Smoothed Inference for Adversarially-Trained Models

Yaniv Nemcovsky*[1], Evgenii Zheltonozhskii*[1], Chaim Baskin*[1], Brian Chmiel*[1,2], Alex M. Bronstein[1], and Avi Mendelson[1]

[1]Technion Israel Institute of Technology
[2]Intel Artificial Intelligence Products Group (AIPG)
yanemcovsky@cs.technion.ac.il; evgeniizh@campus.technion.ac.il;
chaimbaskin@cs.technion.ac.il; brian.chmiel@intel.com; bron@cs.technion.ac.il;
avi.mendelson@cs.technion.ac.il

Abstract

Deep neural networks are known to be vulnerable to inputs with maliciously constructed adversarial perturbations aimed at forcing misclassification. We study randomized smoothing as a way to both improve performance on unperturbed data as well as increase robustness to adversarial attacks. Moreover, we extend the method proposed by He et al. [16] by adding low-rank multivariate noise, which we then use as a base model for smoothing. The proposed method achieves 58.5% top-1 accuracy on CIFAR-10 under PGD attack and outperforms previous works by 4%. In addition, we consider a family of attacks, which were previously used for training purposes in the certified robustness scheme. We demonstrate that the proposed attacks are more effective than PGD against both smoothed and non-smoothed models. Since our method is based on sampling, it lends itself well for trading-off between the model inference complexity and its performance. A reference implementation of the proposed techniques is provided.

1. Introduction

Deep neural networks (DNNs) are showing spectacular performance in a variety of computer vision tasks, but at the same time are susceptible to adversarial examples – small perturbations that alter the prediction [14, 32]. Since the initial discovery of this phenomenon in 2013, growingly stronger defences [14, 14, 16, 21, 26, 30, 31, 39, 41] and counter-attacks [3, 6, 9, 14, 22, 26, 28, 29] were proposed in the literature. Adversarial attacks have also been shown in tasks beyond image classification where they were first discovered, object detection [34], natural language processing [8, 11, 19], reinforcement learning [13], speech-to-text [7], and point cloud classification [38] just to mention a few. Gilmer et al. [12] argued that adversarial examples are an inevitable property of high dimensional data manifolds rather than a weakness of specific models. In view of this, the true goal of an adversarial defence is not to getting rid of the existence of adversarial examples but rather making their search hard. In particular, adding randomness to the network can be particularly successful [5, 16, 24], since information acquired from previous runs cannot be directly applied to a current run.

Contribution. While previous works discuss randomized smoothing in the context of certified robustness [10, 30], in this work we consider it as a viable method to increase both the performance and adversarial robustness of a base model. In particular, we propose a generalization of parametric noise injection (PNI) [16] which we term colored PNI (CPNI), and utilize it as a base model. We demonstrate that randomized smoothing brings significant improvement in accuracy in both the presence and the absense of an adversarial attack.

We discuss several methods of smoothing and ways to optimize a pre-trained model and improve the accuracy of the smoothed classifier. We view smoothing as a transformation of the data space that "smooths" the decision boundaries of the classifier. Such a transformation can be considered as a way of enforcing a smoothness prior of the underlying data distribution classifications boundaries (which is usually the case for naturally occurring distribution).

2. Related work

In these section, we briefly review the notions of white and black box attacks and describe the existing approaches for adversarial defence and certified defence.

*Equal contribution.

Adversarial attacks. Adversarial attacks were first proposed by Szegedy et al. [32], who noted that it was possible to use the gradients of a neural network to discover small perturbations which drastically change its outputs. It is common to divide adversarial attacks into two classes: *white box* attacks, allowing access to the internals of the model (in particular, its gradients); and *black box* attacks allowing access only to the outputs of the model for a given input. Another useful division is into *targeted* attacks – ones that attempt to make the model predict a specific predefined class (target) instead of the real one, as opposed to *untargeted* attacks that only try to degrade the classifier performance.

White box attacks. One of the oldest and simplest white box adversarial attacks is the fast gradient sign method (FGSM) [14], which makes use of special properties of the L_∞ norm and thus utilizes the (normalized) sign of the gradient as an adversarial perturbation:

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \epsilon \cdot \operatorname{sign}(\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L}), \tag{1}$$

where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ denote the clean and the perturbed inputs, respectively, $\mathcal{L}$ is the loss function that the perturbed input shall maximize, and ϵ is the desired attack strength.

Madry et al. [26] proposed to use iterative optimization – specifically, projected gradient ascent, to find stronger adversarial examples:

$$\hat{\boldsymbol{x}}^k = \Pi_{B(x,\epsilon)} \left[\hat{\boldsymbol{x}}^{k-1} + \alpha \cdot \operatorname{sign}(\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L}) \right]. \tag{2}$$

The projection operator Π restricts the perturbed input to be in some vicinity $B(x, \epsilon)$ of the unperturbed input. The interations are initialized with $\hat{\boldsymbol{x}}^0 = \boldsymbol{x}$. This attack, referred to as PGD in the literature, is one of the most powerful L_∞ attacks up to date.

MultiTargeted attack [15] iterates over classes $s \in \mathcal{S}$ and applies a targeted PGD with each s as target. This simple change gives a notable improvement in different settings.

Carlini and Wagner [6] proposed a family of attacks using other norm constraints, in particular, the L_0, L_2 and L_∞ norms. For that, they solve a minimization problem

$$\min \|\boldsymbol{\delta}\| + c\mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}). \tag{3}$$

In contrast to FGSM and PGD, which have a strict bound on the attack norm, the C&W attack always succeeds, but the norm of the perturbation is unbounded.

