ENSEMBLE ADVERSARIAL TRAINING: ATTACKS AND DEFENSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial examples are perturbed inputs designed to fool machine learning models. Adversarial training injects such examples into training data to increase robustness. To scale this technique to large datasets, perturbations are crafted using fast *single-step* methods that maximize a linear approximation of the model's loss.

We show that this form of adversarial training converges to a degenerate global minimum, wherein small curvature artifacts near the data points obfuscate a linear approximation of the loss. The model thus learns to generate weak perturbations, rather than defend against strong ones. As a result, we find that adversarial training remains vulnerable to black-box attacks, where we *transfer* perturbations computed on undefended models, as well as to a powerful novel single-step attack that escapes the non-smooth vicinity of the input data via a small random step.

We further introduce *Ensemble Adversarial Training*, a technique that augments training data with perturbations transferred from other models. On ImageNet, Ensemble Adversarial Training yields models with strong robustness to black-box attacks. In particular, our most robust model won the first round of the *NIPS 2017 competition on Defenses against Adversarial Attacks* (Kurakin et al., 2017c).

1 INTRODUCTION

Machine learning (ML) models are often vulnerable to *adversarial examples*, maliciously perturbed inputs designed to mislead a model at test time (Biggio et al., 2013; Szegedy et al., 2013; Goodfellow et al., 2014b; Papernot et al., 2016a). Furthermore, Szegedy et al. (2013) showed that these inputs *transfer* across models: the same adversarial example is often misclassified by different models, thus enabling simple *black-box* attacks on deployed models (Papernot et al., 2017; Liu et al., 2017).

Adversarial training (Szegedy et al., 2013) increases robustness by augmenting training data with adversarial examples. Madry et al. (2017) showed that adversarially trained models can be made robust to *white-box* attacks (i.e., with knowledge of the model parameters) if the perturbations computed during training closely maximize the model's loss. However, prior attempts at scaling this approach to ImageNet-scale tasks Deng et al. (2009) have proven unsuccessful (Kurakin et al., 2017b).

It is thus natural to ask whether it is possible, at scale, to achieve robustness against the class of *black-box* adversaries Towards this goal, Kurakin et al. (2017b) adversarially trained an Inception v3 model (Szegedy et al., 2016b) on ImageNet using a "single-step" attack based on a linearization of the model's loss (Goodfellow et al., 2014b). Their trained model is robust to single-step perturbations but remains vulnerable to more costly "multi-step" attacks. Yet, Kurakin et al. (2017b) found that these attacks fail to reliably transfer between models, and thus concluded that the robustness of their model should extend to black-box adversaries. Surprisingly, we show that this is not the case.

We demonstrate, formally and empirically, that *adversarial training with single-step methods admits a degenerate global minimum*, wherein the model's loss can not be reliably approximated by a linear function. Specifically, we find that the model's decision surface exhibits sharp curvature near the data points, thus degrading attacks based on a single gradient computation. In addition to the model of Kurakin et al. (2017b), we reveal similar overfitting in an adversarially trained Inception ResNet v2 model (Szegedy et al., 2016a), and a variety of models trained on MNIST (LeCun et al., 1998).

We harness this result in two ways. First, we show that adversarially trained models using single-step methods remain vulnerable to simple attacks. For black-box adversaries, we find that perturbations

crafted on an undefended model often transfer to an adversarially trained one. We also introduce a simple yet powerful single-step attack that applies a small *random* perturbation—to escape the non-smooth vicinity of the data point—before linearizing the model's loss. While seemingly weaker than the Fast Gradient Sign Method of Goodfellow et al. (2014b), our attack significantly outperforms it for a same perturbation norm, for models trained *with or without adversarial training*.

Second, we propose *Ensemble Adversarial Training*, a training methodology that incorporates perturbed inputs transferred from other *pre-trained models*. Our approach *decouples* adversarial example generation from the parameters of the trained model, and increases the *diversity* of perturbations seen during training. We train Inception v3 and Inception ResNet v2 models on ImageNet that exhibit increased robustness to adversarial examples transferred from other holdout models, using various single-step and multi-step attacks (Goodfellow et al., 2014b; Carlini & Wagner, 2017a; Kurakin et al., 2017a; Madry et al., 2017). We also show that our methods globally reduce the *dimensionality* of the space of adversarial examples (Tramèr et al., 2017). Our Inception ResNet v2 model won the first round of the NIPS 2017 competition on Defenses Against Adversarial Attacks (Kurakin et al., 2017c), where it was evaluated on other competitors' attacks in a black-box setting.[1]

2 Related Work

Various defensive techniques against adversarial examples in deep neural networks have been proposed (Gu & Rigazio, 2014; Luo et al., 2015; Papernot et al., 2016c; Nayebi & Ganguli, 2017; Cisse et al., 2017) and many remain vulnerable to adaptive attackers (Carlini & Wagner, 2017a;b; Baluja & Fischer, 2017). Adversarial training (Szegedy et al., 2013; Goodfellow et al., 2014b; Kurakin et al., 2017b; Madry et al., 2017) appears to hold the greatest promise for learning robust models.

Madry et al. (2017) show that adversarial training on MNIST yields models that are robust to white-box attacks, if the adversarial examples used in training closely maximize the model's loss. As we argue in Appendix B, the MNIST dataset is peculiar in that there exists a simple "closed-form" *denoising* procedure (namely feature binarization) which leads to similarly robust models *without adversarial training*. This may explain why robustness to white-box attacks is hard to scale to tasks such as ImageNet (Kurakin et al., 2017b). We believe that the existence of a simple robust baseline for MNIST can be useful for understanding some *limitations* of adversarial training techniques.

Szegedy et al. (2013) found that adversarial examples transfer between models, thus enabling black-box attacks on deployed models. Papernot et al. (2017) showed that black-box attacks could succeed with no access to training data, by exploiting the target model's predictions to *extract* (Tramèr et al., 2016) a surrogate model. Some prior works have hinted that adversarially trained models may remain vulnerable to black-box attacks: Goodfellow et al. (2014b) found that an adversarial maxout network on MNIST has slightly higher error on transferred examples than on white-box examples. Papernot et al. (2017) further showed that a model trained on small perturbations can be evaded by transferring perturbations of larger magnitude. Our finding that adversarial training degrades the accuracy of linear approximations of the model's loss is as an instance of a *gradient-masking* phenomenon (Papernot et al., 2016b), which affects other defensive techniques (Papernot et al., 2016c; Carlini & Wagner, 2017a; Nayebi & Ganguli, 2017; Brendel & Bethge, 2017).

3 The Adversarial Training Framework

We consider a classification task with data $x \in [0,1]^d$ and labels $y_{\text{true}} \in \mathbb{Z}_k$ sampled from a distribution $\mathcal{D}$. We identify a model with an hypothesis h from a space $\mathcal{H}$. On input x, the model outputs class scores $h(x) \in \mathbb{R}^k$. The loss function used to train the model, e.g., cross-entropy, is $L(h(x), y)$.

3.1 Threat Model

For some target model $h \in \mathcal{H}$ and inputs (x, y_{true}) the adversary's goal is to find an *adversarial example* x^{adv} such that x^{adv} and x are "close" yet the model misclassifies x^{adv}. We consider the well-studied class of ℓ_∞ bounded adversaries (Goodfellow et al., 2014b; Madry et al., 2017) that, given

[1] We publicly released our model after the first round, and it could thereafter be targeted using white-box attacks. Nevertheless, a majority of the teams' final round submissions were ensembles containing our model.

some *budget* ϵ, output examples x^{adv} where $\|x^{\text{adv}} - x\|_\infty \leq \epsilon$. As we comment in Appendix B.1, ℓ_∞ robustness is of course not an end-goal for secure ML. We use this standard model to showcase limitations of prior adversarial training methods, and evaluate our proposed improvements.

We distinguish between *white-box* adversaries that have access to the target model's parameters (i.e., h), and *black-box* adversaries that only interact with a model's prediction interface. Formal definitions for these adversaries are in Appendix A. Although security against white-box attacks is the stronger notion (and the one we ideally want ML models to achieve), black-box security is a reasonable and more tractable goal for deployed ML models.

