SN Computer Science manuscript No.
(will be inserted by the editor)

Training Deep Neural Networks for Interpretability and Adversarial Robustness

Adam Noack* · **Isaac Ahern** · **Dejing Dou** ·
Boyang Li

Received: date / Accepted: date

Abstract Deep neural networks (DNNs) have had many successes, but suffer from two major issues: (1) a vulnerability to adversarial examples and (2) a tendency to elude human interpretation. Interestingly, recent empirical and theoretical evidence suggest these two seemingly disparate issues are actually connected. In particular, robust models tend to provide more interpretable gradients than non-robust models. However, whether this relationship works in the opposite direction remains obscure. With this paper, we seek empirical answers to the following question: can models acquire adversarial robustness when they are trained to have interpretable gradients? We introduce a theoretically inspired technique called *Interpretation Regularization* (IR), which encourages a model's gradients to (1) match the direction of interpretable target salience maps and (2) have small magnitude. To assess model performance and tease apart factors that contribute to adversarial robustness, we conduct extensive experiments on MNIST and CIFAR-10 with both ℓ_2 and ℓ_∞ attacks. We demonstrate that networks trained to have interpretable gradients are more robust to adversarial

Conflicts of interest: The work done by Adam Noack was funded by the NSF Center for Big Learning (CBL) and a grant from the Air Force Research Laboratory and Defense Advanced Research Projects Agency, under agreement number FA8750-16-C-0166, subcontract K001892-00-S05. Isaac Ahern's work was funded by the NSF CBL. Dejing Dou was originally funded by the NSF CBL and now works at Baidu. Boyang Li originally worked with Baidu, but now works at Nanyang Technological University.

Adam Noack (* corresponding author)
University of Oregon
E-mail: anoack2@uoregon.edu

Isaac Ahern
University of Oregon
E-mail: iahern@uoregon.edu

Dejing Dou
University of Oregon
E-mail: dou@cs.uoregon.edu

Boyang Li
Nanyang Technological University
E-mail: boyangli@outlook.com

perturbations. Applying the network interpretation technique SmoothGrad [57] yields additional performance gains, especially in cross-norm attacks and under heavy perturbation. The results indicate that the interpretability of the model gradients is a crucial factor for adversarial robustness.

Keywords Machine Learning · Adversarial Robustness · Neural Network Interpretability · Explainable AI · Interpretation Regularization

Declarations

Funding This work was funded by the NSF Center for Big Learning and a grant from the Air Force Research Laboratory and Defense Advanced Research Projects Agency, under agreement number FA8750-16-C-0166, subcontract K001892-00-S05.

Availability of data and material All data and materials we used for our experiments are freely available via PyTorch's torchvision package [48].

Code availability All of the code used for our experiments can be found at this url: `https://github.com/a1noack/interp_regularization`.

Acknowledgements Funding for this project was provided by the National Science Foundation Center for Big Learning and by the Defense Advanced Research Projects Agency's Media Forensics grant.

1 Introduction

Over the past decade, deep neural networks (DNNs) have produced unprecedented results across a wide range of tasks. However, their impressive performance has been clouded by two weaknesses: (1) susceptibility to adversarial perturbations and (2) difficulty in interpreting how they reach their decisions. These weaknesses can erode users' trust in DNNs and limit DNN adoption in security-critical applications, yet our understanding of these two weaknesses is still limited. With this paper, we explore the potential connection between the two phenomena.

Adversarial perturbations are small, almost imperceptible changes to an input that cause a machine learning model to make erroneous predictions [62]. Many attacks that can efficiently find such perturbations have been developed recently, including the fast gradient sign method (FGSM) [23], projected gradient descent (PGD) [38], the Carlini-Wagner attack [10], and many others [45,6,42,13]. In response, many defense techniques have been proposed [23,47,49,17,44,29,37]. Despite the large volume of published work in this area, to date the best defenses remain imperfect, and the cause for the existence of adversarial perturbations continues to be a debated topic [21,52,7,43,28].

A second weakness of DNNs is their opaqueness; even human experts struggle to explain the underlying rationales for DNNs' decisions. The black-box nature of DNNs is especially undesirable in domains such as medicine and law where the reasoning process used to arrive at a decision is often just as important as the decision itself. This need for DNN interpretability has led to the development of interpretation techniques that identify features used by a network to make its prediction [57, 55,2], to visualize the network weights [70,71,5,19], or to calculate training data's influence on the decision [32]. These techniques contribute to the unmasking of the complex mechanisms that underlie DNN behaviors, but by and large DNNs remain incomprehensible black boxes.

Adversarial vulnerability and model opaqueness were previously assumed to be unrelated. However, recent results suggest that the two issues may be connected; specifically, several works have demonstrated, mostly qualitatively, that robust DNNs tend to be interpretable. Tsipras *et al.* [65] found that the loss gradient with respect to the input of adversarially trained networks visually align with human intuition for salient features. Similarly, it has been noticed that gradient regularization [49] and Lipschitz constraints [3], both of which improve adversarial robustness, lead to qualitatively interpretable gradient maps. Etmann *et al.* [18] theoretically showed that, for linear models, Lipschitz regularization causes the gradients to align with the input images. These results constitute a converging collection of evidence that optimizing a network for robustness could result in some degree of interpretability.

With this paper, we explore the other direction of the causality and seek answers to the converse question: *if a network is trained to have interpretable gradients, will it be robust against adversarial attacks?* In the following, we offer some justification for an affirmative answer. At a high level, in order to achieve good adversarial performance, we must maintain high predictive accuracy and curtail the performance degradation caused by adversarial samples at the same time. These two considerations place different requirements on the singular values of the Jacobian. We postulate that an interpretable Jacobian may strike the right balance.