Rony et al. [29] proposed to decouple norm and direction optimization, motivated by the fact that finding the adversarial example in a predefined region is a simpler task. Their DDN attack iteratively changes the norm depending on the success of a previous step:

$$\hat{\boldsymbol{x}}^k = \Pi_{B(x,\epsilon_k)} \left[\hat{\boldsymbol{x}}^{k-1} + \alpha \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L} \right] \tag{4}$$

$$\epsilon_k = (1 + s \cdot \gamma)\epsilon_{k-1}, \tag{5}$$

where $s = -1$ if $\hat{\boldsymbol{x}}^{k-1}$ is misclassified and $s = 1$ otherwise.

Black box attacks. A simplest way to attack a model $\mathcal{F}$ without accessing its gradients is to train a substitute model $\mathcal{F}'$ to predict the outputs of $\mathcal{F}$ [28] and then use its gradients to apply any of the available white box attacks. Liu et al. [25] extended this idea to transferring the adversarial examples from one model (or ensemble of models) to another, not necessary distilled one from another.

Other works propose alternative methods to estimate gradients: ZOO [9] makes a numerical estimation, NATTACK [22] uses natural evolution strategies [36], and BayesOpt [2] employs Gaussian processes.

Adversarial defences. Szegedy et al. [32] proposed to generate adversarial examples during training and use them for training adversarially robust models, optimizing the loss

$$\mathcal{L}_{\text{adv}}(\boldsymbol{x}) = (1 - w) \cdot \mathcal{L}_{CE}(\boldsymbol{x}) + w \cdot \mathcal{L}_{CE}(\hat{\mathbf{x}}), \tag{6}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss, $\hat{\mathbf{x}}$ is the adversarial example and w is a hyperparameter usually set to $w = 0.5$.

The method is particularly convenient if the generation of adversarial examples is fast [14]. Moreover, this method, combined with stronger attacks, provides a powerful baseline for adversarial defences [26], and is utilized as a part of defence procedure in many modern defences.

Xie et al. [39] proposed a feature denoising mechanism inspired by self-attention [33] and non-local blocks [35]. Sarkar et al. [31] introduced loss terms which enforce linear structure and Lipschitz continuity of the network.

Many works proposed improvements over regular adversarial training by applying stronger attacks during the adversarial training phase. For example, Khoury and Hadfield-Menell [21] proposed to use Voronoi cells instead of ϵ-balls as a possible space for adversarial examples in the training phase. The authors argue that this setting has many advantages over regular ϵ-balls, being better suited for low-dimensional data manifold embedded into high dimensional spaces, and providing a full partition of the space. Liu et al. [23] added adversarial noise to all the activations and not only to the input. Jiang et al. [18] proposed to use learning-to-learn framework, training an additional DNN to generate adversarial examples, which is used to adversarially train the classifier, resembling GAN training. Balaji et al. [4] heuristically updated per-image attack strength ϵ_i, decreasing it if attack succeeded and increasing otherwise.

Randomization of the neural network can be a very powerful adversarial defence, since, even providing access to gradients, the attacker does not have access to the network, but rather some randomly perturbed version thereof. One of the first works involving randomization Zheng et al. [42] proposed to improve robustness by reducing distance between two samples differing by a normally distributed variable with a small variance. Zhang and Liang [41] proposed to add

normal noise to the input, which is shown to reduce KL divergence between non-perturbed and adversarial inputs. An important improvement to PGD adversarial training is parametric noise injection (PNI) [16]. PNI improves robustness of the network by injecting Gaussian noise to parameters (weights or activations) and learning the noise strength.

Athalye et al. [3] made an important observation that many defences do not imrove robustness of the defended network but rather obfuscate gradients, making gradient-based optimization methods less effective. They identify common properties of obfuscated gradients and organize them in a checklist. In addition, they propose techniques to overcome common instances of obfuscated gradients: in particualar, approximating non-differentiable functions with a differentiable substitute and using averaging on the randomize ones.

Certified defences. Certified defences, first proposed by Wong and Kolter [37], should give robustness guarantees for a classifier by proving the performance under norm-bounded perturbations. In particular, randomized methods were shown to be especially efficient, and Cohen et al. [10] has shown a tight bound on L_2 certification by applying a random sampling technique. Consequently, Salman et al. [30] used the smoothed classifier to generate stronger "smoothed" adversarial attacks, and utilize them for adversarial training. Such training allows to produce more accurate base classifier, and as a result, improve certified robustness properties. Anonymous [1] proposed adversarial training based on layer-wise provable optimization via convex relaxation, applying adversarial training in non-randomized certification settings and improving non-randomized certification results.

3. Randomized smoothing

Recently, a line of works [10, 30, 37] has shown certain theoretical guarantees on adversarial robustness. In particular, a certified defense proposed by Cohen et al. [10] is based on randomized smoothing and can guarantee the accuracy of 49% on ImageNet for perturbations with the L_2 norm bounded by 0.5. A smooth classifier g assigns x the class label that the base classifier f is most likely to return for $\boldsymbol{x}$ under Gaussian pertubation η,

$$g(\boldsymbol{x}) = \arg\max_y P(f(\boldsymbol{x}+\eta) = y) \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}). \tag{7}$$

Since the outcome of the smoothed classifier cannot be evaluated directly, it is approximated with a Monte-Carlo method, i.e., by averaging over a number of points sampled from the distribution.

Since the smoothing is independent of the architecture of the base classifier, we can take any (robust) model and test the overall improvement provided by smoothing. We will denote by M the number of the samples for Monte-Carlo approximation. Notice that for an input of b samples, the base classifier needs to evaluate a total of bM samples.

In what follows, we describe three different realizations of smoothing, which differ in the way of combining predictions of individual samples into the final prediction.