3.2 Adversarial Training

Following Madry et al. (2017), we consider an adversarial variant of standard Empirical Risk Minimization (ERM), where our aim is to minimize the risk over adversarial examples:

$$h^* = \arg\min_{h \in \mathcal{H}} \quad \mathop{\mathbb{E}}_{(x, y_{\text{true}}) \sim \mathcal{D}} \left[\max_{\|x^{\text{adv}} - x\|_\infty \leq \epsilon} L(h(x^{\text{adv}}), y_{\text{true}}) \right] . \tag{1}$$

Madry et al. (2017) argue that adversarial training has a natural interpretation in this context, where a given attack (see below) is used to approximate solutions to the inner maximization problem, and the outer minimization problem corresponds to training over these examples. Note that the original formulation of adversarial training (Szegedy et al., 2013; Goodfellow et al., 2014b), which we use in our experiments, trains on both the "clean" examples x and adversarial examples x^{adv}.

We consider three algorithms to generate adversarial examples with bounded ℓ_∞ norm. The first two are *single-step* (i.e., they require a single gradient computation); the third is *iterative*—it computes multiple gradient updates. We enforce $x^{\text{adv}} \in [0, 1]^d$ by clipping all components of x^{adv}.

Fast Gradient Sign Method (FGSM). This method (Goodfellow et al., 2014b) linearizes the inner maximization problem in (1):

$$x^{\text{adv}} = x + \varepsilon \cdot \texttt{sign}\left(\nabla_x L(h(x), y_{\text{true}})\right) . \tag{2}$$

Single-Step Least-Likely Class Method (Step-LL). This variant of FGSM introduced by Kurakin et al. (2017a;b) targets the least-likely class, $y_{\text{LL}} = \arg\min\{h(x)\}$:

$$x^{\text{adv}} = x - \varepsilon \cdot \texttt{sign}\left(\nabla_x L(h(x), y_{\text{LL}})\right) . \tag{3}$$

Although this attack only indirectly tackles the inner maximization in (1), Kurakin et al. (2017b) find it to be the most effective for adversarial training on ImageNet.

Iterative Attack (I-FGSM or Iter-LL). This method iteratively applies the FGSM or Step-LL k times with step-size $\alpha \geq \epsilon/k$ and projects each step onto the ℓ_∞ ball of norm ϵ around x. It uses projected gradient descent to solve the maximization in (1). For fixed ϵ, iterative attacks induce higher error rates than single-step attacks, but *transfer* at lower rates (Kurakin et al., 2017a;b).

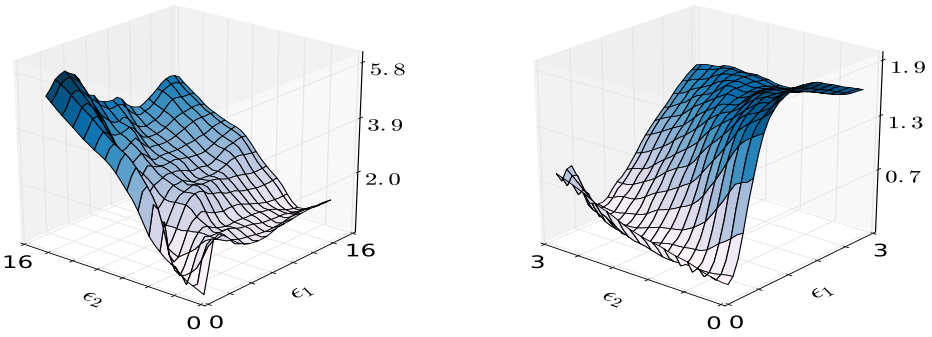
3.3 A Degenerate Global Minimum for Single-Step Adversarial Training

When performing adversarial training with a single-step attack (e.g., the FGSM or Step-LL methods above), we approximate Equation (1) by replacing the solution to the inner maximization problem with the output of the single-step attack. For model families $\mathcal{H}$ with high expressive power, this alternative optimization problem admits at least two substantially different global minima h^*:

Minimum 1: On average, for an input x from $\mathcal{D}$, there is no input x^{adv} close to x (in ℓ_∞ norm) that induces a high loss. In other words, h^* is robust to all ℓ_∞ bounded perturbations.

Minimum 2: The minimizer h^* is a model for which the approximation method underlying the attack (i.e., linearization in our case) poorly fits the model's loss function. As a result, the attack when applied to h^* produces samples x^{adv} that are far from optimal.

Note that this second "degenerate" minimum can be more subtle than a simple case of overfitting to samples produced from single-step attacks. Indeed, we show in Section 4.1 that single-step attacks applied to adversarially trained models create "adversarial" examples *that are easy to classify even for undefended models*. Thus, adversarial training does not simply learn to resist the particular attack used during training, but actually to make that attack perform worse overall.



(a) Loss of model $v3_{adv}$.

(b) Zoom in for small ϵ_1, ϵ_2.

Figure 1: **Gradient masking in single-step adversarial training.** We plot the loss of model $v3_{adv}$ on points $x^* = x + \epsilon_1 \cdot g + \epsilon_2 \cdot g^{\perp}$, where g is the signed gradient and $g^{\perp}$ is an orthogonal adversarial direction. Plot (b) is a zoom of (a) near x. The gradient poorly approximates the global loss.

3.4 Ensemble Adversarial Training

The degenerate minimum described in Section 3.3 is attainable because the learned model's parameters influence the quality of both the minimization and maximization in (1). One solution is to use a stronger adversarial example generation process, at a high performance cost (Madry et al., 2017). Alternatively, Baluja & Fischer (2017) suggest training an adversarial *generator* model as in the GAN framework (Goodfellow et al., 2014a). The power of this generator is likely to require careful tuning, to avoid similar degenerate minima (where the generator or classifier overpowers the other).

We propose a conceptually simpler approach to *decouple* the generation of adversarial examples from the model being trained, while simultaneously drawing an explicit connection with robustness to black-box adversaries. Our method, which we call *Ensemble Adversarial Training*, augments a model's training data with adversarial examples crafted on other *static pre-trained* models. Intuitively, as adversarial examples transfer between models, perturbations crafted on an external model are good approximations for the maximization problem in (1). Moreover, the learned model can not influence the "strength" of these adversarial examples. As a result, minimizing the training loss implies increased robustness to black-box attacks from some set of models. We introduce the following (informal) conjecture, which we will show to be supported by our experimental results (see 4.2):

> *Robustness to attacks transfers: That is, a model that learns to be robust to attacks transfered from some models $\{h_1, \ldots, h_k\}$ is also more robust to black-box attacks from other models.*

4 Experiments

We show the existence of a degenerate minimum, as described in Section 3.3, for the adversarially trained Inception v3 model of Kurakin et al. (2017b). Their model (denoted $v3_{adv}$) was trained on a Step-LL attack with $\epsilon \leq 16/256$. We also adversarially train an Inception ResNet v2 model (Szegedy et al., 2016a) using the same setup. We denote this model by $IRv2_{adv}$. We refer the reader to (Kurakin et al., 2017b) for details on the adversarial training procedure.

We first measure the *approximation-ratio* of the Step-LL attack for the inner maximization in (1). As we do not know the true maximum, we lower-bound it using an iterative attack. For $1{,}000$ random test points, we find that for a standard Inception v3 model, step-LL gets within 19% of the optimum loss on average. This attack is thus a good candidate for adversarial training. Yet, for the $v3_{adv}$ model, the approximation ratio drops to 7%, confirming that the learned model is less amenable to linearization. We obtain similar results for Inception ResNet v2 models. The ratio is 17% for a standard model, and 8% for $IRv2_{adv}$. Similarly, we look at the *cosine similarity* between the perturbations given by a single-step and multi-step attack. The more linear the model, the more similar we expect both perturbations to be. The average similarity drops from 0.13 for Inception v3 to 0.02 for $v3_{adv}$. This effect is not due to the decision surface of $v3_{adv}$ being "too flat" near the data points: the average gradient norm is larger for $v3_{adv}$ (0.17) than for the standard v3 model (0.10).

Table 1: **Error rates (in %) of adversarial examples transferred between models.** We use Step-LL with $\epsilon = 16/256$ for 10,000 random test inputs. Diagonal elements represent a white-box attack. The best attack for each target appears in bold. Similar results for MNIST models appear in Table 7.

