Adversarial Robustness against Multiple and Single l_p-Threat Models via Quick Fine-Tuning of Robust Classifiers

Francesco Croce [1] **Matthias Hein** [1]

Abstract

A major drawback of adversarially robust models, in particular for large scale datasets like ImageNet, is the extremely long training time compared to standard ones. Moreover, models should be robust not only to one l_p-threat model but ideally to all of them. In this paper we propose Extreme norm Adversarial Training (E-AT) for multiple-norm robustness which is based on geometric properties of l_p-balls. E-AT costs up to three times less than other adversarial training methods for multiple-norm robustness. Using E-AT we show that for ImageNet a single epoch and for CIFAR-10 three epochs are sufficient to turn any l_p-robust model into a multiple-norm robust model. In this way we get the first multiple-norm robust model for ImageNet and boost the state-of-the-art for multiple-norm robustness to more than 51% on CIFAR-10. Finally, we study the general transfer via fine-tuning of adversarial robustness between different individual l_p-threat models and improve the previous SOTA l_1-robustness on both CIFAR-10 and ImageNet. Extensive experiments show that our scheme works across datasets and architectures including vision transformers.

1. Introduction

The problem of adversarial examples, that is small adversarial perturbations of the input (Szegedy et al., 2014; Kurakin et al., 2017) changing the decision of a classifier, is a serious obstacle for the use of machine learning in safety-critical systems. Many adversarial defenses have been proposed but most of them could be broken either by stronger attacks (Carlini & Wagner, 2017; Athalye et al., 2018; Mosbach et al., 2018) or using adaptive attacks (Tramèr et al., 2020). Apart from provable adversarial defenses which are however still restricted to rather simple CNNs (Wong et al., 2018; Gowal et al., 2018), the only successful technique so far is adversarial training (Madry et al., 2018) and its improvements (Zhang et al., 2019; Carmon et al., 2019; Wu et al., 2020; 2021). Current state-of-the-art results for l_2- and l_∞-adversarial robustness on CIFAR-10 (Gowal et al., 2020; Rebuffi et al., 2021; Gowal et al., 2021) exploit very large networks like WideResNet-70-16. One can anticipate that this trend of using ever-growing architectures will continue, as we can currently see for ImageNet (Dai et al., 2021; Zhai et al., 2021). Therefore, especially considering that adversarial training is by itself more expensive than standard training, it is increasingly important that robust models can be quickly adapted to other tasks and threat models without the need for an excessively costly re-training from scratch. For this reason, in this paper we focus on fine-tuning which we believe will be an important topic in adversarial robustness in the future.

Moreover, while the community initially focused on adversarial examples for l_∞-perturbations, there has recently been more interest in other l_p-attacks, such as l_1 and l_2, or perceptual threat models (Stutz et al., 2019; Wong & Kolter, 2021; Laidlaw et al., 2021). It is well known that robustness in one l_p-ball does not necessarily generalize to some other l_q-ball for $p \neq q$ (Kang et al., 2019a; Tramèr & Boneh, 2019). However, in safety-critical systems we need robustness against all l_p-norms simultaneously which has triggered recent extensions of adversarial training for multiple l_p-norms (Tramèr & Boneh, 2019; Maini et al., 2020) and provable defenses for all l_p with $p \geq 1$ (Croce & Hein, 2020b). In this paper we show that, using the geometry of the l_p-balls, the computationally expensive multiple norm training procedures of Tramèr & Boneh (2019); Maini et al. (2020), which costs up to three times as much as normal adversarial training, can be replaced by a very effective and simple form of adaptively alternating between the two extreme norms, namely l_1 and l_∞. This scheme achieves similar robustness for the union of the threat models to more costly previous approaches. Additionally, we show that on CIFAR-10 three epochs and on ImageNet even just a single epoch of fine-tuning with our extreme norms adversarial training (E-AT) are sufficient to turn any l_p-robust model

[1]University of Tübingen, Germany. Correspondence to: Francesco Croce <francesco.croce@uni-tuebingen.de>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

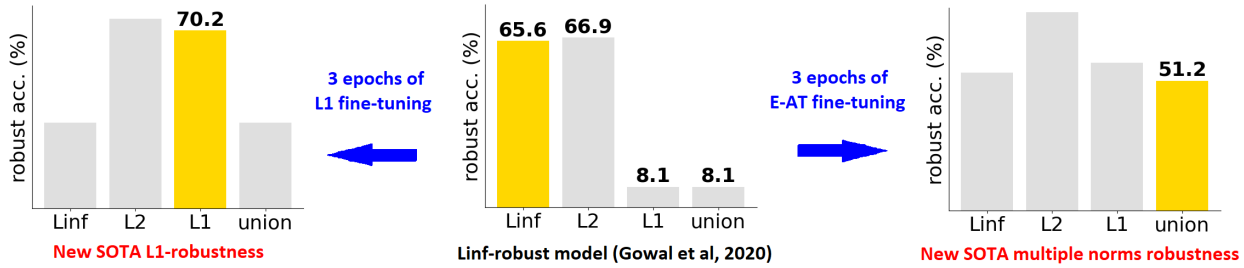

Figure 1. We fine-tune for 3 epochs the WideResNet-70-16 on CIFAR-10 from (Gowal et al., 2020) with highest l_∞-robustness to be either robust wrt l_1 (left) or with our E-AT to be robust wrt the union of the l_∞-, l_2-, and l_1-threat models (right). We achieve state-of-the-art results in both threat models. The plots show the robust accuracy in the individual threat models and in their union for the initial l_∞-robust classifier (middle) and the fine-tuned ones, with the target threat model highlighted.

for $p \in \{1, 2, \infty\}$ into a model which is robust against all l_p-threat models for $p \in \{1, 2, \infty\}$, even if the original classifier was completely non-robust against one of them. Fine-tuning one of the currently most robust networks in the l_∞-threat model (Gowal et al., 2020) for CIFAR-10 we significantly improve the current state-of-the-art performance for multiple norm robustness (robustness over the union of l_1, l_2 and l_∞-balls) without the need of training from scratch this large network with existing expensive techniques. Also, our E-AT fine-tuning scheme yields in just one epoch the first ImageNet model which is robust against multiple l_p-bounded attacks. Finally, we show that fine-tuning, again with only 3 epochs for CIFAR-10 and one epoch for ImageNet, is sufficient to transfer robustness from one threat model to another one, which very quickly yields baselines for all threat models. In this way, starting from l_∞-robust classifiers we achieve SOTA l_1-robust classifiers on CIFAR-10 and ImageNet. These results are quite remarkable as the original classifiers show no or very little l_1-robustness. Fig. 1 summarizes the results of fine-tuning of robust classifiers on CIFAR-10 (see Table 2 and Table 7 for details), and Table 3 for ImageNet shows that our results hold also for fine-tuning of transformer architectures.

2. Related Work

Adversarial training: Nowadays, adversarial training as formulated in Madry et al. (2018) as a min-max optimization problem remains the only general method ensuring adversarial robustness across architectures and datasets (Athalye et al., 2018). Other types of defenses use more sophisticated techniques, typically preventing the direct optimization of the attack. However, adaptive attacks specifically designed for these defenses have often shown that these alternative techniques are non-robust or much less robust than claimed (Tramèr et al., 2020). Recent improvements of adversarial training have been achieved by using different objectives (Zhang et al., 2019), unlabeled data (Carmon et al., 2019), adversarial weights perturbations (Wu et al., 2020) and wider networks (Wu et al., 2021). In Gowal et al. (2020) several recent variants were systematically explored and for a very large architecture they obtain the most robust models for l_∞ and l_2 for CIFAR-10, which we use for our fine-tuning experiments, only slightly improved later on by using particular data augmentation (Rebuffi et al., 2021) or synthetic data (Gowal et al., 2021).

Multiple-norm robustness: It was early on discovered that adversarial robustness against a specific l_p-threat model does typically not transfer to l_q-threat models for $p \neq q$ (see Kang et al. (2019a); Tramèr & Boneh (2019) for extensive studies). On the other hand to achieve really reliable machine learning models l_p-robustness wrt all p is necessary. The first approach to general robustness (Schott et al., 2019) uses multiple variational autoencoders for an analysis by synthesis (ABS) architecture. While ABS is restricted to MNIST, it is robust against l_0, l_2 and l_∞-attacks (although the l_0-robustness has been recently reduced with a stronger black-box attack (Croce et al., 2022)). Tramèr & Boneh (2019); Maini et al. (2020); Madaan et al. (2021) use variants of adversarial training to achieve robustness in multiple norms. Since these are the most similar methods to ours, we present them in detail below. Madaan et al. (2021) additionally proposes a meta-learning approach where one learns optimal noise to augment the samples and uses consistency regularization to enforce similar predictions on clean, augmented and adversarial samples. Finally, Stutz et al. (2020) combine adversarial training with a reject option: while that is effective, the comparison to normal adversarially trained models is difficult as their model is non-robust without the reject option.

Provable robustness: Croce & Hein (2020b) motivated a regularization approach based on the geometry of the l_p-balls which enforces multiple-norm robustness during training: this allows to derive provable guarantees for multiple-norm robustness in contrast to the empirical eval-

uation of adversarial training. However, their approach works only for small network architectures and relatively small radii of the l_p-balls.

Fine-tuning of robust models: Fine-tuning of an existing neural network is a commonly used technique in deep learning (Goodfellow et al., 2016) to quickly adapt a model to a different objective e.g. for language models (Howard & Ruder, 2018). More recently, it has been shown in the area of adversarial robustness that fine-tuning of pre-trained models, possibly using self-supervision, yields better adversarial robustness (Hendrycks et al., 2019; Chen et al., 2020; Xu & Yang, 2020). Jeddi et al. (2020) show that fine-tuning of non-robust models with 10 epochs can yield robust models with the caveat that their robustness evaluation is done using only a single run of PGD with 20 steps. We are not aware of any prior work discussing fine-tuning to transform a robust model wrt a single l_p into one robust wrt multiple threat models or another l_q.

Evaluation of adversarial robustness: Many white-box attacks for l_∞ (Madry et al., 2018; Gowal et al., 2019), l_2 (Madry et al., 2018; Carlini & Wagner, 2017) and l_1 (Chen et al., 2018; Modas et al., 2019; Rony et al., 2021) have been proposed as well as several black box attacks (Brendel et al., 2018; Liu et al., 2019; Cheng et al., 2019; Al-Dujaili & O'Reilly, 2020; Meunier et al., 2019; Zhao et al., 2019) for different threat models. It has recently been shown that AutoAttack (Croce & Hein, 2020c), a parameter-free ensemble of the white-box attacks APGD for the cross-entropy and DLR-loss, FAB-attack (Croce & Hein, 2020a) and the black-box Square Attack (Andriushchenko et al., 2020) is reliably evaluating adversarial robustness for l_2 and l_∞. AutoAttack has recently been extended to l_1 (Croce & Hein, 2021) outperforming all existing state-of-the-art attacks for l_1. Croce & Hein (2020c; 2021) showed that on models defended with adversarial training the two versions of APGD (with budget as in AutoAttack) already give an accurate robustness evaluation. As we have to evaluate very large models always for three threat models, we use those as a strong standard attack in our experiments.

