Robust Neural Networks using Randomized Adversarial Training

Alexandre Araujo, Rafael Pinot, Benjamin Negrevergne, Laurent Meunier, Yann Chevaleyre, Florian Yger, Jamal Atif

Robust Neural Networks using Randomized Adversarial Training

Alexandre Araujo[1, 2] **Rafael Pinot**[1, 3] **Benjamin Negrevergne**[1] **Laurent Meunier**[1, 4]
Yann Chevaleyre[1] **Florian Yger**[1] **Jamal Atif**[1]

[1]PSL, Université Paris-Dauphine [2]Wavestone [3]CEA, Université Paris-Saclay [4]Facebook AI Research

Abstract

Since the discovery of adversarial examples in machine learning, researchers have designed several techniques to train neural networks that are robust against different types of attacks (most notably ℓ_∞ and ℓ_2 based attacks). However, it has been observed that the defense mechanisms designed to protect against one type of attack often offer poor performance against the other. In this paper, we introduce *Randomized Adversarial Training* (RAT), a technique that is efficient both against ℓ_2 and ℓ_∞ attacks. To obtain this result, we build upon adversarial training, a technique that is efficient against ℓ_∞ attacks, and demonstrate that adding random noise at training and inference time further improves performance against ℓ_2 attacks. We then show that RAT is as efficient as adversarial training against ℓ_∞ attacks while being robust against strong ℓ_2 attacks. Our final comparative experiments demonstrate that RAT outperforms all state-of-the-art approaches against ℓ_2 *and* ℓ_∞ attacks.

1 Introduction

Modern neural networks achieve state of the art performances in a variety of domains such as natural language processing (Radford et al., 2018), image recognition (He et al., 2016) and speech recognition (Hinton et al., 2012). However, it has been shown that such neural networks can be vulnerable to *adversarial examples*, i.e. imperceptible variations of legitimate examples crafted to deliberately mislead a machine learning algorithm (Szegedy et al., 2014). In the literature, the magnitude of an acceptable variation is characterized using either the ℓ_∞ or the ℓ_2 norm, and since these two metrics have very distinct properties, the techniques used to generate adversarial examples also differ. In general, ℓ_∞ adversarial examples are easy to generate, but they are also easy to detect due to the large ℓ_2 norm of their perturbation. In contrast generating ℓ_2 adversarial examples is computationally more demanding, but it is also more difficult to generate defense mechanisms that are robust against this type of attacks.

The Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) and other derived techniques (Madry et al., 2018) can be used to generate ℓ_∞ adversarial examples that defeat off-the-shelf neural networks architectures with little computational resources. In the same work, Goodfellow et al. (2015) also proposed a defense mechanism and showed that augmenting batches with ℓ_∞ adversarial examples during training could be used to significantly improve the robustness of the models against these attacks (a.k.a. Adversarial Training). However, they do not discuss the generality of Adversarial Training against other attacks.

Since the introduction of Adversarial Training, more sophisticated attacks have been proposed to craft adversarial perturbations with small ℓ_2 norm (Carlini & Wagner, 2017; Moosavi-Dezfooli et al., 2016) defeating existing defense mechanisms. Given the success of ℓ_∞ Adversarial Training, it would be natural to consider ℓ_2 Adversarial Training as a candidate defense mechanism. Unfortunately for defenders, designing ℓ_2 adversarial training is impractical: state-of-the-art ℓ_2 adversarial training methods require much more computational resources and generating ℓ_2 adversarial examples during training is prohibitive. To cope with this limitation, several researchers have tried to design adversarial training methods based on adversarial examples generated with ℓ_∞ attacks (see e.g (Madry et al., 2018)) but these methods demonstrate surprisingly little benefit in robustness against strong ℓ_2 attacks such as C&W (Carlini & Wagner, 2017).

In this paper, we are concerned with the problem of de-

signing a defense mechanism that is simultaneously robust against ℓ_2 and ℓ_∞ attacks. We first give key insights to understand why ℓ_∞ defense mechanisms perform poorly against ℓ_2 attacks: we observe that the sets of adversarial perturbations that can be generated with existing ℓ_∞ or ℓ_2 attacks are strictly disjoint. Building on this observation we propose a new defense method called *Randomized Adversarial Training* (RAT) that combines Adversarial Training with noise injection at inference time.

Evaluating the robustness of randomized neural networks is a challenging task. Recently Athalye et al. (2018) have pointed flaws in the existing evaluation methodologies and claim that randomized networks should rather be tested against attacks in expectations. Under this new evaluation strategy, several existing methods are not as robust as they first appeared. In this paper, we follow the experimental methodology designed by Athalye et al. (2018) and evaluate RAT under attacks in expectation. Furthermore, we evaluate all our approaches against *untargeted* attacks which is the worst case scenario for defenders (an untargeted attack is successful if the attacker changes the predicted class to any other one). Our method proves to be more robust to expectation attacks than previous noise injection methods.