For a given input $\boldsymbol{x}_0 \in \mathbb{R}^D$ and its one-hot encoded label $\boldsymbol{y}_0 \in \mathbb{R}^K$, we adopt a neural network $f(\cdot)$ with ReLU activation, from which the final prediction is $\hat{\boldsymbol{y}}_0 = F(\boldsymbol{x}_0) = \text{softmax}(f(\boldsymbol{x}_0))$. It is worth noting that the function $f(\boldsymbol{x})$ in the neighborhood of $\boldsymbol{x}$ can be written as a sequence of matrix multiplications and is completely linear because ReLU can be understood as zeroing out matrix rows depending on $\boldsymbol{x}$. The Jacobian of the whole network is denoted by $\mathbf{J}(\boldsymbol{x}_0) = \partial\hat{\boldsymbol{y}}_0/\partial\boldsymbol{x}_0$. During an ℓ_2 adversarial attack, the adversary seeks a small perturbation $\boldsymbol{\delta}$ to the input $\boldsymbol{x}_0$ such that the prediction will change significantly and $\|\boldsymbol{\delta}\|_2$ is smaller than a predefined threshold. Let $\rho(\boldsymbol{x})$ be a lower bound on the change in the confidence norm necessary to flip the prediction of $F(\boldsymbol{x})$. For the attack to be successful, we must have $\|F(\boldsymbol{x}_0+\boldsymbol{\delta}) - F(\boldsymbol{x}_0)\|_2 > \rho(\boldsymbol{x}_0)$. First-order Taylor expansion yields

$$\|\mathbf{J}(\boldsymbol{x}_0)\boldsymbol{\delta}\|_2 > \rho(\boldsymbol{x}_0) \tag{1}$$

Under the singular value decomposition, $\mathbf{J}(\boldsymbol{x}_0) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. We let $\boldsymbol{s}$ be the vector of singular values on the diagonal such that $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{s})$ and derive[1]

$$\left\|\boldsymbol{s}^\top\mathbf{V}^\top\boldsymbol{\delta}\right\|_2 = \|\boldsymbol{s}\|_2\,\|\boldsymbol{\delta}\|_2\,|\cos(\boldsymbol{s},\mathbf{V}^\top\boldsymbol{\delta})| > \rho(\boldsymbol{x}_0) \tag{2}$$

where $\cos(\cdot,\cdot)$ is the cosine of the angle between the two vectors. Since the perturbation $\boldsymbol{\delta}$ is chosen by the adversary, to defend we need to minimize $\|\boldsymbol{s}\|_2$ and $\max_{\boldsymbol{\delta}} |\cos(\boldsymbol{s},\mathbf{V}^\top\boldsymbol{\delta})|$. Jacobian regularization minimizes $\|\mathbf{J}(\boldsymbol{x}_0)\|_F$, which is equivalent to $\|\boldsymbol{s}\|_2$. In order to minimize $\max_{\boldsymbol{\delta}} |\cos(\boldsymbol{s},\boldsymbol{\delta})|$, we should make $\boldsymbol{s}$ as uniform as possible.

[1] Here we assume $K = D$ for simplicity. The common case $K < D$ is very similar but involves more complex notations for matrix truncation.

Nevertheless, minimizing the difference $\|F(\boldsymbol{x}_0 + \boldsymbol{\delta}) - F(\boldsymbol{x}_0)\|_2$ can only minimize the damage caused by adversarial examples. Adversarial performance may still be poor if the network is equally mistaken on normal and adversarial examples. In fact, some analysis show that the prioritization dynamics of the Jacobian's singular values allows gradual acquisition of hierarchical concepts [51] and plays crucial roles in the generalization of neural networks [34]. Making $\boldsymbol{s}$ completely uniform would eliminate such benefits and hurt adversarial performance. If $\boldsymbol{s}$ is not uniform, then it necessarily amplifies certain dimensions of $\mathbf{V}^\top \boldsymbol{x}_0$ while attenuating others, which is a form of feature selection. Ilyas *et al.* [28] have noticed the issue of feature selection and argue that DNNs are brittle because they use non-robust features, which are correlated to the class label but incomprehensible for humans. Intuitively, selectively spending the limited budget of $\|\boldsymbol{s}\|_2$ on features that are invariant to adversarial perturbations and comprehensible to humans should improve adversarial accuracy. Therefore, we offer the conjecture that interpretable gradients may strike the right balance between high predictive performance and low adversarial degradation.

We propose to train models to match interpretable gradients, which we call Interpretation Regularization (IR). In order to obtain interpretable gradients, we extract gradient-based interpretations from adversarially trained robust models, which provide more human-like interpretations than non-robust models, and use them as targets during training. We demonstrate that IR improves model robustness and outperforms Jacobian regularization, despite the fact that our method only acts on one column of the input-output Jacobian rather than the entire Jacobian matrix [29,58,26]. Most importantly, target interpretations extracted by SmoothGrad [57], which are smoother and more interpretable than simple gradients, lead to further robustness gains, especially in difficult cases like cross-norm attacks and large perturbations. This indicates that Interpretation Regularization is more than just distilling existing robust models.

