arXiv:1811.09716v1 [cs.LG] 23 Nov 2018

Robustness via curvature regularization, and vice versa

Seyed-Mohsen Moosavi-Dezfooli*†
seyed.moosavi@epfl.ch

Alhussein Fawzi*‡
afawzi@google.com

Jonathan Uesato‡
juesato@google.com

Pascal Frossard†
pascal.frossard@epfl.ch

Abstract

State-of-the-art classifiers have been shown to be largely vulnerable to adversarial perturbations. One of the most effective strategies to improve robustness is adversarial training. In this paper, we investigate the effect of adversarial training on the geometry of the classification landscape and decision boundaries. We show in particular that adversarial training leads to a significant decrease in the curvature of the loss surface with respect to inputs, leading to a drastically more "linear" behaviour of the network. Using a locally quadratic approximation, we provide theoretical evidence on the existence of a strong relation between large robustness and small curvature. To further show the importance of reduced curvature for improving the robustness, we propose a new regularizer that directly minimizes curvature of the loss surface, and leads to adversarial robustness that is on par with adversarial training. Besides being a more efficient and principled alternative to adversarial training, the proposed regularizer confirms our claims on the importance of exhibiting quasi-linear behavior in the vicinity of data points in order to achieve robustness.

1. Introduction

Adversarial training has recently been shown to be one of the most successful methods for increasing the robustness to adversarial perturbations of deep neural networks [10, 18, 17]. This approach consists in training the classifier on *perturbed* samples, with the aim of achieving higher robustness than a network trained on the original training set. Despite the importance and popularity of this training mechanism, the effect of adversarial training on the geometric properties of the classifier – its loss landscape with respect to the input and decision boundaries – is not well understood. In particular, how do the decision boundaries and loss landscapes of adversarially trained models compare to the ones trained on the original dataset?

In this paper, we analyze such properties and show that one of the main effects of adversarial training is to induce a significant *decrease* in the curvature of the loss function and decision boundaries of the classifier. More than that, we show that such a geometric implication of adversarial training allows us to explain the high robustness of adversarially trained models. To support this claim, we follow a *synthesis* approach, where a new regularization strategy, Curvature Regularization (CURE), encouraging small curvature is proposed and shown to achieve robustness levels that are comparable to that of adversarial training. This highlights the importance of small curvature for improved robustness. In more detail, our contributions are summarized as follows:

- We empirically show that adversarial training induces a significant *decrease in the curvature* of the decision boundary and loss landscape *in the input space*.
- Using a quadratic approximation of the loss function, we establish upper and lower bounds on the robustness to adversarial perturbations with respect to the curvature of the loss. These bounds confirm the existence of a relation between low curvature and high robustness.
- Inspired by the implications of adversarially trained networks on the curvature of the loss function and our theoretical bounds, we propose an efficient regularizer that encourages small curvatures. On standard datasets (CIFAR-10 and SVHN), we show that the proposed regularizer leads to a significant boost of the robustness of neural networks, comparable to that of adversarial training.

The latter step shows that the proposed regularizer can be seen as a more efficient alternative to adversarial training. More importantly, it shows that the effect of adversarial training on the curvature reduction is not a mere by-product, but rather a driving effect that causes the robustness to increase. We stress here that the main focus of this paper is

*The first two authors contributed equally to this work.
†École Polytechnique Fédérale de Lausanne
‡DeepMind

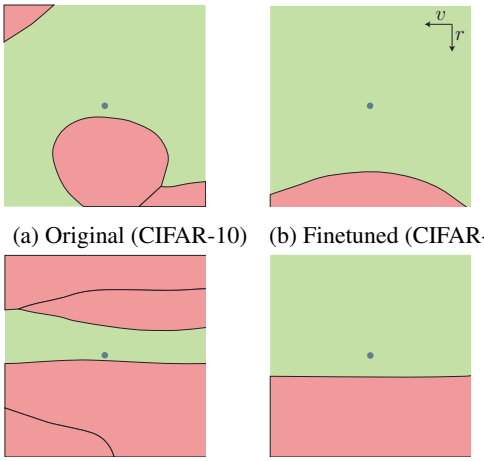
mainly on the latter – analyzing the geometry of adversarial training – rather than outperforming adversarial training.

Related works. The large vulnerability of classifiers to adversarial perturbations has first been highlighted in [3, 22]. Many algorithms aiming to improve the robustness have since then been proposed [10, 21, 17, 5, 1]. In parallel, there has been a large body of work on designing improved attacks [18, 17], which have highlighted that many of the proposed defenses obscure the model rather than make the model truly robust against all attacks [25, 2]. One defense however stands out – adversarial training – which has shown to be empirically robust against all designed attacks. The goal of this paper is to provide an analysis of this phenomenon, and propose a regularization strategy (CURE), which mimics the effect of adversarial training. On the analysis front, many works have analyzed the existence of adversarial examples, and proposed several hypotheses for their existence [7, 9, 23, 6, 13]. In [10], it is hypothesized that networks are not robust as they exhibit a "too linear" behavior. We show here that linearity of the loss function with respect to the inputs (that is, small curvature) is, on the contrary, beneficial for robustness: adversarial training does lead to much more linear loss functions in the vicinity of data points, and we verify that this linearity is indeed the source of increased robustness. We finally note that prior works have attempted to improve the robustness using gradient regularization [11, 16, 20]. However, such methods have not been shown to yield significant robustness on complex datasets, or have not been subject to extensive robustness evaluation. Instead, our main focus here is to study the effect of the *second-order* properties of the loss landscape, and show the existence of a strong connection with robustness to adversarial examples.

2. Geometric analysis of adversarial training

We start our analysis by inspecting the effect of adversarial training on the geometric properties of the decision boundaries of classifiers. To do so, we first compare qualitatively the decision boundaries of classifiers *with* and *without* adversarial training. Specifically, we examine the effect of *adversarial fine-tuning*, which consists in fine-tuning a trained network with a few extra epochs on adversarial examples.[1] We consider the CIFAR-10 [15] and SVHN [19] datasets, and use a ResNet-18 [12] architecture. For fine-tuning on adversarial examples, we use DeepFool [18].