	Source						Source				
Target	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$	**Target**	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$
v4	**60.2**	39.2	31.1	36.6	30.9	v4	**31.0**	14.9	10.2	13.6	9.9
v3	43.8	**69.6**	36.4	42.1	35.1	v3	18.7	**42.7**	13.0	17.8	12.8
$v3_{adv}$	**36.3**	35.6	~~26.6~~	35.2	35.9	$v3_{adv}$	13.6	13.5	~~9.0~~	13.0	**14.5**
IRv2	38.0	38.0	30.8	**50.7**	31.9	IRv2	14.1	14.8	9.9	**24.0**	10.6
$IRv2_{adv}$	**31.0**	30.3	25.7	30.6	~~21.4~~	$IRv2_{adv}$	10.3	**10.5**	7.7	10.4	~~5.8~~
	Top 1						**Top 5**				

We visualize this "gradient-masking" effect (Papernot et al., 2016b) by plotting the loss of $v3_{adv}$ on examples $x^* = x + \epsilon_1 \cdot g + \epsilon_2 \cdot g^\perp$, where g is the signed gradient of model $v3_{adv}$ and $g^\perp$ is a signed vector orthogonal to g. Looking forward to Section 4.1, we actually chose $g^\perp$ to be the signed gradient of another Inception model, from which adversarial examples transfer to $v3_{adv}$. Figure 1 shows that the loss is highly curved in the vicinity of the data point x, and that the gradient poorly reflects the global loss landscape. Similar plots for additional data points are in Figure 4.

We show similar results for adversarially trained MNIST models in Appendix B.2. On this task, *input dropout* (Srivastava et al., 2014) mitigates adversarial training's overfitting problem, in some cases. Presumably, the random input mask *diversifies* the perturbations seen during training (dropout at intermediate layers does not mitigate the overfitting effect). Mishkin et al. (2017) find that input dropout significantly degrades accuracy on ImageNet, so we did not include it in our experiments.

4.1 Attacks Against Adversarially Trained Networks

Kurakin et al. (2017b) found their adversarially trained model to be robust to various single-step attacks. They conclude that this robustness should translate to attacks *transferred* from other models. As we have shown, the robustness to single-step attacks is actually misleading, as the model has learned to *degrade* the information contained in the model's gradient. As a consequence, we find that the $v3_{adv}$ model is substantially more vulnerable to single-step attacks than Kurakin et al. (2017b) predicted, both in a white-box and black-box setting. The same holds for the $IRv2_{adv}$ model.

In addition to the $v3_{adv}$ and $IRv2_{adv}$ models, we consider standard Inception v3, Inception v4 and Inception ResNet v2 models. These models are available in the `TensorFlow-Slim` library (Abadi et al., 2015). We describe similar results for a variety of models trained on MNIST in Appendix B.2.

Black-box attacks. Table 1 shows error rates for single-step attacks transferred between models. We compute perturbations on one model (the source) and transfer them to all others (the targets). When the source and target are the same, the attack is white-box. Adversarial training greatly increases robustness to white-box single-step attacks, but incurs a higher error rate in a black-box setting. Thus, the robustness gain observed when evaluating defended models in isolation is misleading. Given the ubiquity of this pitfall among proposed defenses against adversarial examples Carlini & Wagner (2017a); Brendel & Bethge (2017); Papernot et al. (2016b), we advise researchers *to always consider both white-box and black-box adversaries when evaluating defensive strategies*.

Attacks crafted on adversarial models are found to be weaker even against undefended models (i.e., when using $v3_{adv}$ or $IRv2_{adv}$ as source, the attack transfers with lower probability). This confirms our intuition from Section 3.3: adversarial training does not just overfit to perturbations that affect standard models, but actively degrades the linear approximation underlying the single-step attack.

A new randomized single-step attack. The loss function visualization in Figure 1 shows that sharp curvature artifacts localized near the data points can mask the true direction of steepest ascent. We thus suggest to prepend single-step attacks by a small random step, in order to "escape" the non-smooth vicinity of the data point before linearizing the model's loss. Our new attack, called R+FGSM (alternatively, R+Step-LL), is defined as follows, for parameters ϵ and α (where $\alpha < \epsilon$):

$$x^{\text{adv}} = x' + (\varepsilon - \alpha) \cdot \texttt{sign}\left(\nabla_{x'} J(x', y_{\text{true}})\right), \quad \text{where} \quad x' = x + \alpha \cdot \texttt{sign}(\mathcal{N}(\mathbf{0}^d, \mathbf{I}^d)) \,. \tag{4}$$

Table 2: **Error rates (in %) for Step-LL and R+Step-LL on ImageNet.** We use $\epsilon = 16/256$, $\alpha = \epsilon/2$ on 10,000 random test set inputs. Results for R+FGSM on MNIST models are in Table 7.

	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$
Step-LL	60.2	69.6	26.6	50.7	21.4	31.0	42.7	9.0	24.0	5.8
R+Step-LL	**70.5**	**80.0**	**64.8**	**56.3**	**37.5**	**42.8**	**57.1**	**37.1**	**29.3**	**15.0**
	Top 1					**Top 5**				

Note that the attack requires a single gradient computation. The R+FGSM is a computationally efficient alternative to iterative methods that have high success rates in a white-box setting. Our attack can be seen as a single-step variant of the general PGD method from (Madry et al., 2017).

Table 2 compares error rates for the Step-LL and R+Step-LL methods (with $\epsilon = 16/256$ and $\alpha = \epsilon/2$). The extra random step yields a stronger attack for all models, even those without adversarial training. This suggests that a model's loss function is generally less smooth near the data points.

We find that the addition of this random step hinders transferability (see Table 9). We also tried adversarial training using R+FGSM on MNIST, using a similar approach as (Madry et al., 2017). We adversarially train a CNN (model A in Table 5) for 100 epochs, and attain $> 90.0\%$ accuracy on R+FGSM samples. However, training on R+FGSM provides only little robustness to iterative attacks. For the PGD attack of (Madry et al., 2017) with 20 steps, the model attains 18.0% accuracy.

4.2 Ensemble Adversarial Training

We now evaluate our Ensemble Adversarial Training strategy described in Section 3.4. We recall our intuition: by augmenting training data with adversarial examples crafted from *static pre-trained models*, we decouple the generation of adversarial examples from the model being trained, so as to avoid the degenerate minimum described in Section 3.3. Moreover, our hope is that robustness to attacks transferred from some fixed set of models will generalize to other black-box adversaries.

Table 3: **Models used for Ensemble Adversarial Training on ImageNet.** The ResNets (He et al., 2016) use either 50 or 101 layers. IncRes stands for Inception ResNet (Szegedy et al., 2016a).

Trained Model	Pre-trained Models	Holdout Models
Inception v3 ($v3_{adv\text{-}ens3}$)	Inception v3, ResNet v2 (50)	Inception v4
Inception v3 ($v3_{adv\text{-}ens4}$)	Inception v3, ResNet v2 (50), IncRes v2	ResNet v1 (50)
IncRes v2 ($IRv2_{adv\text{-}ens}$)	Inception v3, IncRes v2	ResNet v2 (101)

We train Inception v3 and Inception ResNet v2 models (Szegedy et al., 2016a) on ImageNet, using the pre-trained models shown in Table 3. In each training batch, we rotate the source of adversarial examples between the currently trained model and one of the pre-trained models. We select the source model *at random* in each batch, to diversify examples across epochs. The pre-trained models' gradients can be precomputed for the full training set. *The per-batch cost of Ensemble Adversarial Training is thus lower than that of standard adversarial training*: using our method with $n-1$ pre-trained models, only every n^{th} batch requires a forward-backward pass to compute adversarial gradients. We use synchronous distributed training on 50 machines, with minibatches of size 16 (we did not pre-compute gradients, and thus lower the batch size to fit all models in memory). Half of the examples in a minibatch are replaced by Step-LL examples. As in (Kurakin et al., 2017b), we use RMSProp with a learning rate of 0.045, decayed by a factor of 0.94 every two epochs.

To evaluate how robustness to black-box attacks generalizes across models (our conjecture in Section 3.4), we transfer various attacks crafted on three different holdout models (see Table 3). We use the **Step-LL**, **R+Step-LL**, **FGSM**, **I-FGSM** and the **PGD** attack from Madry et al. (2017) using the hinge-loss function from Carlini & Wagner (2017a). Our results are in Table 4. For each model, we report the *worst-case* error rate over all black-box attacks transfered from each of the holdout models (15 attacks in total). Additional results on MNIST are in Table 8.