2.1. Adversarial training for the union of l_1-, l_2- and l_∞-balls

Let us denote by $f_\theta : \mathbb{R}^d \to \mathbb{R}^K$ the classifier parameterized by $\theta \in \mathbb{R}^n$, with input $x \in \mathbb{R}^d$ and $f_\theta(x) \in \mathbb{R}^K$ where K is the number of classes of the task. Let further $\mathcal{D} = \{(x_i, y_i)\}_i$ be the training set, with y_i the correct label of x_i, and $\mathcal{L} : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ a given loss function. The aim is to enforce adversarial robustness in all multiple l_p-balls simultaneously, i.e., defining $B_p(\epsilon_p) = \{x \in \mathbb{R}^d : \|x\|_p \leq \epsilon_p\}$, the threat model is the union of the individual l_p-balls, i.e. $\Delta = B_1(\epsilon_1) \cup B_2(\epsilon) \cup B_\infty(\epsilon_\infty)$. Δ is a non convex set, since in practice the ϵ_p are chosen such that no l_p-ball contains any of the others. In adversarial training the worst case loss for input perturbations in the threat model, $\max_{\delta \in \Delta} \mathcal{L}(f_\theta(x_i + \delta), y_i)$, is minimized. Efficiently maximizing the loss $\mathcal{L}$ in the union of threat models is non-trivial and different approaches have been proposed.

MAX: Tramèr & Boneh (2019) suggest to run the three attacks for each $B_p(\epsilon_p)$ for $p \in \{1, 2, \infty\}$ independently and then use the one which realizes the highest loss, that is

$$\max_{p \in \{1,2,\infty\}} \max_{\delta \in B_p(\epsilon_p)} \mathcal{L}(f_\theta(x_i + \delta), y_i).$$

This training optimizes directly the worst case in the union but comes at the price of being nearly three times as expensive as normal adversarial training wrt a single l_p-ball.

AVG: Additionally, Tramèr & Boneh (2019) suggest to run the three attacks independently but replace the inner maximization problem with

$$\sum_{p \in \{1,2,\infty\}} \max_{\delta \in B_p(\epsilon_p)} \mathcal{L}(f_\theta(x_i + \delta), y_i),$$

and thus averaging the updates of all l_p-balls with the motivation of not "wasting" the computed attacks, in particular when the attained loss values are rather similar and thus the max is ambiguous. This has similar cost to MAX.

MSD: Maini et al. (2020) argue that the correct way to maximize the loss in the union is to test during the PGD attacks all the three steepest ascent updates corresponding to the three norms (sign of the gradient for l_∞, normalized gradient for l_2 and a smoothed l_1-step by using the top-k components in magnitude of the gradient) and then take the step which yields the highest loss. Maini et al. (2020) report that MSD outperforms both AVG and MAX, also in terms of a more stable training. As all three updates (forward passes) are tested but only one backward pass is needed (gradient is the same) this costs roughly two times as much as normal adversarial training.

SAT: Madaan et al. (2021) introduce Stochastic Adversarial Training (SAT) which randomly samples $p \in \{1, 2, \infty\}$ for each batch and performs PGD only for the corresponding l_p-norm. While SAT has the same cost as standard adversarial training, Madaan et al. (2021) report that it does not perform very well.

3. Fast Multiple-Norm Robustness via Extreme Norms Adversarial Training and Fine-Tuning

3.1. Multiple-norm robustness via fast fine-tuning of existing robust models

Prior works (Tramèr & Boneh, 2019; Kang et al., 2019a) observed that models adversarially trained wrt l_∞ give non

Table 1. **CIFAR-10 - Other methods vs E-AT for fine-tuning:** We fine-tune with different methods for multiple norms for 3 epochs the RN-18 robust wrt l_∞ (mean and standard deviation of the clean and robust accuracy over 5 seeds is reported). We report clean performance, robust accuracy in each l_p-threat model, their average and the robust accuracy in their union (all values in percentage).

model	*clean*	l_∞ ($\epsilon_\infty = \frac{8}{255}$)	l_2 ($\epsilon_2 = 0.5$)	l_1 ($\epsilon_1 = 12$)	*average*	*union*	*time/epoch*
RN-18 l_∞-AT	83.7	48.1	59.8	7.7	38.5	7.7	151 s
+ SAT	83.5 $\pm$ 0.2	43.5 $\pm$ 0.2	68.0 $\pm$ 0.4	47.4 $\pm$ 0.5	53.0 $\pm$ 0.2	41.0 $\pm$ 0.3	161 s
+ AVG	84.2 $\pm$ 0.4	43.3 $\pm$ 0.4	68.4 $\pm$ 0.6	46.9 $\pm$ 0.6	52.9 $\pm$ 0.4	40.6 $\pm$ 0.4	479 s
+ MAX	82.2 $\pm$ 0.3	45.2 $\pm$ 0.4	67.0 $\pm$ 0.7	46.1 $\pm$ 0.4	52.8 $\pm$ 0.3	42.2 $\pm$ 0.6	466 s
+ MSD	82.2 $\pm$ 0.4	44.9 $\pm$ 0.3	67.1 $\pm$ 0.6	47.2 $\pm$ 0.6	53.0 $\pm$ 0.4	42.6 $\pm$ 0.2	306 s
+ E-AT	82.7 $\pm$ 0.4	44.3 $\pm$ 0.6	68.1 $\pm$ 0.5	48.7 $\pm$ 0.5	53.7 $\pm$ 0.3	42.2 $\pm$ 0.8	160 s

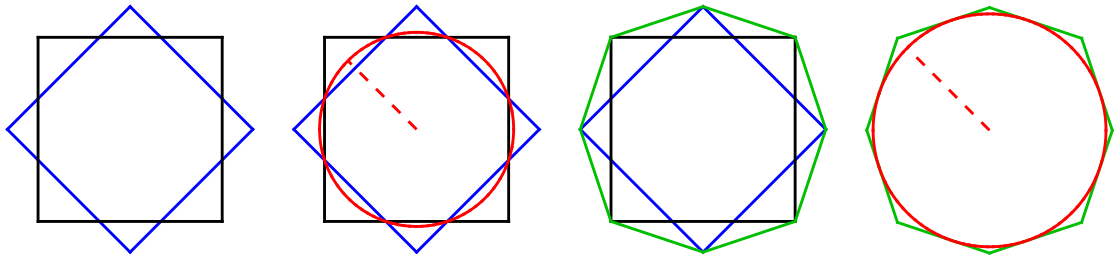

Figure 2. Visualization of the l_2-ball contained in the union resp. the convex hull of the union of l_1- and l_∞-balls in $\mathbb{R}^2$. **First:** co-centric l_1-ball (blue) and l_∞-ball (black). **Second:** in red the largest l_2-ball contained in the union of l_1- and l_∞-ball. **Third:** in green the convex hull of the union of the l_1- and l_∞-ball. **Fourth:** the largest l_2-ball (red) contained in the convex hull. The l_2-ball contained in the convex hull is significantly larger than that in the union of l_1- and l_∞-ball.

trivial robustness to l_2-attacks, although lower than what one gets directly training against such attacks, and vice versa. This is confirmed by our evaluation in App. C.7, where we also notice that l_1-AT provides good robust accuracy in l_2. On the other hand, training for l_∞ resp. l_1 does not yield particular robustness to the dual norm, which is reasonable since the perturbations generated in the two threat models are very different, while the l_2-threat model is an intermedicate case which yields partial robustness against l_∞ and l_1. Therefore, we propose to use models trained for robustness wrt a single norm as good initializations to achieve, within a small computational budget, multiple norms robustness. We test this by fine-tuning for 3 epochs an l_∞-robust model on CIFAR-10 with the existing methods for multiple norms robustness and our E-AT. Table 1 shows that a short fine-tuning (details in Sec. 3.4) of a PreAct ResNet-18 (He et al., 2016) trained with adversarial training wrt l_∞ is effective in achieving competitive robustness in the union, not far from those of full training from random initialization (see Table 5). However, the most effective methods, MAX and MSD, are 2-3x slower than standard adversarial training, while SAT is as fast as l_∞-AT but performs slightly worse. E-AT aims at achieving the same results as MAX and MSD in the union while having complexity on par with SAT. Moreover, we report the average robustness in the 3 threat models, as done in prior works, where E-AT achieves the best results. In App. C.1 we show that similar observations can be made when fine-tuning models initially l_2- or l_1-robust, and (Sec. 3.4) are not specific to CIFAR-10 but generalize to ImageNet.

All previous methods assume that for achieving robustness to multiple norms each threat models has to be used at training time. In the following we first present an argument, using recent results from Croce & Hein (2020b), suggesting that this need not be the case. Based on this analysis, we introduce our extreme norms adversarial training (E-AT) which achieves multiple norm robustness at the same price as training for a single l_p. Finally, we fine-tune with E-AT large robust models on CIFAR-10 and ImageNet.

3.2. Geometry of the union of l_p-balls and their convex hull

The main insight we use for E-AT is that a linear classifier which is robust in both an l_1- and an l_∞-ball is also robust wrt the largest l_p-ball for $1 \leq p \leq \infty$ which fits into the convex hull of the union of the l_1- and l_∞-ball. This ball is significantly larger than the largest l_p-ball contained into the union of the l_1- and l_∞-ball (see Fig. 2). Thus it is sufficient to be robust wrt the two "extreme" norms l_1 and l_∞ to ensure robustness. While this is exact for affine classifiers, we conjecture that for neural networks this will at least hold approximately true (note that typical ReLU-networks yield piecewise affine classifiers (Arora et al., 2018)) and for the model it is the most efficient way in terms of capacity to be l_1- and l_∞-robust.

We now state the main results from Croce & Hein (2020b) which are the basis for our E-AT. We work in the non-trivial setting, where the balls are not included in each other as otherwise the problem of enforcing multiple norms (including l_1 or l_∞) robustness boils down again to single norm robustness. For this, it has to hold $\epsilon_1 \in (\epsilon_\infty, d\epsilon_\infty)$: for CIFAR-10, $\epsilon_\infty = \frac{8}{255}$ and dimension $d = 3072$ yield an upper bound $\epsilon_1 \leq 96.38$ which is far higher than $\epsilon_1 = 12$ commonly used for l_1-threat models. We denote by $U_{1,\infty}(\epsilon_1, \epsilon_\infty) = B_1(\epsilon_1) \cup B_\infty(\epsilon_\infty)$ the union of the l_1- and l_∞-balls. One can then ask the obvious question: how much l_p-robustness do I get for $1 < p < \infty$ from a classifier robust in $U_{1,\infty}(\epsilon_1, \epsilon_\infty)$?