It is worth emphasizing that the paper's contribution is in highlighting the connection between interpretability and adversarial robustness. Interpretation Regularization does not and is not intended to provide a practical adversarial defense because it requires an adversarially trained robust model to supply a target interpretation. More specifically, our contributions are:

- We empirically investigate if networks optimized to have interpretable gradients are robust to adversarial attacks. We find that simply requiring the model to match interpretations extracted from a robust model can improve robustness. Applying the network interpretation technique SmoothGrad further reinforces robustness.
- To explain the experimental results, we analyze the connection between Jacobian regularization and Interpretation Regularization. We identify two factors—the suppression of the gradient and the selective use of features guided by high-quality interpretations—that contribute to the effectiveness of Interpretation Regularization and explain model behaviors.

2 Related Work

In this section, we provide a brief review of the vast literature on DNN interpretation, adversarial attacks, and adversarial defenses.

2.1 Interpretation of DNNs

Numerous methods have been proposed to interpret and understand different aspects of DNNs. For example, the representations learned in the network layers can be probed and visualized [70,71,5,19]. The influence of training data on the model's prediction can be estimated [32]. In this paper, we are mostly concerned with interpretations in the form of features' contribution to the model's prediction. When the input is an image, the measure of feature contribution is often referred to as an *importance map* or a *salience map*.

The gradient of the network output with respect to the input provides a simple yet effective method for generating salience maps [4]. The following Taylor expansion approximates the model behavior $f(\boldsymbol{x})$ around $\boldsymbol{x}$.

$$f(\boldsymbol{x}+\boldsymbol{\varepsilon}) = f(\boldsymbol{x}) + \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}^{\top} \boldsymbol{\varepsilon} + o(\boldsymbol{\varepsilon}^{\top}\boldsymbol{\varepsilon}) \tag{3}$$

where $\boldsymbol{\varepsilon}$ represents a small change to $\boldsymbol{x}$. The relative importance of the feature x_i can then be captured by the absolute value $|\frac{\partial f(\boldsymbol{x})}{\partial x_i}|$, which measures how $f(\boldsymbol{x})$ changes when a small change is applied to x_i. While such interpretations highlight salient features of an image, the simple gradient often exhibits a large degree of visual noise and does not always correspond to human intuition regarding feature contribution. This has motivated the development of more elaborate salience map generation techniques in order to induce more structured and visually meaningful interpretations. These include Gradient × Input [55], Integrated Gradients [61], Deep Taylor Decomposition [40], DeepLIFT [55], Guided Backprop [59], and GradCAM / Guided GradCAM [53]. SmoothGrad [57] and VarGrad [1] compute Monte Carlo expectations of the first and second moments of the gradient when noise is added to the input image. Contrastive explanations [15] identify how absent components contribute to the prediction.

Evaluation of the generated salience maps is an important and challenging topic. [31] analyzes behaviors of interpretation methods acting on simple linear models. [1] proposes that salience methods should satisfy include sensitivity towards model and label perturbation. [30] argues they should be invariant with respect to uniform mean shifts of the input. Several popular methods (Integrated Gradients, Guided Backprop, Guided GradCAM, etc) do not satisfy these apparently reasonable requirements.

2.2 Adversarial Attacks

The threat model [8] describes the attacker's goals, knowledge, and capabilities. In terms of goals, untargeted attacks do not care about the model's exact predictions as long as they are incorrect, whereas targeted attacks aim to force a particular erroneous prediction. In terms of knowledge, white-box attacks have access to the model's loss gradients, whereas black-box attacks do not. The attacker's capabilities may be modeled as the amount of perturbation they are allowed to make, usually measured using the ℓ_0, ℓ_2, or ℓ_∞ metric.

Among white-box attacks, FGSM [23] provided a proof-of-concept by adding an ε-scaled sign vector of the loss gradient to the input image. Projected gradient descent (PGD) [38] provides a more powerful iterated optimization approach. Whenever the perturbation magnitude exceeds the attacker's budget, the perturbed input is projected back to the allowed range. The Jacobian-based Saliency Map Attack (JSMA) [46] modifies pixels that have large gradients. DeepFool [41] applies a local linear approximation in iterated optimization. Carlini and Wagner [10] used constrained optimization and reparameterization to effectively search for adversarial samples.

Effective attacks can be built even when gradient information is not available. [45] builds a dataset by querying the target model and use the dataset to train a substitute network from which gradient can be obtained for the attack. Carefully constructed adversarial examples can be transferred across models [36,68] and across images [42]. In addition, gradient-free attacks [13,67,27] do not rely on gradient information. Brendel *et al.* [6] proposed a hard-label attack, which starts from an adversarial point and iteratively reduces the distance to the natural image. [9] demonstrates that methods detecting adversarial examples can be defeated as well.

2.3 Adversarial Defenses

Adversarial training [62,23] is one of the first proposed defenses and remain the most effective. Madry *et al.* [38] show that if the adversary is able to effectively solve the inner maximization problem, the DNN can adjust its parameters to withstand worst-case perturbations. Extensions of adversarial training have been proposed [64,66,54]. [60] builds robustness by applying label smoothing to ℓ_∞ adversarial training. [33] helps to mitigate its negative effect on standard accuracy [65]. Others [22,24,39,50] attempt to detect adversarial examples before feeding them to the network.

As adversaries often exploit noisy and extreme gradients [56], a class of techniques regularize the gradients of the network in order to gain robustness. Ross *et al.* [49] propose a variation of double backpropagation [17], and show that regularizing the loss gradient is an effective defense against FGSM, JSMA, and the targeted gradient sign method. Similarly, Jakubovitz *et al.* [29] propose to regularize the input-logits Jacobian matrix. Furthermore, Parseval Networks [14] constrain the Lipschitz constant of each layer. Cross-Lipschitz regularization [25] forces the differences between gradients of each class score function to be small.