Our contribution: We introduce RAT, a Randomized Adversarial Training defense mechanism that combines adversarial training with noise injection at inference time, and demonstrate its efficiency against all state-of-the-art attacks. We then conduct experiments to highlight the difference between ℓ_∞ and ℓ_2 attacks. Finally, we show that RAT achieves state-of-the-art robustness against ℓ_∞ attacks in expectation, and outperforms existing defense mechanisms against ℓ_2 attacks. In other words, RAT is both more general and more robust than existing techniques. To the best of our knowledge, this is the first time that a defense mechanism obtains such good results against both types of attacks.

2 Related Work

Since the discovery of adversarial examples by Szegedy et al. (2014), a number of work have investigated the design of *attacks* to generate adversarial examples, and corresponding *defenses* to train models that are robust against these attacks. In their seminal work, Goodfellow et al. (2015) introduced the *Fast Gradient Sign Method* (FGSM), a technique that computes the gradient of the loss function with respect to an input x and perturbs x in the direction of the sign of the gradient to create an adversarial example. In the case of a linear classifier, their approach maximizes the change in the output for perturbations on the inputs that are bounded with the ℓ_∞ norm. Then they argue that neural networks are locally linear (a property that is desirable to ease the training of deep neural nets), and demonstrate empirically that FGSM is able to generate good adversarial examples for neural networks. To protect against such attacks, Goodfellow et al. (2015) also introduced the adversarial training defense technique that we describe in the introduction (training networks using examples adversarial generated with FGSM). This defense offers satisfying protection against FGSM generated adversarial examples.

The FGSM attack can be seen as a one-step gradient method and Kurakin et al. (2017); Madry et al. (2018) have proposed variants of FGSM that perform multiple gradient descent steps. More specifically, Madry et al. (2018) introduced *Projected Gradient Descent* (PGD), an attack that can be used to generate adversarial examples defeating FGSM-based adversarial training. Naturally, Madry et al. (2018) also experimented adversarial training using adversarial examples generated with PGD. Adversarial training does offer an increased protection level against PGD adversarial examples, albeit at a higher computational cost. More recently, a number of defenses have been proposed (Song et al., 2018; Dhillon et al., 2018; Samangouei et al., 2018) and demonstrate good empirical results against some of the attacks, but do not offer a general improvement in robustness according to the thorough robustness study conducted by Athalye et al. (2018).

Several ℓ_2 attacks have also been developed (Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017). In this paper, we mainly experiment with the ℓ_2 *C&W attack* by Carlini & Wagner (2017) which obtains the best results. This is an iterative and unbounded attack (but in practice the ℓ_2 distortion remains low). The attack is very efficient against all networks but requires significant computational resources (generating a single adversarial example can takes minutes or even hours). Nevertheless this attack remains the most effective attack and the best existing defense mechanism so far (Madry et al. (2018)) only offers marginal protection against it.

An important idea to better protect models against strong ℓ_2 attacks is to add noise at inference. The presence of noise complicates the design of adversarial examples because the precise outcome of the model is not known by the attacker. The first method explicitly using this kind of technique was introduced by Xie et al. (2018) and a number of alternative noise based protection methods have been proposed (Liu et al., 2018; Rakin et al., 2018) offers some protection. However, Athalye et al. (2017) have remarked that randomization based protection mechanisms should be evaluated against attacks in expectation. Most existing noise-based protection mechanisms are weak under this evaluation methodology. Neverthe-

less, randomization remains the only viable mechanism against ℓ_2 attacks.

In addition, the robustness of randomized neural networks have also been studied from a theoretical perspective and these studies suggest that it is a good lead. For example, Fawzi et al. (2016) was able to compute lower and upper bounds on the minimal perturbation that fools a classifier in a randomly chosen direction. Finally Pinot et al. (2019) have demonstrated why injecting noise from exponential family at inference adds robustness, based on an in-depth analysis of the link between probability metrics (Wasserstein, Kullback Leibler, etc) and neural networks.

3 Preliminaries

In this section, we introduce preliminary definitions on randomized neural networks and recall the principles underlying the different attacks that we consider in this paper.

3.1 Randomized neural networks

Let us first precise what we mean by a *Neural network* and a *Randomized neural network*.

Definition 1. *An N layer neural network takes an input x and outputs a logit vector for x as follows:*

$$F_\theta(x) := \phi^{(N)} \circ \cdots \circ \phi^{(1)}(x)$$

θ is the parameter set of the network and $\phi^{(i)}$ are the layers of the network parametrized by θ. Those layers can either be convolutional, dense, or of any other form.