Proposition 3.1 (Croce & Hein (2020b)) *If $d \geq 2$ and $\epsilon_1 \in (\epsilon_\infty, d\epsilon_\infty)$, then*

$$\min_{x \in \mathbb{R}^d \setminus U_{1,\infty}(\epsilon_1, \epsilon_\infty)} \|x\|_p = \left(\epsilon_\infty^p + \frac{(\epsilon_1 - \epsilon_\infty)^p}{(d-1)^{p-1}} \right)^{\frac{1}{p}}. \quad (1)$$

Thus a classifier which is robust for the union $U_{1,\infty}(\epsilon_1, \epsilon_\infty)$ has automatically a non-trivial robustness for all intermediate l_p-norms. For us the case $p = 2$ is most interesting which given that $\epsilon_1 \gg \epsilon_\infty$ can be tightly upper bounded as

$$\epsilon_2 := \min_{x \in \mathbb{R}^d \setminus U_{1,\infty}(\epsilon_1, \epsilon_\infty)} \|x\|_2 \leq \sqrt{\epsilon_\infty^2 + \frac{\epsilon_1^2}{d-1}}. \quad (2)$$

For CIFAR-10 one gets $\epsilon_2 \leq 0.2188$. As the radius of the l_2-threat model is usually chosen as $\epsilon_2 = 0.5$, robustness in the union alone would not be sufficient to achieve the desired robustness wrt l_p for $p \in \{1, 2, \infty\}$. However, if we consider affine classifiers then a guarantee for $B_1(\epsilon_1)$ *and* $B_\infty(\epsilon_\infty)$ implies a guarantee with respect to the convex hull C of their union $B_1(\epsilon_1) \cup B_\infty(\epsilon_\infty)$ as an affine classifier generates a half-space and thus only the extreme points of $B_1(\epsilon_1)$ resp. $B_\infty(\epsilon_\infty)$ matter (see Figure 2 for an illustration).

Theorem 3.1 (Croce & Hein (2020b)) *Let C be the convex hull of $B_1(\epsilon_1) \cup B_\infty(\epsilon_\infty)$. If $d \geq 2$ and $\epsilon_1 \in (\epsilon_\infty, d\epsilon_\infty)$, then*

$$\min_{x \in \mathbb{R}^d \setminus C} \|x\|_p = \frac{\epsilon_1}{\left(\epsilon_1/\epsilon_\infty - \alpha + \alpha^q \right)^{1/q}}, \quad (3)$$

where $\alpha = \frac{\epsilon_1}{\epsilon_\infty} - \lfloor \frac{\epsilon_1}{\epsilon_\infty} \rfloor$ and $\frac{1}{p} + \frac{1}{q} = 1$.

As standard architectures using ReLU activation function yield a piecewise affine classifier one can anticipate that this result gives at least a rule of thumb on the expected l_p-robustness when one is l_1- and l_∞-robust. Again with the choice of $\epsilon_1, \epsilon_\infty$ from above one gets for the radius of the l_2-ball that fits into the convex hull C of the union of $B_1(\epsilon_1)$ and $B_\infty(\epsilon_\infty)$:

$$\epsilon_2 := \min_{x \in \mathbb{R}^d \setminus C} \|x\|_2 = \frac{\epsilon_1}{\sqrt{\epsilon_1/\epsilon_\infty - \alpha + \alpha^2}} \approx 0.6178. \quad (4)$$

Thus for a desired l_2-robustness with radius less than 0.6178 it is sufficient for an affine classifier, and at least plausible for a ReLU network, to enforce l_1-robustness with $\epsilon_1 = 12$ and l_∞-robustness with $\epsilon_\infty = \frac{8}{255}$. This motivates our extreme norms adversarial training (E-AT) and fine-tuning.

3.3. Extreme norms adversarial training (E-AT)

In light of the geometrical argument presented in the previous section, we propose to train only on adversarial perturbations for the l_∞- and l_1-threat models if the l_2-radius obtained from Theorem 3.1 is larger than the radius ϵ_2 of the l_2-threat model. In this case it is sufficient to just train for the extremes l_1 and l_∞ in order to achieve robustness also to the intermediate l_p-attacks with $p \in (1, \infty)$. Since we seek a method as expensive as standard adversarial training, for each batch we do either the l_1- or the l_∞-attack. For full training from a random initialization simply alternating or sampling uniformly at random from the l_1- and l_∞-attack works already well. However, for very quick fine-tuning, e.g. just one epoch in the case of ImageNet, for multiple norm robustness from an existing classifier robust wrt a single threat model, one has to take into account the existing robustness of the model. Thus we use an adaptive sampling strategy based on the running averages, reset at every epoch, of the robust training errors rerr_1 and rerr_∞ (note that these running averages are computed just from averaging the robust error on the batches where the respective attack has been performed, thus no extra attacks are necessary), such that the probability for sampling the l_p-threat model is

$$\frac{\text{rerr}_p}{\text{rerr}_1 + \text{rerr}_\infty}, \quad \text{for } p \in \{1, \infty\}. \quad (5)$$

The motivation for this sampling scheme is that the robust error in the union Δ is mainly influenced by the worst threat model. We show in more details the effect of the biased sampling in E-AT fine-tuning in Sec. C.4.

3.4. Scaling up multiple-norm robust models

The fact that multiple-norm robustness can be achieved via a short fine-tuning allows to use large architectures, which would be hard and expensive to train from random initialization in this more difficult threat model. Moreover, fine-tuning with E-AT has similar computational cost per epoch as standard adversarial training, and since we aim at efficiency, we therefore use it as main tool on large models.

Experimental details: In the following we fine-tune

Table 2. **CIFAR-10 - 3 epochs of E-AT fine-tuning on l_p-robust models:** We fine-tune with E-AT models robust wrt a single l_p-norm, and report the robust accuracy on 1000 test points for all threat models and the difference to the initial classifier. (*) uses extra data.

	model	*clean*		l_∞ ($\epsilon_\infty = \frac{8}{255}$)		l_2 ($\epsilon_2 = 0.5$)		l_1 ($\epsilon_1 = 12$)		*union*	
Fine-tuning l_∞-robust models	RN-50 - l_∞	88.7		50.9		59.4		5.0		5.0	
	(Engstrom et al., 2019) + FT	86.2	-2.5	46.0	-4.9	70.1	10.7	49.2	44.2	43.4	38.4
	WRN-34-20 - l_∞	87.2		56.6		63.7		8.5		8.5	
	(Gowal et al., 2020) + FT	88.3	1.1	49.3	-7.3	71.8	8.1	51.2	42.7	46.2	37.7
	WRN-28-10 - l_∞ (*)	90.3		59.1		65.7		8.0		8.0	
	(Carmon et al., 2019) + FT	90.3	0.0	52.6	-6.5	74.7	9.0	54.0	46.0	48.7	40.7
	WRN-28-10 - l_∞ (*)	89.9		62.9		67.2		10.8		10.8	
	(Gowal et al., 2020) + FT	91.2	1.3	53.9	-9.0	76.0	8.8	56.9	46.1	50.1	39.3
	WRN-70-16 - l_∞ (*)	90.7		65.6		66.9		8.1		8.1	
	(Gowal et al., 2020) + FT	91.6	0.9	54.3	-11.3	78.2	11.3	58.3	50.2	51.2	43.1
Fine-tuning l_2-robust models	RN-50 - l_2	91.5		29.7		70.3		27.0		23.0	
	(Engstrom et al., 2019) + FT	87.8	-3.7	43.1	13.4	70.8	0.5	50.2	23.2	41.7	18.7
	RN-50 - l_2 (*)	91.1		37.7		73.4		31.2		28.8	
	(Augustin et al., 2020) + FT	87.0	-4.1	47.2	9.5	70.4	-3.0	54.1	22.9	46.0	17.2
	WRN-70-16 - l_2 (*)	94.1		43.1		81.7		34.6		32.4	
	(Gowal et al., 2020) + FT	91.2	-2.9	51.9	8.8	79.2	-2.5	58.8	24.2	49.7	17.3
Fine-tuning l_1-robust models	RN-18 - l_1	87.1		22.0		64.8		60.3		22.0	
	(Croce & Hein, 2021) + FT	83.5	-3.6	40.3	18.3	68.1	3.3	55.7	-4.6	40.1	18.1

with E-AT[1] for 3 epochs on CIFAR-10 and 1 epoch on ImageNet-1k, starting with learning rate 0.05 or 0.01, depending on the model, and decreasing by a factor of 10 every 1/3 of the total number of finetuning epochs. We do 10 steps of APGD in adversarial training for CIFAR-10, while 5 and 15 with l_∞ and l_1 respectively on ImageNet as optimizing in the l_1-ball requires more iterations in that case. When the model was originally trained with extra data beyond the training set on CIFAR-10, we use the 500k images introduced by Carmon et al. (2019) as additional data for fine-tuning (see also Sec. A).

CIFAR-10: RobustBench (Croce et al., 2021) provides a collection of the currently most robust classifiers. We took a subset of the most robust models, among those which do not use synthetic data, for l_2- and l_∞-norm and the l_1-robust one from Croce & Hein (2021) (all are trained with the same radii ϵ_p as in our experiment). Note that we use the classifiers from Gowal et al. (2020) (instead of those from Rebuffi et al. (2021)) since those were the best available ones at the time of the start of this project. We present in Table 2 the results. First of all the fine-tuning works for all tested architectures and results in many cases in stronger robustness in the union than for the specifically trained WideResNet-28-10 models (see Table 14). In particular, the most robust l_∞-model from Gowal et al. (2020) with 65.6% l_∞-robustness and only 8.1% l_1-robustness can be fine-tuned to a multiple-norm robust model with 51.2% robustness which is up to our knowledge the best reported multiple-norm robustness. While in general it is expected that larger architectures and extra data improve robustness (see e.g. RobustBench leaderboards), we could achieve such improvement without the high computational cost (and potential instabilities) of training large networks on an extended dataset from scratch. Very interesting is that the l_2-robustness of 78.2% is quite close to the 81.7% l_2-robustness of the specifically l_2-trained model from Gowal et al. (2020). Moreover, the l_1-robustness of 58.3% is close to the best reported one of 60.3% (Croce & Hein, 2021) (however we improve this a lot in the next section) and the model has even higher clean accuracy. Clearly, this comes at the price of a significant loss in l_∞ but this is to be expected. Striking is that fine-tuning the l_2-robust model from Gowal et al. (2020) results in a very similar result. Finally, we observe that the l_∞-threat model is the most challenging one, and fine-tuning l_∞-robust models yields the best robust accuracy in the union (when comparing models with the same architecture). In a nutshell, E-AT fine-tuning of existing l_p-robust models yields very efficient and competitive baselines for future research in this area.

ImageNet: We start with the l_2- resp. l_∞-robust models from Engstrom et al. (2019), Bai et al. (2021) and Debenedetti (2022) including the vision transformers DeiT small (Touvron et al., 2021) and XCiT small (El-Nouby et al., 2021) We use $\epsilon_2 = 2$ for the experiments as the robust accuracy is still in a reasonable range of 40% and

[1]Code available at https://github.com/fra31/robust-finetuning.