2.4 Relationship Between Adversarial Robustness and Interpretability

Recently, it has been observed that robust networks tend to be more interpretable. Anil *et al.* [3] remark that networks trained with Lipschitz constraints have gradients that appear more interpretable. Similarly, Ross *et al.* [49] find that gradient regularized networks have qualitatively more interpretable gradient maps. Others [65,11] note that the simple gradient salience maps generated from adversarially trained models are more interpretable than those generated from non-robust models. Tsipras *et al.* [65] provide the hypothesis that models that can withstand adversarial examples have necessarily learned to rely on features invariant to adversarial perturbations. Because humans are naturally invariant to these perturbations, robust models tend to function more similarly to the human vision system than non-robust models. Etmann *et al.* [18] provide theoretical justification that robust linear tend to have gradients that are co-linear with the input images, which is related to interpretability.

A couple of works explored the relation between model robustness and interpretability from different perspectives than ours. [16] explain the role of individual neurons with adversarial examples. [20] show that interpretations of neural networks are not immune from adversarial attacks.

Using an approach similar to Generative Adversarial Networks, Chan *et al.* [12] force the Jacobian of the network to contain information needed to reconstruct a natural image. The resulting network becomes robust to certain ℓ_∞ PGD attacks, especially when some adversarial training has been added. However, it remains obscure if the complex training procedure yields interpretable networks. To the best of our knowledge, no work has demonstrated that forcing a model's gradients to be interpretable improves the model's robustness.

3 Approach

The objective of our experiments is to determine if it is possible to make a model robust to adversarial perturbations by optimizing the model to have interpretable gradients. To this end, we supplement the standard cross-entropy loss function with two regularization terms that together encourage the simple gradient salience map for each data point to agree with an interpretable target interpretation.

We introduce the following notations. A data point, drawn from the data distribution D, consists of an input $\boldsymbol{x} \in \mathbb{R}^D$ and a label $\boldsymbol{y} \in \mathbb{R}^K$. Here $\boldsymbol{y}$ is a K-dimensional one-hot vector that contains a single 1 at the correct class y_c and zeros at the other $K-1$ positions. The neural network $f_{\boldsymbol{\theta}}(\cdot)$ has parameters $\boldsymbol{\theta} \in \mathbb{R}^M$ and outputs the logits before the final softmax operation, so that the model prediction for $\boldsymbol{x}$ can be written as $\hat{\boldsymbol{y}} = \text{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{x}))$. Additionally, the input-logits Jacobian matrix $\mathcal{J}(\boldsymbol{x}) \in \mathbb{R}^{K \times D}$ is computed as $\mathcal{J}(\boldsymbol{x}) = \partial f_{\boldsymbol{\theta}}(\boldsymbol{x}) / \partial \boldsymbol{x}$.[2] Of particular interest is the slice of the Jacobian matrix corresponding to the correct label, $\mathcal{J}_c(\boldsymbol{x}) = \langle \partial f_{\boldsymbol{\theta}}(\boldsymbol{x}) / \partial \boldsymbol{x} \rangle_{[c,:]}$, also known as the simple gradient salience map [4].

With standard supervised training, the optimal parameters $\boldsymbol{\theta}^*$ are found by minimizing the cross-entropy loss.

$$\mathcal{L}_{XE}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_i y_i \log \hat{y}_i = y_c \log \hat{y}_c \tag{4}$$

Assuming the availability of target interpretations $I(\boldsymbol{x})$ for each $\boldsymbol{x}$ (covered in Section 3.1), we can add two regularization terms to the standard loss in order to (1) encourage the network to align its gradients with the target interpretations and (2)

[2] $\mathcal{J}(\boldsymbol{x})$ is distinct from the input-output Jacobian matrix $\mathbf{J}(\boldsymbol{x})$. They are related by $\mathbf{J}(\boldsymbol{x}) = (\text{diag}(\hat{\boldsymbol{y}}) - \hat{\boldsymbol{y}}\hat{\boldsymbol{y}}^\top)\mathcal{J}(\boldsymbol{x})$.

restrict the magnitude of its simple gradient salience maps.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbf{D}} [\mathcal{L}_{XE}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}) - \lambda_d \cos(\mathcal{J}_c(\boldsymbol{x}), I(\boldsymbol{x})) + \lambda_m \|\mathcal{J}_c(\boldsymbol{x})\|_F] \tag{5}$$

In the above, $\cos(\mathcal{J}_c(\boldsymbol{x}), I(\boldsymbol{x}))$ is the cosine of the angle between the vectorized target interpretation $I(\boldsymbol{x})$ and the vectorized simple gradient $\mathcal{J}_c(\boldsymbol{x})$. The coefficients λ_d and λ_m control the regularization strengths. Given interpretable target interpretations, these terms encourage the gradients of the model to be interpretable.

3.1 Generating Target Interpretations

Using the SmoothGrad method [57], we extract target salience maps for each data point from a pretrained neural network (details in Section 3.2). We choose SmoothGrad over other interpretation methods for two reasons. First, it satisfies the basic sensitivity and invariance properties [1,30] discussed in Section 2.1, which assert that the interpretation is properly sensitive to the model and data distributions. Second, SmoothGrad can be understood as a method for canceling out the influence of small perturbations on the interpretation, which has the effect of drawing the interpretation closer to what humans find meaningful [57].