Fig. 1 illustrates normal cross-sections of the decision boundaries before and after adversarial fine-tuning for classifiers trained on CIFAR-10 and SVHN datasets. Specifically, the classification regions are shown in the plane spanned by (r, v), where r is the normal to the decision boundary and v corresponds to a random direction. In addition to inducing a larger distance between the data point and the decision boundary (hence resulting in a higher robustness), observe that the decision regions of fine-tuned networks are flatter and more regular. In particular, note that the curvature of the decision boundaries decreased after fine-tuning.



(a) Original (CIFAR-10) (b) Finetuned (CIFAR-10)

(c) Original (SVHN) (d) Fine-tuned (SVHN)

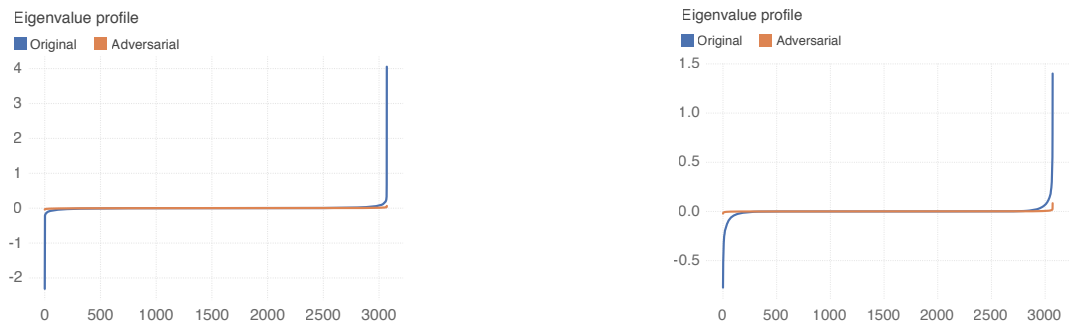
Figure 1: Random normal cross-sections of the decision boundary for ResNet-18 classifiers trained on CIFAR-10 (first row) and SVHN (second row). The first column is for classifiers trained on the original dataset, and the second column shows the boundaries after adversarial fine-tuning on 20 epochs for CIFAR-10 and 10 epochs for SVHN. The green and red regions represent the correct class and incorrect classes, respectively. The point at the center shows the datapoint, while the lines represent the different decision boundaries (note that the red regions can include different incorrect classes).

To further quantify this phenomenon, we now compute the *curvature profile* of the loss function (with respect to the inputs) before and after adversarial fine-tuning. Formally, let ℓ denote the function that represents the loss of the network with respect to the inputs; e.g., in the case of cross-entropy, $\ell(x) = \text{XEnt}(f_\theta(x), y)$, where y is the true label of image $x \in \mathbb{R}^d$, and $f_\theta(x)$ denotes the logits.[2] The curvature profile corresponds to the set of eigenvalues of the Hessian matrix

$$H = \left(\frac{\partial^2 \ell}{\partial x_i \partial x_j} \right) \in \mathbb{R}^{d \times d}$$

[1] While adversarial fine-tuning is distinct from vanilla adversarial training, which consists in training on adversarial images *from scratch*, we use an adversarially fine-tuned network in this paper as it allows to single out the effect of training on adversarial examples, as opposed to other uncontrolled phenomenon happening in the course of vanilla adversarial training.

[2] We omit the label y from ℓ for simplicity, as the label can be understood from the context.



(a) CIFAR-10

(b) SVHN

Figure 2: Curvature profiles, which correspond to sorted eigenvalues of the Hessian, of the original and the adversarially fine-tuned networks. Note that the number of eigenvalues is equal to $32 \times 32 \times 3 = 3072$, which corresponds to the number of input dimensions. The ResNet-18 architecture is used.

where $x_i, i = 1, \ldots, d$ denote the input pixels. We stress on the fact that the above Hessian is with respect to the inputs, and not the weights of the network. To compute these eigenvalues in practice, we note that Hessian vector products are given by the following for any z;

$$Hz = \frac{\nabla \ell(x + hz) - \nabla \ell(x)}{h} \text{ for } h \to 0. \qquad (1)$$

We then proceed to a finite difference approximation by choosing a finite h in Eq. (1). Besides being more efficient than generating the full Hessian matrix (which would be prohibitive for high-dimensional datasets), the finite difference approach has the benefit of measuring *larger-scale* variations of the gradient (where the scale is set using the parameter h) in the neighborhood of the datapoint, rather than an infinitesimal point-wise curvature. This is crucial in the setting of adversarial classification, where we analyze the loss function in a small neighbourhood of data points, rather than the asymptotic regime $h \to 0$ which might capture very local (and not relevant) variations of the function.[3]

Intuitively, small eigenvalues (in absolute value) of H indicate a small curvature of the graph of ℓ around x, hence implying that the classifier has a "locally linear" behaviour in the vicinity of x. In contrast, large eigenvalues (in absolute value) imply a high curvature of the loss function in the neighbourhood of image x. For example, in the case where the eigenvalues are exactly zero, the function becomes locally linear, hence leading to a flat decision surface.