Table 3. **ImageNet - Results of one epoch of E-AT fine-tuning of existing robust models:** We use existing models trained to be robust wrt a single l_p-ball (either l_∞ or l_2) and fine-tune them for a single epoch for multiple-norm robustness with our E-AT scheme.

	model	*clean*		l_∞ ($\epsilon_\infty = \frac{4}{255}$)		l_2 ($\epsilon_2 = 2$)		l_1 ($\epsilon_1 = 255$)		union	
Fine-tuning l_∞-robust models	RN-50 - l_∞	62.9		29.8		17.7		0.0		0.0	
	(Engstrom et al., 2019) + FT	58.0	-4.9	27.3	-2.5	41.1	23.4	24.0	24.0	21.7	21.7
	RN-50 - l_∞	68.2		36.7		15.6		0.0		0.0	
	(Bai et al., 2021) + FT	60.1	-8.1	29.2	-7.5	42.1	26.5	24.5	24.5	22.6	22.6
	DeiT-S - l_∞	66.4		35.6		40.1		3.1		3.1	
	(Bai et al., 2021) + FT	62.6	-3.8	32.2	-3.4	46.1	6.0	24.8	21.7	23.6	20.5
	XCiT-S - l_∞	72.8		41.7		45.3		2.7		2.7	
	(Debenedetti, 2022) + FT	68.0	-4.8	36.4	-5.3	51.3	6.0	28.4	25.7	26.7	24.0
Fine-tuning l_2-robust models	RN-50 - l_2	58.7		25.0		40.5		14.0		13.5	
	(Engstrom et al., 2019) + FT	56.7	-2.0	26.7	1.7	41.0	0.5	25.4	11.4	23.1	9.6

Table 4. **ImageNet - 1 epoch of fine-tuning of existing robust models:** We use existing models trained to be robust wrt a single l_p-ball (either l_∞ or l_2) and fine-tune them for a single epoch for multiple-norm robustness with SAT and E-AT.

model	*clean*	l_∞	l_2	l_1	*union*
RN-50 - l_∞-AT	62.9	29.8	17.7	0.0	0.0
+ SAT	59.4	26.5	38.8	21.1	19.4
+ E-AT	58.0	27.3	41.1	24.0	21.7
RN-50 - l_2-AT	58.7	25.0	40.5	14.0	13.5
+ SAT	57.7	25.9	41.6	23.2	21.1
+ E-AT	56.7	26.7	41.0	25.4	23.1

together with our choice of $\epsilon_1 = 255$ and the standard $\epsilon_\infty = \frac{4}{255}$ the l_2-radius from Theorem 3.1 is almost exactly 2. The initial l_∞-models are completely non-robust for l_1 but achieve, after fine-tuning, over 24% l_1-robust accuracy and also the l_2-robust accuracy improves, at the price of a relatively small loss in l_∞-robust and clean accuracy. Interestingly, the DeiT-S and XCiT-S models has already high robustness wrt l_2, unlike the RN-50s, which further improves thanks to E-AT, and the latter attains the best robustness in the union. For the l_2-robust model all robust accuracies improve as the original model was trained for $\epsilon = 3$. Up to our knowledge no multiple-norm robustness has been reported before for ImageNet and thus these results are an important baseline. Finally, we show in Table 4 that both SAT and E-AT are effective for fine-tuning on ImageNet, and E-AT achieves the best robustness in the union (we omit the other methods since they are computationally more expensive). We also observe that in this case l_1 is the most challenging threat model, and the best robustness in the union is achieved when fine-tuning the classifier (among those using RN-50 as architecture) trained wrt l_2 which already has non-trivial robustness wrt l_1.

Additional experiments: Appendix C contains further studies and details about fine-tuning with E-AT, e.g. we report runtime and show that fine-tuning a naturally trained model does not provide competitive robustness and leads to low clean accuracy. Moreover, we show the stability of the scheme over random seeds, that increasing the number of epochs progressively improves the robustness in the union, and that even models trained to be robust wrt perceptual metrics can be used for E-AT fine-tuning.

3.5. Full training for multiple norm robustness from random initialization

We evaluate the performance of the different methods for multiple-norm robustness when applied for full training from random initialization on CIFAR-10 (using the same ϵ_p as above). Table 5 reports the results, averaged over 3 runs, in every threat model: MAX and MSD attain the best robustness in the union, and E-AT is close to them and outperforms SAT. We recall that E-AT is 2-3x less expensive than MSD and MAX (see Table 1). We also include the performance of models trained to be robust for single norms, which do not show high robustness in the union of the threat models. More details and further experiments with WideResNet-28-10 as architecture can be found in App. A and App. C.7.

3.6. Robustness against unseen non l_p-bounded adversarial attacks

We investigate on CIFAR-10 to which extent adversarial robustness achieved with multiple norms training generalizes to unseen and possibly very different threat models. We select three sparse attacks (l_0-bounded, patches and frames) and five adversarial corruptions (fog, snow, Gabor noise, elastic, l_∞-JPEG) from Kang et al. (2019b). Additionally, we compute the accuracy of the classifiers on the common corruptions, i.e. not adversarially optimized, of CIFAR-10-C (Hendrycks & Dietterich, 2019). Table 6 shows the results against such attacks of WRN-28-10 ad-

Table 5. **CIFAR-10 - Training from random initialization:** For full training of PreAct ResNet-18 with different methods for multiple norm robustness, we show robust accuracy in each l_p-threat model, their average, and the robustness in their union.

model	*clean*	l_∞	l_2	l_1	*avg.*	*union*
l_∞-AT	84.0	48.1	59.7	6.3	38.0	6.3
l_2-AT	88.9	27.3	68.7	25.3	40.5	20.9
l_1-AT	85.9	22.1	64.9	59.5	48.8	22.1
SAT	83.9	40.7	68.0	54.0	54.2	40.4
AVG	84.6	40.8	68.4	52.1	53.8	40.1
MAX	80.4	45.7	66.0	48.6	53.4	44.0
MSD	81.1	44.9	65.9	49.5	53.4	43.9
E-AT	81.9	43.0	66.4	53.0	54.2	42.4

versarially trained either wrt single norms or for multiple norm robustness. Additionally, we include the PAT model from Laidlaw et al. (2021), which uses a RN-50 as architecture, is obtained with perceptual adversarial training, and has been shown to be robust to unseen attacks, and a normally trained model (NAT) as baseline. The models trained wrt multiple norms show much higher robustness in both the union (worst case robustness over all threat models, excluding common corruptions) and average robustness against these attacks: in particular, MAX and E-AT achieve almost twice as high robustness in the union than the best of the models not trained for multiple-norm robustness, with only little loss in clean accuracy compared to the other robust classifiers. Moreover, for almost all threat models, these classifiers attain the highest robustness or are close to it, while the best among the single l_p-robust models varies. Therefore training for multiple norm robustness allows to generalize to some extent beyond the threat model used during training. Finally, E-AT outperforms even the PAT model which is trained wrt LPIPS and aims at generalization to unseen attacks. More details in App. C.8.

3.7. Effect of varying the radii of the l_∞- and l_1-ball on the robustness wrt l_2

We study the effect of varying the radii of the l_∞- and l_1-balls in E-AT on the robustness wrt l_2 of the resulting classifier. We fine-tune with E-AT the l_∞-robust RN-18 (trained with $\epsilon_\infty = 8/255$) for 3 epochs with different pairs $(\epsilon_\infty, \epsilon_1)$ such that $\sqrt{\epsilon_\infty \cdot \epsilon_1}$ is constant at ≈ 0.62, i.e. inducing, ignoring α, the same ϵ_2 according to Eq. (4) (convex hull of the union), but very different ϵ_2 for Eq. (2) (union only), ranging from 0.13 (16/255, 6) to 0.44 (4/255, 24). Fig. 3 shows the l_2-robustness curves computed with FAB (Croce & Hein, 2020a). For all combinations the l_2-robustness curves are similar, even for the smaller ϵ_1 (the small drops are related to the lower clean accuracy). Thus the statement of Theorem 3.1 (convex hull) formulated for linear classifiers holds empirically also for deep networks.

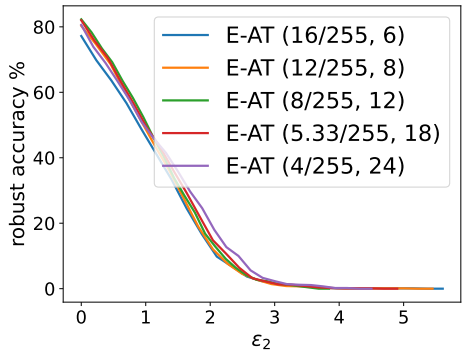

Figure 3. l_2-robustness curves of models obtained fine-tuning an l_∞-robust RN-18 with E-AT on CIFAR-10 with different combinations $(\epsilon_\infty, \epsilon_1)$ of radii of the l_∞- and l_1-ball.

4. Fine-Tuning l_p-Robust Models to Become l_q-Robust for $p \neq q$

Motivated by our results for the multiple-norm threat model we study to which extent we can fine-tune an l_p-robust model to a l_q-robust model with $p \neq q$. Again the emphasis is on an extremely short fine-tuning time so that this is much faster than full adversarial training.

CIFAR-10: We fine-tune for 3 epochs the most l_∞-robust model at $\epsilon_\infty = \frac{8}{255}$ of Gowal et al. (2020) with adversarial training wrt l_2 and l_1 with $\epsilon_2 = 0.5$ and $\epsilon_1 = 12$. Table 7 shows that fine-tuning for l_1-robustness yields 70.2% l_1-robust accuracy which is 9.9% more than the previously most robust model with a small computational cost (to be fair with a larger architecture and using extra data). Also we get a strikingly high l_2-robust accuracy for the l_2-fine-tuned model of 80.0% not far away from the 81.7% which ones gets by training for l_2 from scratch. Surprisingly, fine-tuning the l_2-robust model of Gowal et al. (2020) wrt l_1 does not outperform the l_1-robustness achieved by fine-tuning their l_∞-robust model. Interestingly, fine-tuning the l_1-robust PreAct ResNet-18 for l_2 yields a better l_2-robustness than l_2-training from scratch, see Table 14. This shows again that fine-tuning of existing models requires minimal effort and already provides strong baselines for adversarial robustness obtained by adversarial training from scratch and in some cases even outperforms them.