Convergence speed. Convergence of Ensemble Adversarial Training is slower than for standard adversarial training, a result of training on "hard" adversarial examples and lowering the batch size. Kurakin et al. (2017b) report that after 187 epochs ($150k$ iterations with minibatches of size 32),

Table 4: **Error rates (in %) for Ensemble Adversarial Training on ImageNet.** Error rates on clean data are computed over the full test set. For 10,000 random test set inputs, and $\epsilon = 16/256$, we report error rates on white-box Step-LL and the *worst-case error* over a series of black-box attacks (*Step-LL, R+Step-LL, FGSM, I-FGSM, PGD*) transferred from the holdout models in Table 3. For both architectures, we mark methods tied for best in bold (based on 95% confidence).

	Top 1			Top 5		
Model	**Clean**	**Step-LL**	**Max. Black-Box**	**Clean**	**Step-LL**	**Max. Black-Box**
v3	**22.0**	69.6	51.2	**6.1**	42.7	24.5
$v3_{adv}$	**22.0**	**26.6**	40.8	**6.1**	**9.0**	17.4
$v3_{adv\text{-}ens3}$	23.6	30.0	**34.0**	7.6	**10.1**	**11.2**
$v3_{adv\text{-}ens4}$	24.2	43.3	**33.4**	7.8	19.4	**10.7**
IRv2	**19.6**	50.7	44.4	**4.8**	24.0	17.8
$IRv2_{adv}$	**19.8**	**21.4**	34.5	**4.9**	**5.8**	11.7
$IRv2_{adv\text{-}ens}$	**20.2**	26.0	**27.0**	**5.1**	7.6	**7.9**

the $v3_{adv}$ model achieves 78% accuracy. Ensemble Adversarial Training for models $v3_{adv\text{-}ens3}$ and $v3_{adv\text{-}ens4}$ converges after 280 epochs ($450k$ iterations with minibatches of size 16). The Inception ResNet v2 model is trained for 175 epochs, where a baseline model converges at around 160 epochs.

White-box attacks. For both architectures, the models trained with Ensemble Adversarial Training are slightly less accurate on clean data, compared to standard adversarial training. Our models are also more vulnerable to white-box single-step attacks, as they were only partially trained on such perturbations. Note that for $v3_{adv\text{-}ens4}$, the proportion of white-box Step-LL samples seen during training is $1/4$ (instead of $1/3$ for model $v3_{adv\text{-}ens3}$). The negative impact on the robustness to white-box attacks is large, for only a minor gain in robustness to transferred samples.

Ensemble Adversarial Training is not robust to *white-box* Iter-LL and R+Step-LL samples: the error rates are similar to those for the $v3_{adv}$ model, and omitted for brevity (see Kurakin et al. (2017b) for Iter-LL attacks and Table 2 for R+Step-LL attacks). Kurakin et al. (2017b) conjecture that larger models are needed to attain robustness to such attacks. Yet, against black-box adversaries, these attacks are only a concern insofar as they reliably transfer between models.

Black-box attacks. Ensemble Adversarial Training significantly boosts robustness to *all* attacks transferred from the holdout models, thus supporting our conjecture from Section 3.4 that robustness transfers across models. For the $IRv2_{adv\text{-}ens}$ model, the accuracy loss (compared to IRv2's accuracy on clean data) is **7.4% (top 1)** and **3.1% (top 5)**. We find that the strongest attacks in our test suite (i.e., with highest transfer rates) are single-step attacks. Black-box R+Step-LL or iterative attacks are less effective, as they do not transfer with high probability (see (Kurakin et al., 2017b) and Table 9). We also tried the I-FGSM crafted on an ensemble of all three holdout models, as in (Liu et al., 2017) but we also found this attack to be inferior to the transfered single-step attacks.

Our results have little variance with respect to the attack parameters (e.g., smaller ϵ) or to the use of other holdout models for black-box attacks (e.g., we obtain similar results by attacking the $v3_{adv\text{-}ens3}$ and $v3_{adv\text{-}ens4}$ models with the IRv2 model). We also find that $v3_{adv\text{-}ens3}$ is not vulnerable to perturbations transferred from $v3_{adv\text{-}ens4}$. We obtain similar results on MNIST (see Appendix B.2), thus demonstrating the applicability of our approach to different datasets and model architectures.

Our Inception ResNet v2 model was included as a baseline defense in the NIPS 2017 competition on Adversarial Examples (Kurakin et al., 2017c). The model was evaluated on other users' black-box attacks, and *finished 1st among* 70 *submissions* in the first development round. After the first round, we released our model publicly, which enabled other users to launch white-box attacks against it. Nevertheless, a majority of the teams' final submissions were ensembles that included our model.

Decreasing gradient masking. Ensemble Adversarial Training decreases the magnitude of the gradient masking effect described previously. For the $v3_{adv\text{-}ens3}$ and $v3_{adv\text{-}ens4}$ models, we find that the loss incurred on a Step-LL attack gets within respectively 13% and 18% of the optimum loss (we recall that for models v3 and $v3_{adv}$, the approximation ratio was respectively 19% and 7%). Similarly, for the $IRv2_{adv\text{-}ens}$ model, the ratio improves from 8% (for $IRv2_{adv}$) to 14%. As expected, not solely training on a white-box single-step attack decreases the gradient masking effect.

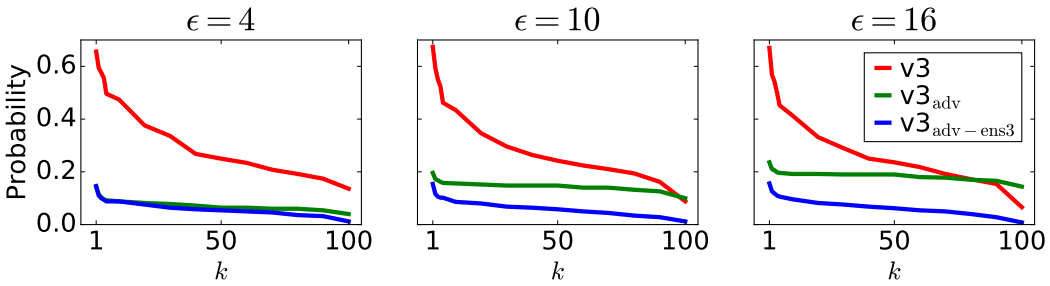


Figure 2: **The dimensionality of the adversarial cone**. For 500 correctly classified points x, and for $\epsilon \in \{4, 10, 16\}$, we plot the probability that we find *at least* k orthogonal vectors r_i such that $\|r_i\|_\infty = \epsilon$ and $x + r_i$ is misclassified. For $\epsilon \geq 10$, model v3_{adv} shows a *bimodal* phenomenon: most points x either have 0 adversarial directions or more than 90.

Finally, we revisit the "Gradient-Aligned Adversarial Subspace" (GAAS) method of Tramèr et al. (2017). Their method estimates the size of the space of adversarial examples in the vicinity of a point, by finding a set of *orthogonal* perturbations of norm ϵ that are all adversarial. We note that adversarial perturbations do not technically form a "subspace" (e.g., the $\mathbf{0}$ vector is not adversarial). Rather, they may form a "cone", the dimension of which varies as we increase ϵ. By linearizing the loss function, estimating the dimensionality of this cone reduces to finding vectors r_i that are strongly aligned with the model's gradient $g = \nabla_x L(h(x), y_{\text{true}})$. Tramèr et al. (2017) give a method that finds k orthogonal vectors r_i that satisfy $g^\top r_i \geq \epsilon \cdot \|g\|_2 \cdot \frac{1}{\sqrt{k}}$ (this bound is tight). We extend this result to the ℓ_∞ norm, an open question in (Tramèr et al., 2017). In Section D, we give a randomized combinatorial construction (Colbourn, 2010), that finds k orthogonal vectors r_i satisfying $\|r_i\|_\infty = \epsilon$ and $\mathbb{E}\left[g^\top r_i\right] \geq \epsilon \cdot \|g\|_1 \cdot \frac{1}{\sqrt{k}}$. We show that this result is tight as well.

For models v3, v3_{adv} and $\text{v3}_{\text{adv-ens3}}$, we select 500 correctly classified test points. For each x, we search for a maximal number of orthogonal adversarial perturbations r_i with $\|r_i\|_\infty = \epsilon$. We limit our search to $k \leq 100$ directions per point. The results are in Figure 2. For $\epsilon \in \{4, 10, 16\}$, we plot the proportion of points that have at least k orthogonal adversarial perturbations. For a fixed ϵ, the value of k can be interpreted as the dimension of a "slice" of the cone of adversarial examples near a data point. For the standard Inception v3 model, we find over 50 orthogonal adversarial directions for 30% of the points. The v3_{adv} model shows a curious bimodal phenomenon for $\epsilon \geq 10$: for most points ($\approx 80\%$), we find no adversarial direction aligned with the gradient, which is consistent with the gradient masking effect. Yet, for most of the remaining points, the adversarial space is very high-dimensional ($k \geq 90$). Ensemble Adversarial Training yields a more robust model, with only a small fraction of points near a large adversarial space.