Definition 2. *A randomized neural network is a neural network described as follows:*

$$\tilde{F}_\theta(x) := \phi^{(N)} \circ \cdots \circ (\phi^{(1)}(x) + b)$$

Where b is a multivariate Gaussian vector $\mathcal{N}(0, \sigma I)$.

According to Pinot et al. (2019), it is possible to use any kind of noise injection in the network as long as it comes from an exponential family. It is possible to inject noise on any on the N layers' activation. For the sake of simplicity, we restrict our study to the injection of Gaussian noise on the first layer of the network.

3.2 Adversarial attacks

As discussed in Section 1, attacks can be divided in two categories: ℓ_2 and ℓ_∞ attacks. Finding a good attack is a bi-criteria optimization problem, in which the attacker tries to minimize the norm of the perturbation while trying to maximize the loss function. There are to different ways of implementing this: (1) by minimizing the norm under constraint on the loss, or (2) by maximizing the loss under constraint on the norm. There is no consensus so far and the state-of-the-art attacks that we describe in the rest of this section are implemented following either (1) or (2).

Attack 1. *(Goodfellow et al., 2015)*

Fast Gradient Method with ℓ_p-norm (FGM_p) aims to solve the following type (2) optimization problem

$$\underset{||r||_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(F_\theta(x+r), y)$$

where $\mathcal{L}$ is the loss function of the classifier (e.g. cross entropy), F_θ is the logit predicted by the network and y the true label of x. Assuming ϵ to be small, the previous problem can be approximated by solving the following:

$$\underset{||r||_p \leq \epsilon}{\operatorname{argmax}} \nabla_x \mathcal{L}(F_\theta(x), y)^\top r$$

When $p \neq \infty$, this corresponds to the FGM_p attack. When considering $p = \infty$, the solution to this problem is $r = \epsilon \text{sign}(\nabla_x \mathcal{L})$, which corresponds to the $FGSM$ attack.

Attack 2. *(Carlini & Wagner, 2017)*

To craft an ℓ_2 adversarial example, Carlini & Wagner (2017) solves the following type (1) optimization problem

$$\underset{r}{\operatorname{argmin}} ||r||_2 + c \times g(x+r)$$

where $g(x+r) < 0$ if and only if the $f(x+r) \neq y$ (y here is the true label of x), and f the underlying classifier of the neural network F_θ. Carlini & Wagner (2017) use a change of variable $r = \tanh(w) - x$ to ensure that $-1 \leq x + r \leq 1$, a binary search to optimize the constant c, and Adam or SGD to compute an approximated solution.

C&W attack is an ℓ_2 attack since the objective function aims to minimize the ℓ_2 norm of the perturbation. C&W can also be implemented as a ℓ_∞ attack but this attack is not as effective as classical ℓ_∞ attacks.

Attack 3. *(Kurakin et al., 2016; Madry et al., 2018)*

The PGD attack is a generalization of the iterative FGSM attack (Kurakin et al., 2016). The goal of the adversary is to solve the following type (2) problem

$$\underset{||r||_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(F_\theta(x+r), y)$$

In practice, the authors propose an iterative method to compute a solution:

$$x^{t+1} = \prod_{x \oplus r} (x^t + \alpha \operatorname{sign}(\nabla_x \mathcal{L}(F_\theta(x^t), y)))$$

Where $x \oplus r$ is the Minkowski sum between $\{x\}$ and $\{r$ s.t. $||r||_p \leq \epsilon\}$, α a gradient step size, $\prod_S$ is the projection operator on S and x^0 is randomly chosen in $x \oplus r$.

As for Fast Gradient Methods, PGD can be implemented either with $p = \infty$ or $p = 2$. In the rest of this paper, we denote by PGD^t, a PGD attack with t steps.

Remark 1. *When the network is randomized, the good practice concerning evaluation is to use the Expectation Over Transformation (Athalye et al., 2017; Carlini et al., 2019) during the attack. If $\tilde{F}_\theta(x)$ is a randomized network with x a given input then the Expectation Over Transformation (EOT) is $\mathbb{E}\left(\tilde{F}_\theta(x)\right)$, where the expectation is taken over the randomness of $\tilde{F}_\theta$. In practice, the expectation is estimated by Monte Carlos type methods. As advised by Athalye et al. (2017) we use EOT to evaluate the performances of randomized methods.*

4 Randomized Adversarial Training

Though seemingly simple, Adversarial Training based on PGD or FGSM is one of the only defenses that practically proved robust (Athalye et al., 2018). This method has been extensively studied and demonstrates state-of-the-art results in ℓ_∞ setting. Unfortunately it gives very poor results against state-of-the-art ℓ_2 attacks such as C&W. In this section we first explain why Adversarial Training based on ℓ_∞ attacks does protect models against ℓ_2 attacks and vice-versa. Then, we introduce RAT, a defense strategy that mixes ℓ_∞ adversarial training with randomization in order to offer simultaneous protection against powerful ℓ_2 and ℓ_∞ attacks.