ImageNet: We fine-tune the l_2-RN-50 and the l_∞-robust transformers DeiT-S and XCiT-S used in the previous section to the other threat model respectively and wrt l_1. The results are in Table 7 (those for a RN-50 robust wrt l_∞ in App. D). For all models it is possible to achieve within one epoch of fine-tuning a non-trivial robustness in every threat model. In particular, note that those originally trained for the l_∞-threat model have very low robustness wrt l_1, while after fine-tuning XCiT-S achieves 33.9% robust accuracy, even higher than what is obtained from the l_2-robust RN-50

Table 6. **CIFAR-10 - Robustness against non l_p-bounded attacks:** We test the robustness of WRN-28-10 trained in different threat models against unseen types of attacks. Moreover, we add the PAT model from Laidlaw et al. (2021), which uses RN-50 as architecture.

model	*clean*	*comm. corr.*	l_0	*patches*	*frames*	*fog*	*snow*	*gabor*	*elastic*	*jpeg*	*avg.*	*union*
NAT	94.4	71.6	0.1	8.1	2.6	47.3	3.9	35.0	0.2	0.0	12.2	0.0
l_∞-AT	81.9	72.6	7.3	21.6	26.2	36.0	35.9	52.5	59.4	5.1	30.5	2.0
l_2-AT	87.8	79.2	13.2	25.0	17.7	44.9	22.1	43.5	56.6	14.0	29.6	4.5
l_1-AT	83.5	75.0	40.9	41.3	21.1	35.6	20.6	41.2	53.3	25.5	34.9	8.6
PAT	82.6	76.9	23.3	37.9	21.7	53.5	25.6	41.8	53.5	13.7	33.9	8.0
SAT	80.5	72.0	38.7	36.7	29.3	33.5	29.0	49.8	57.0	37.4	38.9	13.8
AVG	82.0	73.6	39.7	36.8	30.8	37.2	21.1	49.9	58.1	30.4	38.0	10.9
MAX	80.1	71.3	35.1	34.6	32.7	34.5	35.0	53.4	58.5	33.5	39.7	15.3
MSD	81.0	71.7	36.9	35.0	31.8	34.6	26.4	51.5	59.7	33.4	38.7	12.9
E-AT	79.1	71.3	39.5	37.7	30.5	34.8	33.4	50.2	58.6	38.7	40.4	15.9

Table 7. **Fine-tuning l_p-robust models to another threat model:** For each norm we fine-tune the most robust models wrt the other ones for 3 epochs for CIFAR-10 and 1 epoch for ImageNet and report clean and robust accuracy for all threat models. Even for the threat models where the robustness of the original model is low, the fine-tuning is sufficient to yield robustness almost at the same level of the specialized models with same architecture. For each threat model (column) we highlight in blue the model trained for the specific norm, in orange those only fine-tuned in the target norm. The values of the thresholds ϵ are the same used the multiple norms experiments.

CIFAR-10

	clean	l_∞	l_2	l_1
WRN-70-16 (Gowal et al., 2020) - l_∞ (*)				
original	90.7	65.6	66.9	8.1
+ FT wrt l_2	92.8	47.4	80.0	34.0
+ FT wrt l_1	92.4	33.9	74.7	**70.2**
WRN-70-16 (Gowal et al., 2020) - l_2 (*)				
original	94.1	43.1	81.7	34.6
+ FT wrt l_∞	92.3	58.5	73.5	11.4
+ FT wrt l_1	92.8	29.2	75.7	68.9
RN-18 (Croce & Hein, 2021) - l_1				
original	87.1	22.0	64.8	60.3
+ FT wrt l_∞	82.7	44.2	66.6	25.4
+ FT wrt l_2	88.0	31.0	69.8	39.7

ImageNet

	clean	l_∞	l_2	l_1
DeiT-S (Bai et al., 2021) - l_∞				
original	66.4	35.6	40.1	3.1
+ FT wrt l_2	66.5	31.2	46.1	9.6
+ FT wrt l_1	61.0	23.9	42.9	30.1
XCiT-S (Debenedetti, 2022) - l_∞				
original	72.8	41.7	45.3	2.7
+ FT wrt l_2	71.5	35.9	51.4	9.5
+ FT wrt l_1	65.8	25.2	47.1	**33.9**
RN-50 (Engstrom et al., 2019) - l_2				
original	58.7	25.0	40.5	14.0
+ FT wrt l_∞	59.1	31.5	40.1	7.5
+ FT wrt l_1	56.8	18.0	37.1	28.7

i.e. 28.7%. Note that compared to fine-tuning for multiple-norm robustness (see Table 3), fine-tuning XCiT-S specifically for l_2 yields almost the same robust accuracy but 3.5% better clean performance, while, when fine-tuning it for l_1, the l_1-robust accuracy is 5.5% higher. Up to our knowledge our ImageNet models are the first ones for which l_1-robustness is reported.

5. Conclusion

Based on the geometry of the l_p-balls we have introduced E-AT, a novel training scheme for multiple-norm robustness which achieves comparable adversarial robustness in the union while being significantly faster. We also show for the first time that fine-tuning can be used to transfer adversarial robustness from a single l_p-threat model to the multiple norms one, and that one can even obtain an l_q-robust classifier with a quick fine-tuning of an l_p-robust one with $p \neq q$. This yields strong baselines for future research. We have in this way generated models with SOTA performance for multiple-norm and l_1-robustness on CIFAR-10 and the first models on ImageNet with significant multiple-norm as well as l_1-robustness. This shows that fine-tuning is an excellent technique to avoid the increasingly high costs of training large adversarially robust models from scratch.

Acknowledgements

We acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A), the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645, and by DFG grant 389792660 as part of TRR 248.

References

Al-Dujaili, A. and O'Reilly, U.-M. There are no bit parts for sign bits in black-box attacks. In *ICLR*, 2020.

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

Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. Understanding deep neural networks with rectified linear unit. In *ICLR*, 2018.

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

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

Bai, Y., Mei, J., Yuille, A., and Xie, C. Are transformers more robust than CNNs? In *NeurIPS*, 2021.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *NeurIPS*, pp. 11190–11201. 2019.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018.

Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., and Wang, Z. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020.

Cheng, M., Le, T., Chen, P.-Y., Yi, J., Zhang, H., and Hsieh, C.-J. Query-efficient hard-label black-box attack: An optimization-based approach. In *ICLR*, 2019.

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

Croce, F. and Hein, M. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In *ICLR*, 2020b.

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

Croce, F. and Hein, M. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In *ICML*, 2021.

Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Croce, F., Andriushchenko, M., Singh, N. D., Flammarion, N., and Hein, M. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *AAAI*, 2022.

Dai, Z., Liu, H., Le, Q. V., and Tan, M. Coatnet: Marrying convolution and attention for all data sizes. In *NeurIPS*, 2021.

Debenedetti, E. Adversarially robust vision transformers. Master's thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), 4 2022.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021.

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

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. preprint, arXiv:1810.12715v3, 2018.

Gowal, S., Uesato, J., Qin, C., Huang, P.-S., Mann, T., and Kohli, P. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint, arXiv:1910.09338*, 2019.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593v2*, 2020.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. Improving robustness using generated data. In *NeurIPS*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *ECCV*, 2016.

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

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

Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. In *ACL*, 2018.

Jeddi, A., Shafiee, M. J., and Wong, A. A simple fine-tuning is all you need: Towards robust deep learning via adversarial fine-tuning. *arXiv preprint, arXiv:2012.13628*, 2020.

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

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

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. In *ICLR Workshop*, 2017.

Laidlaw, C., Singla, S., and Feizi, S. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2021.

Liu, S., Chen, P.-Y., Chen, X., and Hong, M. signSGD via zeroth-order oracle. In *ICLR*, 2019.

Madaan, D., Shin, J., and Hwang, S. J. Learning to generate noise for multi-attack robustness. In *ICML*, 2021.

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

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

Meunier, L., Atif, J., and Teytaud, O. Yet another but more efficient black-box adversarial attack: tiling and evolution strategies. *arXiv preprint, arXiv:1910.02244*, 2019.

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

Mosbach, M., Andriushchenko, M., Trost, T., Hein, M., and Klakow, D. Logit pairing methods can fool gradient-based attacks. In *NeurIPS 2018 Workshop on Security in Machine Learning*, 2018.

Rao, S., Stutz, D., and Schiele, B. Adversarial training against location-optimized adversarial patches. In *ECCV Workshop on the Dark and Bright Sides of Computer Vision: Challenges and Oppurtinities for Privacy and Security*, 2020.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Data augmentation can improve robustness. In *NeurIPS*, 2021.

Rony, J., Granger, E., Pedersoli, M., and Ayed, I. B. Augmented Lagrangian adversarial attacks. In *ICCV*, 2021.

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

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

Stutz, D., Hein, M., and Schiele, B. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *ICML*, 2020.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, pp. 2503–2511, 2014.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

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

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

Wong, E. and Kolter, J. Z. Learning perturbation sets for robust machine learning. In *ICLR*, 2021.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *NeurIPS*, 2018.

Wu, B., Chen, J., Cai, D., He, X., and Gu, Q. Do wider neural networks really help adversarial robustness? In *NeurIPS*, 2021.

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

Xu, C. and Yang, M. Adversarial momentum-contrastive pre-training. *arXiv preprint, arXiv:2012.13154v2*, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, pp. 87.1–87.12, 2016.

Zajac, M., Zołna, K., Rostamzadeh, N., and Pinheiro, P. O. Adversarial framing for image and video classification. In *AAAI*, pp. 10077–10078, 2019.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. *arXiv:2106.04560*, 2021.

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

Zhao, P., Liu, S., Chen, P.-Y., Hoang, N., Xu, K., Kailkhura, B., and Lin, X. On the design of black-box adversarial examples by leveraging gradient-free optimization and operator splitting method. In *ICCV*, pp. 121–130, 2019.

A. Experimental Details

For the comparison of training schemes we use for multiple-norm robustness we train PreAct ResNet-18 (He et al., 2016) with softplus activation function for 80 epochs with initial learning rate of 0.05 reduced by a factor of 10 after 70 epochs. When training WideResNet-28-10 (Zagoruyko & Komodakis, 2016) we use a cyclic schedule for the learning rate with maximum value 0.1 for 30 epochs. We use SGD optimizer with momentum of 0.9 and weight decay of $5 \cdot 10^{-4}$, batch size of 128. We use random cropping and horizontal flipping as augmentation techniques. For adversarial training of models robust wrt a single norm and with SAT and our novel scheme E-AT we use APGD with default parameters, while for the re-trained AVG, MAX, and MSD we use PGD for l_∞ (step size $\epsilon_\infty/4$) and l_2 (step size $\epsilon/3$), SLIDE (Tramèr & Boneh, 2019) for l_1 (standard parameters). For all methods we use 10 steps for the inner maximization problem in adversarial training (note that AVG and MAX repeat the attack for all threat models, and MSD tests multiple steps, thus they are more expensive). For all schemes we select the best performing checkpoint for the comparison (that is the one with the highest robustness) when using the piecewise schedule, the final checkpoint with the cyclic schedule. Moreover, we use the TRADES-XENT loss (TRADES loss (Zhang et al., 2019) with adversarial points maximizing the cross-entropy loss) since Gowal et al. (2020) show that this gives slightly better robustness on CIFAR-10 without additional data, while we use standard adversarial training (Madry et al., 2018) for PreAct ResNet-18. We train the classifiers of MNG-AC (Madaan et al., 2021) with the original code, where we set $\epsilon_1 = 12$ and rescale the step size linearly. Finally, for the runtime comparison we run each method on a single Tesla V100 GPU.

For fine-tuning on CIFAR-10 we use 3 epochs and the same setup as for full training except for the learning rate schedule, since in this case we use as initial value the best performing one in $\{0.01, 0.05\}$ (the larger value works best for the smaller networks) and reduce it by a factor of 10 at the beginning of each epoch. When the model was originally trained with extra data beyond the training set on CIFAR-10, we use the 500k images introduced by (Carmon et al., 2019) as additional data for fine-tuning, and each batch is split equally between standard and extra images, and we count 1 epoch when the whole standard training set has been used: note that in this way, using only 3 epochs not the whole pseudo-labelled dataset is exploited.