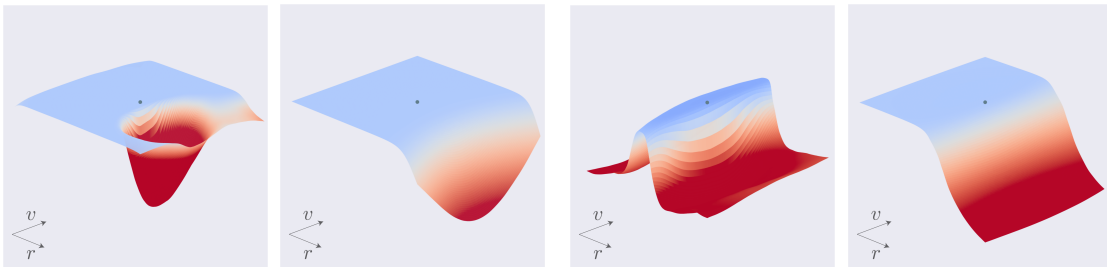
We compute the curvature profile at 100 random test samples, and show the average curvature in Fig. 2 for CIFAR-10 and SVHN datasets. Note that adversarial fine-tuning has led to a strong decrease in the curvature of the loss in the neighborhood of data points. To further illustrate qualitatively this significant decrease in curvature due to adversarial training, Fig. 3 shows the loss surface before and after adversarial training along normal and random directions r and v. Observe that while the original network has large curvature in certain directions, the effect of adversarial training is to "regularize" the surface, resulting in a smoother, lower curvature (i.e., linear-like) loss.

[3]For example, using ReLU non-linearities result in a piecewise linear neural network as a function of the inputs. This implies that the Hessian computed at the logits is exactly 0. This result is however very local; using the finite-difference approximation, we focus on larger-scale neighbourhoods.

	FGSM	ℓ_∞-DF	PGD(7)	PGD(20)
Original	38.0%	11.0%	0.5%	0.2%
Fine-tuned	61.0%	57.5%	57.2%	56.9%

Table 1: Adversarial accuracies for original and fine-tuned network on CIFAR-10, where adversarial examples are computed with different attacks; FGSM [10], DF [18] and PGD [17]. Perturbations are constrained to have ℓ_∞ norm smaller than $\epsilon = 4$ (images have pixel values in $[0, 255]$).

We finally note that this effect of adversarial training on the loss surface has the following somewhat paradoxical implication: while adversarially trained models are *more robust* to adversarial perturbations (compared to original networks), they are also *easier to fool*, in the sense that simple attacks are as effective as complex ones. This is in stark contrast with original networks, where complex networks involving many gradient steps (e.g., PGD(20)) are much more effective than simple methods (e.g., FGSM). See Table 1. The comparatively small gap between the adversarial accuracies for different attacks on adversarially trained models is a direct consequence of the significant decrease of the curvature of the loss, thereby requiring a small number of gradient steps to find adversarial perturbations.



(a) Original (CIFAR-10) (b) Fine-tuned (CIFAR-10) (c) Original (SVHN) (d) Fine-tuned (SVHN)

Figure 3: Illustration of the negative of the loss function; i.e., $-\ell(s)$ for points s belonging to a plane spanned by a normal direction r to the decision boundary, and random direction v. The original sample is illustrated with a blue dot. The light blue part of the surface corresponds to low loss (i.e., corresponding to the classification region of the sample), and the red part corresponds to the high loss (i.e., adversarial region).

3. Analysis of the influence of curvature on robustness

While our results show that adversarial training leads to a decrease in the curvature of the loss, the relation between adversarial robustness and curvature of the loss remains unclear. To elucidate this relation, we consider a simple binary classification setting between class 1 and -1. Recall that $\ell(\cdot, 1)$ denotes the function that represents the loss of the network with respect to an input from class 1. For example, in the setting where the log-loss is considered, we have $\ell(x, 1) = -\log(p(x))$, where $p(x)$ denotes the output of softmax corresponding to class 1. In that setting, x is classified as class 1 iff $\ell(x, 1) \leq \log(2)$. For simplicity, we assume in our analysis that x belongs to class 1 without loss of generality, and hence omit the second argument in ℓ in the rest of this section. We assume that the function ℓ can be locally well approximated using a quadratic function; that is, for "sufficiently small" r, we can write:

$$\ell(x + r) \approx \ell(x) + \nabla\ell(x)^T r + \frac{1}{2} r^T H r,$$

where $\nabla\ell(x)$ and H denote respectively the gradient and Hessian of ℓ at x. Let x be a point classified as class 1; i.e., $\ell(x) \leq t$, where t denotes the loss threshold (e.g., $t = \log(2)$ for the log loss). For this datapoint x, we then define r^* to be the minimal perturbation in the ℓ_2 sense[4], which fools the classifier assuming the quadratic approximation holds; that is,

$$r^* := \arg\min_r \|r\| \text{ s.t. } \ell(x) + \nabla\ell(x)^T r + \frac{1}{2} r^T H r \geq t.$$

In the following result, we provide upper and lower bounds on the magnitude of r^* with respect to properties of the loss function at x.

[4]We use the ℓ_2 norm for simplicity. Using the equivalence of norms in finite dimensional spaces, our result allows us to also bound the magnitude of ℓ_∞ adversarial perturbations.

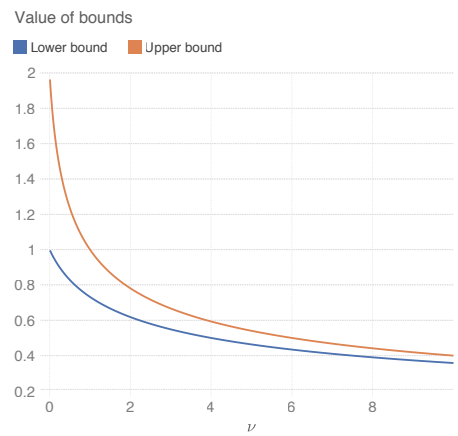


Figure 4: Illustration of upper and lower bounds in Eq. (2) and (3) on the robustness with respect to curvature ν. We have set $\|\nabla\ell(x)\| = 1, c = 1, \nabla\ell(x)^T v = 0.5$ in this example.