3.1. Prediction smoothing

In this case, we generate predictions for each of the M samples independently and then perform voting; we output the most frequent prediction among the samples:

$$g(\boldsymbol{x}) = \arg\max_y \sum_{i=1}^{M} \mathbb{1}\left[\arg\max_{y'} f_{i,y'}(\boldsymbol{x}) = y\right], \tag{8}$$

where $g(\boldsymbol{x})$ is output of smoothed model and $f_{i,y}(\boldsymbol{x})$ is probability of class y predicted for the i-th sample $\boldsymbol{x} + \boldsymbol{\eta}_i$.

This is the smoothing method that was previously discussed by Cohen et al. [10] and Salman et al. [30] for certified robustness. By taking into account the M votes, the accuracy of the classifier was shown to increase on both clean and adversarial inputs. In this method, one only cares about the classification of each of the M samples, and no importance is given to their classifications certainties.

3.2. Soft prediction smoothing

In this case, we calculate the expectation of probability of each class and use them for prediction generation:

$$g(\boldsymbol{x}) = \arg\max_c \sum_{i=1}^{M} \mathrm{softmax}(f_{i,c}), \tag{9}$$

This method was previously mentioned by Salman et al. [30] as a way to apply adversarial training to a smoothed classifier. Since the probabilities for each class are now differentiable, this allows to train the classifier end-to-end. In contrast to prediction smoothing, we now fully take into account the classification probabilities of each of the M samples. However, this defence is easier to overcome for attackers. Often, even if the attack is not successful, the probability of competing classes increases significantly, which means that the even unsuccessful attack on a base model does affect the prediction of the smooth model.

3.3. Weighed smoothing

The two former methods can be generalized as follows: In its most general form, the smoothed model output can be written as

$$g(\boldsymbol{x}) = \arg\max_y \sum_{i=1}^{M} \mathrm{V}(f_{i,y}), \tag{10}$$

where V is some voting function: the indicator function for prediction smoothing, and softmax for soft prediciton

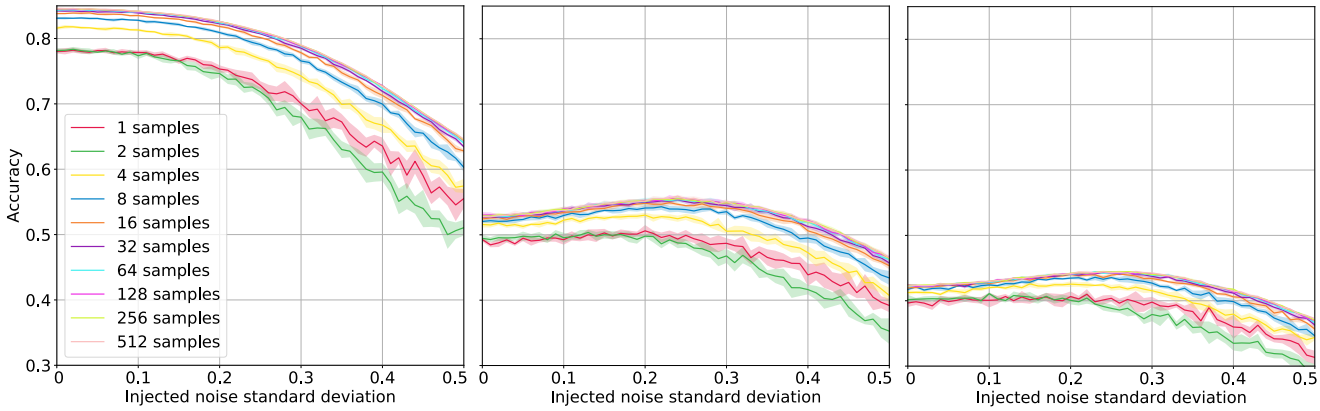

Figure 1: Accuracy as a function of injected noise strength with different number of samples for a CPNI base model with prediction smoothing for different attacks: no attack (left); PGD (middle); and EPGD (right).

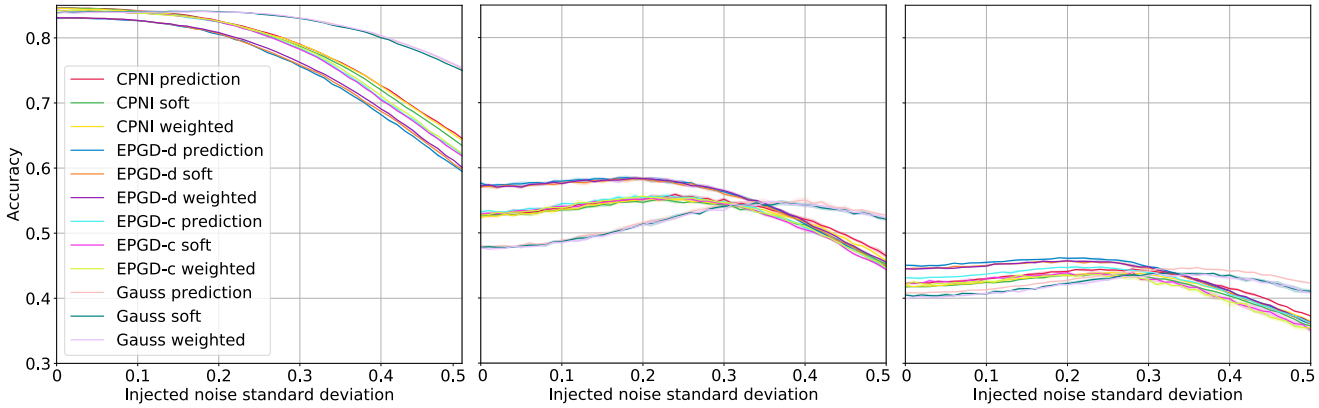

Figure 2: Accuracy as a function of injected noise strength with 8 Monte Carlo samples for all base model and all smoothing methods for different attacks: no attack (left); PGD (middle); and EPGD (right).

smoothing. We propose to assign some weight to top-k predictions using, for example,

$$V(k) = 2^{1-k}, \text{ or} \tag{11}$$

$$V_c(k) = \begin{cases} 1 & k = 1 \\ c & k = 2 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

In particular, V_c expresses the dependency on the second prominent class noted by Cohen et al. [10]. It is also possible to take into account the prior of each class in the definition of V. Assigning weights is a compromise between the hard and soft prediction approaches in the sense that we take into account the classifications probabilities of each of the M samples.