4.1 No free lunch in adversarial training

To understand why ℓ_∞ adversarial training does not protect against ℓ_2 attacks, we first need to characterize the adversarial examples generated with ℓ_∞ attacks.

ℓ_∞ adversarial examples: let ϵ be the maximum amount of acceptable noise measured using the ℓ_∞ norm and r be the perturbation of an input example $x \in \mathbb{R}^d$ with $\|r\|_\infty \leq \epsilon$. By computing the sign of the gradient of the loss, FGSM finds perturbations r in the *corners* of the ℓ_∞ ball of radius ϵ, as illustrated in Figure 1 (left). These perturbations remain visually imperceptible because they have bounded ℓ_∞ norm, but they are large when measured with the ℓ_2 norm: $\|r\|_2 = \epsilon \times \sqrt{d}$. PGD^n, the other ℓ_∞ attack, generalizes FGSM by performing n iterative calls to FGSM. By doing so, after a single gradient step, the perturbations r also have a norm of $\|r\|_2 = \epsilon \times \sqrt{d}$. When the dimensionality d is large (as it is usually the case for most deep learning applications) all the perturbations are likely clustered around the corners of the ℓ_∞ ball, depicted in red in Figure 1, and far away from the ℓ_2 ball.

Let us now consider adversarial examples generated with ℓ_2 attacks.

ℓ_2 adversarial examples: we choose ϵ' to be the maximum acceptable noise measured in ℓ_2 norm that an ℓ_2 attack is allowed to generate. Therefore all ℓ_2 attacks must lie within the ℓ_2 ball of radius ϵ', the blue area depicted in Figure 1 (left). Remark that the ℓ_2 ball does not include the corners of the ℓ_∞ ball unless we set $\epsilon' \geq \epsilon\sqrt{d}$. Thus, in high dimension (large values of d), unless ϵ' is set to a very large value, the ℓ_2 ball will not include the corners of the ℓ_∞ ball. As a consequence, the sets of ℓ_∞ and ℓ_2 adversarial examples are mutually disjoints.

Because the two sets of adversarial examples are disjoint, training models with ℓ_∞ adversarial examples, can only protect ℓ_2 adversarial examples that are inside the ℓ_∞ ball. The other adversarial examples remain unprotected. We illustrate this phenomenon using Figure 1 (middle): the cross at the center is the original example x and the square around the cross represents all acceptable ℓ_∞ adversarial perturbations $\|r\|_\infty \leq \epsilon$. By training a classifier that labels x and all the ℓ_∞ adversarial examples correctly (materialized by the red line), we do not guarantee that the classifier is protected against adversarial examples that are in the cap (i.e. the ℓ_2 adversarial examples that are in the ℓ_2 ball but not in the ℓ_∞ ball).

In large dimension, the red corners are very far away from the ℓ_2 ball. Therefore we hypothesize that in this setting, a large proportion of the ℓ_2 adversarial examples remain unprotected. To verify this hypothesis, we measure the proportion of adversarial examples inside of the ℓ_2 ball before and after ℓ_∞ adversarial training. The results are presented in Figure 2 (left: without adversarial training, right: with adversarial training).

On both charts, the blue area represents the proportion of adversarial training examples that are inside the ℓ_∞ ball. The red area represents the adversarial examples that are outside the ℓ_∞ ball but still inside the ℓ_2 ball (valid ℓ_2 adversarial examples). Finally the brown beige area represents the adversarial examples that are beyond the ℓ_2 bound. The radius ϵ' of the ℓ_2 ball varies along the x-axis from ϵ' to $\epsilon'\sqrt{d}$.

On the left chart (without adversarial training) most ℓ_2 adversarial examples generated by C&W are also inside the ℓ_∞ ball. On the right chart much fewer adversarial examples that remain in the ℓ_∞ ball, this is the expected consequence of ℓ_∞ adversarial training. However, most of these adversarial examples have shifted in the area

