Adversarial Robustness Against the Union of Multiple Perturbation Models

Pratyush Maini [1] **Eric Wong** [2] **J. Zico Kolter** [3 4]

Abstract

Owing to the susceptibility of deep learning systems to adversarial attacks, there has been a great deal of work in developing (both empirically and certifiably) robust classifiers. While most work has defended against a single type of attack, recent work has looked at defending against multiple perturbation models using simple aggregations of multiple attacks. However, these methods can be difficult to tune, and can easily result in imbalanced degrees of robustness to individual perturbation models, resulting in a sub-optimal worst-case loss over the union. In this work, we develop a natural generalization of the standard PGD-based procedure to incorporate multiple perturbation models into a single attack, by taking the worst-case over all steepest descent directions. This approach has the advantage of directly converging upon a trade-off between different perturbation models which minimizes the worst-case performance over the union. With this approach, we are able to train standard architectures which are simultaneously robust against ℓ_∞, ℓ_2, and ℓ_1 attacks, outperforming past approaches on the MNIST and CIFAR10 datasets and achieving adversarial accuracy of 47.0% against the union of (ℓ_∞, ℓ_2, ℓ_1) perturbations with radius = (0.03, 0.5, 12) on the latter, improving upon previous approaches which achieve 40.6% accuracy.

1. Introduction

Machine learning algorithms have been shown to be susceptible to *adversarial examples* (Szegedy et al., 2014) through the existence of data points which can be adversarially perturbed to be misclassified, but are "close enough" to the original example to be imperceptible to the human eye. Methods to generate adversarial examples, or "attacks", typically rely on gradient information, and most commonly use variations of projected gradient descent (PGD) to maximize the loss within a small perturbation region, usually referred to as the adversary's perturbation model. A number of heuristic defenses have been proposed to defend against this phenomenon, e.g. distillation (Papernot et al., 2016) or logit-pairing (Kannan et al., 2018). However, as time goes by, the original robustness claims of these defenses typically don't hold up to more advanced adversaries or more thorough attacks (Carlini & Wagner, 2017; Engstrom et al., 2018; Mosbach et al., 2018). One heuristic defense that seems to have survived (to this day) is to use *adversarial training* against a PGD adversary (Madry et al., 2018), and remains quite popular due to its simplicity and apparent empirical robustness. The method continues to perform well in empirical benchmarks even when compared to recent work in provable defenses, though it comes with no formal guarantees.

While adversarial training has primarily been used to learn models robust to a single perturbation model, some recent work has looked at empirically defending against multiple perturbation models simultaneously. Schott et al. (2019) proposed a variational autoencoder based architecture to learn an MNIST classifier which was robust to multiple perturbation models, while Tramèr & Boneh (2019) proposed simple aggregations of different adversaries for adversarial training against multiple perturbation models.

While these approaches can achieve varying degrees of robustness to the considered adversarial perturbation models, in practice it is quite difficult to achieve an optimal trade-off which minimizes the worst-case error in the union of perturbation models. Rather, these approaches tend to converge to suboptimal local minima, resulting in a model that is highly robust to certain perturbation models while failing to defend against others, and the robust performance can often vary substantially across datasets. This results in poor and unpredictable robust performance against the worst-case attack, and indicates that the optimization procedure actually fails to minimize the worst-case loss in the union of the perturbation models.

[1]Department of Computer Science and Engineering, IIT Delhi, India [2]Machine Learning Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [3]Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA [4]Bosch Center for Artificial Intelligence, Pittsburgh, Pennsylvania, USA. Correspondence to: Pratyush Maini <pratyush.maini@gmail.com>.

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

We believe that achieving robustness to multiple perturbations is an essential step towards the eventual objective of universal robustness and our work further motivates research in this area. In this work, we make three main contributions towards learning models which are adversarially robust to multiple perturbation models. First, we demonstrate the inconsistency of previous approaches across datasets, showing that they converge to suboptimal tradeoffs which may not actually minimize the robust objective of worst-case loss over the combined perturbation model. Second, we propose a modified PGD-based algorithm called "Multi Steepest Descent" (MSD) for adversarial training, which naturally incorporates different gradient-based perturbation models into a single unified adversary to directly solve the inner optimization problem of finding the worst-case loss. Third, we show empirically that our approach improves upon past work by finding trade-offs between the perturbation models which significantly improve the worst-case robust performance against multiple perturbation models on both MNIST and CIFAR10. Specifically, on MNIST, our model achieves 58.4% adversarial accuracy against the union of all three attacks $(\ell_\infty, \ell_2, \ell_1)$ for $\epsilon = (0.3, 2.0, 10)$ respectively, substantially improving upon both the ABS models and also simpler aggregations of multiple adversarial attacks, which at best achieve 42.1% robust accuracy. Additionally, unlike past work, we also train a CIFAR10 model against the union of all three attacks $(\ell_\infty, \ell_2, \ell_1)$, which achieves 47.0% adversarial accuracy for $\epsilon = (0.03, 0.5, 12)$ and improves upon the simpler aggregations of multiple attacks which can achieve 40.6% robust accuracy under this perturbation model. In all cases, we find that our approach is able to consistently reduce the worst-case error under the unified perturbation model. Code for reproducing all the results can be found at: https://github.com/locuslab/robust_union.