For fine-tuning on ImageNet we use 1 epoch, initial learning rate of 0.01 (0.0005 for XCiT-S), reduced by a factor of 10 every $1/3$ of training steps. We follow the setup of (Engstrom et al., 2019) for data augmentation and setting batch size to 256 (except for the RN-50 from Bai et al. (2021) and XCiT-S for which we use 192 to fit into the GPU memory) and weight decay to 10^{-4}. For adversarial training we use APGD with 5 steps for l_∞ and l_2, 15 steps for l_1 since optimizing in the l_1-ball intersected with the box constraints is more challenging, see (Croce & Hein, 2021).

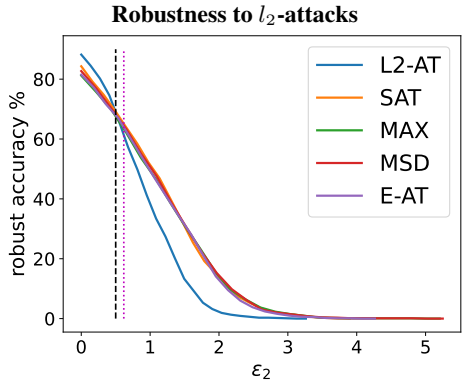

Figure 4. l_2-robustness curves of a model trained with l_2-adversarial training (AT) for $\epsilon_2 = 0.5$ and methods for multiple norm robustness on CIFAR-10. Our E-AT is expected to yield robustness at $\epsilon_2 = 0.62$. **Although l_2-attacks are not used for training, our extreme-adv. training scheme E-AT yields l_2-robustness similar to the other methods and even to the one obtained with specific l_2-adversarial training.**

B. Additional Analysis, Evaluation and Experiments for E-AT

We here analyze in more details our E-AT scheme and expand the comparison to existing methods presented above.

B.1. Robustness wrt l_2 of E-AT

To show the effect of E-AT on l_2-robustness we plot in Fig. 4 the robust accuracy wrt l_2 computed with FAB (Croce & Hein, 2020a), which minimizes the size of the perturbations, as a function of the threshold ϵ_2 for a PreAct ResNet-18 trained with either l_2-AT at $\epsilon_2 = 0.5$ or methods for multiple norm robustness (see complete results for such models in Sec. C.7). Theorem 3.1 suggests that the extreme norms training provides robustness at $\epsilon_2 \approx 0.62$, which is confirmed by the plots. Although no l_2-attack has been used during training by the E-AT model, it has robustness wrt l_2 similar to that both of the techniques for multiple norms which use l_2-perturbations at training time and of the classifier specifically trained for such threat model.

B.2. Confirmation that adversarial training with the extreme norms is sufficient: Analysis of MAX and MSD training

Both MAX and MSD schemes perform adversarial training considering all the threat models simultaneously, and

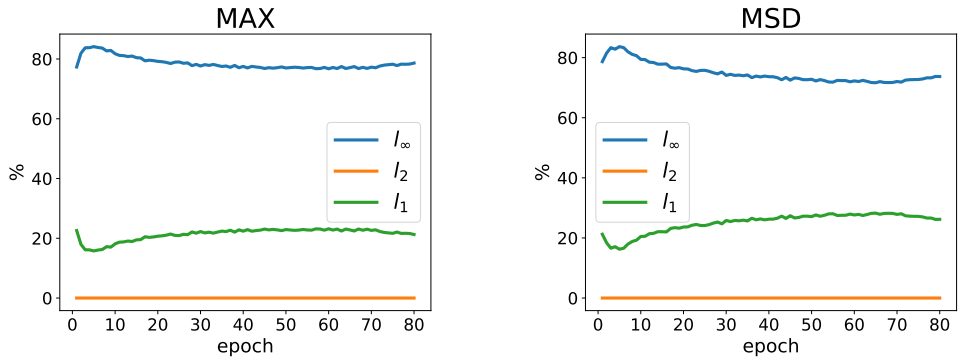

Figure 5. **CIFAR-10, ResNet18. Left:** For MAX-training we show for each epoch during training the percentage of points attaining for the indicated l_p-threat model ($p \in \{1, 2, \infty\}$) the highest loss over the the three threat models. **Right:** For MSD-training we show the percentage of steps taken wrt each threat model over epochs (note that MSD does the steepest descent step for each l_p-threat model and then realizes the one yielding maximal loss).

we analyze here their training procedure in more detail. Fig. 5 shows for the MAX strategy how many times, in percentage, for each epoch the points computed for each threat model realize the maximal loss over the three attacks (l_1, l_2, l_∞), and then are subsequently used for the update of the model. Similarly, for MSD, we show the frequency with which a step wrt each l_p-norm is taken when computing the adversarial points (average over all iterations and training points). In both cases the l_∞-threat model is the most used one with the l_1-threat model being used $3-4$ times less often. However, the l_2-threat model is almost never chosen. This empirically confirms the analysis from Theorem 3.1 which shows that training only wrt l_∞ and l_1 is (at least for a linear classifier) sufficient to achieve l_2-robustness for the chosen ϵ_2 and thus during training no extra updates wrt the l_2-threat model are necessary. This is in line with the results reported in Fig. 4 which show that for different thresholds the l_2-robustness achieved by E-AT-training is similar to that of standard l_2-training (for generating Fig. 4 we use FAB attack (Croce & Hein, 2020a) to compute robust accuracy at varying ϵ_2).

C. Fine-Tuning with E-AT for Multiple-Norm Robustness

C.1. Multiple-norm robustness via fast fine-tuning

We here show that with short fine-tuning of an l_p-robust model it is possible to achieve competitive multiple-norm robustness for any $p \in \{\infty, 2, 1\}$. Table 8 shows that all methods for robustness in the union are able to improve in 3 epochs the robustness of l_p-robust classifiers to multiple threat models on CIFAR-10. Moreover, one can see that the l_∞-threat model is the most challenging, and fine-tuning robust models wrt it yields the best robust accuracy in the union. Moreover, E-AT outperforms SAT which has roughly the same computational cost (the training time per epoch can be found in Table 14).

C.2. Fine-tuning natural models

We fine-tune with E-AT for 3 epochs PreAct ResNet-18 either naturally trained or robust wrt a single l_p-norm. Table 9 shows clean and robust accuracy for each threat model for the initial classifier and after E-AT fine-tuning: while for all robust models the fine-tuning yields values competitive with the full training for multiple norms (see Table 14), starting from a standard model leads to significantly lower both clean performance and robustness in the union of the three l_p-balls.

C.3. Runtime with large models

We reported in Table 1 and Table 14 the runtime per epoch of E-AT. For larger architectures the computational cost increases significantly, and adversarial training with the WideResNet-70-16, the largest one we consider, on CIFAR-10 takes, in our experiments, around 6100 s per epoch when using only the training set and over 10000 s if the unlabelled data is used (since twice more training steps are effectively used). Moreover, fine-tuning on ImageNet takes around 6 h per epoch for l_∞ and l_2 (5 steps of adversarial training), 20 h per epoch for l_1 (15 steps), 13.5 h per epoch with E-AT (with the architectures we consider). This shows how transferring robustness with fine-tuning might allow to obtain classifiers robust wrt different threat models fast and at much lower computational cost.

C.4. Effect of biased sampling scheme

We compare the biased sampling scheme of E-AT introduced in Eq. (5) to a uniform one, where one chooses uniformly at random which l_p, with $p \in \{1, \infty\}$, attack to use

Table 8. **CIFAR-10 - Fine-tuning for multiple-norm robustness:** We fine-tune with different methods for multiple norms for 3 epochs the PreAct ResNet-18 robust wrt individual norms (mean and standard deviation of the clean and robust accuracy over 5 seeds is reported).

model	*clean*	l_∞ ($\epsilon_\infty = \frac{8}{255}$)	l_2 ($\epsilon_2 = 0.5$)	l_1 ($\epsilon_1 = 12$)	*average*	*union*
RN-18 - l_∞-AT	83.7	48.1	59.8	7.7	38.5	7.7
+ SAT	83.5 ± 0.2	43.5 ± 0.2	68.0 ± 0.4	47.4 ± 0.5	53.0 ± 0.2	41.0 ± 0.3
+ AVG	84.2 ± 0.4	43.3 ± 0.4	68.4 ± 0.6	46.9 ± 0.6	52.9 ± 0.4	40.6 ± 0.4
+ MAX	82.2 ± 0.3	45.2 ± 0.4	67.0 ± 0.7	46.1 ± 0.4	52.8 ± 0.3	42.2 ± 0.6
+ MSD	82.2 ± 0.4	44.9 ± 0.3	67.1 ± 0.6	47.2 ± 0.6	53.0 ± 0.4	42.6 ± 0.2
+ E-AT unif.	82.6 ± 0.6	44.4 ± 0.4	68.0 ± 0.3	48.5 ± 1.0	53.6 ± 0.4	42.2 ± 0.3
+ E-AT	82.7 ± 0.4	44.3 ± 0.6	68.1 ± 0.5	48.7 ± 0.5	53.7 ± 0.3	42.2 ± 0.8
RN-18 - l_2-AT	88.2	29.8	68.6	27.5	42.0	23.1
+ SAT	86.8 ± 0.3	38.2 ± 0.4	69.6 ± 0.7	49.1 ± 0.6	52.3 ± 0.3	37.2 ± 0.4
+ AVG	86.9 ± 0.2	37.8 ± 0.4	70.2 ± 0.6	48.3 ± 0.5	52.1 ± 0.4	36.8 ± 0.4
+ MAX	85.1 ± 0.8	42.1 ± 0.3	69.4 ± 0.2	45.6 ± 0.5	52.4 ± 0.2	40.0 ± 0.3
+ MSD	85.3 ± 0.4	42.0 ± 0.7	69.2 ± 0.2	44.0 ± 0.4	51.8 ± 0.4	39.0 ± 0.5
+ E-AT unif.	85.9 ± 0.5	40.0 ± 0.6	69.4 ± 0.7	50.3 ± 0.4	53.2 ± 0.3	38.9 ± 0.8
+ E-AT	85.8 ± 0.7	40.7 ± 0.9	69.5 ± 0.5	49.5 ± 0.5	53.3 ± 0.6	39.4 ± 0.7
RN-18 - l_1-AT	87.1	22.0	64.8	60.3	49.0	22.0
+ SAT	85.1 ± 0.3	38.4 ± 0.6	68.3 ± 0.4	55.0 ± 0.7	53.9 ± 0.2	38.2 ± 0.6
+ AVG	86.0 ± 0.2	38.7 ± 0.5	68.4 ± 0.3	55.6 ± 0.6	54.2 ± 0.2	38.5 ± 0.5
+ MAX	82.2 ± 0.3	42.7 ± 0.5	66.8 ± 0.5	48.1 ± 0.4	52.5 ± 0.3	41.5 ± 0.4
+ MSD	81.8 ± 0.7	42.7 ± 0.5	66.8 ± 0.4	47.9 ± 0.6	52.5 ± 0.2	41.5 ± 0.5
+ E-AT unif.	83.5 ± 0.7	39.8 ± 0.5	68.0 ± 0.2	55.8 ± 0.5	54.6 ± 0.2	39.6 ± 0.5
+ E-AT	83.6 ± 0.6	40.5 ± 0.4	68.1 ± 0.1	55.3 ± 0.4	54.6 ± 0.2	40.3 ± 0.3

for adversarial training for each batch. This is referred to as E-AT unif. in Table 8, where one case see that the biased sampling schemes yields slightly better results when fine-tuning the l_2- and in particular the l_1-robust model: we hypothesize that, since l_∞ is the most challenging threat model, it is important to use it more often at training time when the initial model is non robust wrt l_∞. Moreover, we test E-AT unif. for full training (see Table 14).