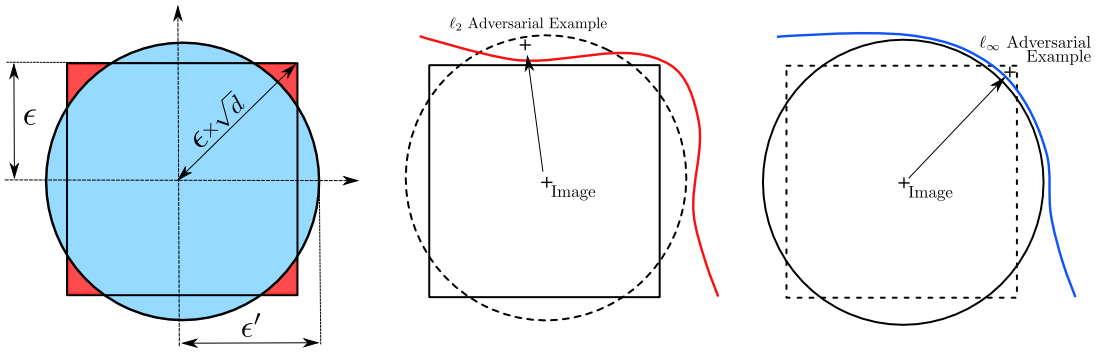

Figure 1: Left: 2D representation of the ℓ_∞ and ℓ_2 balls of respective radius ϵ and ϵ'. Middle: a classifier trained with ℓ_∞ adversarial perturbations (materialized by the red line) remains vulnerable to ℓ_2 attacks. Right: a classifier trained with ℓ_2 adversarial perturbations (materialized by the blue line) remains vulnerable to ℓ_∞ adversarial examples (note, this case is impractical).

in the ℓ_2 ball that is not in the ℓ_∞ ball (i.e. the cap). These examples are equally good from the ℓ_2 perspective, and the impact of ℓ_∞ adversarial training in terms of ℓ_2 robustness is effectively null.

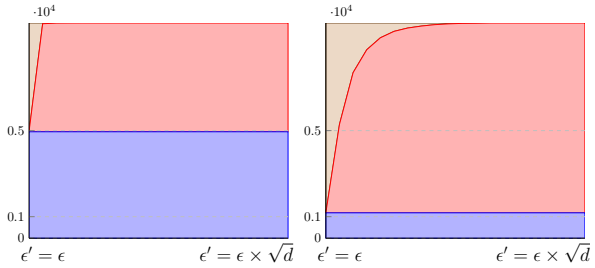

Figure 2: Comparison of the number of adversarial examples found by C&W, inside the ℓ_∞ ball (lower, blue area), outside the ℓ_∞ ball but inside the ℓ_2 ball (middle, red area) and outside the ℓ_2 ball (upper gray area). ϵ is set to 0.3 and ϵ' varies along the x-axis. Left: without adversarial training, right: with adversarial training. Most adversarial examples, have shifted from the ℓ_∞ ball to the cap of the ℓ_2 ball, but remain at the same ℓ_2 distance from the original example.

Finally, remark that even though it is impractical, ℓ_2 adversarial training would also be inefficient against ℓ_∞ adversarial attacks for similar reasons, as illustrated in the Figure 1 (right).

We conclude that the protection mechanisms are complementary, and that mixing ℓ_∞ and ℓ_2 protection mechanisms is necessary to obtain a strong and general protection mechanism. Building on this insight, we introduce RAT and LeRAT, two protection mechanisms that include defense techniques against both types of attacks.

4.2 RAT & LeRAT

To construct a defense strategy that is both robust to ℓ_2 and ℓ_∞ attacks, we propose Randomized Adversarial Training (RAT), described in Algorithm 1.

The main difference between AT and RAT is that the latter is designed to train randomized networks instead of deterministic networks. Recall[1] that each time a randomized network $\tilde{F}_\theta$ is used to make a prediction on an example or to compute a gradient, a new noise vector b is drawn and added to the activation of the first layer of the network, thus producing a randomized prediction (respectively a randomized gradient). In RAT, this randomization is done both when computing the adversarial attack (line 5 in Algorithm 1) and when updating the weights of the network (line 7). Note that for the sake of limiting the computational cost of the training phase, EOT method isn't used to compute adversarial examples.

Algorithm 1 RAT

1: Initialize θ
2: **for** $i \in [1, nb\_steps]$ **do**
3: $B_i := (x_k, y_k)_k$ the current mini-batch
4: **for** $x_k \in B_i$ **do**
5: Compute adversarial attack x'_k of x_k against $\tilde{F}_\theta$ using an arbitrary method
6: **end for**
7: Update θ following the criterion: $\frac{1}{|B_i|} \sum_k \mathcal{L}(\tilde{F}_\theta(x'_k), y_k)$
8: **end for**
9: **return** Network $\tilde{F}_\theta$

[1] see definition 2

Learned Randomized Adversarial Training : *Learned Randomized Adversarial Training* (LeRAT) is a natural variation of RAT were internal noise injection parameters are learned during training instead of being chosen by grid search. The rational of this procedure is that learning the noise injection should help the network finding the good amount of noise to defend. One should note that this procedure can only be used with the adversarial loss as a classical loss would instantly shut down the noise injection.

5 RAT the robust

In this section, we first discuss the robustness of neural networks trained with RAT against ℓ_∞ attacks. Then we discuss their performances against stronger ℓ_2 attacks and provide a number of empirical insights to better understand the behaviour of protection mechanisms against this type of attacks. Finally, we discuss the generality of the RAT approach. All code and datasets will be release on our website upon acceptance of this paper.