The SmoothGrad method first samples N points around a given input $\boldsymbol{x}$ from the standard Gaussian distribution and takes the mean of the simple gradient salience maps generated for each sample. Formally, having drawn N independent $e_i \sim \mathcal{N}(0, \sigma^2)$, the interpretation $I^{\text{SmG}}(\boldsymbol{x})$ is computed as the Monte Carlo expectation.

$$I^{\text{SmG}}(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{J}_c(\boldsymbol{x} + e_i) \tag{6}$$

In order to filter out small values that are usually ignored by a human observer, we further threshold the target salience map with its standard deviation. For each interpretation $I^{\text{SmG}}(\boldsymbol{x})$, we compute the pixel-level standard deviation σ_{S} and mean μ_{S}. Any value in $I^{\text{SmG}}(\boldsymbol{x})$ falling within the range $[\mu_{\text{S}} - \phi\sigma_{\text{S}}, \mu_{\text{S}} + \phi\sigma_{\text{S}}]$ is set to zero. ϕ is a hyperparameter that determines the filtering strength. Figure 1 contains examples of generated target interpretations. It can be observed that the thresholding operation erases the noisy components but retains the important parts of the interpretation.

3.2 Adversarial Training

We create a robust neural network using adversarial training, one of the earliest and still most reliable defenses. The purpose of this model is to supply the target interpretations and serve as the upper bound for robustness in the experiments.

We adopt a PGD adversary that iteratively adds perturbations to an input sample to fool a model. Formally, we let $\boldsymbol{x}_t$ denote the input after t iterations of transformation and $\boldsymbol{x}_0 = \boldsymbol{x}$. After each perturbation is added, the data point is projected to the nearest point within an ℓ_2 hypersphere with the radius ε around $\boldsymbol{x}_0$. This operation

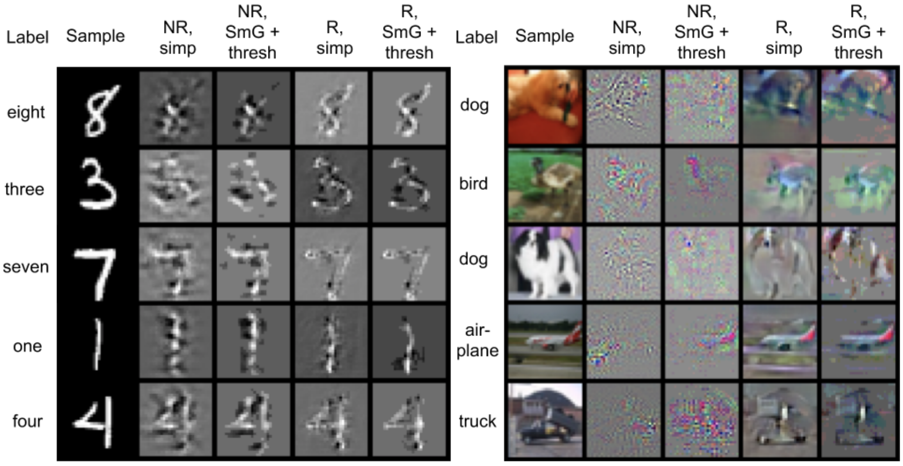

Fig. 1: MNIST (left half) and CIFAR-10 (right half) salience maps from adversarially trained networks (R) and networks with standard training (NR). “simp” and “SmG” denote target interpretations generated using simple gradient and SmoothGrad, respectively. The threshold ϕ is set to 1.

is denoted by the function $\text{clip}_{\boldsymbol{x}_0,\varepsilon}(\cdot)$. The $\text{sign}(\boldsymbol{z})$ function maps the vector $\boldsymbol{z}$ to the element-wise sign $\{-1, 1\}$. The iterative optimization can be characterized as

$$\boldsymbol{x}_{t+1} = \text{clip}_{\boldsymbol{x}_0,\varepsilon}(\boldsymbol{x}_t + \varepsilon \ \text{sign}(\mathcal{L}_{XE}(\boldsymbol{x}_t, \boldsymbol{y}, \boldsymbol{\theta}))), \tag{7}$$

$$\text{clip}_{\boldsymbol{x}_0,\varepsilon}(\boldsymbol{x}') = \begin{cases} \boldsymbol{x}_0 + \frac{\boldsymbol{x}'-\boldsymbol{x}_0}{\|\boldsymbol{x}'-\boldsymbol{x}_0\|_2}\varepsilon & \text{if } \|\boldsymbol{x}' - \boldsymbol{x}_0\|_2 > \varepsilon \\ \boldsymbol{x}' & \text{otherwise.} \end{cases} \tag{8}$$

After the adversarial examples are created, they are given the original labels and used in place of the original samples for the training of a robust model.

3.3 Jacobian Regularization

Jacobian regularization [29] is another defense technique, which supplements the original cross entropy loss with a regularization term.

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathbf{D}} [\mathcal{L}_{XE}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) + \lambda_J \|\mathcal{J}(\boldsymbol{x})\|_F] \tag{9}$$

where $\|\cdot\|_F$ is the Frobenius norm and λ_J is a hyperparameter determining the strength of the regularization. Contrasting Eq. 9 with Eq. 5, the regularization term in Jacobian Regularization suppresses all entries in the Jacobian matrix, whereas Interpretation Regularization is only concerned with one slice of the Jacobian that corresponds to the correct label.