Theorem 1. *Let x be such that $c := t - \ell(x) \geq 0$, and let $g = \nabla\ell(x)$. Assume that $\nu := \lambda_{\max}(H) \geq 0$, and let u be the eigenvector corresponding to ν. Then, we have*

$$\frac{\|g\|}{\nu}\left(\sqrt{1 + \frac{2\nu c}{\|g\|^2}} - 1\right) \leq \|r^*\| \tag{2}$$

$$\leq \frac{|g^T u|}{\nu}\left(\sqrt{1 + \frac{2\nu c}{(g^T u)^2}} - 1\right) \tag{3}$$

The above bounds can further be simplified to:

$$\frac{c}{\|g\|} - 2\nu\frac{c^2}{\|g\|^3} \leq \|r^*\| \leq \frac{c}{|g^T u|}$$

Proof. **Lower bound.** Let $\alpha := \|r^*\|$. We note that α

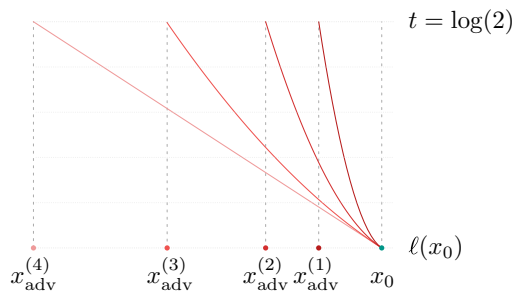


Figure 5: Geometric illustration in 1d of the effect of curvature on the adversarial robustness. Different loss functions (with varying curvatures) are illustrated at the vicinity of data point x_0, and $x_{\text{adv}}^{(i)}$ indicate the points at which such losses exceed t (where t is the misclassification threshold). All curves have the same loss and gradient at x_0. Note that increasing curvature leads to smaller adversarial examples (i.e., smaller $|x_0 - x_{\text{adv}}^{(i)}|$).

satisfies

$$-c + \|g\|\alpha + \frac{\nu}{2}\alpha^2 \geq -c + g^T r^* + \frac{1}{2}(r^*)^T H r^* \geq 0.$$

Solving the above second-order inequality, we get $\alpha \geq \frac{\|g\|}{\nu}\left(\sqrt{1 + \frac{2\nu c}{\|g\|^2}} - 1\right)$ or $\alpha \leq -\frac{\|g\|}{\nu}\left(\sqrt{1 + \frac{2\nu c}{\|g\|^2}} + 1\right)$. However, since $\alpha \geq 0$, the first inequality holds, which precisely corresponds to the lower bound.

Upper bound. Let $\alpha \geq 0$. Define $r := \alpha u$, and let us find the minimal $|\alpha|$ such that

$$-c + g^T r + \frac{1}{2} r^T H r = -c + \alpha g^T u + \frac{\alpha^2 \nu}{2} \geq 0.$$

We note that the above inequality holds for any $|\alpha| \geq |\alpha_{\min}|$, with $|\alpha_{\min}| = \frac{|g^T u|}{\nu}\left(\sqrt{1 + \frac{2\nu c}{(g^T u)^2}} - 1\right)$. Hence, we have that $\|r^*\| \leq |\alpha_{\min}|$, which concludes the proof of the upper bound. The simplified bounds are proven using the inequality $1 + \frac{x}{2} - \frac{x^2}{2} \leq \sqrt{1+x} \leq 1 + \frac{x}{2}$. □

Remark 1. ***Increasing* robustness with *decreasing* curvature.** Note that upper and lower bounds on the robustness in Eq. (2), (3) *decrease* with increasing curvature ν. To see this, Fig. 4 illustrates the dependence of the bounds on the curvature ν. In other words, under the second order approximation, this shows that *small curvature* (i.e., small eigenvalues of the Hessian) is beneficial to obtain classifiers with higher robustness (when the other parameters are kept fixed). This is in line with our observations from Section 2, where robust models are observed to have a smaller curvature than networks trained on original data. Fig. 5 provides intuition to the decreasing robustness with increasing curvature in a one-dimensional example.

Remark 2. Dependence on the gradient. In addition to the dependence on the curvature ν, note that the upper and lower bounds depend on the gradient $\nabla\ell(x)$. In particular, these bounds *decrease* with the norm $\|\nabla\ell(x)\|$ (for a fixed direction). Hence, under the second order approximation, this suggests that the robustness decreases with larger gradients. However, as previously noted in [25, 2], imposing small gradients might provide a false sense of robustness. That is, while having small gradients can make it hard for gradient-based methods to attack the network, the network can still be intrinsically vulnerable to small perturbations.

Remark 3. Bound tightness. Note that the upper and lower bounds match (and hence bounds are exact) when the gradient $\nabla\ell(x)$ is collinear to the largest eigenvector u. Interestingly, this condition seems to be approximately satisfied in practice, as the average normalized inner product $\frac{|\nabla\ell(x)^T u|}{\|\nabla\ell(x)\|_2}$ for CIFAR-10 is equal to 0.43 before adversarial fine-tuning, and 0.90 after fine-tuning (average over 1000 test points). This inner product is significantly larger than the inner product between two typical vectors uniformly sampled from the sphere, which is approximately $\frac{1}{\sqrt{d}} \approx 0.02$. Hence, the gradient aligns well with the direction of largest curvature of the loss function in practice, which leads to approximately tight bounds.

4. Improving robustness through curvature regularization

While adversarial training leads to a regularity of the loss in the vicinity of data points, it remains unclear whether this regularity is the *main* effect of adversarial training, which confers robustness to the network, or it is rather a *byproduct* of a more sophisticated phenomenon. To answer this question, we follow here a *synthesis* approach, where we derive a regularizer which mimics the effect of adversarial training on the loss function – encouraging small curvatures.