3.4. Defence by random noise injection

To get some intuition on the effect of the noise injection on adversarial attacks, we perform an analysis of a simple classification model – a support vector machine (SVM) $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$ with the parameters $\boldsymbol{w}$ and b. We look at the formulation of SVM under adversarial attacks with zero-mean random noise injected into the input. We assume that the attacker is aware of this noise, but is limited to observing the effect of a single realization thereof.

We start from the expectation of the SVM objective on a single input sample x_i:

$$\mathbb{E}_{\boldsymbol{\eta}} \max_{\boldsymbol{\delta}} \max \Big[1 - y_i(\boldsymbol{w} \cdot (\boldsymbol{x}_i + \boldsymbol{\eta}_i - \boldsymbol{\delta}_i) + b), 0\Big], \tag{13}$$

where $\boldsymbol{\eta}_i$ is the injected noise and $\boldsymbol{\delta}_i$ is the adversarial noise. Denoting $\boldsymbol{\delta}'_i = \boldsymbol{\delta}_i - \boldsymbol{\eta}_i$, we can, using the result of Xu et al. [40], write the latter as

$$\mathbb{E}_{\boldsymbol{\eta}} \max_{\boldsymbol{\delta}} \boldsymbol{w} \cdot \boldsymbol{\delta}'_i + \max \Big[1 - y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b), 0\Big] \tag{14}$$

Since the expectation of $\boldsymbol{\eta}$ is $\mathbf{0}$, $\mathbb{E}_{\boldsymbol{\eta}} \boldsymbol{w} \cdot \boldsymbol{\eta}_i = 0$, leading to

$$\max_{\boldsymbol{\delta}} \boldsymbol{w} \cdot \boldsymbol{\delta}_i + \max \Big[1 - y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b), 0\Big], \tag{15}$$

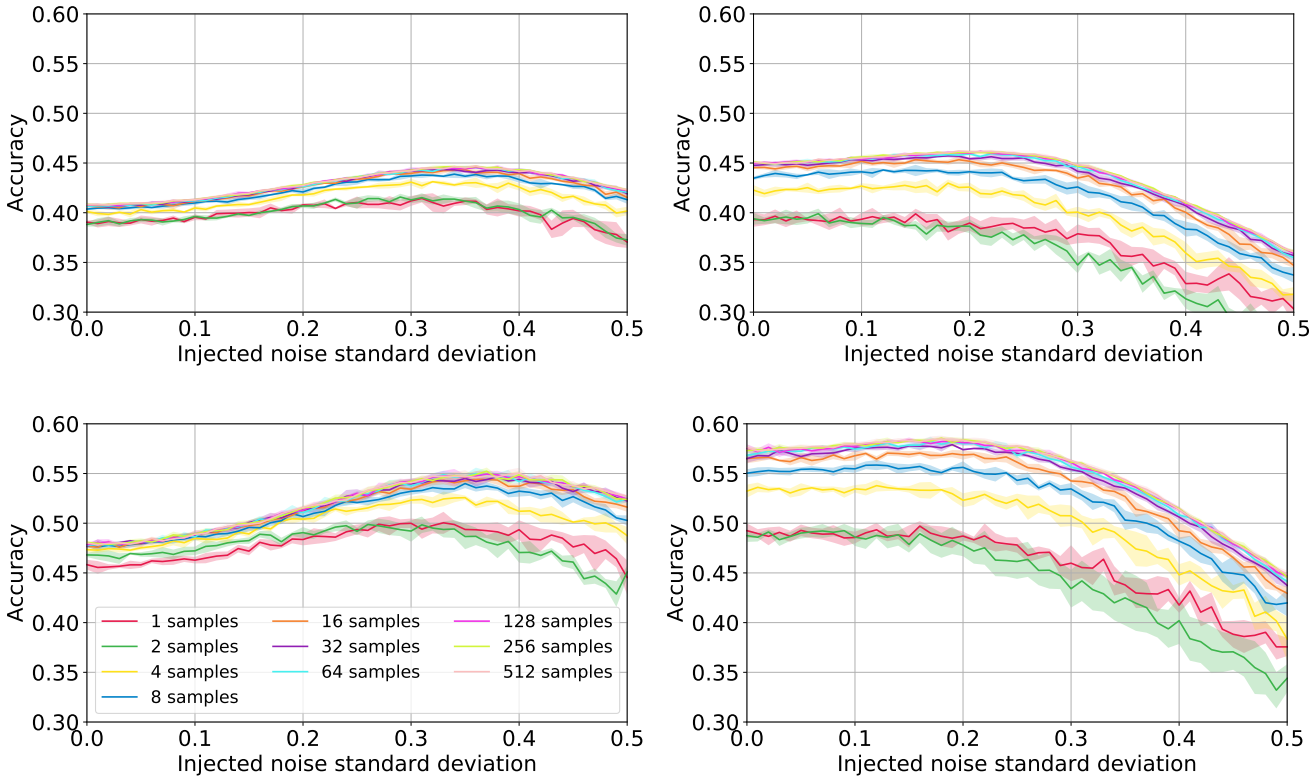

Figure 3: Accuracy as a function of injected noise strength with different number of samples for Gauss (left) and EPGD-diverged (right) with prediction smoothing for different attacks: EPGD (up) and PGD (down).

which is nothing but the SVM objective without injected noise. Thus, the expectation of an adversarial attack will be the same as that of an attack without the noise injection, and it is unclear whether the attacker can devise a better strategy to breach the defense effect provided by the noise injection.