5 Conclusion and Future Work

Previous work on adversarial training at scale has produced encouraging results, showing strong robustness to (single-step) adversarial examples (Goodfellow et al., 2014b; Kurakin et al., 2017b). Yet, these results are misleading, as the adversarially trained models remain vulnerable to simple black-box and white-box attacks. Our results, generic with respect to the application domain, suggest that adversarial training can be improved by *decoupling* the generation of adversarial examples from the model being trained. Our experiments with Ensemble Adversarial Training support our conjecture from Section 3.4, that robustness to attacks from some models *transfers* to other models.

We did not consider black-box adversaries that attack a model via other means than by transferring examples from a local model (see Appendix A for a formal definition). For instance, generative techniques (Baluja & Fischer, 2017) might provide an avenue for stronger attacks. Moreover, *adaptive* adversaries (also defined in Appendix A) could try to exploit queries to the target model's prediction function in their attack, as demonstrated in (Papernot et al., 2017). If queries to the target model yield full *prediction confidences*, an adversary could estimate the target's gradient at a given point (e.g., using finite-differences) and fool the target with our R+FGSM attack. If queries only return the predicted label, the attack does not apply. Exploring the impact of these classes of black-box attacks and evaluating their scalability to complex tasks is an interesting avenue for future work.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML-KDD*, pp. 387–402. Springer, 2013.

Wieland Brendel and Matthias Bethge. Comment on" biologically inspired protection of deep networks from adversarial attacks". *arXiv preprint arXiv:1704.01547*, 2017.

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

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *arXiv preprint arXiv:1705.07263*, 2017b.

Moustapha Cisse, Bojanowski Piotr, Grave Edouard, Dauphin Yann, and Usunier Nicolas. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.

Charles J Colbourn. *CRC handbook of combinatorial designs*. CRC press, 2010.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2017a.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017b.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Nips 2017: Defense against adversarial attack, 2017c. URL `https://www.kaggle.com/c/nips-2017-defense-against-adversarial-attack`.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.

Yan Luo, Xavier Boix, Gemma Roig, Tomaso Poggio, and Qi Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of convolution neural network advances on the imagenet. *Computer Vision and Image Understanding*, 2017.

Aran Nayebi and Surya Ganguli. Biologically inspired protection of deep networks from adversarial attacks. *arXiv preprint arXiv:1703.09202*, 2017.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pp. 372–387. IEEE, 2016a.

Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016b.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pp. 582–597. IEEE, 2016c.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security (ASIACCS)*, pp. 506–519. ACM, 2017.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016a.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pp. 2818–2826, 2016b.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Usenix Security*, 2016.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

A THREAT MODEL: FORMAL DEFINITIONS

We provide formal definitions for the threat model introduced in Section 3.1. In the following, we explicitly identify the hypothesis space $\mathcal{H}$ that a model belongs to as describing the model's *architecture*. We consider a target model $h \in \mathcal{H}$ trained over inputs (x, y_{true}) sampled from a data distribution $\mathcal{D}$. More precisely, we write

$$h \leftarrow \texttt{train}(\mathcal{H}, X_{\text{train}}, Y_{\text{train}}, r) \ ,$$

where `train` is a *randomized training procedure* that takes in a description of the model architecture $\mathcal{H}$, a training set $X_{\text{train}}, Y_{\text{train}}$ sampled from $\mathcal{D}$, and randomness r.

Given a set of test inputs $X, Y = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ from $\mathcal{D}$ and a budget $\epsilon > 0$, an adversary $\mathcal{A}$ produces adversarial examples $X^{\text{adv}} = \{x_1^{\text{adv}}, \ldots, x_m^{\text{adv}}\}$, such that $\|x_i - x_i^{\text{adv}}\|_\infty \leq \epsilon$ for all $i \in [1, m]$. We evaluate success of the attack as the error rate of the target model over X^{adv}:

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}(\arg\max h(x_i^{\text{adv}}) \neq y_i) \ .$$

We assume $\mathcal{A}$ can sample inputs according to the data distribution $\mathcal{D}$. We define three adversaries.

Definition 1 (White-Box Adversary)**.** *For a target model $h \in \mathcal{H}$, a white-box adversary is given access to all elements of the training procedure, that is* `train` *(the training algorithm), $\mathcal{H}$ (the model architecture), the training data X_{train}, Y_{train}, the randomness r and the parameters h. The adversary can use attacks such as those in Section 3.2 to find adversarial inputs.*

White-box access to the internal model weights corresponds to a very strong adversarial model. We thus also consider the following relaxed and arguably more realistic notion of a *black-box* adversary.

Definition 2 (*Non-Adaptive* Black-Box Adversary)**.** *For a target model $h \in \mathcal{H}$, a non-adaptive black-box adversary only gets access to* `train` *(the target model's training procedure) and $\mathcal{H}$ (the model architecture). The adversary then chooses a procedure* $\texttt{train}'$ *and model architecture $\mathcal{H}'$ and trains a local model h' over samples from the data distribution $\mathcal{D}$. The adversary crafts adversarial examples X^{adv} on the local model h' using white-box attack strategies and applies these crafted inputs to the target model.*

Most importantly, a black-box adversary does not learn the randomness r used to train the target, nor the target's parameters h. The black-box adversaries in our paper are actually slightly *stronger* than the ones defined above, in that they use the same training data $X_{\text{train}}, Y_{\text{train}}$ as the target model.

We provide $\mathcal{A}$ with the target's training procedure `train` to capture knowledge of *defensive strategies* applied at training time, e.g., adversarial training Szegedy et al. (2013); Goodfellow et al. (2014b) or ensemble adversarial training (see Section 4.2). The local training procedure $\texttt{train}'$ and model architecture $\mathcal{H}'$ chosen by the adversary may be different from those used to train the target. For ensemble adversarial training, $\mathcal{A}$ also knows the architectures of all pre-trained models. In this work, we always mount black-box attacks that train a local model with a *different* architecture than the target model. We actually find that black-box attacks on adversarially trained models are stronger in this case (see Table 1).

The main focus of our paper is on non-adaptive black-box adversaries as defined above. For completeness, we also formalize a stronger notion of *adaptive* black-box adversaries that additionally issue prediction queries to the target model Papernot et al. (2017).

Definition 3 (*Adaptive* Black-Box Adversary)**.** *For a target model $h \in \mathcal{H}$, an adaptive black-box adversary only gets access to* `train` *(the target model's training procedure) and $\mathcal{H}$ (the model architecture). The adversary issues (adaptive) oracle queries to the target model. That is, for arbitrary inputs $x \in [0, 1]^d$, the adversary obtains $y = \arg\max h(x)$. The adversary then chooses a procedure* $\texttt{train}'$ *and model architecture $\mathcal{H}'$, trains a surrogate model over tuples (x, y) obtained from querying the target model, and uses this local model to craft adversarial examples.*

Papernot et al. Papernot et al. (2017) show that such attacks are possible even if the adversary only gets access to a small number of samples from $\mathcal{D}$. Note that if the target model's prediction interface returns the full vector of class scores $h(x)$, adaptive black-box adversaries could use queries to the target model to estimate the model's gradient (e.g., using finite differences), and then apply the attacks in Section 3.2. We further discuss adaptive black-box attack strategies in Section 5.

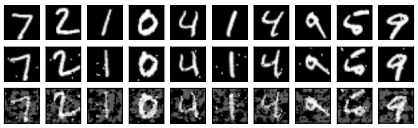


Figure 3: **Adversarial Examples on MNIST.** (top) clean examples. (middle) inputs are rotated by $20°$ and 5 random pixels are flipped. (bottom) The I-FGSM with $\epsilon = 0.3$ is applied.

B EXPERIMENTS ON MNIST

We re-iterate our ImageNet experiments on MNIST. For this simpler task, Madry et al. (2017) show that training on iterative attacks conveys robustness to white-box attacks with bounded ℓ_∞ norm. Our goal is not to attain similarly strong white-box robustness on MNIST, but to show that our observations on limitations of single-step adversarial training, extend to other datasets than ImageNet.