Hoffman *et al.* [26] demonstrate that the Frobenius norm of the Jacobian can be approximated using random projections.

$$\|\mathcal{J}(\boldsymbol{x})\|_F \approx \sqrt{\frac{1}{n_{proj}} \sum_{\mu=1}^{n_{proj}} \left[\frac{\partial(\boldsymbol{v}^{\mu} \cdot f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{\partial x}\right]^2} \tag{10}$$

Here the random projection vector $\boldsymbol{v}^{\mu}$ is drawn from the $(K-1)$-dimensional unit sphere for every training iteration. In practice, Hoffman *et al.* show that even a single random projection vector is effective at suppressing the Jacobian norm.

4 Experiments

In this section, we present three experiments. In the first two experiments, we compare Interpretation Regularization with adversarial training, Jacobian regularization, and ablated variants on MNIST and CIFAR-10. After that, we further explore the role of the target interpretation by using target interpretations permuted to different degrees. Finally, we discuss the results and their implications.

4.1 General Setup

We first describe some general setup that applies to all experiments. The pixel values for each image were normalized to the range $[0,1]$. We used ℓ_2 adversarial training (AT) throughout, as preliminary results showed Interpretation Regularization works better with interpretation targets from ℓ_2 AT than ℓ_∞ AT. Note that, on MNIST, ℓ_2 PGD adversaries tend to be less effective than ℓ_∞ adversaries and provide weaker defenses [38,35]. We attack all networks using both ℓ_2 and ℓ_∞ attacks.

The target interpretations are generated in the following manner. We followed the original recommendation for SmoothGrad [57] and set the noise level at $\sigma = 0.15$ and number of samples N to 50. The filtering threshold was set to $\phi = 1$ (See Section 3.1 for details). We extract simple gradient and SmoothGrad saliency maps from ℓ_2 adversarially trained networks as well as non-robust networks that are trained only on natural images. In addition, we also create a *complete* random permutation (permutation probability at 1.0) of the robust SmoothGrad interpretation. For each dataset, we create baselines using adversarial training and Jacobian regularization. In order to facilitate meaningful comparisons, all networks were trained to have roughly the same validation loss on natural images.

Due to space limitations, we use some shorthands in tables and figures to denote these configurations. “R” and “NR” denote target interpretations generated from robust and non-robust or standard trained networks, respectively. Simple gradient and SmoothGrad are denoted by “simp” and “SmG”, respectively. “perm” means that each target interpretation was completely randomly permuted. “IR” indicates Interpretation Regularization, “AT” indicates adversarial training, and “JR” stands for Jacobian regularization [26].

4.2 MNIST

The MNIST experiments are set up in the following ways. The convolutional neural network (CNN), taken from [63], has two convolutional layers of 32 and 64 filters of size 5×5 and stride 1. Each convolutional layer is followed by max-pooling with a 2×2 kernel and a stride of 2. The last pooling layer feeds into a dense layer with 1024 neurons. Dropout with $p = 0.5$ is applied before the final dense layer of 10 neurons and the softmax operation. All layers except the last employ the ReLU activation function. All experiments employ SGD with momentum at 0.9, an initial learning rate of 0.01 that decayed to zero using the cosine schedule, a batch size of 50, and the maximum number of epochs was 100.

Following [65], we extract robust interpretations from an adversarially trained CNN with a randomly initialized PGD adversary using an ℓ_2 radius of 1.5 and 40 iterations of PGD. Tspiras *et al.* [65] qualitatively showed that training against this adversary produced networks that had interpretable simple gradient salience maps. This network, along with a second network trained with a PGD adversary with an ℓ_2 radius of 2.5 and 40 iterations of PGD, serve as baselines.

Using the robust network's simple gradient maps as the target interpretations (R, simp), we performed a grid search to find the best combination of λ_d (which controls the strength of the gradient alignment with $I(x)$) and λ_m (which controls the magnitude of $\|\mathcal{J}_{c(\boldsymbol{x})}(\boldsymbol{x})\|_F$). λ_d at 3.0 and λ_m at 0.15 produced the best results. The other four sets of target interpretations simply reused these values and did not employ any additional tuning. For baselines, we created adversarially trained networks with radii of 1.5 and 2.5. For Jacobian regularization, we performed a search across λ_J and found 0.32 to produce good results.

4.3 CIFAR-10

For all experiments with the CIFAR-10 dataset, the Wide ResNet (WRN) 28×10 architecture [69], a large network with 36.5 million parameters, was used. We adopted weight decay of $5\mathrm{e}{-4}$, dropout rate of 0.3, SGD with Nesterov momentum at 0.9, an initial learning rate of 0.1 decaying to zero under the cosine schedule, batch size of 128, and 200 training epochs. As data augmentation, the input images were randomly cropped and horizontally flipped during training, and each target interpretation was transformed in the same way as its corresponding input image.

We obtain robust networks using adversarial training with PGD, two different ℓ_2 radii of $80/255$ and $320/255$, and 7 PGD iterations. Again, this is the same setup which showed qualitatively interpretable gradients in [65]. For Jacobian regularization, we adopt the approximation from [26] with $n_{proj} = 1$ to save computation and do a search across λ_J, finding $\lambda_J = 0.1$ and 0.03 to produce good results. For the simple gradient map of the robust network (R, simp), λ_d at 0.75 and λ_m at 0.005 or 0.02 were found by a grid search to produce good results and validation losses on natural images that were comparable with the two adversarially trained networks. These hyperparameters were used across the other four Interpretation Regularization experiments with no further tuning.