The effect of noise injection is similar to random gradient obfuscation. Let $\boldsymbol{x}$ be some input point, $\boldsymbol{\eta}$ a realization of the random perturbation, $\boldsymbol{\delta}$ the adversarial attack that would have been chosen for $\boldsymbol{x}$, and $\boldsymbol{\delta}'$ the adversarial attack that would have been chosen for $\boldsymbol{x} + \boldsymbol{\eta}$. Since we add noise to the input in each forward iteration, the adversary computes $\boldsymbol{\delta}'$ instead of $\boldsymbol{\delta}$ which has some random distribution around the true adversarial direction. Denoting by Π_a the projection on the direction chosen, by $\Pi_\perp$ projection on the space orthogonal to this direction, and by $\|\boldsymbol{\delta}\|_p \leq \epsilon$ the L_p-bound on the attack strength, yields

$$\Pi_\perp(\boldsymbol{\delta}') = \Pi_\perp(\boldsymbol{\eta}) \equiv \boldsymbol{\eta_0} \tag{16}$$

$$\|\Pi_a(\boldsymbol{\delta}')\|_p = \left(\epsilon^p - \|\boldsymbol{\eta_0}\|_p^p\right)^{1/p} \tag{17}$$

For $p = \infty$, the second term equals ϵ. However, the first term is a random variable that moves us further away from the adversarial direction and therefore decreases the probability of a successful adversarial attack. This effect accumulates when the adversary attack computes the gradients multiple times (such as in PGD).

4. Training a smoothed classifier

An obvious extension of the discussed method is to incorporate the smoothing into the training procedure. We consider two approaches to training of the smoothed classifier: one based on prediction smoothing, and another one based on soft prediction smoothing. In both cases, we start from an adversarially pre-trained base model and fine-tune it to improve the robustness of the smooth classifier.

4.1. Prediction smoothing training

In the case of prediction smoothing (or any other case in which V is not differentiable), we cannot train the smoothed model directly. We, therefore, want to train the base model in a way that optimizes the loss of smoothed model. The 0-1 loss of n training samples is

$$\mathcal{L}_{01} = \sum_{i=1}^{n} \ell_{01}(y_i, \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \tag{18}$$

with the point-wise terms

$$\ell_{01}(y_i, \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) = 1 - \mathbb{1}\left[y_i = \arg\max_{y \in \mathcal{Y}} P_\eta\big[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\eta}) = y\big]\right]$$

is minimized over the model parameters $\boldsymbol{\theta}$.

Denoting for brevity $P_y = P_\eta\big[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\eta}) = y\big]$, we can approximate the indicator as

$$\begin{aligned} &\mathbb{1}\left[y_i = \arg\max_{y \in \mathcal{Y}} P_y\right] = \mathbb{1}\left[P_{y_i} \geq \max_{y' \in \mathcal{Y} \setminus \{y_i\}} P_{y'}\right] \quad (19) \\ \approx &\left[P_{y_i} - \max_{y' \in \mathcal{Y} \setminus \{y_i\}} P_{y'}\right]_+ = \max_{y \in \mathcal{Y}} \left[P_y - \max_{y' \in \mathcal{Y} \setminus \{y_i\}} P_{y'}\right], \end{aligned}$$

where we approximate the Heavyside function $\mathbb{1}$ with a better-behaving ReLU function $[\cdot]_+$ on the interval $[-1, 1]$. The last equality follows from the fact that if y_i is not the most probable class then $y = y'$. The expression in Eq. (20) resembles the bound Cohen et al. [10] has suggested for the radius of certification from adversarial attacks.

We now show a relation to training the base model under perturbation, similarly denoting $\mathbb{1}_y = \mathbb{1}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\eta}) = y]$:

$$\ell_{01}(y_i, \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) = 1 - \max_{y \in \mathcal{Y}} \mathbb{E}_\eta \mathbb{1}_y + \max_{y' \in \mathcal{Y} \setminus \{y_i\}} \mathbb{1}_{y'}. \quad (20)$$

Written in this form, the 0-1 loss is now amenable to Monte-Carlo approximation, however working which such a non-convex loss is problematic. We therefore bound the $\max$ over the expectation by the expectation over the $\max$,

$$\ell_{01}(y_i, \tilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \leq 1 - \mathbb{E}_\eta \max_{y \in \mathcal{Y}} \mathbb{1}_y + \mathbb{E}_\eta \max_{y' \in \mathcal{Y} \setminus \{y_i\}} \mathbb{1}_{y'}.$$