B.1 A NOTE ON ℓ_∞ ROBUSTNESS ON MNIST

The MNIST dataset is a simple baseline for assessing the potential of a defense, but the obtained results do not always generalize to harder tasks. We suggest that this is because achieving robustness to ℓ_∞ perturbations admits a simple "closed-form" solution, given the near-binary nature of the data. Indeed, for an average MNIST image, over 80% of the pixels are in $\{0, 1\}$ and only 6% are in the range $[0.2, 0.8]$. Thus, for a perturbation with $\epsilon \leq 0.3$, binarized versions of x and x^{adv} can differ in at most 6% of the input dimensions. By binarizing the inputs of a standard CNN trained *without adversarial training*, we obtain a model that enjoys robustness similar to the model trained by Madry et al. (2017). Concretely, for a white-box I-FGSM attack, we get at most 11.4% error.

The existence of such a simple robust representation begs the question of why learning a robust model with adversarial training takes so much effort. Finding techniques to improve the performance of adversarial training, even on simple tasks, could provide useful insights for more complex tasks such as ImageNet, where we do not know of a similarly simple "denoising" procedure.

These positive results on MNIST for the ℓ_∞ norm also leave open the question of defining a general norm for adversarial examples. Let us motivate the need for such a definition: we find that if we first *rotate* an MNIST digit by $20°$, and then use the I-FGSM, our rounding model and the model from Madry et al. (2017) achieve only 65% accuracy (on "clean" rotated inputs, the error is $< 5\%$). If we further randomly "flip" 5 pixels per image, the accuracy of both models drops to under 50%. Thus, we successfully evade the model by slightly extending the threat model (see Figure 3).

Of course, we could augment the training set with such perturbations. *An open question is whether we can enumerate all types of "adversarial" perturbations.* In this work, we focus on the ℓ_∞ norm to illustrate our findings on the limitations of single-step adversarial training on ImageNet and MNIST, and to showcase the benefits of our Ensemble Adversarial Training variant. Our approach can easily be extended to consider multiple perturbation metrics. We leave such an evaluation to future work.

B.2 RESULTS

We repeat experiments from Section 4 on MNIST. We use the architectures in Table 5. We train a standard model for 6 epochs, and an adversarial model with the FGSM ($\epsilon = 0.3$) for 12 epochs.

Table 5: **Neural network architectures used in this work for the MNIST dataset.** Conv: convolutional layer, FC: fully connected layer.

A	B	C	D
Conv(64, 5, 5) + Relu	Dropout(0.2)	Conv(128, 3, 3) + Tanh	[FC(300) + Relu, Dropout(0.5)] $\times 4$
Conv(64, 5, 5) + Relu	Conv(64, 8, 8) + Relu	MaxPool(2,2)	FC + Softmax
Dropout(0.25)	Conv(128, 6, 6) + Relu	Conv(64, 3, 3) + Tanh	
FC(128) + Relu	Conv(128, 5, 5) + Relu	MaxPool(2,2)	
Dropout(0.5)	Dropout(0.5)	FC(128) + Relu	
FC + Softmax	FC + Softmax	FC + Softmax	

Table 6: **Approximation ratio between optimal loss and loss induced by single-step attack on MNIST.** Architecture B' is the same as B without the input dropout layer.

A	A_{adv}	B	B_{adv}	B^*	B^*_{adv}	C	C_{adv}	D	D_{adv}
17%	0%	25%	8%	23%	1%	25%	0%	49%	16%

During adversarial training, we avoid the *label leaking* effect described by Kurakin et al. (2017b) by using the model's predicted class $\arg\max h(x)$ instead of the true label y_{true} in the FGSM,

We first analyze the "degenerate" minimum of adversarial training, described in Section 3.3. For each trained model, we compute the *approximation-ratio* of the FGSM for the inner maximization problem in equation (1). That is, we compare the loss produced by the FGSM with the loss of a strong iterative attack. The results appear in Table 6. As we can see, for all model architectures, adversarial training degraded the quality of a linear approximation to the model's loss.

We find that *input dropout* Srivastava et al. (2014) (i.e., randomly dropping a fraction of input features during training) as used in architecture B limits this unwarranted effect of adversarial training.[2] If we omit the input dropout (we call this architecture B^*) the single-step attack degrades significantly. We discuss this effect in more detail below. For the fully connected architecture D, we find that the learned model is very close to linear and thus also less prone to the degenerate solution to the min-max problem, as we postulated in Section 3.3.

Attacks. Table 7 compares error rates of undefended and adversarially trained models on white-box and black-box attacks, as in Section 4.1. Again, model B presents an anomaly. For all other models, we corroborate our findings on ImageNet for adversarial training: (1) black-box attacks trump white-box single-step attacks; (2) white-box single-step attacks are significantly stronger if prepended by a random step. For model B_{adv}, the opposite holds true. We believe this is because input dropout increases diversity of attack samples similarly to Ensemble Adversarial Training.

Table 7: **White-box and black-box attacks against standard and adversarially trained models.** For each model, the strongest single-step white-box and black box attacks are marked in bold.

	white-box		**black-box**				
	FGSM	R+FGSM	$FGSM_A$	$FGSM_B$	$FGSM_{B^*}$	$FGSM_C$	$FGSM_D$
A	64.7	**69.7**	-	**61.5**	53.2	46.8	41.5
A_{adv}	2.2	**14.8**	6.6	**10.7**	8.8	6.5	8.3
B	85.0	**86.0**	45.7	-	69.9	59.9	**85.9**
B_{adv}	**11.6**	11.1	6.4	**8.9**	8.5	4.9	6.1
B^*	**75.7**	74.1	44.3	**72.8**	-	46.0	62.6
B^*_{adv}	4.3	**40.6**	16.1	14.7	15.0	**17.9**	9.1
C	**81.8**	**81.8**	40.2	55.8	49.5	-	**59.4**
C_{adv}	3.7	**17.1**	9.8	**29.3**	21.5	11.9	21.9
D	92.4	**95.4**	61.3	**74.1**	68.9	65.1	-
D_{adv}	25.5	**47.5**	**32.1**	30.5	29.3	28.2	21.8

While training with input dropout helps avoid the degradation of the single-step attack, it also significantly delays convergence of the model. Indeed, model B_{adv} retains relatively high error on white-box FGSM examples. Adversarial training with input dropout can be seen as comparable to training with a randomized single-step attack, as discussed in Section 4.1.

The positive effect of input dropout is architecture and dataset specific: Adding an input dropout layer to models A, C and D confers only marginal benefit, and is outperformed by Ensemble Adversarial Training, discussed below. Moreover, Mishkin et al. (2017) find that input dropout significantly degrades accuracy on ImageNet. We thus did not incorporate it into our models on ImageNet.

Ensemble Adversarial Training. To evaluate Ensemble Adversarial Training 3.4, we train two models per architecture. The first, denoted $[\text{A-D}]_{\text{adv-ens}}$, uses a single pre-trained model of the same

[2] We thank Arjun Bhagoji, Bo Li and Dawn Song for this observation.

Table 8: **Ensemble Adversarial Training on MNIST.** For black-box robustness, we report the maximum and average error rate over a suite of 12 attacks, comprised of the FGSM, I-FGSM and PGD (Madry et al., 2017) attacks applied to models A,B,C and D. We use $\epsilon = 16$ in all cases. For each model architecture, we mark the models tied for best (at a 95% confidence level) in bold.

	Clean	**FGSM**	**Max. Black Box**	**Avg. Black Box**
A_{adv}	**0.8**	**2.2**	10.8	7.7
$A_{adv\text{-}ens}$	**0.8**	7.0	**6.6**	5.2
$A_{adv\text{-}ens3}$	**0.7**	5.4	**6.5**	**4.3**
B_{adv}	**0.8**	**11.6**	8.9	**5.5**
$B_{adv\text{-}ens}$	**0.7**	**10.5**	**6.8**	**5.3**
$B_{adv\text{-}ens3}$	**0.8**	14.0	8.8	**5.1**
C_{adv}	**1.0**	3.7	29.3	18.7
$C_{adv\text{-}ens}$	**1.3**	**1.9**	17.2	10.7
$C_{adv\text{-}ens3}$	**1.4**	3.6	**14.5**	**8.4**
D_{adv}	**2.6**	25.5	32.5	23.5
$D_{adv\text{-}ens}$	**2.6**	**21.5**	38.6	28.0
$D_{adv\text{-}ens3}$	**2.6**	29.4	**29.8**	**15.6**

type (i.e., $A_{adv\text{-}ens}$ is trained on perturbations from another model A). The second model, denoted $[A\text{-}D]_{adv\text{-}ens3}$, uses 3 pre-trained models ($\{A, C, D\}$ or $\{B, C, D\}$). We train all models for 12 epochs.