Curvature regularization (CURE) method. Recall that H denotes the Hessian of the loss ℓ at datapoint x. We denote by $\lambda_1, \ldots, \lambda_d$ the eigenvalues of H. Our aim is to penalize large eigenvalues of H; we therefore consider a regularizer $L_r = \sum_i p(\lambda_i)$, where p is a non-negative function, which we set to be $p(t) = t^2$ to encourage all eigenvalues to be small. For this choice of p, L_r corresponds to the Frobenius norm of the matrix H. We further note that

$$L_r = \sum_i p(\lambda_i) = \text{trace}(p(H)) = \mathbb{E}(z^T p(H) z) = \mathbb{E}\|Hz\|^2,$$

where the expectation is taken over $z \sim \mathcal{N}(0, I_d)$. By using a finite difference approximation of the Hessian, we have $Hz \approx \frac{\nabla\ell(x+hz) - \nabla\ell(x)}{h}$, where h denotes the discretization step, and controls the scale on which we require the varia-

Table 2: Adversarial and clean accuracy for CIFAR-10 for original, regularized and adversarially trained models. Performance is reported for ResNet and WideResNet models, and the perturbations are computed using PGD(20). Perturbations are constrained to have ℓ_∞ norm less than $\epsilon = 8$ (where pixel values are in $[0, 255]$).

	ResNet-18		WideResNet-28×10	
	Clean	Adversarial	Clean	Adversarial
Normal training	94.9%	0.0%	94.6%	0.0%
CURE	81.2%	36.3%	83.1%	41.4%
Adversarial training [17]	79.4%	43.7%	87.3%	45.8%

tion of the gradients to be small. Hence, L_r becomes

$$L_r = \frac{1}{h^2}\mathbb{E}\left\|\nabla\ell(x+hz) - \nabla\ell(x)\right\|^2.$$

The above regularizer involves computing an expectation over $z \sim \mathcal{N}(0, I_d)$, and penalizes large curvatures along all directions equally. Rather than approximating the above with an empirical expectation of $\|Hz\|^2$ over isotropic directions drawn from $\mathcal{N}(0, I_d)$, we instead *select* directions which are known to lead to high curvature (e.g., [14, 8]), and minimize the curvature along such chosen directions. The latter approach is more efficient, as the computation of each matrix-vector product Hz involves one backward pass; focusing on high-curvature directions is therefore essential to minimize the overall curvature without having to go through each single direction in the input space. This selective approach is all the more adapted to the very sparse nature of curvature profiles we see in practice (see Fig. 2), where only a few eigenvalues are large. This provides further motivation for identifying large curvature directions and penalizing the curvature along such directions.

Prior works in [8, 14] have identified gradient directions as high curvature directions. In addition, empirical evidence reported in Section 3 (Remark 3) shows a large inner product between the eigenvector corresponding to maximum eigenvalue and the gradient direction; this provides further indication that the gradient is pointing in high curvature directions, and is therefore a suitable candidate for z. We set in practice $z = \frac{\text{sign}(\nabla\ell(x))}{\|\text{sign}(\nabla\ell(x))\|}$, and finally consider the regualizer [5]

$$L_r = \|\nabla\ell(x+hz) - \nabla\ell(x)\|^2,$$

where the $\frac{1}{h^2}$ is absorbed by the regularization parameter. Our fine-tuning procedure then corresponds to minimizing the regularized loss function $\ell + \gamma L_r$ with respect to the weight parameters, where γ controls the weight of the regularization relative to the loss term.

[5]The choice of $z \propto \nabla\ell(x)$ leads to almost identical results. We have chosen to set $z \propto \text{sign}(\nabla\ell(x))$, as we are testing the robustness of the classifier to ℓ_∞ perturbations. Hence, setting z be the sign of the gradient is more relevant, as it constrains the z direction to belong to the hypercube of interest.

We stress that the proposed regularization approach significantly departs from adversarial training. In particular, while adversarial training consists in minimizing *the loss on perturbed points* (which involves solving an optimization problem), our approach here consists in imposing regularity *of the gradients* on a sufficiently small scale (i.e., determined by h). Previous works [17] have shown that adversarial training using a weak attack (such as FGSM [10], which involves a single gradient step) does *not* improve the robustness. We show that our approach, which rather imposes gradient regularity (i.e., small curvature) along such directions, does lead to a significant improvement in the robustness of the network.

We use two pre-trained networks, ResNet-18 [12] and WResNet-28x10 [26], on the CIFAR-10 and SVHN datasets, where the pixel values are in $[0, 255]$. For the optimization of the regularized objective, we use the Adam optimizer with a decreasing learning rate between [10^{-4}, 10^{-6}] for a duration of 20 epochs starting from a pre-trained network. We linearly increase the value of h from 0 to 1.5 during the first 5 epochs, and from there on, we use a fixed value of $h = 1.5$. For γ, we set it to 4 and 8 for ResNet-18 and WResNet-28 respectively.

Results. We evaluate the regularized networks with a strong PGD attack of 20 iterations, as it has been shown to outperform other adversarial attack algorithms [17]. The adversarial accuracies of the regularized networks are reported in Table 2 for CIFAR-10, and in the supp. material for SVHN. Moreover, the adversarial accuracy as a function of the perturbation magnitude ϵ is reported in Fig. 6.

Observe that, while networks trained on the original dataset are not robust to perturbations as expected, performing 20 epochs of fine-tuning with the proposed regularizer leads to a significant boost in adversarial performance. In particular, the performance with the proposed regularizer is comparable to that of adversarial training reported in [17]. This result hence shows the importance of the curvature decrease phenomenon described in this paper in explaining the success of adversarial training.

In addition to verifying our claim that small curvature

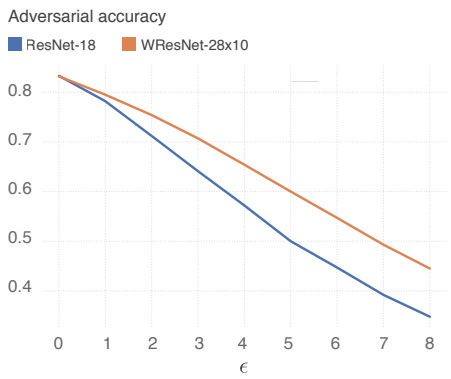


Figure 6: Adversarial accuracy versus perturbation magnitude ϵ computed using PGD(20), for ResNet-18 and WResNet-28x10 trained with CURE on CIFAR-10. See [17] for the curve corresponding to adversarial training. Curve generated for 2000 random test points.