Since the classification events are disjoint we obtain

$$\begin{aligned} \mathcal{L}_{01} &= \mathbb{E}_\eta \left[\sum_{i=1}^{n} \left(1 - \sum_{y \in \mathcal{Y}} \mathbb{1}_y + \sum_{y' \in \mathcal{Y} \setminus \{y_i\}} \mathbb{1}_{y'}\right)\right] \\ &= \mathbb{E}_{\boldsymbol{\eta}} \left[\sum_{i=1}^{n} 1 - \mathbb{1}_{y_i}\right] = \mathbb{E}_{\boldsymbol{\eta}} \sum_{i=1}^{n} \ell_{01}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)), \quad (21) \end{aligned}$$

which is the 0-1 loss of the base classifier under Gaussian perturbation. This means that by injecting Gaussian noise into input of the base classifier, we minimize the loss of the smoothed one. $\boldsymbol{\eta}$ is not restricted to the Gaussian distributions as long as the above expectations exist; we could combine Gaussian noise injection with adversarial training:

$$\begin{aligned} \boldsymbol{\eta} &\sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) + B \cdot \boldsymbol{\delta} \\ B &\sim \mathrm{Ber}(q), \quad (22) \end{aligned}$$

where $\boldsymbol{\delta}$ is the adversarial attack and q is the weight of adversarial samples in the training procedure. Such adversarial training under Gaussian perturbations relates to minimizing the 0-1 loss of the smoothed classifier under adversarial attack. We refer to this method as "Gauss" further in text.

4.2. Soft smoothing training

The smoothing of the classifier gives rise to a new family of attacks, which target the smoothed classifier rather than the base model. In particular, in the case of soft smoothing, the output of the smoothed classifier is differentiable and thus can be used as a source of gradient for white box attack. Salman et al. [30] discussed this family of attacks as a way to improve the generalization of a base model and thus improve certified accuracy of smooth model.

We consider adversarial training with such attacks as a way to directly optimize the smoothed model by training the base model. We expect it to increase the adversarial robustness of both the base and the smoothed models. Specifically, we consider PGD attacks, where at each iteration we compute the gradients based on several randomized samples. We chose to limit the number of samples to 8, since further improvement in the predicting ability is counter-balanced by the increase in the computational complexity.

Perceptually aligned gradients. The aforementioned soft smoothing attacks were previously explored in various ways. Kaur et al. [20] have shown that targeted attacks of this kind have perceptually-aligned gradients, i.e. they perturb the input to make it resemble a member of a different class, even for models that were not adversarially trained. In contrast, untargeted adversarial attacks exhibit this phenomenon only on adversarially-trained models. In general, perceptual alignment of the gradients is an indirect evidence of model robustness [17, 27]: we expect such a perturbation, if strong enough, to be able to attack even a perfect classifier. Such phenomena might indicate that adversarial training of the model makes the adversarial attacks converge to soft smoothing-based attacks.

Soft smoothing-based attacks. We, therefore, consider using such attacks not only as a part of the training, but rather as a viable way to produce effective adversarial attacks. We could also expect such attacks to have lower variance, as we are taking the expectation over a few samples, which allows to find more effective perturbations with less effort. In particular, we refer to such PGD-basd attacks as "expectaton PGD" (EPGD). Notice that an EPGD attack with k PGD steps and m samples requires computing the gradients $k \times m$ times, making its computational cost comparable with that of a PGD attack with $k \times m$ steps.

4.3. Noise strength learning

He et al. [16] learned the variance of the noise injected into the layers. This approach can be applied to the noise injected to the input, in case we are not interested in certification properties of the defence. In this scenario, we explore the tradeoff between the adversarial robustness and the accu-

Table 1: Results of CPNI. Mean and standard deviation is calculated over 10 runs for our experiments (upper half), and over 5 runs for experiments by He et al. [16] (lower half). Noise is injected either to weights ("W") or output activations ("A-a"). Best results for PNI and CPNI are set in **bold**.

Method	Accuracy, mean±std%	
	Clean	PGD
PNI-W	82.84 ± 0.22	$\mathbf{46.11 \pm 0.43}$
CPNI-W	78.48 ± 0.41	$\mathbf{48.84 \pm 0.55}$
CPNI-A-a	$\mathbf{83.41 \pm 0.14}$	45.47 ± 0.18
CPNI-W+A-a	77.07 ± 0.40	46.07 ± 0.45
PNI-W	84.89 ± 0.11	45.94 ± 0.11
PNI-W+A-a	$\mathbf{85.12 \pm 0.10}$	43.57 ± 0.12

Table 2: Comparison of our method to prior art on CIFAR-10.

Method	Accuracy, mean±std%	
	Clean	PGD
Madry et al. [26][a]	83.84	39.14 ± 0.05
Jiang et al. [18]	85.31 ± 0.41	53.42 ± 1.07
He et al. [16]	82.84 ± 0.22	46.11 ± 0.43
Balaji et al. [4]	91.34	48.53
Sarkar et al. [31]	87.65	54.77
Smooth (our)	81.07 ± 0.01	$\mathbf{58.56 \pm 0.14}$

[a] results by He et al. [16]

racy on clean samples, taking into account the quality of the base classifier and amount of Monte Carlo samples.

4.4. Parametric noise injection and colored noise

Our method can be applied on top of any defense or even with no defense at all. In particular, we implement a variant of PNI, a defense proposed by He et al. [16], which we call CPNI. The change is based on the fact that allowing the noise to be correlated (colored) improves the performance of PNI. In particular, we use multivariate Gaussian with a factorized covariance matrix of the form $\boldsymbol{\sigma} = \boldsymbol{D} + \boldsymbol{L}^\top \boldsymbol{L}$, where $\boldsymbol{D}$ is diagonal matrix and $\boldsymbol{L}$ is $n \times r$ matrix. In particular, $r = 5$ enhances the adversarial accuracy by at least 2%, as shown in Table 1.