To evaluate the robustness of the protection mechanisms against ℓ_∞ and ℓ_2 attacks we follow the rigorous experimental protocol proposed by Athalye et al. (2018), that is: we evaluate the randomized networks using EOT attacks. We also consider untargeted attacks (i.e. a missclassification from one target to any other will be considered as a failure for the defender), and using the white-box attack setting (i.e. we assume that the attacker has unlimited access to the weights of the network). Overall, this represents the most conservative setting to evaluate protection mechanisms. See Section 5.4 for a comprehensive description of our experimental setting.

5.1 Robustness against ℓ_∞ attacks

We evaluate RAT and LeRat against ℓ_∞ attacks generated with FGSM and its stronger variant: PGD (Madry et al., 2018). We also compare it against state-of-the-art Adversarial Training as well as randomized networks proposed by Pinot et al. (2019). We first experiment with a varying ℓ_2 bound on the perturbation. The results are presented in Figure 3.

Unsurprisingly, the accuracy lowers for all protection mechanisms when we raise the value of ϵ from 0.05 to 0.3. (i.e. when we tolerate more perturbed adversarial examples.) On this setting, LeRAT obtains the best robustness together with state-of-the-art ℓ_∞ protection mechanism, R.N. demonstrate lower performance on this setting but RAT demonstrate reasonable performances especially on strong PGD attacks.

In a second set of experiments we fix the bound $\epsilon = 0.1$ and increase the cardinality of the sample used to compute the attack in expectation (more samples means stronger EOT attacks). The results are shown in Figure 4 (a) and (b).

As we can see plain R.N. offers a lower protection against strong EOT attacks. In contrast LeRAT is not impacted by the stronger attacks, and RAT offers a good compromise.

5.2 Robustness against ℓ_2 attacks (C&W)

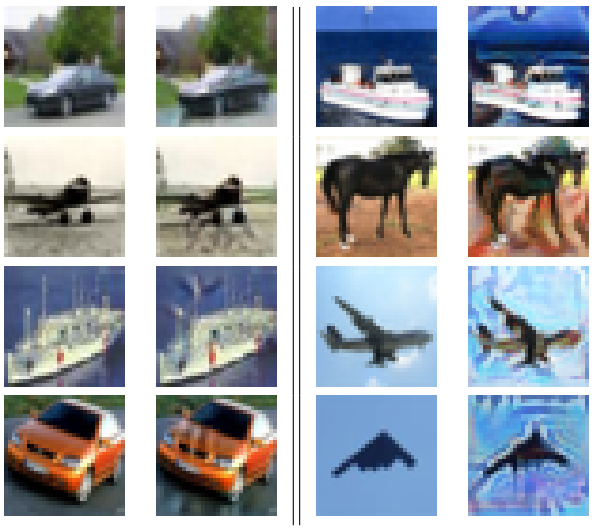

Table 1: Original images and the corresponding adversarial examples generated by C&W attack against our approach. On the left of the double line (ℓ_2 norm of the perturbation = 4), the perturbations easily noticeable. On the right (ℓ_2 norm = 10, worst adversarial examples) the perturbation can be noticed, even without comparing with the original image.

We now evaluate our methods against strong ℓ_2 attacks using the C&W attack. Given an input image, the C&W attack finds the minimum perturbation that will lead a classifier to a misclassification. We evaluate our approach using this setting (as in (Athalye et al., 2018)) and then discuss other relevant aspects the evaluation. The results are presented in Figure 4 (c).

First, we can see that adversarial training (A.T.) offers no protection whatsoever against ℓ_2 attacks. We can also see that all randomized networks outperform again A.T. trained with ℓ_∞ adversarial examples. As we explained in Section 4, ℓ_∞ adversarial training cannot offer proper protection against ℓ_2 attacks. Finally, we can see that the robustness of LeRAT is significantly lower than the other two approaches.

Remark that in this experimental the norm of the perturbation is not bounded, therefore there are no guarantees