2. Related work

After their original introduction, one of the first widely-considered attacks against deep networks had been the Fast Gradient Sign Method (Goodfellow et al., 2015), which showed that a single, small step in the direction of the sign of the gradient could sometimes fool machine learning classifiers. While this worked to some degree, the Basic Iterative Method (Kurakin et al., 2017) (now typically referred to as the PGD attack) was significantly more successful at creating adversarial examples, and now lies at the core of many papers. Since then, a number of improvements and adaptations have been made to the base PGD algorithm to overcome heuristic defenses and create stronger adversaries. Adversarial attacks were thought to be safe under realistic transformations (Lu et al., 2017) until the attack was augmented to be robust to them (Athalye et al., 2018b). Adversarial examples generated using PGD on surrogate models can transfer to black box models (Papernot et al., 2017). Utilizing core optimization techniques such as momentum can greatly improve the attack success rate and transferability, and was the winner of the NIPS 2017 competition on adversarial examples (Dong et al., 2018). Uesato et al. (2018) showed that a number of ImageNet defenses were not as robust as originally thought, and Athalye et al. (2018a) defeated many of the heuristic defenses submitted to ICLR 2018 shortly after the reviewing cycle ended, all with stronger PGD variations.

Throughout this cycle of attack and defense, some defenses were uncovered that remain robust to this day. The aforementioned PGD attack, and the related defense known as adversarial training with a PGD adversary (which incorporates PGD-attacked examples into the training process) has so far remained empirically robust (Madry et al., 2018). Verification methods to certify robustness properties of networks were developed, utilizing techniques such as SMT solvers (Katz et al., 2017), SDP relaxations (Raghunathan et al., 2018b), and mixed-integer linear programming (Tjeng et al., 2019), the last of which has recently been successfully scaled to reasonably sized networks. Other work has folded verification into the training process to create provably robust networks (Wong & Kolter, 2018; Raghunathan et al., 2018a), some of which have also been scaled to even larger networks (Wong et al., 2018; Mirman et al., 2018; Gowal et al., 2018). Although some of these could potentially be extended to apply to multiple perturbations simultaneously, most of these works have focused primarily on defending against and verifying only a *single* type of adversarial perturbation at a time.

Last but most relevant to this work are adversarial defenses that are robust against multiple types of attacks simultaneously. Schott et al. (2019) used multiple variational autoencoders to construct a complex architecture called analysis by synthesis (ABS) for the MNIST dataset that is not as easily attacked by ℓ_∞, ℓ_2, and ℓ_0 adversaries. The ABS model has two variations, one which is robust to ℓ_0 and ℓ_2 but not ℓ_∞ attacks and other which is robust to ℓ_∞ and ℓ_0 but not ℓ_2 attacks. Similarly, Tramèr & Boneh (2019) study the theoretical and empirical trade-offs of adversarial robustness in various settings when defending against aggregations of multiple adversaries, however they find that the ℓ_∞ perturbation model interferes with other perturbation models on MNIST (ℓ_1 and ℓ_2) and they study a rotation and translation adversary instead of an ℓ_2 adversary for CIFAR10. Croce & Hein (2019) propose a provable adversarial defense against all ℓ_p norms for $p \geq 1$ using a regularization term. Finally, while not studied as a defense, Kang et al. (2019) study the transferability of adversarial robustness between models trained against different perturbation models, while Jordan et al. (2019) study combination attacks with low perceptual distortion.

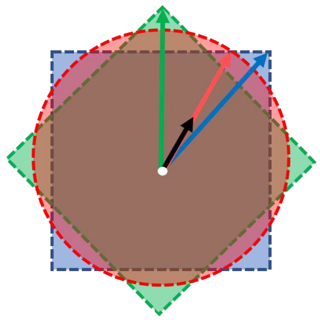

Figure 1: A depiction of the steepest descent directions for ℓ_∞, ℓ_2, and ℓ_1 norms. The gradient is the black arrow, and the α radius step sizes and their corresponding steepest descent directions ℓ_∞, ℓ_2, and ℓ_1 are shown in blue, red, and green respectively.

3. Overview of adversarial training

Adversarial training is an approach to learn a classifier which minimizes the worst-case loss within some perturbation region (the perturbation model). Specifically, for some network f_θ parameterized by θ, loss function ℓ, and training data $\{x_i, y_i\}_{i=1\ldots n}$, the robust optimization problem of minimizing the worst-case loss within ℓ_p norm-bounded perturbations with radius ϵ is

$$\min_\theta \sum_i \max_{\delta \in \Delta_{p,\epsilon}} \ell(f_\theta(x_i + \delta), y_i), \tag{1}$$

where $\Delta_{p,\epsilon} = \{\delta : \|\delta\|_p \leq \epsilon\}$ is the ℓ_p ball with radius ϵ centered around the origin. To simplify the notation, we will abbreviate $\ell(f_\theta(x+\delta), y) = \ell(x+\delta; \theta)$.

3.1. Solving the inner optimization problem

We first look at solving the inner maximization problem, namely

$$\max_{\delta \in \Delta_{p,\epsilon}} \ell(x + \delta; \theta). \tag{2}$$

This is the problem addressed by the "attackers" in the space of adversarial examples, hoping that the classifier can be tricked by the optimal perturbed image, $x + \delta^\star$. Typical solutions solve this problem by running a form of projected gradient descent, which iteratively takes steps in the gradient direction to increase the loss followed by a projection step back onto the feasible region, the ℓ_p ball. Since the gradients at the example points themselves (i.e., $\delta = 0$) are typically too small to make efficient progress, more commonly used is a variation called *projected steepest descent*.