		Adversarial Accuracy				
Training Technique	**Standard Accuracy**	PGD40, ℓ_2 norm	PGD40, ℓ_∞ norm			
		$\varepsilon = 1.50$	0.10	0.20	0.25	0.30
Standard Training	99.50	78.41	78.91	9.32	3.57	1.71
AT (PGD40, ℓ_2, ε=1.5)	99.39	**89.88**	**96.60**	73.01	32.07	5.46
AT (PGD40, ℓ_2, ε=2.5)	98.29	88.06	94.23	**76.39**	**49.98**	10.94
JR	98.12	82.64	91.15	60.58	29.41	6.54
IR (R, perm, SmG)	97.28	78.75	88.11	54.36	26.26	7.58
IR (NR, simp)	98.05	79.57	87.82	50.20	23.71	2.29
IR (NR, SmG)	98.04	81.22	90.39	55.98	28.70	4.98
IR (R, simp)	98.12	84.24	91.60	64.03	*37.90	*10.86
IR (R, SmG)	98.18	*85.25	*92.35	*66.92	41.22	**11.52**

Table 1: Mean MNIST adversarial accuracies on the test set averaged over 3 random restarts of the attacks. The first, second, and third highest accuracies for each adversary are bolded, underlined, and asterisked, respectively.

4.4 MNIST Permuted Interpretation

To further investigate the effects of the target interpretation, we conduct an additional experiment with new sets of permuted target interpretations. We first extract the SmoothGrad interpretation from the robust network, which was adversarially trained on the MNIST dataset as in Section 4.2. After that, we randomly permute from 10% to 100% of pixels in each interpretation to obtain ten new sets of interpretations. Each of these sets is then used as target interpretations for Interpretation Regularization. Figure 3 shows some example target interpretations from each of the ten sets. In this way, the mean and standard deviations of the pixel values in each target interpretation is held constant, but the semantic patterns in the target interpretations are disrupted to varying extents.

The optimal λ_d and λ_m from Section 4.2 were used without any changes. In other words, the only hyperparameter changing across networks being trained in this experiment was the permutation probability for the target interpretations. The results for this experiment can be found in Fig. 2.

4.5 Results

Tables 1 and 2 report the standard and adversarial accuracies under different ℓ_2 and ℓ_∞ norm constraints on MNIST and CIFAR-10. Since all adversarial training was performed with an ℓ_2 adversary and the target interpretations extracted accordingly, the attacks from ℓ_∞ adversaries create challenging defense transfer scenarios for the defense methods.

With unperturbed data, standard training achieves the highest accuracy and all defense techniques degrade the performance. The adversarial attacks prove effective, resulting in substantial performance degradation of the standard model. Three of the four attacks on CIFAR-10 brought the standard model's accuracy to below 1%. For

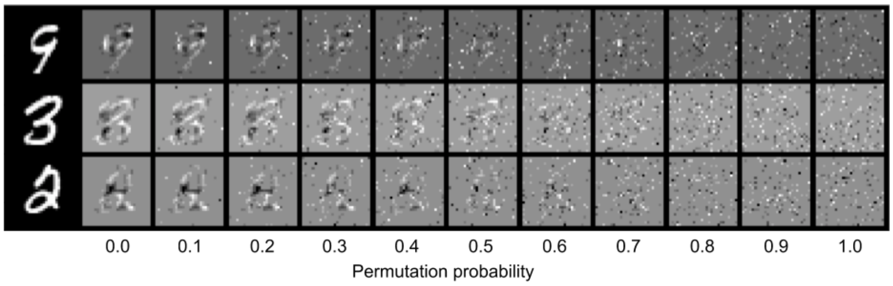

Fig. 2: Three SmoothGrad target interpretations with varying degrees of permutation. The original images (with labels of 9, 3, and 2, from top to bottom) are in the leftmost column.

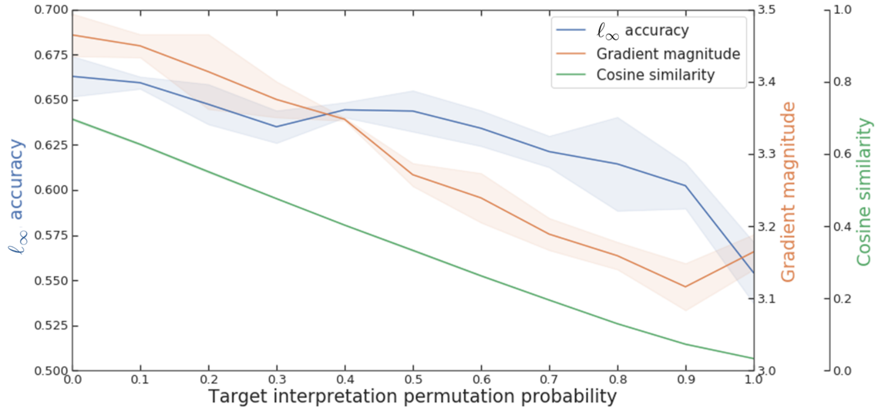

Fig. 3: Adversarial accuracy under differently permuted target interpretations. On the y-axis: PGD-40 ℓ_∞ ($\varepsilon = .2$) adversarial accuracy, gradient magnitudes $\|\mathcal{J}_c(\boldsymbol{x})\|_F$, and average cosine similarity between target interpretations and the gradients. The x-axis represents the permutation probability for the target interpretations. 95% confidence interval shading over $n = 3$ independently trained networks.

