>WideResNet-34-20</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Pang et al. (2020)</td>
<td>86.70</td>
<td>53.57</td>
<td>N</td>
<td>WideResNet-34-20</td>
<td>arXiv, Nov 2020</td>
</tr>
<tr>
<td>Zhang et al. (2020)</td>
<td>84.52</td>
<td>53.51</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>ICML 2020</td>
</tr>
<tr>
<td>Rice et al. (2020)</td>
<td>85.34</td>
<td>53.42</td>
<td>N</td>
<td>WideResNet-34-20</td>
<td>ICML 2020</td>
</tr>
<tr>
<td>Zhang et al. (2020)</td>
<td>82.22</td>
<td>52.86</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>arXiv, Nov 2020</td>
</tr>
<tr>
<td>Qin et al. (2019)</td>
<td>86.28</td>
<td>52.84</td>
<td>N</td>
<td>WideResNet-40-8</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Chen et al. (2020a)</td>
<td>86.04</td>
<td>51.56</td>
<td>N</td>
<td>ResNet-50</td>
<td>CVPR 2020</td>
</tr>
<tr>
<td>Chen et al. (2020a)</td>
<td>85.32</td>
<td>51.12</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Silawarin et al. (2020)</td>
<td>86.84</td>
<td>50.72</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>arXiv, Mar 2020</td>
</tr>
<tr>
<td>Engstrom et al. (2019)</td>
<td>87.03</td>
<td>49.25</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>resNet-50</td>
</tr>
<tr>
<td>Singh et al. (2019)</td>
<td>87.80</td>
<td>49.12</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Xiao et al. (2019)</td>
<td>86.21</td>
<td>47.41</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Zhang et al. (2019)</td>
<td>87.20</td>
<td>44.83</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Madry et al. (2019)</td>
<td>87.14</td>
<td>44.04</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>ICML 2018</td>
</tr>
<tr>
<td>Pang et al. (2020a)</td>
<td>80.89</td>
<td>43.48</td>
<td>N</td>
<td>ResNet-32</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Wang et al. (2020)</td>
<td>83.34</td>
<td>43.21</td>
<td>N</td>
<td>ResNet-18</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Shafahi et al. (2019)</td>
<td>86.11</td>
<td>41.47</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Ding et al. (2020)</td>
<td>84.36</td>
<td>41.44</td>
<td>N</td>
<td>WideResNet-28-4</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Aitzmon et al. (2019)</td>
<td>81.30</td>
<td>40.22</td>
<td>N</td>
<td>ResNet-18</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Moosavi-Jazi et al. (2019)</td>
<td>83.11</td>
<td>38.50</td>
<td>N</td>
<td>ResNet-18</td>
<td>CVPR 2019</td>
</tr>
<tr>
<td>Zhang &amp; Wang (2019)</td>
<td>89.98</td>
<td>36.64</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>NeurIPS 2019</td>
</tr>
<tr>
<td>Jing et al. (2019)</td>
<td>78.91</td>
<td>34.95</td>
<td>N</td>
<td>ResNet-20</td>
<td>ICCV 2019</td>
</tr>
<tr>
<td>Xiao et al. (2020)</td>
<td>79.28</td>
<td>18.50</td>
<td>N</td>
<td>DenseNet-121</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Jin &amp; Renad (2020)</td>
<td>90.84</td>
<td>1.35</td>
<td>N</td>
<td>ResNet-18</td>
<td>arXiv, Mar 2020</td>
</tr>
<tr>
<td>Mustafa et al. (2019)</td>
<td>89.16</td>
<td>0.28</td>
<td>N</td>
<td>ResNet-110</td>
<td>ICCV 2019</td>
</tr>
<tr>
<td>Chan et al. (2020)</td>
<td>93.79</td>
<td>0.26</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Allaria et al. (2020)</td>
<td>91.03</td>
<td>0.00</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>arXiv, Jun 2020</td>
</tr>
<tr>
<td>Standard</td>
<td>94.78</td>
<td>0.00</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 2: Leaderboard for the \( \ell_2 \)-threat model.

<table>
<thead>
<tr>
<th>model</th>
<th>clean</th>
<th>AA</th>
<th>extra</th>
<th>architecture</th>
<th>venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gowal et al. (2020)</td>
<td>94.74</td>
<td>80.53</td>
<td>Y</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Gowal et al. (2020)</td>
<td>90.90</td>
<td>74.50</td>
<td>N</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>Wu et al. (2020a)</td>
<td>88.51</td>
<td>73.66</td>
<td>N</td>
<td>WideResNet-34-10</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>Augustin et al. (2020)</td>
<td>91.08</td>
<td>72.91</td>
<td>Y</td>
<td>ResNet-50</td>
<td>GitHub, Sep 2019</td>
</tr>
<tr>
<td>Engstrom et al. (2019)</td>
<td>90.83</td>
<td>69.24</td>
<td>N</td>
<td>ResNet-50</td>
<td>GitHub, Sep 2019</td>
</tr>
<tr>
<td>Rice et al. (2020)</td>
<td>87.28</td>
<td>67.68</td>
<td>N</td>
<td>ResNet-18</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Rony et al. (2019)</td>
<td>89.05</td>
<td>66.44</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>CVPR 2019</td>
</tr>
<tr>
<td>Ding et al. (2020)</td>
<td>88.02</td>
<td>66.09</td>
<td>N</td>
<td>WideResNet-28-4</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>Standard</td>
<td>94.78</td>
<td>0.00</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 3: Leaderboard for common corruptions.

<table>
<thead>
<tr>
<th>model</th>
<th>clean</th>
<th>corr.</th>
<th>extra</th>
<th>architecture</th>
<th>venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Hendrycks et al. (2020)</td>
<td>95.83</td>
<td>89.09</td>
<td>N</td>
<td>ResNeXt-29-32x4d</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>2 Hendrycks et al. (2020)</td>
<td>95.08</td>
<td>88.82</td>
<td>N</td>
<td>WideResNet-40-2</td>
<td>ICLR 2020</td>
</tr>
<tr>
<td>3 Kireev et al. (2021)</td>
<td>94.77</td>
<td>88.53</td>
<td>N</td>
<td>ResNet-18</td>
<td>arXiv, Mar 2021</td>
</tr>
<tr>
<td>4 Gowal et al. (2020)</td>
<td>94.74</td>
<td>87.68</td>
<td>Y</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>5 Kireev et al. (2021)</td>
<td>93.24</td>
<td>85.04</td>
<td>N</td>
<td>ResNet-18</td>
<td>arXiv, Mar 2021</td>
</tr>
<tr>
<td>6 Gowal et al. (2020)</td>
<td>90.90</td>
<td>84.90</td>
<td>N</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>7 Kireev et al. (2021)</td>
<td>93.10</td>
<td>84.10</td>
<td>N</td>
<td>ResNet-18</td>
<td>arXiv, Mar 2021</td>
</tr>
<tr>
<td>8 Gowal et al. (2020)</td>
<td>91.10</td>
<td>81.84</td>
<td>Y</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>9 Kireev et al. (2021)</td>
<td>92.46</td>
<td>80.46</td>
<td>N</td>
<td>ResNet-18</td>
<td>arXiv, Mar 2021</td>
</tr>
<tr>
<td>10 Gowal et al. (2020)</td>
<td>85.29</td>
<td>76.37</td>
<td>N</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
<tr>
<td>11 Standard</td>
<td>94.78</td>
<td>73.46</td>
<td>N</td>
<td>WideResNet-28-10</td>
<td>N/A</td>
</tr>
</tbody>
</table>