Steepest descent For some norm $\|\cdot\|_p$ and step size α, the direction of steepest descent on the loss function ℓ for a perturbation δ is

$$v_p(\delta) = \arg\max_{\|v\|_p \leq \alpha} v^T \nabla \ell(x + \delta; \theta). \tag{3}$$

Then, instead of taking gradient steps, steepest descent uses the following iteration

$$\delta^{(t+1)} = \delta^{(t)} + v_p(\delta^{(t)}). \tag{4}$$

In practice, the norm used in steepest descent is typically taken to be the same ℓ_p norm used to define the perturbation region $\Delta_{p,\epsilon}$. However, depending on the norm used, the direction of steepest descent can be quite different from the actual gradient (Figure 1). Note that a single steepest descent step with respect to the ℓ_∞ norm reduces to $v_\infty(x) = \alpha \cdot \text{sign}(\nabla \ell(x + \delta; \theta))$, better known in the adversarial examples literature as the Fast Gradient Sign Method (Goodfellow et al., 2015).

Projections The second component of projected steepest descent for adversarial examples is to project iterates back onto the ℓ_p ball around x. Specifically, projected steepest descent performs the following iteration

$$\delta^{(t+1)} = \mathcal{P}_{\Delta_{p,\epsilon}}\left(\delta^{(t)} + v_p(\delta^{(t)})\right) \tag{5}$$

where $\mathcal{P}_{\Delta_{p,\epsilon}}(\delta)$ is the standard projection operator that finds the perturbation $\delta' \in \Delta_{p,\epsilon}$ that is "closest" in Euclidean space to the input δ, defined as

$$\mathcal{P}_{\Delta_{p,\epsilon}}(\delta) = \arg\min_{\delta' \in \Delta_{p,\epsilon}} \|\delta - \delta'\|_2^2. \tag{6}$$

Visually, a depiction of this procedure (steepest descent followed by a projection onto the perturbation region) for an ℓ_2 adversary can be found in Figure 1. If we instead project the steepest descent directions with respect to the ℓ_∞ norm onto the ℓ_∞ ball of allowable perturbations, the projected steepest descent iteration reduces to

$$\begin{aligned} \delta^{(t+1)} &= P_{\Delta_{\infty,\epsilon}}(\delta^{(t)} + v_\infty(\delta^{(t)})) \\ &= \operatorname*{clip}_{[-\epsilon,\epsilon]}\left(\delta^{(t)} + \alpha \cdot \text{sign}(\nabla \ell(x + \delta^{(t)}; \theta))\right) \end{aligned} \tag{7}$$

where $\text{clip}_{[-\epsilon,+\epsilon]}$ "clips" the input to lie within the range $[-\epsilon, \epsilon]$. This is exactly the Basic Iterative Method used in Kurakin et al. (2017), typically referred to in the literature as an ℓ_∞ PGD adversary.

3.2. Solving the outer optimization problem

We next look at how to solve the outer optimization problem, or the problem of learning the weights θ that minimize the loss of our classifier. While many approaches have been proposed in the literature, we will focus on a heuristic called adversarial training, which has generally worked well in practice.

Adversarial training Although solving the min-max optimization problem may seem daunting, a classical result known as Danskin's theorem (Danskin, 1967) says that the gradient of a maximization problem is equal to the gradient of the objective evaluated at the optimum. For learning models that minimize the robust optimization problem from Equation (1), this means that

$$\nabla_\theta \left(\sum_i \max_{\delta \in \Delta_{p,\epsilon}} \ell(x_i + \delta; \theta) \right) = \sum_i \nabla_\theta \ell(x_i + \delta^*(x_i); \theta) \tag{8}$$

where $\delta^*(x_i) = \arg\max_{\delta \in \Delta_{p,\epsilon}} \ell(x_i+\delta; \theta)$. In other words, this means that in order to backpropagate through the robust optimization problem, we can solve the inner maximization and backpropagate through the solution. Adversarial training does this by empirically maximizing the inner problem with a PGD adversary. Note that since the inner problem is not solved exactly, Danskin's theorem does not strictly apply. However, in practice, adversarial training does seem to provide good empirical robustness, at least when evaluated against the ℓ_p perturbation model it was trained against.

4. Adversarial training for multiple perturbation models

We can now consider the core of this work, adversarial training procedures against multiple perturbation models. More formally, let $\mathcal{S}$ represent a set of perturbation models, such that $p \in \mathcal{S}$ corresponds to the ℓ_p perturbation model $\Delta_{p,\epsilon}$, and let $\Delta_{\mathcal{S}} = \bigcup_{p \in \mathcal{S}} \Delta_{p,\epsilon}$ be the union of all perturbation models in $\mathcal{S}$. Note that the ϵ chosen for each ball is *not* typically the same, but we still use the same notation ϵ for simplicity, since the context will always make clear which ℓ_p-ball we are talking about. Then, the generalization of the robust optimization problem in Equation (1) to multiple perturbation models is

$$\min_\theta \sum_i \max_{\delta \in \Delta_{\mathcal{S}}} \ell(x_i + \delta; \theta). \tag{9}$$

The key difference is in the inner maximization, where the worst-case adversarial loss is now taken over *multiple* ℓ_p perturbation models. In order to perform adversarial training, using the same motivational idea from Danskin's theorem, we can backpropagate through the inner maximization by first finding (empirically) the optimal perturbation,

$$\delta^* = \arg\max_{\delta \in \Delta_{\mathcal{S}}} \ell(x + \delta; \theta). \tag{10}$$

To find the optimal perturbation over the union of perturbation models, we begin by discussing simple generalizations of standard adversarial training, which will use aggregations of PGD solutions for individual adversaries to approximately solve the inner maximization over multiple adversaries. The computational complexity of these approaches are a constant factor times than the complexity of standard adversarial training, where the constant is equal to the number of adversaries. We will focus the exposition primarily on adversarial training based approaches as these are most related to our proposed method, and we refer the reader to Schott et al. (2019) for more detail on the analysis by synthesis approach.