We evaluate our models on black-box attacks crafted on models A,B,C,D (for a fair comparison, we do not use the same pre-trained models for evaluation, but retrain them with different random seeds). The attacks we consider are the FGSM, I-FGSM and the PGD attack from Madry et al. (2017) with the loss function from Carlini & Wagner (2017a)), all with $\epsilon = 0.3$. The results appear in Table 8. For each model, we report the worst-case and average-case error rate over all black-box attacks.

Ensemble Adversarial Training significantly increases robustness to black-box attacks, except for architecture B, which we previously found to not suffer from the same overfitting phenomenon that affects the other adversarially trained networks. Nevertheless, model $B_{adv\text{-}ens}$ achieves slightly better robustness to white-box and black-box attacks than B_{adv}. In the majority of cases, we find that using a single pre-trained model produces good results, but that the extra diversity of including three pre-trained models can sometimes increase robustness even further. Our experiments confirm our conjecture that robustness to black-box attacks generalizes across models. Indeed, we find that when training with three external models, we attain very good robustness against attacks initiated from models with the same architecture (as evidenced by the average error on our attack suite), but also increased robustness to attacks initiated from the fourth holdout model

C TRANSFERABILITY OF RANDOMIZED SINGLE-STEP PERTURBATIONS.

In Section 4.1, we introduced the R+Step-LL attack, an extension of the Step-LL method that prepends the attack with a small random perturbation. In Table 9, we evaluate the transferability of R+Step-LL adversarial examples on ImageNet. We find that the randomized variant produces perturbations that transfer at a much lower rate (see Table 1 for the deterministic variant).

Table 9: **Error rates (in %) of randomized single-step attacks transferred between models on ImageNet.** We use R+Step-LL with $\epsilon = 16/256, \alpha = \epsilon/2$ for 10,000 random test set samples. The white-box attack always outperforms black-box attacks.

	Source						**Source**				
Target	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$	**Target**	v4	v3	$v3_{adv}$	IRv2	$IRv2_{adv}$
v4	**70.5**	37.2	23.2	34.0	24.6	v4	**42.8**	14.3	6.3	11.9	6.9
v3	42.6	**80.0**	26.7	38.5	27.6	v3	18.0	**57.1**	8.0	15.6	8.6
$v3_{adv}$	31.4	30.7	**64.8**	30.4	34.0	$v3_{adv}$	10.7	10.4	**37.1**	10.1	12.9
IRv2	36.2	35.7	23.0	**56.3**	24.6	IRv2	12.8	13.6	6.1	**29.3**	7.0
$IRv2_{adv}$	26.8	26.3	25.2	26.9	**37.5**	$IRv2_{adv}$	8.0	8.0	7.7	8.3	**15.0**
	Top 1						**Top 5**				

D GRADIENT ALIGNED ADVERSARIAL SUBSPACES FOR THE ℓ_∞ NORM

Tramèr et al. (2017) consider the following task for a given model h: for a (correctly classified) point x, find k orthogonal vectors $\{r_1, \ldots, r_k\}$ such that $\|r_i\|_2 \leq \epsilon$ and all the $x + r_i$ are adversarial (i.e., $\arg\max h(x + r_i) \neq y_{\text{true}}$). By linearizing the model's loss function, this reduces to finding k orthogonal vectors r_i that are maximally aligned with the model's gradient $g = \nabla_x L(h(x), y_{\text{true}})$. Tramèr et al. (2017) left a construction for the ℓ_∞ norm as an open problem.

We provide an optimal construction for the ℓ_∞ norm, based on *Regular Hadamard Matrices* (Colbourn, 2010). Given the ℓ_∞ constraint, we find orthogonal vectors r_i that are maximally aligned with the *signed* gradient, $\texttt{sign}(g)$. We first prove an analog of (Tramèr et al., 2017, Lemma 1).

Lemma 4. *Let $v \in \{-1, 1\}^d$ and $\alpha \in (0, 1)$. Suppose there are k orthogonal vectors $r_1, \ldots r_n \in \{-1, 1\}^d$ satisfying $v^\top r_i \geq \alpha \cdot d$. Then $\alpha \leq k^{-\frac{1}{2}}$.*

Proof. Let $\hat{r_i} = \frac{r_i}{\|r_i\|_2} = \frac{r_i}{\sqrt{d}}$. Then, we have

$$d = \|v\|_2^2 \geq \sum_{i=1}^{k} |v^\top \hat{r_i}|^2 = d^{-1} \sum_{i=1}^{k} |v^\top r_i|^2 \geq d^{-1} \cdot k \cdot (\alpha \cdot d)^2 = k \cdot \alpha^2 \cdot d \,, \tag{5}$$

from which we obtain $\alpha \leq k^{-\frac{1}{2}}$. □

This result bounds the number of orthogonal perturbations we can expect to find, for a given alignment with the signed gradient. As a warm-up consider the following trivial construction of k orthogonal vectors in $\{-1, 1\}^d$ that are "somewhat" aligned with $\texttt{sign}(g)$. We split $\texttt{sign}(g)$ into k "chunks" of size $\frac{d}{k}$ and define r_i to be the vector that is equal to $\texttt{sign}(g)$ in the ith chunk and zero otherwise. We obtain $\texttt{sign}(g)^\top r_i = \frac{d}{k}$, a factor $\sqrt{k}$ worse than the the bound in Lemma 4.

We now provide a construction that meets this upper bound. We make use of Regular Hadamard Matrices of order k (Colbourn, 2010). These are square matrices H_k such that: (1) all entries of H_k are in $\{-1, 1\}^k$; (2) the rows of H_k are mutually orthogonal; (3) All row sums are equal to $\sqrt{k}$.

The order of a Regular Hadamard Matrix is of the form $4u^2$ for an integer u. We use known constructions for $k \in \{4, 16, 36, 64, 100\}$.

Lemma 5. *Let $g \in \mathbb{R}^d$ and k be an integer for which a Regular Hadamard Matrix of order k exists. Then, there is a randomized construction of k orthogonal vectors $r_1, \ldots r_n \in \{-1, 1\}^d$, such that $\texttt{sign}(g)^\top r_i = d \cdot k^{-1/2}$. Moreover, $\mathbb{E}[g^\top r_i] = k^{-1/2} \cdot \|g\|_1$.*

Proof. We construct k orthogonal vectors $r_1, \ldots, r_k \in \{-1, 1\}^d$, where r_i is obtained by repeating the ith row of H_k d/k times (for simplicity, we assume that k divides d. Otherwise we pad r_i with zeros). We then multiply each r_i component-wise with $\texttt{sign}(g)$. By construction, the k vectors $r_i \in \{-1, 1\}^d$ are mutually orthogonal, and we have $\texttt{sign}(g)^\top r_i = \frac{d}{k} \cdot \sqrt{k} = d \cdot k^{-1/2}$, which is tight according to Lemma 4.