C.5. Results using different number of epochs

Table 10 shows the effect of our E-AT-fine-tuning for different numbers of epochs on a RN-18 either robust wrt l_∞ or naturally trained. For the robust model, with longer training the clean accuracy progressively improves, as well as the robustness in the union of the threat models. In particular, the models fine-tuned for 15 epochs has robust accuracy similar to that achieved by the MAX-training (43.2% compared to 43.3%, see Table 14), while still being significantly faster even considering the training time of the initial model. When starting from a standard model, E-AT fine-tuning leads to a large drop in clean accuracy while the robustness in the union remains lower than what can be achieved with robust models, even when using 15 epochs. Similarly, we fine-tune for 3 epochs, instead of 1 as done above, the l_2-robust model on ImageNet from Engstrom et al. (2019) with our E-AT. Table 11 shows that the longer fine-tuning improves all the performance metrics between 0.6% and 1.3%. Moreover, since we use a single epoch of fine-tuning on ImageNet, we test its effect on CIFAR-10. In Table 12 we fine-tune with E-AT models adversarially trained wrt a single norm for 1 epoch: this is sufficient to significantly increase the robustness in the union of the threat models, which gets close to that obtained with the standard 3 epochs (differences are in the range 0.6% to 2.5%). In particular, the l_∞-robust classifier is again the most suitable for the fine-tuning, since it has been trained in the most challenging threat model.

C.6. Fine-tuning perceptually robust models

We test the effect of E-AT fine-tuning on a model trained to be robust to perturbations which are aligned with human perception. In particular, we use the classifier obtained with perceptual adversarial training (PAT), i.e. wrt the LPIPS metric, from Laidlaw et al. (2021), and compare it to two models with the same architecture (ResNet-50) adversarially trained wrt l_∞ and l_2. Table 13 shows the robustness in every threat model for the original models and those obtained with 3 epochs of E-AT fine-tuning. The PAT classifier has initially the highest robustness in the union, confirming the obersevation of Laidlaw et al. (2021) that PAT provides some robustness to unseen attacks. After fine-tuning, all three models achieve similar worst-case robustness, with the classifier originally l_∞-robust being slightly better. This shows that our E-AT fine-tuning is effective even when applied to models adversarially trained not wrt an l_p-norm.

Table 9. **CIFAR-10 - 3 epochs of fine-tuning with E-AT :** We report the results of fine-tuning PreAct ResNet-18 models to become robust wrt the union of the threat models. Fine-tuning any l_p-robust model leads to competitive clean and robust accuracy to full training, differently from using a naturally trained model.

model	*clean*		l_∞ ($\epsilon_\infty = \frac{8}{255}$)		l_2 ($\epsilon_2 = 0.5$)		l_1 ($\epsilon_1 = 12$)		*union*	
RN-18 - standard	94.4		0.0		0.0		0.0		0.0	
+ FT	66.6	-27.8	29.9	29.9	50.1	50.1	38.5	38.5	29.8	29.8
RN-18 - l_∞	83.7		48.1		59.8		7.7		7.7	
+ FT	82.3	-1.4	43.4	-4.7	68.0	8.2	48.0	40.3	41.2	33.5
RN-18 - l_2	88.2		29.8		68.6		27.5		23.1	
+ FT	85.4	-2.8	40.6	10.8	69.8	1.2	48.7	21.2	39.1	16.0
RN-18 - l_1	87.1		22.0		64.8		60.3		22.0	
+ FT	83.5	-3.6	40.3	18.3	68.1	3.3	55.7	-4.6	40.1	18.1

Table 10. **CIFAR-10 - Fine-tuning for more epochs:** We show the effect of fine-tuning for different number of epochs (3 is the standard we use) the PreAct ResNet-18 (standard or robust wrt l_∞).

model	*clean*		l_∞ ($\epsilon_\infty = \frac{8}{255}$)		l_2 ($\epsilon_2 = 0.5$)		l_1 ($\epsilon_1 = 12$)		*union*	
RN-18 - l_∞	83.7		48.1		59.8		7.7		7.7	
+ 3 epochs FT	82.3	-1.4	43.4	-4.7	68.0	8.2	48.0	40.3	41.2	33.5
+ 5 epochs FT	83.0	-0.7	45.2	-2.9	68.8	9.0	50.1	42.4	43.1	35.4
+ 7 epochs FT	83.1	-0.6	44.6	-3.5	68.7	8.9	50.4	42.7	42.6	34.9
+ 10 epochs FT	84.0	0.3	44.9	-3.2	69.2	9.4	51.0	43.3	42.8	35.1
+ 15 epochs FT	84.6	0.9	44.9	-3.2	69.5	9.7	52.1	44.4	43.2	35.5
RN-18 - standard	94.4		0.0		0.0		0.0		0.0	
+ 3 epochs FT	66.6	-27.8	29.9	29.9	50.1	50.1	38.5	38.5	29.8	29.8
+ 5 epochs FT	70.6	-23.8	33.8	33.8	55.2	55.2	44.4	44.4	33.4	33.4
+ 7 epochs FT	72.1	-22.3	36.1	36.1	58.9	58.9	45.9	45.9	35.6	35.6
+ 10 epochs FT	75.4	-19.0	37.1	37.1	61.0	61.0	47.9	47.9	36.9	36.9
+ 15 epochs FT	76.0	-18.4	40.2	40.2	61.6	61.6	49.2	49.2	40.0	40.0

C.7. Full training for multiple norm robustness

We report in Table 14 the detailed results of training for multiple norms robustness from random initialization with PreAct ResNet-18 (He et al., 2016) and WideResNet-28-10 (Zagoruyko & Komodakis, 2016). For both architectures MAX and MSD attain the best results in the union. However, E-AT is only slightly worse while being 2-3 times less expensive. We include MNG-AC from Madaan et al. (2021) for the larger architecture as that is used also in the original paper. Moreover, since it has PreAct ResNet-18 as architecture, we additionally include the original MSD model of Maini et al. (2020), marked with (*), which obtains 41.4% robustness in the union, whereas with our reimplementation we get 43.9%, improving their results significantly. Note that they reported in their paper 47.0% robustness in the union, while our APGD-based evaluation reduces this to 41.4% which shows that our robustness evaluation is significantly stronger. Finally, we add the results of E-AT without biased sampling scheme (E-AT unif.): this achieves worse results than E-AT on the WRN-28-10, showing the effectiveness of the biased sampling.

C.8. Robustness against unseen non l_p-bounded attacks

To test robustness against l_0-attacks we use Sparse-RS (Croce et al., 2022) with a budget of 18 pixels and 10k queries. We adopt patches of size 5×5 pixels, optimized with the PGD-based attack from Rao et al. (2020) (without constraints on the position of the patch on the images), and frames of width 1 pixel (Zajac et al., 2019), again optimized with PGD: in both cases we use 10 random restarts of 100 iterations. For the adversarial corruptions we use the original implementation (Kang et al., 2019b) with 100 iterations and search for a budget ϵ for which the models show different levels of robustness (in details, for fog $\epsilon = 128$, snow $\epsilon = 0.5$, Gabor noise $\epsilon = 60$, elastic $\epsilon = 0.125$, l_∞-JPEG $\epsilon = 0.25$). Finally, we average the classification accuracy over the 5 severities of the common corruptions. All the statistics are on 1000 test points.

Table 11. **ImageNet - Fine-tuning for more epochs:** We fine-tune the l_2-robust model from (Engstrom et al., 2019) for either 1 or 3 epochs with our E-AT scheme.

model	*clean*		l_∞ ($\epsilon_\infty = \frac{4}{255}$)		l_2 ($\epsilon_2 = 2$)		l_1 ($\epsilon_1 = 255$)		union	
RN-50 - l_2 (Engstrom et al., 2019)	58.7		25.0		40.5		14.0		13.5	
+ 1 epochs FT	56.7	-2.0	26.7	1.7	41.0	0.5	25.4	11.4	23.1	9.6
+ 3 epochs FT	57.4	-1.3	27.8	2.8	41.6	1.1	26.7	12.7	23.7	10.2

Table 12. **CIFAR-10 - Fine-tuning for 1 epoch:** We show the effect of fine-tuning for a single (compared to the standard 3) models robust wrt a single norm.

model	*clean*		l_∞ ($\epsilon_\infty = \frac{8}{255}$)		l_2 ($\epsilon_2 = 0.5$)		l_1 ($\epsilon_1 = 12$)		*union*	
RN-50 (Engstrom et al., 2019) - l_∞	88.7		50.9		59.4		5.0		5.0	
+ 1 epochs FT	84.8	-3.9	46.6	-4.3	68.3	8.9	47.2	42.2	42.8	37.8
+ 3 epochs FT	86.2	-2.5	46.0	-4.9	70.1	10.7	49.2	44.2	43.4	38.4
RN-50 (Engstrom et al., 2019) - l_2	91.5		29.7		70.3		27.0		23.0	
+ 1 epochs FT	85.9	-5.6	41.8	12.1	69.6	-0.7	47.6	20.6	39.7	16.7
+ 3 epochs FT	87.8	-3.7	43.1	13.4	70.8	0.5	50.2	23.2	41.7	18.7
RN-18 (Croce & Hein, 2021) - l_1	87.1		22.0		64.8		60.3		22.0	
+ 1 epochs FT	78.9	-8.2	37.7	15.7	62.9	-1.9	51.3	-9.0	37.6	15.6
+ 3 epochs FT	83.5	-3.6	40.3	18.3	68.1	3.3	55.7	-4.6	40.1	18.1

D. Fine-tuning to a different threat model: additional results

Table 15. **Fine-tuning l_p-robust models to another threat model:** we fine-tune for 1 epoch models trained wrt l_p with adversarial training wrt another l_q.

	ImageNet			
	clean	l_∞	l_2	l_1
RN-50 (Engstrom et al., 2019) - l_∞				
original	62.9	29.8	17.7	0.0
+ FT wrt l_2	62.9	25.5	41.5	8.4
+ FT wrt l_1	57.7	18.0	37.6	27.4

We additionally report in Table 15 the results of fine-tuning for 1 epoch the RN-50 (Engstrom et al., 2019) robust wrt l_∞ with adversarial training wrt others l_q-threat models: similarly to the other models this is sufficient to obtain significant robustness in the new threat model.