4.1. Simple combinations of multiple perturbations

First, we study two simple approaches to generalizing adversarial training to multiple perturbation models, which can learn robust models and do not rely on complicated architectures. While these methods work to some degree, we later find empirically that these methods do not necessarily minimize the worst-case performance, can converge to unexpected tradeoffs between multiple perturbation models, and can have varying dataset-dependent performance.

MAX: Worst-case perturbation One way to generalize adversarial training to multiple perturbation models is to use each perturbation model independently, and train on the adversarial perturbation that achieved the maximum loss. Specifically, for each adversary $p \in \mathcal{S}$, we solve the innermost maximization with an ℓ_p PGD adversary to get an approximate worst-case perturbation δ_p,

$$\delta_p = \arg\max_{\delta \in \Delta_{p,\epsilon}} \ell(x + \delta; \theta), \tag{11}$$

and then approximate the maximum over all adversaries as

$$\delta^* \approx \arg\max_{\delta_p} \ell(x + \delta_p; \theta). \tag{12}$$

When $|\mathcal{S}| = 1$, then this reduces to standard adversarial training. Note that if each PGD adversary solved their subproblem from Equation (11) exactly, then this is the optimal perturbation $\delta^\star$. This method corresponds to the "max" strategy from Tramèr & Boneh (2019).

AVG: Augmentation of all perturbations Another way to generalize adversarial training is to train on all the adversarial perturbations for all $p \in \mathcal{S}$ to form a larger adversarial dataset. Specifically, instead of solving the robust problem for multiple adversaries in Equation (9), we instead solve

$$\min_\theta \sum_i \sum_{p \in \mathcal{S}} \max_{\delta \in \Delta_{p,\epsilon}} \ell(x_i + \delta; \theta) \tag{13}$$

by using individual ℓ_p PGD adversaries to approximate the inner maximization for each perturbation model. This reduces to standard adversarial training when $|\mathcal{S}| = 1$ and corresponds to the "avg" strategy from Tramèr & Boneh (2019).

While these methods work to some degree, (which is shown later in Section 5), both of these approaches solve the inner

Algorithm 1 Multi steepest descent for learning classifiers that are simultaneously robust to ℓ_p attacks for $p \in \mathcal{S}$

Input: classifier f_θ, data x, labels y
Parameters: ϵ_p, α_p for $p \in \mathcal{S}$, maximum iterations T, loss function ℓ
$\delta^{(0)} = 0$
for $t = 0 \dots T-1$ **do**
  for $p \in \mathcal{S}$ **do**
    $\delta_p^{(t+1)} = P_{\Delta_{p,\epsilon}}(\delta^{(t)} + v_p(\delta^{(t)}))$
  end for
  $\delta^{(t+1)} = \arg\max_{\delta_p^{(t+1)}} \ell(f_\theta(x + \delta_p^{(t+1)}), y)$
end for
return $\delta^{(T)}$

maximization problem independently for each adversary. Consequently, each individual PGD adversary is myopic to its own perturbation model and does not take advantage of the fact that the perturbation region is enlarged by other perturbation models. To leverage the full information provided by the union of perturbation regions, we propose a modification to standard adversarial training, which combines information from all considered perturbation models into a single PGD adversary that is potentially stronger than the combination of independent adversaries.

4.2. Multi Steepest Descent

To create a PGD adversary with full knowledge of the perturbation region, we propose an algorithm that incorporates the different perturbation models within each step of projected steepest descent. Rather than generating adversarial examples for each perturbation model with separate PGD adversaries, the core idea is to create a single adversarial perturbation by simultaneously maximizing the worst-case loss over all perturbation models at each projected steepest descent step. We call our method *multi steepest descent* (MSD), which can be summarized as the following iteration:

$$\begin{aligned} \delta_p^{(t+1)} &= P_{\Delta_{p,\epsilon}}(\delta^{(t)} + v_p(\delta^{(t)})) \;\; \text{for} \;\; p \in \mathcal{S} \\ \delta^{(t+1)} &= \arg\max_{\delta_p^{(t+1)}} \ell(x + \delta_p^{(t+1)}) \end{aligned} \tag{14}$$

The key difference here is that at each iteration of MSD, we choose a projected steepest descent direction that maximizes the loss over all attack models $p \in \mathcal{S}$, whereas standard adversarial training and the simpler approaches use comparatively myopic PGD subroutines that only use one perturbation model at a time. The full algorithm is in Algorithm 1, and can be used as a drop in replacement for standard PGD adversaries to learn robust classifiers with adversarial training. We direct the reader to Appendix A for a complete description of steepest descent directions and projection operators for ℓ_∞, ℓ_2, and ℓ_1 norms.[1]