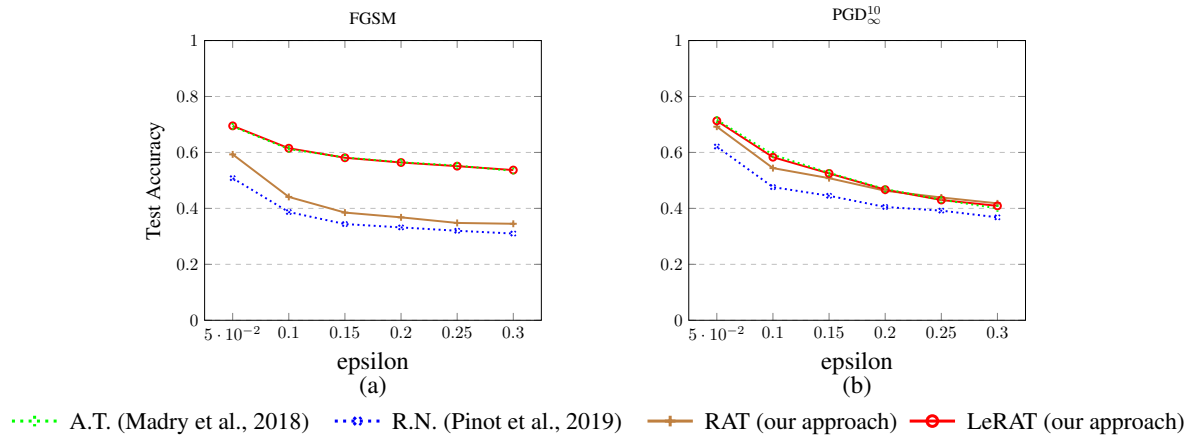

Figure 3: These figures represent the test accuracy on the CIFAR10 dataset given the bound ϵ of the ℓ_∞ attacks. The value range from $5 \cdot 10^{-2}$ to 0.3 and the higher the value, the more the images is perturbed. We can see that A.T. perform equally with LeRAT while RAT out perform R.N.

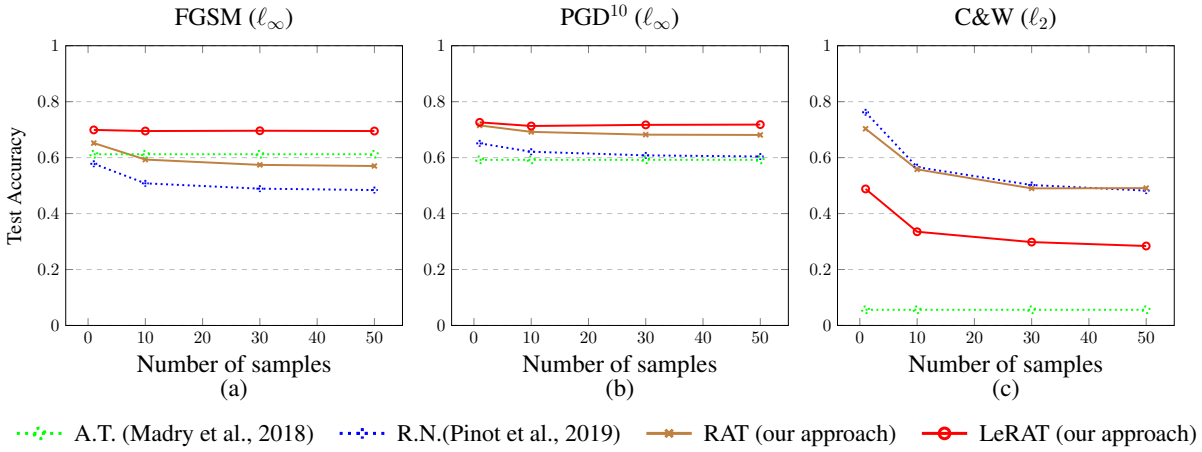

Figure 4: This figure shows the accuracy of different models against FGSM ℓ_∞ attacks (a), PGD^{10} ℓ_∞ attacks (b) and C&W ℓ_2 attacks (c). All attacks are EOT attacks estimated using an increasing sample size. LeRAT outperforms other approaches on ℓ_∞. RAT and R.N. outperform LeRAT on ℓ_2 attacks. RAT offer the best tradeoff.

that the actual adversarial example is below the threshold of human perception. In Figure 5, we have plotted statistics about the adversarial examples generated by C&W to defeat each defense mechanism.

The statistics in Figure 5 show that the average ℓ_2 norm of adversarial examples required to defeat all randomized networks is surprisingly high. In Figure 1 we provide several adversarial examples of variable ℓ_2 norm. As we can see, adversarial examples with an ℓ_2 norm above 4 do defeat the defense mechanism but they are easily identified with a human eye. In other words, defeating these defense mechanisms may require perturbation that are too big to remain undetected by humans.

5.3 General performance against strong attacks and general discussion

We now report the performance of different approaches against stronger attacking settings. For ℓ_∞ we use a PGD^{10} attack with a large ϵ bound of 0.3. For ℓ_2, we use C&W with 40 iterations. In both case, we use 50 Monte-Carlo iterations to estimate the best EOT perturbation with high accuracy. The results are summarized in Table 2. As we can see, RAT and LeRAT offer the best robustness against ℓ_∞ and RAT offers the best robustness against ℓ_2 attacks. RAT outperforms the state-of-the-art against all existing defense mechanisms by a significant margin and offers the best compromise overall.