5. Experiments

We study the performance of each of the proposed smoothing methods (prediction, soft prediction, and weighed smoothing) over 4 different base models: adversarially-trained CPNI-W, and three additional models which were obtained by fine-tuning the base model with training methods described in Section 4: EPGD and Gauss. For EPGD, we selected two models chosen based on clean validation set performance: one with the best validation accuracy (labelled *EPGD-converged*) and with the worst validation accuracy (*EPGD-diverged*). The motivation to consider the latter arose since PGD and EPGD are highly similar in nature to the point that PGD attacks converge to EPGD with adversarial training, therefore training until convergence could result in a model very similar to CPNI. We conjecture that this is a sign of some form of overfitting of the smooth model, which requires further investigation. For Gauss we tuned the CPNI model based on the noise maximizing performance, $\sigma = 0.24$. Our main focus is to explore the behavior of the different smoothing methods, and thus we have not tried to achieve best possible results by optimizing the hyper-parameters of both the base and the smoothed classifier.

Experimental settings. On CIFAR-10, we trained the CPNI base model with ResNet-20 for 400 epochs and chose the model with the highest performance on a clean validation set. We used SGD with the learning rate 0.1, reduced by 10 at epochs 200 and 300, weight decay 10^{-4}, and colored noise factorized covariance of rank $r = 5$. The obtained model was fine-tuned with either EPGD adversarial training or Gauss training for up to 200 epochs.

Comparison to other adversarial defences. We compared the best-performing instance of the proposed defence (EPGD-diverged with prediction smoothing and 256 samples) to the current state-of-the-art in adversarial defences on CIFAR-10. This configuration achieves an improvement of 4% over the best previous work and 10% over the PNI baseline, as shown in Table 2.

Black box attacks. We tested our defence against black box attacks, in particular, transferable attack [25]. For that purpose, we trained another instance of CPNI-W model and used it as a source model in three different configurations: PGD without smoothing, PGD with smoothing, and EPGD with smoothing. Results are reported in Table 4. Our model performs well even if the source model is not smoothed, which is an argument against a randomized gradient obfuscation effect.

Ablation study. We compared a different number of Monte Carlo samples: 2^n for $0 \leqslant n \leqslant 9$. Since the difference between 16 and 512 samples is within one standard deviation, we have not systematically studied any further increase in the number of samples, which provides diminishing returns. However, in case where accuracy is more important than the runtime, it might be feasible to use more samples, since we have achieved as high as 59.3% accuracy on PGD for 1024 samples with the EPGD-diverged model.

Table 3: Results of proposed defence for white box attacks (PGD and EPGD) on CIFAR-10. Mean and standard deviation over 5 runs is presented in form of mean±std.

Base Model	Smoothing Method	Iterations	Noise	Accuracy, mean±std		
				Clean	PGD	EPGD
CPNI-W	Prediction	512	0.0	$\mathbf{84.63 \pm 0.05}$	52.8 ± 0.29	42.22 ± 0.01
CPNI-W	Prediction	512	0.24	81.44 ± 0.06	55.92 ± 0.22	44.43 ± 0.11
EPGD-diverged	Prediction	256	0.0	83.05 ± 0.01	57.31 ± 0.22	45.11 ± 0.14
EPGD-diverged	Prediction	256	0.18	81.07 ± 0.01	$\mathbf{58.56 \pm 0.14}$	45.98 ± 0.15
EPGD-diverged	Prediction	512	0.19	80.8 ± 0.01	58.32 ± 0.03	$\mathbf{46.21 \pm 0.08}$
EPGD-diverged	Soft	512	0.0	83.13 ± 0.07	57.07 ± 0.15	44.47 ± 0.13
EPGD-diverged	Soft	512	0.19	80.98 ± 0.07	58.32 ± 0.33	45.61 ± 0.18
EPGD-diverged	Weighted	256	0.19	80.98 ± 0.07	58.47 ± 0.18	45.5 ± 0.2
EPGD-converged	Prediction	256	0.22	81.82 ± 0.05	55.9 ± 0.37	44.85 ± 0.13
Gauss	Prediction	256	0.37	81.25 ± 0.06	55.23 ± 0.37	44.53 ± 0.11

Table 4: Results of proposed defence applied on CPNI-W with prediction smoothing for transferable attacks (PGD, PGD-s (PGD with prediction smoothing in source model), and EPGD) on CIFAR-10. Mean and standard deviation are calculated over 5 runs.

Iterations	Noise	Accuracy, mean±std		
		PGD	PGD-s	EPGD
4	0.02	59.24 ± 0.35	58.88 ± 0.15	53.92 ± 0.32
8	0.02	60.38 ± 0.10	60.39 ± 0.14	55.16 ± 0.32

For each number of Monte Carlo samples, we considered multiple noise standard deviations in the range $[0, 0.5]$. Higher noise levels are not considered due to significant degradation of the model performance. For each method, we report clean (unperturbed) validation accuracy, as well as adversarial accuracy for PGD and EPGD. We then report a number of the best results for each base model and smoothing method in Table 3, and a comparison of different setups in Figs. 1 to 3. All our best results were obtained with prediction smoothing, even for a low number of samples. Even though the advantage is of the order of a single standard deviation, this performance difference is persistent among base models, number of iterations, and attack types, and thus is likely to be systematic. This could indicate that the attacker can utilize additional information provided by other methods better than the defender. In particular, as shown in Table 3, *EPGD-diverged* has achieved the accuracy under both PGD and EPGD attacks. The fact that EPGD reduces performance of smoothed models proves that some part of the defensive effect of smoothing is a consequence of gradient obfuscation.

Fig. 2 demonstrates that there is a single optimal value of noise strength for each base model, independent of the smoothing method. This value is almost same for PGD and EPGD, probably due to similarity of these two attacks. For the CPNI and EPGD models, accuracy on clean data reduces with the noise strength. The Gauss model is more resilient to noise injection, acquiring maximum for around 40% higher standard deviation of the injected noise. This can be related to the fact that the certified radius is twice smaller than the standard deviation of the noise used in the smoothed classifier.