5. Results

In this section, we present experimental results on using generalizations of adversarial training to achieve simultaneous robustness to ℓ_∞, ℓ_2, and ℓ_1 perturbations on the MNIST and CIFAR10 datasets. Our primary goal is to show that adversarial training can be used to directly minimize the worst-case loss over the union of perturbation models to achieve competitive results by avoiding any trade-off that biases one particular perturbation model at the cost of the others. Our results improve upon the state-of-the-art in three key ways. First, we can continue to use simple, standard architectures for image classifiers, without relying on complex architectures or input binarization as done by Schott et al. (2019). Second, our method is able to learn a single model (on both MNIST and CIFAR10) which optimizes the worst-case performance over the union of all three perturbation models, whereas previous approaches are only robust against two at a time, or have performance which is dataset dependent. Finally, we provide the first CIFAR10 model trained to be simultaneously robust against ℓ_∞, ℓ_2, and ℓ_1 adversaries, in comparison to previous work which trained a model robust to ℓ_∞, ℓ_1, and rotation/translation attacks (Tramèr & Boneh, 2019).

We train models using MSD, MAX and AVG approaches for both MNIST and CIFAR10 datasets. We additionally train models against individual PGD adversaries to measure the changes and tradeoffs in universal robustness. Since the analysis by synthesis model is not scalable, we do not include it in our experimentation for CIFAR10. We perform an extensive evaluation of these models with a broad suite of both gradient and non-gradient based attacks using Foolbox[2] (the same attacks used by Schott et al. (2019)), and also incorporate all the PGD-based adversaries discussed in this paper. All aggregate statistics that combine multiple attacks compute the worst-case error rate over all attacks for *each* example, in order to reflect the worst-case loss over the combined perturbation model.

Summaries of these results at specific thresholds can be found in Tables 1 and 2, where B-ABS and ABS refer to binarized and non-binarized versions of the analysis by synthesis models from Schott et al. (2019), P_p refers to a model trained against a PGD adversary with respect to the p-norm, MAX and AVG refer to models trained using the worst-case and data augmentation generalizations of adversarial training, and MSD refers to models trained using

[1] The pure ℓ_1 steepest descent step is inefficient since it only updates one coordinate at a time. It can be improved by taking steps on multiple coordinates, similar to that used in Tramèr & Boneh (2019), and is also explained in Appendix A.

[2] https://github.com/bethgelab/foolbox (Rauber et al., 2017)

multi steepest descent. Full tables containing the complete breakdown of these numbers over all individual attacks used in the evaluation are in Appendix B. We report the results against individual attacks and perturbation models for completeness, however we note that the original goal and motivation of all these algorithms is to minimize the robust optimization objective from Equation (9). While there may be different implicit tradeoffs between individual perturbation models that can be difficult to compare, the robust optimization objective, or the performance against the union of *all* attacks, provides a single common metric that all approaches are optimizing.

5.1. Experimental setup

Architectures and hyperparameters For MNIST, we use a four layer convolutional network with two convolutional layers consisting of 32 and 64 5×5 filters and 2 units of padding, followed by a fully connected layer with 1024 hidden units, where both convolutional layers are followed by 2×2 Max Pooling layers and ReLU activations (this is the same architecture used by Madry et al. (2018)). This is in contrast to past work on MNIST, which relied on per-class variational autoencoders to achieve robustness against multiple perturbation models (Schott et al., 2019), which was also not easily scalable to larger datasets. Since our methods have the same computational complexity as standard adversarial training, they also easily apply to standard CIFAR10 architectures, and in this paper we use the well known pre-activation version of the ResNet18 architecture consisting of nine residual units with two convolutional layers each (He et al., 2016).

A complete description of the hyperparameters used is in Appendix C. All reported ϵ are for images scaled to be between the range $[0, 1]$. All experiments were run on modest amounts of GPU hardware (e.g. a single 1080ti).

Attacks used for evaluation To evaluate the model, we incorporate the attacks from Schott et al. (2019) along with our PGD based adversaries, and provide a short description of the same here. Note that we exclude attacks based on gradient estimation, since the gradient for the standard architectures used here are readily available.

For ℓ_∞ attacks, although we find the ℓ_∞ PGD adversary to be quite effective, for completeness, we additionally use the Foolbox implementations of Fast Gradient Sign Method (Goodfellow et al., 2015), PGD attack (Madry et al., 2018), and Momentum Iterative Method (Dong et al., 2018).

For ℓ_2 attacks, in addition to the ℓ_2 PGD adversary, we use the Foolbox implementations of the same PGD adversary, the Gaussian noise attack (Rauber et al., 2017), the boundary attack (Brendel et al., 2017), DeepFool (Moosavi-Dezfooli et al., 2016), the pointwise attack (Schott et al., 2019), DDN based attack (Rony et al., 2018), and C&W attack (Carlini & Wagner, 2017).

For ℓ_1 attacks, we use both the ℓ_1 PGD adversary as well as additional Foolbox implementations of ℓ_0 attacks at the same radius, namely the salt & pepper attack (Rauber et al., 2017) and the pointwise attack (Schott et al., 2019). Note that an ℓ_1 adversary with radius ϵ is strictly stronger than an ℓ_0 adversary with the same radius, and so we choose to explicitly defend against ℓ_1 perturbations instead of the ℓ_0 perturbations considered by Schott et al. (2019).

We make **10 random restarts** for each of the results mentioned hereon for both MNIST and CIFAR10 [3]. We encourage future work in this area to incorporate the same, since the success of all attacks, specially decision based or gradient free ones, is observed to increase significantly over restarts.