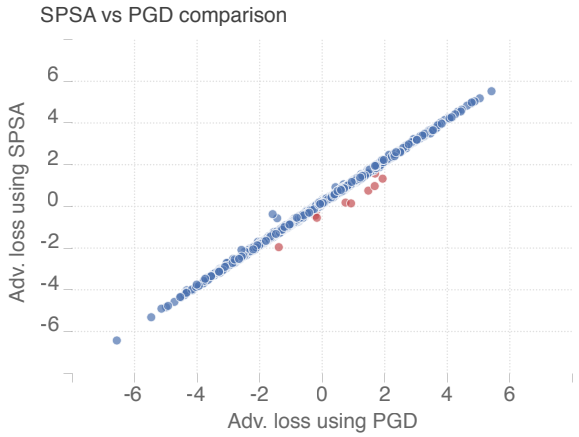
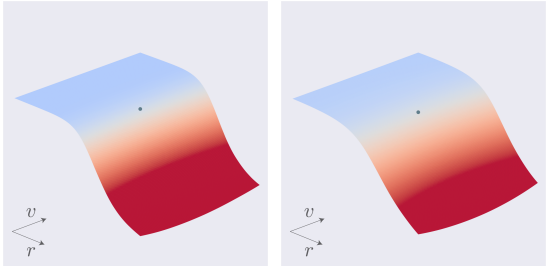


Figure 7: Analysis of gradient masking in a network trained with CURE. Adversarial loss computed with SPSA (y-axis) vs. adversarial loss with PGD(100) (x-axis) on a batch of 1000 datapoints. Adversarial loss corresponds to the difference of logits on true and adversarial class. Each point in the scatter plot corresponds to a single test sample. Negative loss indicates that the data point is misclassified. Points close to the line $y = x$ indicate that both attacks identified similar adversarial perturbations. Points below the line, shown in red, indicate points for which SPSA identified stronger adversarial perturbation than PGD. Note that overall, SPSA and PGD identified similarly perturbations.

confers robustness to the network (and that it is the underlying effect in adversarial training), we note that the proposed regularizer has practical value, as it is efficient to compute and can therefore be used as an alternative to adversarial training. In fact, the proposed regularizer requires 2 backward passes to compute, and is used in fine-tuning for 20 epochs. In contrast, one needs to run adversarial training



(a) ResNet-18 (b) WideResNet-28

Figure 8: Similar plot to Fig. 3, but where the loss surfaces of the network obtained with CURE are shown.

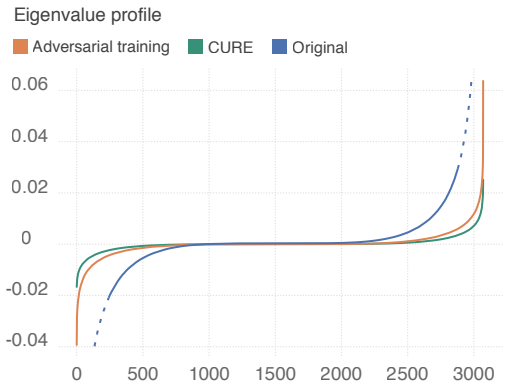


Figure 9: Curvature profile for a network fine-tuned using adversarial training and CURE. The ResNet-18 architecture on CIFAR-10 is used. For comparison, we also report the profile for the original network (same as Fig. 2), where we clipped the values to fit in the y range.

against a *strong* adversary in order to reach good robustness [17], and start the adversarial training procedure from scratch. We note that strong adversaries generally require around 10 backward passes, making the proposed regularization scheme a more efficient alternative. We note however that the obtained results are slightly worse than adversarial training; we hypothesize that this might be either due to higher order effects in adversarial training not captured with our second order analysis or potentially due to a sub-optimal choice of hyper-parameters γ and h.

Stronger attacks and verifying the absence of gradient masking. To provide further evidence on the robustness of the network fine-tuned with CURE, we attempt to find perturbations for the network with more complex attack algorithms. For the WideResNet-28x10, we obtain an adversarial accuracy of 41.1% on the test set when using PGD(40) and PGD(100). This is only slightly worse than the result reported in Table 2 with PGD(20). This shows that increasing the complexity of the attack does not lead to

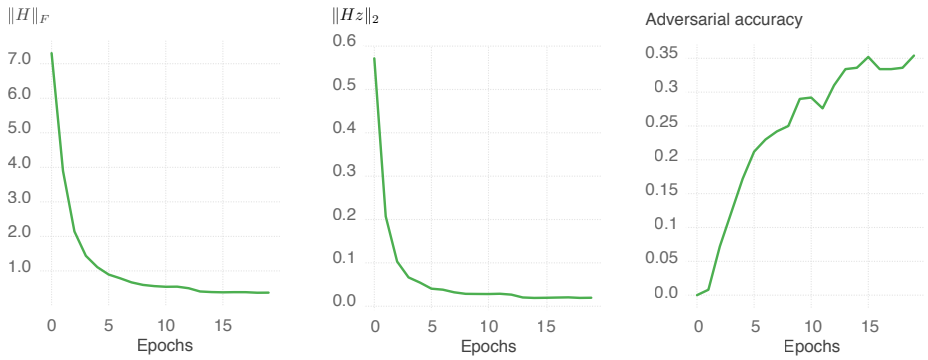


Figure 10: Evolution throughout the course of CURE fine-tuning for a ResNet-18 on CIFAR-10. The curves are averaged over 1000 datapoints. **Left:** estimate of Frobenius norm , **Middle:** $\|Hz\|$, where $z = \text{sign}(\nabla\ell(x))/\|\text{sign}(\nabla\ell(x))\|_2$ and **Right:** adversarial accuracy computed using PGD(20). The Frobenius norm is estimated with $\|H\|_F^2 = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\|Hz\|^2$, where the expectation is approximated with an empirical expectation over 100 samples $z_i \sim \mathcal{N}(0, I)$.

a significant decrease in the adversarial accuracy. Moreover, we evaluate the model against a gradient-free optimization method (SPSA), similar to the methodology used in [25], and obtained an adversarial accuracy of 44.5%. We compare moreover in Fig. 7 the *adversarial loss* (which represents the difference between the logit scores of the true and adversarial class) computed using SPSA and PGD for a batch of test data points. Observe that both methods lead to comparable adversarial loss (except on a few data points), hence further justifying that CURE truly improves the robustness, as opposed to masking or obfuscating gradients. Hence, just like adversarial training which was shown empirically to lead to networks that are robust to all tested attacks in [25, 2], our experiments show that the regularized network has similar robustness properties.