As the weight of the gradient g may not be uniformly distributed among its d components, we apply our construction to a random permutation of the signed gradient. We then obtain

$$\mathbb{E}[g^\top r_i] = \mathbb{E}\left[\sum_{j=1}^{d} |g^{(j)}| \cdot \texttt{sign}(g^{(j)}) \cdot r_i^{(j)}\right] \tag{6}$$

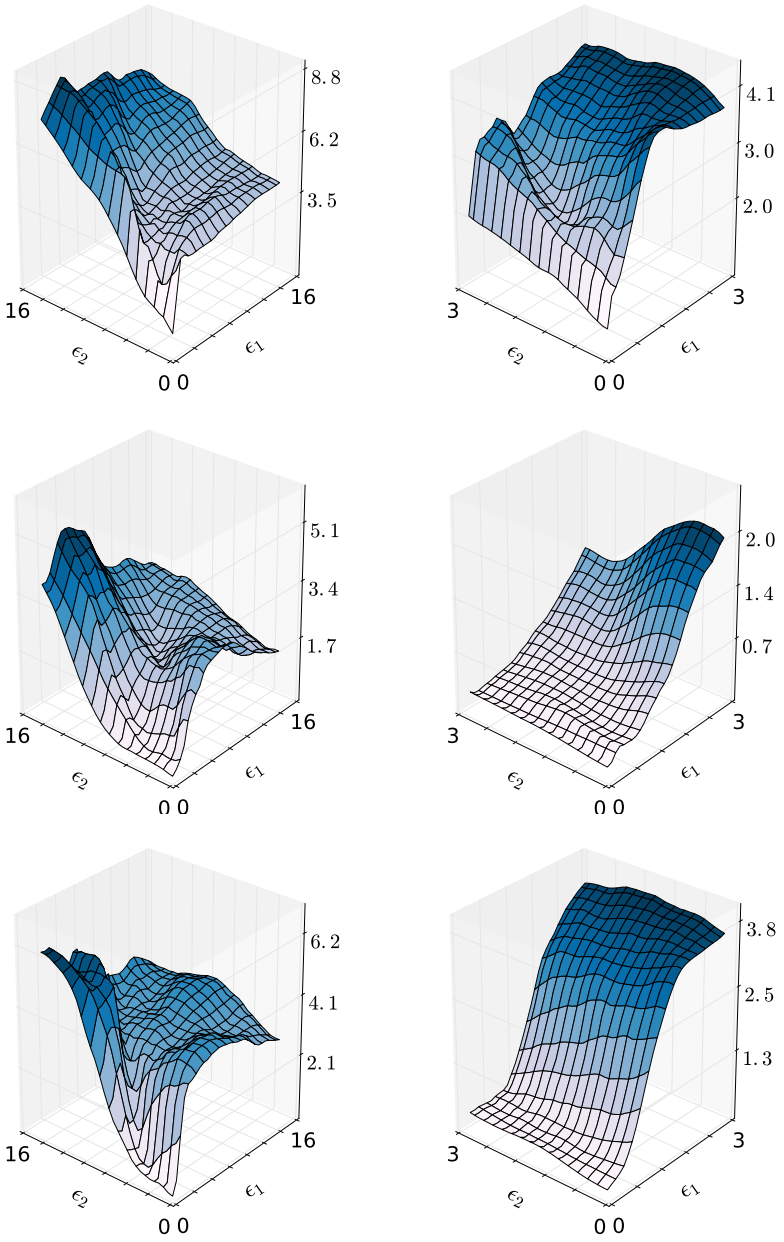
$$= \sum_{j=1}^{d} |g^{(j)}| \cdot \mathbb{E}\left[\texttt{sign}(g^{(j)}) \cdot r_i^{(j)}\right] = k^{-1/2} \cdot \|g\|_1 \,. \tag{7}$$

□

It can be shown that the bound in Lemma 5 can be attained if and only if the r_i are constructed from the rows of a Regular Hadamard Matrix (Colbourn, 2010). For general integers k for which no such matrix exists, other combinatorial designs may be useful for achieving looser bounds.

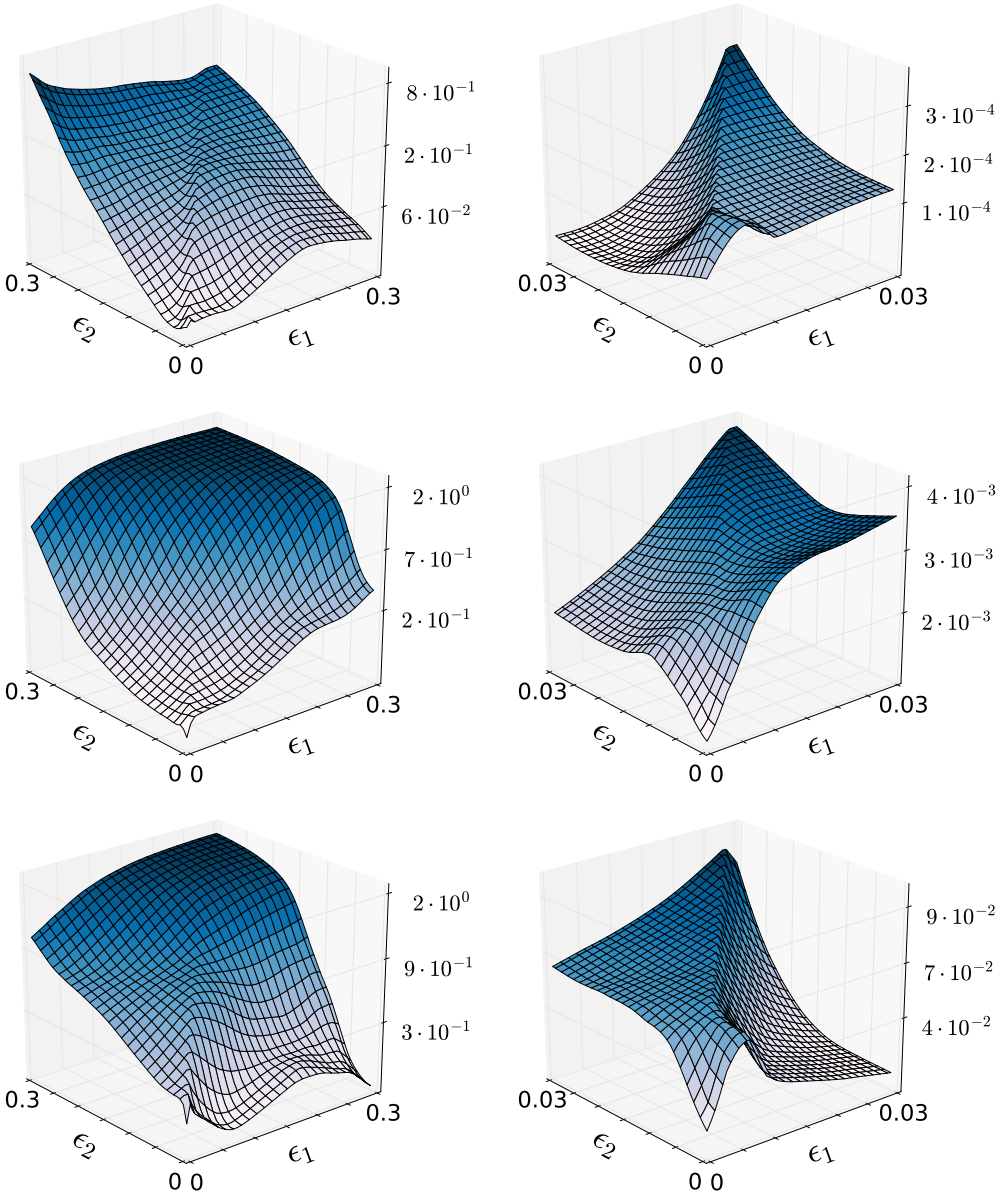
E ILLUSTRATIONS OF GRADIENT MASKING IN ADVERSARIAL TRAINING

In Section 3.3, we show that adversarial training introduces spurious curvature artifacts in the model's loss function around data points. As a result, one-shot attack strategies based on first-order approximations of the model loss produce perturbations that are non-adversarial. In Figures 4 and 5 we show further illustrations of this phenomenon for the Inception $v3_{adv}$ model trained on ImageNet by Kurakin et al. (2017b) as well as for the model A_{adv} we trained on MNIST.



(a) Loss surface for model $v3_{adv}$. (b) Zoom in of the loss for small ϵ_1, ϵ_2.

Figure 4: **Additional illustrations of the local curvature artifacts introduced by adversarial training on ImageNet.** We plot the loss of model $v3_{adv}$ on samples of the form $x^* = x + \epsilon_1 \cdot g + \epsilon_2 \cdot g^{\perp}$, where g is the signed gradient of $v3_{adv}$ and $g^{\perp}$ is an orthogonal adversarial direction, obtained from an Inception v4 model. The right-side plots are zoomed in versions of the left-side plots.



(a) Loss surface for model A_{adv} (log-scale).

(b) Zoom in of the loss for small ϵ_1, ϵ_2.

Figure 5: **Illustrations of the local curvature artifacts introduced by adversarial training on MNIST.** We plot the loss of model A_{adv} on samples of the form $x^* = x + \epsilon_1 \cdot g + \epsilon_2 \cdot g^{\perp}$, where g is the signed gradient of model A_{adv} and $g^{\perp}$ is an orthogonal adversarial direction, obtained from model B. The right-side plots are zoomed in versions of the left-side plots.