E. Experiments on MNIST

We further test the different techniques on the MNIST dataset. We use the same CNN of Maini et al. (2020) as architecture and $\epsilon_\infty = 0.3$, $\epsilon_2 = 2$ and $\epsilon_1 = 10$ as thresholds at which evaluating robustness, as done by Maini et al. (2020). We note that while it is an easier dataset, MNIST is challenging when it comes to adversarial training since it presents unexpected phenomena: e.g. Tramèr & Boneh (2019) noted that l_∞-adversarial training induces gradient obfuscation when using attack wrt l_2 and l_1, and both Tramèr & Boneh (2019); Maini et al. (2020) had to use many PGD-steps (up to 100), and Tramèr & Boneh (2019) even a ramp-up schedule for the ϵ during training. While other modifications to the training setup might be beneficial for some or all the methods, we just increased the number of APGD-steps to 50 for l_1 (see more details below). In Table 16 we compare E-AT to SAT, AVG, MAX and MSD (for the last three we use the models provided by Maini et al. (2020)). First, E-AT outperforms the available classifiers trained with AVG, MAX and MSD, meaning that even on MNIST it is a strong baseline. However, in this case, SAT, which trains on all types of perturbations, achieves better results than E-AT on average: E-AT has higher variance over runs but the best run (over multiple seeds) is close to the best one of SAT in terms of robustness in the union (55.3% vs 54.4%). Interestingly, SAT has much higher robustness wrt l_2 compared to E-AT, but this is somehow expected since Eq. (3) would "predict" robustness for E-AT at $\epsilon_2 \approx 1.7$ while $\epsilon_2 = 2$ is used for testing, and this is precise only for linear models. Thus the slightly worse performance of E-AT compared to SAT for the chosen radii of the threat models is to be expected from our geometric analysis.

Moreover, since we have shown that fine-tuning an l_p-robust model with E-AT yields high multiple norms robustness, and given that E-AT from random initialization is weak mostly wrt l_2, we fine-tune the l_2-AT classifier with E-AT. This, with just 3 or 5 epochs, significantly outperforms SAT (up to +2.3% robustness in the union), while preserving l_2-robustness. In total, we improve the previous SOTA for multiple-norm robustness for MNIST from

Table 13. **CIFAR-10 - E-AT fine-tuning of perceptually robust models:** We use E-AT to fine-tune for 3 epochs the PAT model, robust wrt LPIPS, and compare to fine-tuning l_p-robust models.

model	*clean*		l_∞ ($\epsilon_\infty = \frac{8}{255}$)		l_2 ($\epsilon_2 = 0.5$)		l_1 ($\epsilon_1 = 12$)		*union*	
RN-50 - l_∞	88.7		50.9		59.4		5.0		5.0	
(Engstrom et al., 2019) + FT	86.2	-2.5	46.0	-4.9	70.1	10.7	49.2	44.2	43.4	38.4
RN-50 - l_2	91.5		29.7		70.3		27.0		23.0	
(Engstrom et al., 2019) + FT	87.8	-3.7	43.1	13.4	70.8	0.5	50.2	23.2	41.7	18.7
RN-50 - PAT	82.6		31.1		62.4		33.6		27.7	
(Laidlaw et al., 2021) + FT	83.7	1.1	43.7	12.6	68.5	6.1	50.7	17.1	42.3	14.6

48.7% (MSD) to 57.5% (E-AT fine-tuning of an l_2-robust model with 5 epochs) which is a significant improvement. Note that in this case we increase the radii ϵ_∞ and ϵ_1 to 0.33 and 14 respectively to preserve the l_2-robustness: in fact, with such values Eq. (3) yields $\epsilon_2 \approx 2.16$. However, training from random initialization, in the standard setup, with the larger thresholds leads to worse robustness in the union. We hypothesize that this is due to the increased difficulty of the task to learn: it is known that even single norm adversarial training is problematic when increasing the value of ϵ (Ding et al., 2020).

Experimental details: For training we use 30 epochs with cyclic learning rate (maximum value 0.05, also used for fine-tuning) and no data augmentation (other settings as for CIFAR-10). As mentioned, we use in adversarial training for multiple norms (SAT, E-AT unif. and E-AT) 50 steps of APGD for l_1, and to reduce the training cost we decrease to 5 those for l_∞. Moreover, for training, in l_1-APGD we increase the parameter to control the initial sparsity of the updates to 0.1 (default is 0.05). For evaluation, we use the full AutoAttack, since on MNIST FAB (Croce & Hein, 2020a) and Square Attack (Andriushchenko et al., 2020) are at times stronger than PGD-based attacks, as shown in the original papers.

Table 14. **CIFAR-10 - Comparison of different full training schemes:** We compare methods for multiple-norm robustness on training from random initialization.

method	*clean*	l_∞ ($\epsilon_\infty = \frac{8}{255}$)	l_2 ($\epsilon_2 = 0.5$)	l_1 ($\epsilon_1 = 12$)	*average*	*union*	*time/epoch*
WideResNet-28-10							
l_∞-AT	82.6 ± 0.5	52.0 ± 0.7	59.7 ± 0.2	9.1 ± 0.2	40.3 ± 0.4	9.1 ± 0.2	922 s
l_2-AT	88.2 ± 0.4	35.9 ± 0.2	70.9 ± 0.4	36.1 ± 0.2	47.6 ± 0.2	31.3 ± 0.2	928 s
l_1-AT	83.7 ± 0.2	30.7 ± 0.7	65.1 ± 0.5	61.6 ± 0.3	52.5 ± 0.5	30.7 ± 0.7	949 s
SAT	80.5 ± 0.6	45.9 ± 0.5	66.7 ± 0.3	55.9 ± 0.5	56.2 ± 0.4	45.7 ± 0.6	925 s
MNG-AC	81.3 ± 0.3	43.5 ± 0.7	66.9 ± 0.2	57.6 ± 0.8	56.0 ± 0.4	43.3 ± 0.7	1500 s
AVG	82.5 ± 0.4	45.4 ± 1.1	68.0 ± 0.9	55.0 ± 0.2	56.1 ± 0.7	45.1 ± 1.1	2771 s
MAX	79.9 ± 0.1	48.4 ± 0.7	65.3 ± 0.3	50.2 ± 0.6	54.6 ± 0.5	47.4 ± 0.8	2479 s
MSD	80.6 ± 0.3	48.0 ± 0.2	65.6 ± 0.3	51.7 ± 0.4	55.1 ± 0.2	46.9 ± 0.1	1554 s
E-AT unif.	79.7 ± 0.2	45.4 ± 0.5	66.0 ± 0.5	55.6 ± 0.5	55.7 ± 0.4	45.1 ± 0.7	939 s
E-AT	79.9 ± 0.7	46.6 ± 0.2	66.2 ± 0.6	56.0 ± 0.4	56.3 ± 0.3	46.4 ± 0.3	921 s
PreAct ResNet-18							
l_∞-AT	84.0 ± 0.3	48.1 ± 0.2	59.7 ± 0.4	6.3 ± 1.0	38.0 ± 0.3	6.3 ± 1.0	151 s
l_2-AT	88.9 ± 0.6	27.3 ± 1.8	68.7 ± 0.1	25.3 ± 1.6	40.5 ± 1.1	20.9 ± 1.8	153 s
l_1-AT	85.9 ± 1.1	22.1 ± 0.1	64.9 ± 0.5	59.5 ± 0.8	48.8 ± 0.4	22.1 ± 0.1	195 s
SAT	83.9 ± 0.8	40.7 ± 0.7	68.0 ± 0.4	54.0 ± 1.2	54.2 ± 0.8	40.4 ± 0.7	161 s
AVG	84.6 ± 0.3	40.8 ± 0.7	68.4 ± 0.7	52.1 ± 0.4	53.8 ± 0.1	40.1 ± 0.8	479 s
MAX	80.4 ± 0.5	45.7 ± 0.9	66.0 ± 0.4	48.6 ± 0.8	53.4 ± 0.5	44.0 ± 0.7	466 s
MSD (*)	82.1	43.1	64.5	46.5	51.4	41.4	-
MSD	81.1 ± 1.1	44.9 ± 0.6	65.9 ± 0.6	49.5 ± 1.2	53.4 ± 0.4	43.9 ± 0.8	306 s
E-AT unif.	82.2 ± 1.8	42.7 ± 0.7	67.5 ± 0.5	53.6 ± 0.1	54.6 ± 0.2	42.4 ± 0.6	163 s
E-AT	81.9 ± 1.4	43.0 ± 0.9	66.4 ± 0.6	53.0 ± 0.3	54.2 ± 0.4	42.4 ± 0.7	160 s

Table 16. **MNIST - Comparison of full training schemes and fine-tuning with E-AT for multiple norm robustness:** We train classifier (architecture as in Maini et al. (2020)) on MNIST with different training scheme. For SAT and E-AT we report, together with the statistics over multiple random seeds, the results of the best run. Additionally, we show the results of fine-tuning the l_2-AT model with E-AT for different numbers of epochs, which achieves the best results. (*) AVG, MAX and MSD classifiers are those provided by Maini et al. (2020).

model	*clean*	l_∞ ($\epsilon_\infty = 0.3$)	l_2 ($\epsilon_2 = 2$)	l_1 ($\epsilon_1 = 10$)	*union*
l_∞-AT	98.9 ± 0.12	90.0 ± 0.45	8.4 ± 1.49	6.0 ± 0.65	4.2 ± 0.73
l_2-AT	98.8 ± 0.17	0.0 ± 0.05	70.7 ± 0.26	59.1 ± 0.37	0.0 ± 0.05
l_1-AT	98.8 ± 0.09	0.0 ± 0.00	45.8 ± 0.42	77.2 ± 0.14	0.0 ± 0.00
SAT	98.6 ± 0.17	62.0 ± 0.86	65.7 ± 1.69	61.3 ± 1.29	53.9 ± 1.25
AVG (*)	99.1	58.6	60.8	22.5	21.1
MAX (*)	98.6	39.4	59.9	25.6	20.3
MSD (*)	98.2	63.7	66.6	51.0	48.7
E-AT unif.	98.8 ± 0.12	67.1 ± 3.03	50.2 ± 4.93	62.0 ± 4.59	45.9 ± 4.43
E-AT	98.7 ± 0.12	69.0 ± 3.66	56.4 ± 3.94	61.1 ± 4.03	50.6 ± 3.89
l_2-AT + E-AT (3 ep.)	96.9 ± 0.32	57.5 ± 0.92	67.8 ± 0.58	62.0 ± 1.16	54.6 ± 0.59
l_2-AT + E-AT (5 ep.)	97.4 ± 0.21	60.6 ± 2.19	65.9 ± 0.56	63.8 ± 0.93	56.2 ± 1.16
best run					
SAT	98.8	62.7	67.2	62.6	55.3
E-AT unif.	98.9	67.4	56.3	63.6	51.6
E-AT	98.8	71.0	58.4	62.0	54.4
l_2-AT + E-AT (3 ep.)	97.3	58.0	67.9	62.9	55.7
l_2-AT + E-AT (5 ep.)	97.5	61.6	66.2	64.0	57.5