Curvature and robustness. We now analyze the network obtained using CURE fine-tuning, and show that the obtained network has similar geometric properties to the adversarially trained one. Fig. 8 shows the loss surface in a plane spanned by (r, v), where r and v denote respectively a normal to the decision boundary and a random direction. Note that the loss surface obtained with CURE is qualitatively very similar to the one obtained with adversarial training (Fig. 3), whereby the loss has a more linear behavior in the vicinity of the data point. Quantitatively, Fig. 9 compares the curvature profiles for the networks trained with CURE and adversarial fine-tuning. Observe that both profiles are very similar. We also report the evolution of the adversarial accuracy and curvature quantities in Fig. 10 during fine-tuning with CURE. Note that throughout the fine-tuning process, the curvature decreases while the adversarial accuracy increases, which further shows the link between robustness and curvature. Note also that, while we explicitly regularized for $\|Hz\|$ (where z is a fixed direction

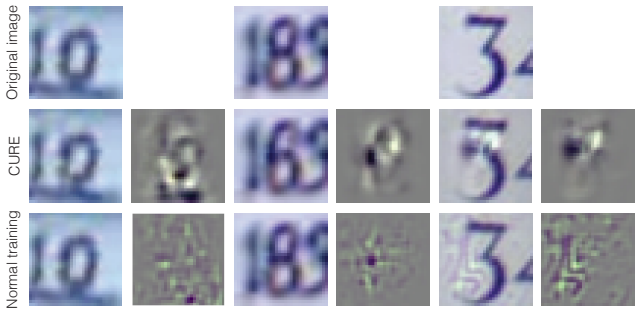


Figure 11: Visualizations of perturbed images and perturbations on SVHN for the ResNet-18 classifier.

for each data point) as a proxy for $\|H\|_F$, the network does show that the intended target $\|H\|_F$ decreases in the course of training, hence further suggesting that $\|Hz\|$ acts as an efficient proxy of the global curvature.

Qualitative evaluation of adversarial perturbations. We finally illustrate some adversarial examples in Fig. 11 for networks trained on SVHN. Observe that the network trained with CURE exhibits visually meaningful adversarial examples, as perturbed images do resemble images from the adversary class. A similar observation for adversarially trained models has been made in [24].

5. Conclusion

Guided by the analysis of the geometry of adversarial training, we have provided empirical and theoretical evidence showing the existence of a strong correlation between small curvature and robustness. To validate our analysis, we proposed a new regularizer (CURE), which directly encourages small curvatures (in other words, promotes local linearity). This regularizer is shown to significantly improve

the robustness of deep networks and even achieve performance that is comparable to adversarial training. In light of prior works attributing the vulnerability of classifiers to the "linearity of deep networks", this result is somewhat surprising, as it shows that one needs to decrease the curvature (and not increase it) to improve the robustness. In addition to validating the importance of controlling the curvature for improving the robustness, the proposed regularizer also provides an efficient alternative to adversarial training. In future work, we plan to leverage the proposed regularizer to train provably robust networks.

Acknowledgements

A.F. would like to thank Neil Rabinowitz and Avraham Ruderman for the fruitful discussions. S.M and P.F would like to thank the Google Faculty Research Award, and the Hasler Foundation, Switzerland, in the framework of the ROBERT project.

References

[1] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations (ICLR)*, 2017.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402, 2013.

[4] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018.

[5] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2017.

[6] A. Fawzi, H. Fawzi, and O. Fawzi. Adversarial vulnerability for any classifier. In *Neural Information Processing Systems (NIPS)*, 2018.

[7] A. Fawzi, O. Fawzi, and P. Frossard. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*, 2015.

[8] A. Fawzi, S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto. Empirical study of the topology and geometry of deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres. In *International Conference on Learning Representations (ICLR)*, 2018.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[11] S. Gu and L. Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

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

[13] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2263–2273, 2017.

[14] S. Jetley, N. Lord, and P. Torr. With friends like these, who needs adversaries? In *Neural Information Processing Systems (NIPS)*, 2018.

[15] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[16] C. Lyu, K. Huang, and H.-N. Liang. A unified gradient regularization family for adversarial examples. In *IEEE International Conference on Data Mining (ICDM)*, 2015.

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.

[20] A. S. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

[21] U. Shaham, Y. Yamada, and S. Negahban. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*, 2015.

[22] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[23] T. Tanay and L. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[24] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[25] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning (ICML)*, 2018.

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

A. Supplementary material

A.1. Results of applying CURE on the SVHN dataset

We fine-tune a pre-trained ResNet-18 using our method, CURE, on SVHN dataset. The learning rate is varying between $[10^{-4}, 10^{-6}]$ for a duration of 20 epochs. The value of γ is set to 4, 8, and 12 for 10, 5, and 5 epochs respectively. Also, for SVHN, we fix $h = 1.25$.

	ResNet-18	
	Clean	Adversarial
Normal training	96.3%	0.9%
CURE	91.1%	28.4%
Adv. training (reported in [4])	93%	33%

Table 3: Adversarial and clean accuracy for SVHN for original, regularized and adversarially trained models. Performance is reported for a ResNet-18 model, and the perturbations are computed using PGD(10) with $\epsilon = 12$.