Model	ℓ_∞ Attack	ℓ_2 Attack
Adv Training (A.T.) (Madry et al., 2018)	0.400	0.056
Randomized Network (R.N.) (Pinot et al., 2019)	0.356	0.489
Randomized Adv Training (RAT)	**0.409**	**0.499**
Learned Randomized Adv Training (LeRAT)	0.400	0.296

Table 2: Comparison of the robustness of all approaches against ℓ_∞ and ℓ_2 attacks. Adversarial Training offers good protection against ℓ_∞ attacks but performs poorly against ℓ_2 attacks. The ℓ_∞ attack is computed with PGD^{10} with an epsilon of 0.3, the ℓ_2 attack is made with C&W with 40 iterations. All attacks against randomized network are computed using a Monte Carlo with 50 iterations in order to estimate the expected value of the network. Randomize networks offer good protection against ℓ_2 attacks but is suboptimal against ℓ_∞ attacks. Our approach (RAT) offers the best protection against all attacks.

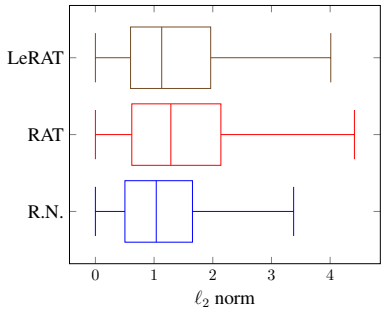

Figure 5: This chart shows statistics about the adversarial examples generated by the C&W attack to defeat different defenses. The adversarial examples generated to fight the RAT defense have higher norm (and thus are easier to identify) than for other defenses.

5.4 Details of the evaluation methodology

Dataset For our experiments, we use the CIFAR10 dataset (Krizhevsky et al., 2009), which consist of 60 000 32x32 colour images with 10 classes. Theses dataset are split with 50 000 images for the training set and 10 000 images for evaluation.

Baseline We used the Wide Resnet architecture introduced by (Zagoruyko & Komodakis, 2016) for all our experiments with 10 and 28 as widen factor and depth. Every input is normalized between -1 and +1. Our model is learned on 200 epochs with a batch size of 200 and a starting learning rate of 0.1. During the training, the learning rate decreases to 0.02, 0.004 and 0.00008 respectively after 7500, 15000 and 20 000 steps. We also use random crop and random flip as data augmentation techniques. For regularization, we use dropout and 0.002 weight decay. To train the network, we use the cross entropy loss with momentum 0.9. This network architecture achieves a 0.95 accuracy on CIFAR10.

Randomized Adversarial Training For adversarial training, we use the procedure described in algorithm 1 with *Projected Gradient Descent* as an attack algorithm. PGD is implemented with 1 iteration, and a global perturbation intensity of 0.03, and gradient step size of 0.02. In order to choose the variance of the injected noise, we computed a grid search that lead us to choose noise drawn from a Gaussian with mean 0 and variance 1 as a good trade off between accuracy and robustness.

Evaluation under attack To evaluate our approach, we compute the accuracy under attack with FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018), as ℓ_∞ attacks and C&W (Carlini & Wagner, 2017), as ℓ_2. For PGD, we use 10 iterations with a gradient step size of 0.02. For C&W, we use 9 iteration for the binary search and 40 iterations overall. As stated in (Athalye et al., 2017), attacks against randomized defense need to be computed with the expected value of the loss or the logits. In our experiment, we estimate the expected value of the attack with a Monte Carlo.

6 Conclusion

We conducted a thorough experimental analysis and provided insights to understand the behaviour of several defense mechanisms under different types of attacks. Building on these insights, we introduce two novel techniques named RAT et LeRAT that can be used to train robust networks against ℓ_∞ and ℓ_2 attacks. Then we compare them against existing state-of-the-art techniques: adversarial training which is state-of-the-art for ℓ_∞ attacks and Randomized Networks which is state-of-the-art on ℓ_2 attacks. Our experiments show that our approach outperform adversarial training on ℓ_∞ attacks and randomized networks on ℓ_2 attacks, resulting in an approach that is both more general and more robust than the existing approaches.

References

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to 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. 274–283, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

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

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Dhillon, G. S., Azizzadenesheli, K., Bernstein, J. D., Kossaifi, J., Khanna, A., Lipton, Z. C., and Anandkumar, A. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.

Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pp. 1632–1640, 2016.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10. *Canadian Institute for Advanced Research*, 2009.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *European Conference on Computer Vision*, pp. 381–397. Springer, 2018.

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.

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.

Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., and Atif, J. Theoretical evidence for adversarial robustness through randomization: the case of the exponential family. *arXiv preprint arXiv:1902.01148*, 2019.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. Technical report, OpenAi, 2018.

Rakin, A. S., He, Z., and Fan, D. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. *arXiv preprint arXiv:1811.09310*, 2018.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 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.

Xie, C., Wang, J., Zhang, Z., Ren, Z., and Yuille, A. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.