5.2. MNIST

We first present results on the MNIST dataset, which are summarized in Table 1 (a more detailed breakdown over each individual attack is in Appendix B.1). Complete robustness curves over a range of epsilons over each perturbation model can be found in Figure 2. Although we reproduce the simpler approaches here, a more detailed discussion of how these results compare with those presented by Tramèr & Boneh (2019) can be found in Appendix D.

Suboptimal trade-offs While considered an "easy" dataset, we first note that most of the previous approaches for multiple perturbation models on MNIST are only able to defend against two out of three perturbation models at a time, resulting in a suboptimal trade-off between different perturbation models which has poor overall performance against the worst-case attack in the combined perturbation model. Despite relying on a significantly more complex architecture, the B-ABS model is weak against ℓ_2 attacks while the ABS model is weak against ℓ_∞ attacks. Meanwhile, the AVG model is weak against strong ℓ_1 decision-based attacks. The MAX and MSD models achieve relatively better trade-offs, with the MSD model performing the best with a robust accuracy rate of 58.4% against the union of (ℓ_∞, ℓ_2, ℓ_1) perturbations with radius $\epsilon = (0.3, 2.0, 10)$, which is over a 15% improvement in comparison to the MAX model.

Gradient Masking in MNIST models We find that even though models trained via the MAX and AVG approaches provide reasonable robustness against *first-order* attacks

[3] All attacks were run on a subset of the first 1000 test examples with 10 random restarts, with the exception of Boundary Attack, which by default makes 25 trials per iteration and DDN based Attack which does not benefit from the same owing to a deterministic initialization of δ.

Table 1: Summary of adversarial accuracy results for MNIST (higher is better)

	P_∞	P_2	P_1	B-ABS[4]	ABS[4]	MAX	AVG	MSD
Clean Accuracy	99.1%	99.2%	99.3%	99%	99%	98.6%	99.1%	98.3%
ℓ_∞ attacks ($\epsilon = 0.3$)	90.3%	0.4%	0.0%	77%	8%	51.0%	65.2%	62.7%
ℓ_2 attacks ($\epsilon = 2.0$)	13.6%	69.2%	38.5%	39%	80%	61.9%	60.1%	67.9%
ℓ_1 attacks ($\epsilon = 10$)	4.2%	43.4%	70.0%	82%	78%	52.6%	39.2%	65.0%
All Attacks	3.7%	0.4%	0.0%	39%	8%	42.1%	34.9%	**58.4**%

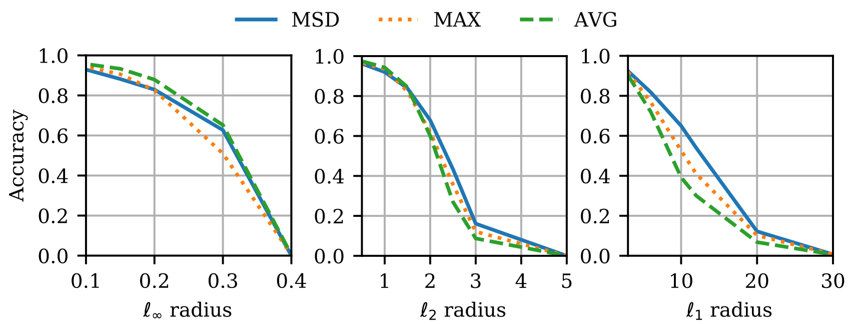

Figure 2: Robustness curves showing the adversarial accuracy for the MNIST model trained with MSD, AVG, MAX against ℓ_∞ (left), ℓ_2 (middle), and ℓ_1 (right) perturbation models over a range of epsilon.

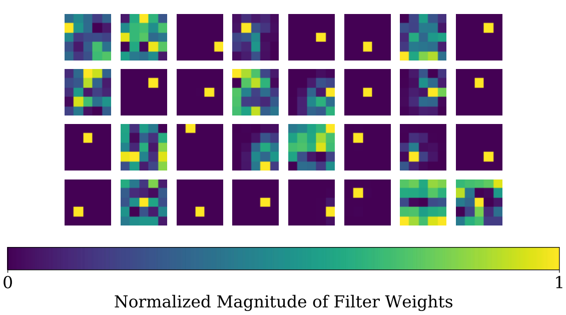

Figure 3: A view of each of the (5x5) learned filters of the first layer of a CNN robust to ℓ_∞ attacks. The singular sharp values are characteristic features of models robust to ℓ_∞ attacks.

(breakdown of attacks in Appendix B.1), they can be vulnerable to gradient-free attacks like the Pointwise Attack and Boundary Attack. This indicates the presence of masked gradients that prevent *first-order* adversaries from finding the optimal steepest descent direction (Athalye et al., 2018a), similar to how ℓ_∞ trained models are weak against decision-based attacks in other norms as also observed by Schott et al. (2019) and Tramèr & Boneh (2019). We analyze the learned weights of the first layer filters of the CNN models trained on the MNIST, and observe a strong correlation of the presence of thresholding filters (Figure 3) with the susceptibility to decision-based ℓ_1 and ℓ_2 adversaries. Further analysis of the learned filter weights for all the models can be found in Appendix E, where we observe that by reducing the number of thresholding filters, the MSD model is able to perform better against decision based adversaries, whereas learning filter patterns similar to that of an ℓ_∞ robust model correlates with susceptibility of MAX and AVG training methods to gradient-free adversaries.