A.2. Curvature profile of CURE

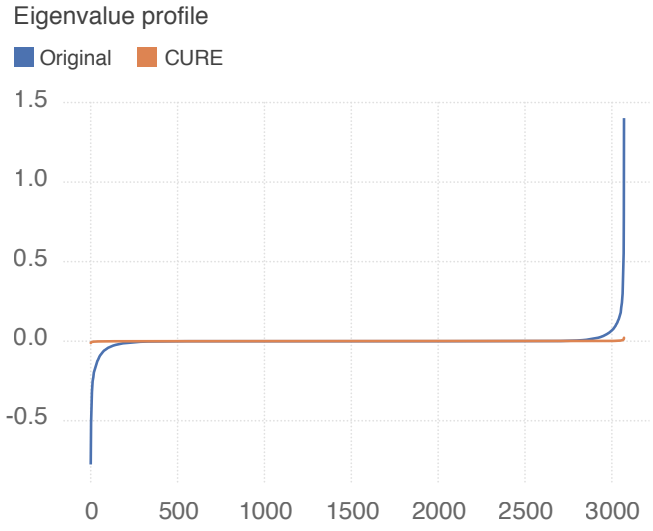
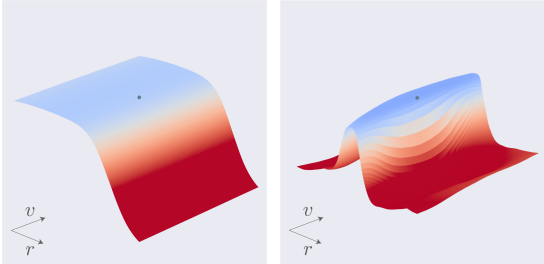


Figure 12: Curvature profiles for a ResNet-18 model trained on SVHN and its fine-tuned version using CURE.

A.3. Loss surface visualization



(a) CURE (b) Original network

Figure 13: Illustration of the negative of the loss surface of the original and the fine-tuned networks trained on SVHN; i.e., $-\ell(s)$ for points s belonging to a plane spanned by a normal direction r to the decision boundary, and random direction v. The original sample is illustrated with a blue dot. The light blue part of the surface corresponds to low loss (i.e., corresponding to the classification region of the sample), and the red part corresponds to the high loss (i.e., adversarial region).

A.4. Evolution of curvature and robustness

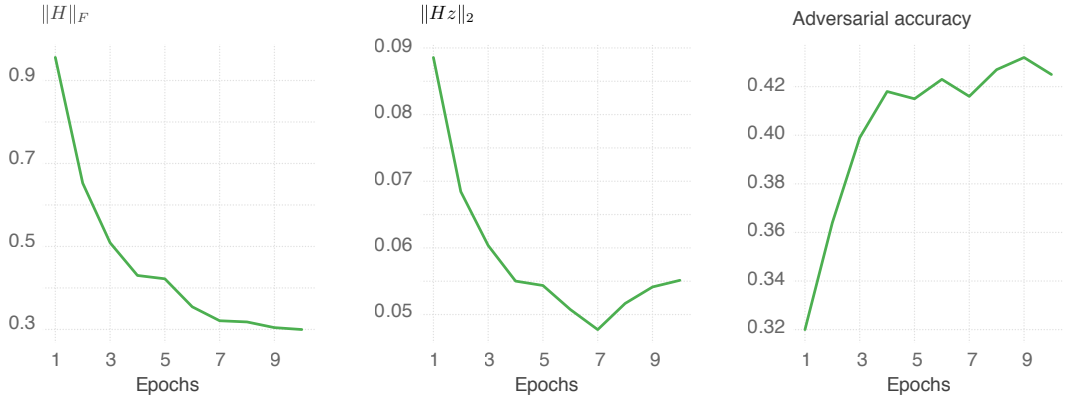


Figure 14: Evolution throughout the course of our CURE fine-tuning for a ResNet-18 on SVHN. The curves are averaged over 1000 datapoints. **Left:** estimate of Frobenius norm , **Middle:** $\|Hz\|$, where $z = \text{sign}(\nabla\ell(x)) = \|\text{sign}(\nabla\ell(x))\|_2$, and **Right:** adversarial accuracy computed using PGD(10) with $\epsilon = 8$. The Frobenius norm is estimated with $\|H\|_F^2 = \mathbb{E}_{z\sim\mathcal{N}(0,I)}\|Hz\|^2$, where the expectation is approximated with an empirical expectation over 100 samples $z_i \sim \mathcal{N}(0, I)$.

A.5. Adversarial accuracy

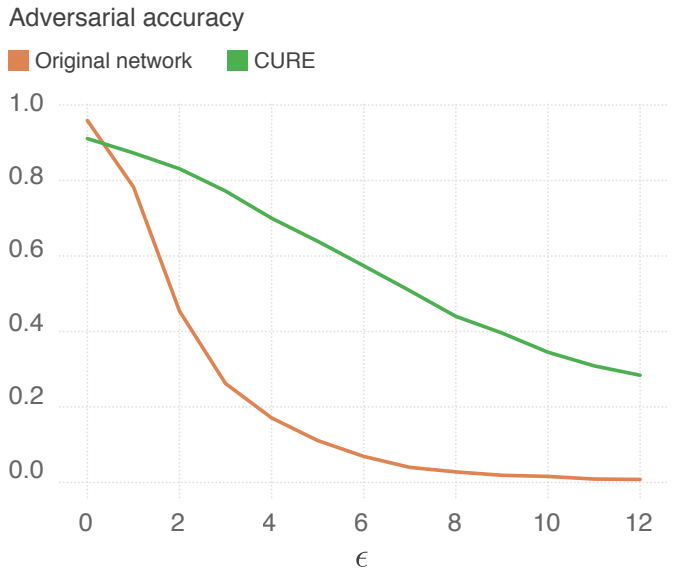


Figure 15: Adversarial accuracy versus perturbation magnitude ϵ computed using PGD(10), for ResNet-18 trained with CURE on SVHN. Curve generated for 2000 random test points.