In contrast to the certified robustness scheme [10], we have not observed improvement of the Gauss model over CPNI-W, which might be a result of the interference of the CPNI-induced noise and the Gaussian noise.

In addition, we noted that smoothing without noise addition improves clean accuracy by as much as 3%. From Fig. 2 it is clear that *EPGD-converged* and CPNI base model result in similar performance for all setups, except EPGD attack.

Performance for different attack strength We evaluated our best-performing model against PGD attack with different strength, ϵ, to study the effect of transferring defence on attacks of different strength. Results are shown in Fig. 4.

6. Conclusions

We proposed an adversarial defense based on randomized smoothing, which shows state-of-the-art results for white-box attacks, namely PGD, on CIFAR-10 with a relatively small number of iterations. We also confirm the efficiency of our defense against black box attacks, by successfully defending against transferring adversarial examples from various models. Our method offers a practical trade-off between the inference time and model performance and can be incorporated into any adversarial defense.

In addition, we proposed to utilize a family of attacks that take smoothing into account against smoothed classifiers.

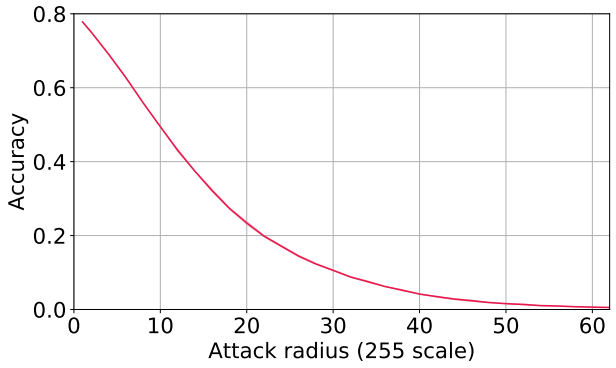

Figure 4: Accuracy of CPNI-W model with prediction smoothing over 8 iterations under PGD attack with different attack radius. Standard deviation is smaller than line width.

We showed its superiority over existing attacks and demonstrated the importance of smoothing against those attack. We show that adversarial training of smoothed classifier is a non-trivial task and study several approaches to it.

Acknowledgments

The research was funded by Hyundai Motor Company through HYUNDAI-TECHNION-KAIST Consortium, ERC StG RAPID, and Hiroshi Fujiwara Technion Cyber Security Research Center.

References

[1] Anonymous. Adversarial training and provable defenses: Bridging the gap. In *Submitted to International Conference on Learning Representations*, 2020). under review.

[2] Anonymous. Bayesopt adversarial attack. In *Submitted to International Conference on Learning Representations*, 2020). under review.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018). PMLR.

[4] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019).

[5] Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural networks. *arXiv preprint arXiv:1810.00363*, 2018).

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, May 2017).

[7] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018).

[8] Akshay Chaturvedi, Abijith KP, and Utpal Garain. Exploring the robustness of nmt systems to nonsensical inputs. *arXiv preprint arXiv:1908.01165*, 2019).

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 15–26, New York, NY, USA, 2017). ACM.

[10] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320, Long Beach, California, USA, 09–15 Jun 2019). PMLR.

[11] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018).

[12] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018).

[13] Adam Gleave, Michael Dennis, Neel Kant, Cody Wild, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019).

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

[15] Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*, 2019).

[16] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019).

[17] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019).

[18] Haoming Jiang, Zhehui Chen, Yuyang Shi, Bo Dai, and Tuo Zhao. Learning to defense by learning to attack. *arXiv preprint arXiv:1811.01213*, 2018).

[19] Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2019).

[20] Simran Kaur, Jeremy Cohen, and Zachary C Lipton. Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*, 2019).

[21] Marc Khoury and Dylan Hadfield-Menell. Adversarial training with voronoi constraints. *arXiv preprint arXiv:1905.01019*, 2019).

[22] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. NATTACK: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3866–3876, Long Beach, California, USA, 09–15 Jun 2019). PMLR.

[23] Aishan Liu, Xianglong Liu, Chongzhi Zhang, Hang Yu, Qiang Liu, and Junfeng He. Training robust deep neural networks via adversarial noise propagation. *arXiv preprint arXiv:1909.09034*, 2019).

[24] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018).

[25] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016).

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018).

[27] Preetum Nakkiran. A discussion of 'adversarial examples are not bugs, they are features': Adversarial examples are just bugs, too. *Distill*, 2019). https://distill.pub/2019/advex-bugs-discussion/response-5.

[28] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, New York, NY, USA, 2017). ACM.

[29] Jérôme Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019).

[30] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *arXiv preprint arXiv:1906.04584*, 2019).

[31] Anindya Sarkar, Nikhil Kumar Gupta, and Raghu Iyengar. Enforcing linearity in dnn succours robustness and adversarial image generation. *arXiv preprint arXiv:1910.08108*, 2019).

[32] 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).

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017).

[34] Derui Wang, Chaoran Li, Sheng Wen, Surya Nepal, and Yang Xiang. Daedalus: Breaking non-maximum suppression in object detection via adversarial examples. *arXiv preprint arXiv:1902.02067*, 2019).

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018).

[36] Daan Wierstra, Tom Schaul, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3381–3387, June 2008).

[37] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5286–5295, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018). PMLR.

[38] Chong Xiang, Charles R. Qi, and Bo Li. Generating 3d adversarial point clouds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019).

[39] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 501–509, June 2019).

[40] Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009).

[41] Yuchen Zhang and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pages 684–693. PMLR, 16–18 Apr 2019).

[42] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4480–4488, June 2016).