Unreliable training of MAX and AVG To give the MAX and AVG approaches the best chance at succeeding, we searched over a wide range of hyperparameters (which are described in Appendix C.2). However, we frequently observe that these training runs result in masked gradients as described earlier, and are seemingly unable to balance the right trade-off between multiple attacks. In Figure 4, we show the sensitivity of different training methods to training time hyperparameter choices. The worst case accuracy is evaluated using the worst case over three gradient based attacks (PGD attacks in $\ell_\infty, \ell_2, \ell_1$ space) and one gradient-free attack (pointwise attack in ℓ_1 space). The MAX training method achieves greater than 40% robust accuracy in only 10% of all the hyperparameter configurations tried. The sensitivity was even higher for the AVG method on the MNIST

[4]Results are reported directly from Schott et al. (2019), which used epsilon balls of radii (0.3,1.5,12) for (ℓ_∞, ℓ_2, ℓ_0) adversaries. They used an ℓ_0 perturbation region of a higher radius and evaluated against ℓ_0 attacks. So the reported number is a near estimate of the ℓ_1 adversarial accuracy. They used an ℓ_2 perturbation model of a lower radius = 1.5. Further, they do not perform attack restarts and the adversarial accuracy against all attacks is an upper bound based on the reported accuracies for individual perturbation models. Finally, all ABS results were computed using numerical gradient estimation, since gradients are not readily available.

Table 2: Summary of adversarial accuracy results for CIFAR10 (higher is better)

	P_∞	P_2	P_1	MAX	AVG	MSD
Clean accuracy	83.3%	90.2%	73.3%	81.0%	84.6%	81.1%
ℓ_∞ attacks ($\epsilon = 0.03$)	50.7%	28.3%	0.2%	44.9%	42.5%	48.0%
ℓ_2 attacks ($\epsilon = 0.5$)	57.3%	61.6%	0.0%	61.7%	65.0%	64.3%
ℓ_1 attacks ($\epsilon = 12$)	16.0%	46.6%	7.9%	39.4%	54.0%	53.0%
All attacks	15.6%	27.5%	0.0%	34.9%	40.6%	**47.0**%

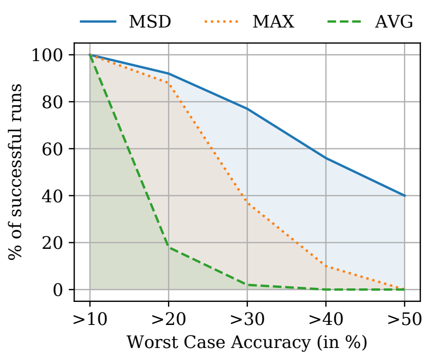

Figure 4: Among all the models trained using the MSD, MAX and AVG methods during our hyperparameter search, we plot the percentage of models for each method that achieve robust accuracies greater than a particular threshold (against the union of $\ell_\infty, \ell_1, \ell_2$ attacks).

dataset. Also, note that nearly all models attain greater than 50% robust accuracy when only attacked by gradient-based adversaries, and the performance drop is largely attributed to the gradient-free attack.

However, MSD is comparatively much easier to tune and achieves greater than 50% accuracy in around 40% of the runs. Moreover, we find that MSD offers a natural way to counteract any unwanted bias towards one perturbation type by adjusting the relative step-sizes of individual descent directions, whereas doing the same for the MAX and AVG approaches does not help.

We note that in order to train the MAX and AVG approaches reasonably well on the MNIST dataset (Table 1), we had to set the radius of the ℓ_1 ball to 12 for AVG and increase the number of PGD ℓ_1 attack restarts during training for MAX. These methods help make the PGD ℓ_1 attack relatively *stronger* by changing the perturbation model, and re-aligns the optimal trade-offs when the training process is unable to naturally capture them. We observe that *small starts*, as employed by Tramèr & Boneh (2019) to make their models work better, may have a similar effect of re-aligning the strength of various perturbation models. Rather than "fixing" the balance between different perturbation models by changing the individual attacks used for training, MSD is able to achieve the right trade-off by directly balancing them, leading to greater reliability and consistency when compared to the MAX and AVG approaches.

5.3. CIFAR10

Next, we present results on the CIFAR10 dataset, which are summarized in Table 2 (a more detailed breakdown over each individual attack is in Appendix B.2). Our MSD approach reaches the best performance against the union of attacks, and achieves 47.0% (individually 48.0%, 64.3%, 53.0%) adversarial accuracy against the union of $(\ell_\infty, \ell_2, \ell_1)$ perturbations of size $\epsilon = (0.03, 0.5, 12)$. We note that the P_1 model trained against an ℓ_1 PGD adversary is not very robust when evaluated against decision-based attacks, even though it can defend reasonably well against the ℓ_1 PGD attack in isolation (Table 4 in Appendix B.2). Complete robustness curves over a range of epsilons over each perturbation model can be found in Figure 5. The specific heuristic adjustments made to obtain the best-performing MAX and AVG models are detailed in Appendix C.2. Although we reproduce the simple adversarial training approaches here, a direct comparison of how these results compare to those reported by (Tramèr & Boneh, 2019) can be found in Appendix D. Furthermore, while adversarial defenses are generally not intended to be robust to attacks outside of the perturbation model, we show some experiments exploring this aspect in Appendix F, namely the performance on the CIFAR10-C dataset (CIFAR10 with common corruptions) as well as exploring what happens when one defends against only two adversaries and evaluates on a third, unseen adversary.

Dataset variability In addition to converging to suboptimal trade-offs between different adversaries as seen on MNIST, we find that the performance of simpler versions of adversarial training for multiple perturbations can also vary significantly based on the dataset. While the MAX approach performed better than AVG on MNIST, in the CIFAR10 setting we find that these roles are swapped: the MAX approach converged to a suboptimal local minima which is 5.7% less robust against the union of perturbation models than AVG. Once again, this highlights the inconsistency of the simpler generalizations of adversarial training: depending on the problem setting, they may converge to suboptimal

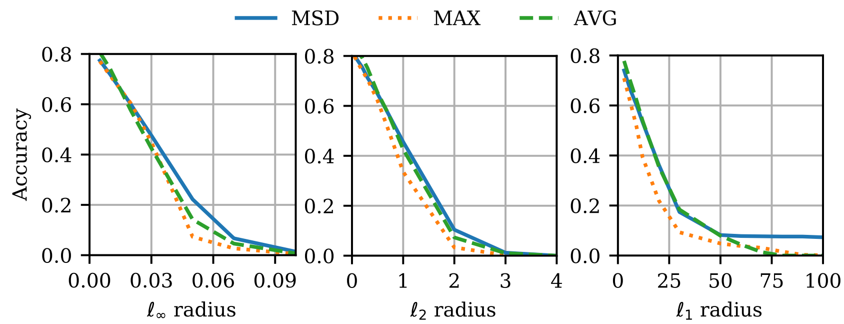

Figure 5: Robustness curves showing the adversarial accuracy for the CIFAR10 model trained with MSD, AVG, MAX against ℓ_∞ (left), ℓ_2 (middle), and ℓ_1 (right) perturbation models over a range of epsilon.

local optima which do not minimize the robust optimization objective from Equation (9). On the other hand, in both problem settings, we find MSD consistently converges to a local optimum which is better at minimizing the worst-case loss in the union of the perturbation models, achieving 47.0% robust accuracy, improving upon the best-performing simpler method of AVG by 6.4%.

6. Conclusion

In this paper, we showed that previous approaches aimed towards learning models which are adversarially robust to multiple perturbation models can be highly variable (across parameters and datasets), and difficult to tune, thereby converging to suboptimal local minima with trade-offs which do not defend against the union of multiple perturbation models. On the other hand, by incorporating the different perturbation models directly into the direction of steepest descent, our proposed approach of MSD consistently outperforms past approaches across both MNIST and CIFAR10. The approach inherits the scalability and generality of adversarial training, without relying on specific complex architectures, and is able to better accomplish the robust optimization objective. We recommend using MSD to directly minimize the worst-case performance among multiple perturbation models.

Acknowledgements

Eric Wong was funded by support from the Bosch Center for AI, under contract 0087016732PCR, and a fellowship from the Siebel Scholars Foundation. Pratyush Maini was supported by a fellowship from the Khorana Program for Scholars, aided jointly by the Department of Science & Technology Govt. of India and the US Department of State.

References

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018a. URL https://arxiv.org/abs/1802.00420.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018b. PMLR. URL http://proceedings.mlr.press/v80/athalye18b.html.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017.

Chen, P.-Y., Sharma, Y., Zhang, H., Yi, J., and Hsieh, C.-J. Ead: Elastic-net attacks to deep neural networks via adversarial examples, 2017.

Croce, F. and Hein, M. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. *CoRR*, abs/1905.11213, 2019. URL http://arxiv.org/abs/1905.11213.

Danskin, J. M. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 1967.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 272–279, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390191. URL http://doi.acm.org/10.1145/1390156.1390191.

Engstrom, L., Ilyas, A., and Athalye, A. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018. URL http://arxiv.org/abs/1810.12715.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Jordan, M., Manoj, N., Goel, S., and Dimakis, A. G. Quantifying perceptual distortion of adversarial examples. *arXiv preprint arXiv:1902.08265*, 2019.

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

Kannan, H., Kurakin, A., and Goodfellow, I. J. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018. URL http://arxiv.org/abs/1803.06373.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An efficient smt solver for verifying deep neural networks. *arXiv preprint arXiv:1702.01135*, 2017.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *ICLR Workshop*, 2017. URL https://arxiv.org/abs/1607.02533.

Lu, J., Sibai, H., Fabry, E., and Forsyth, D. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning (ICML)*, 2018. URL https://www.icml.cc/Conferences/2018/Schedule?showEvent=2477.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Mosbach, M., Andriushchenko, M., Trost, T., Hein, M., and Klakow, D. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 582–597. IEEE, 2016.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4944-4. doi: 10.1145/3052973.3053009. URL http://doi.acm.org/10.1145/3052973.3053009.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=Bys4ob-Rb.

Raghunathan, A., Steinhardt, J., and Liang, P. S. Semidefinite relaxations for certifying robustness to adversarial examples. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10900–10910. Curran Associates, Inc., 2018b. URL http://papers.nips.cc/paper/8285-semidefinite-relaxations-for-certifying-robustness-to-adversarial-examples.pdf.

Rauber, J., Brendel, W., and Bethge, M. Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*, 2017. URL http://arxiv.org/abs/1707.04131.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. *CoRR*, abs/1811.09600, 2018. URL http://arxiv.org/abs/1811.09600.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1EHOsC9tX.

Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyGIdiRqtm.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. *arXiv preprint arXiv:1904.13000*, 2019.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5025–5034, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/uesato18a.html.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5283–5292, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8410–8419. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8060-scaling-provable-adversarial-defenses.pdf.

Zagoruyko, S. and Komodakis, N. Wide residual networks, 2016.