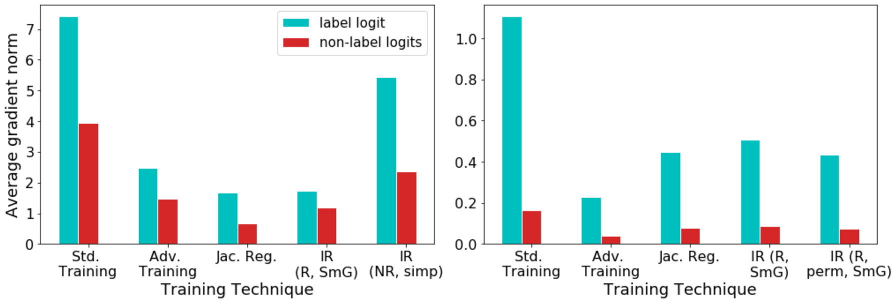

Fig. 4: Mean Frobenius norms of the Jacobians for the label logit, $\|\mathcal{J}_c(\boldsymbol{x})\|_F$, and non-label logits $\|\mathcal{J}_{\neq c}(\boldsymbol{x})\|_F$ for MNIST (left) and CIFAR-10 (right).

Training Technique	**Standard Accuracy**	**Adversarial Accuracy**			
		PGD40, ℓ_2 norm		PGD40, ℓ_∞ norm	
		ε = 80/255	$320/255$	$4/255$	$8/255$
Standard Training	95.00	11.84	0.00	0.74	0.01
AT (PGD7, ℓ_2, ε=320/255)	76.04	**67.57**	**58.29**	**59.05**	**38.4**
JR	78.93	61.41	8.35	42.22	11.23
IR (R, simp)	78.68	$\underline{63.74}$	$\underline{12.56}$	47.18	16.28
IR (R, SmG)	78.97	62.64	12.42	$\underline{48.17}$	$\underline{18.78}$
AT (PGD7, ℓ_2, ε=80/255)	90.34	**75.14**	**20.44**	**59.43**	**25.37**
JR	85.41	58.96	3.32	32.27	2.51
IR (NR, simp)	81.43	45.04	0.18	16.69	0.40
IR (NR, SmG)	84.39	53.84	0.56	28.00	1.84
IR (R, perm, SmG)	85.70	58.20	1.91	29.19	1.87
IR (R, simp)	85.39	62.45	5.35	39.93	8.2
IR (R, SmG)	85.69	$\underline{63.71}$	$\underline{7.98}$	$\underline{46.64}$	$\underline{14.25}$

Table 2: Mean CIFAR-10 adversarial accuracies on test set averaged over 3 random restarts of the attacks. Within each group of models, the highest and second highest accuracies for each adversary are bolded and underlined, respectively.

most attacks, adversarial training yields the highest adversarial performance and Interpretation Regularization is the second best. However, in the two highest difficulty setting on MNIST (ℓ_∞ norms 0.25 and 0.30), IR begins to surpass AT and becomes the most robust network under 0.30. In addition, IR outperforms JR on all attacks. The performance differences between the two methods range from 1.2% to 11.81% on MNIST and from 2.34% to 14.37% on CIFAR-10.

The best IR performance is achieved, in almost all cases, by IR(R, SmG), which uses interpretations from SmoothGrad and the AT network. The performance of IR(R,simp) and IR(R, SmG) diverge the most, by 6.67% and 6.05%, when the interpretations are derived from ℓ_2 AT with a low $\varepsilon = 80/255$ and attacked by the ℓ_∞ adversary. That is, when the adversarial training and attack have the most mismatch. On the other hand, IR(R,simp) have a slight edge of 1.1% or less over IR(R, SmG) when the adversarial training use an ℓ_2 adversaries with a large $\varepsilon = 320/255$ and the attacks come from ℓ_2 adversaries. Finally, interpretations from non-robust models offer some robustness over standard training, but they compare unfavorably with JR or permuted interpretations.

Figure 3 shows the effects of random permutation on the interpretation. The overall trend is quite clear: greater permutation causes lower adversarial accuracy. Furthermore, as the permutation increases, the network becomes less and less able to align its gradients with the target interpretations. Not included in the graph are the standard accuracies of the networks; these accuracies trend monotonically downward as well, beginning at with an average of 98.13% and ending with an average of 97.37%, and the Pearson correlation coefficient between permutation probability and standard accuracy is -0.93.

4.6 Discussion

Disentangling the effects of Jacobian norms and target interpretations. In the introduction, we show that in order to defend against an arbitrary perturbation $\boldsymbol{\delta}$, we could suppress the Jacobian's singular values $\|\boldsymbol{s}\|_2$, which is equivalent to the Jacobian's Frobenius norm. This gives us Jacobian regularization. To verify that the suppression has happened, we plot the Frobenius norm of the input-logits Jacobians in Figure 4. We separate the Jacobian slices that correspond to the correct class, $\mathcal{J}_c(\boldsymbol{x})$, and those of the incorrect classes $\mathcal{J}_{\neq c}(\boldsymbol{x})$. The results are averaged over all training samples. For the incorrect class slices, the results are also averaged over all output logits that differ from the ground truth.

We find that almost all defense methods reduce the norms of Jacobians compared to standard training. Previously, we also observed that IR with permuted target interpretations can provide some adversarial robustness, even though it performs worse than JR. We attribute this effect to the fact that, even with a completely uninformative target interpretation, IR still decreases the Frobenius norm of Jacobians, which can improve robustness. In addition, the results show that for most models, the correct-class Jacobian norm was much larger than wrong-class Jacobian norms. This explains why IR is effective when it only constrains the correct-class Jacobian norm whereas JR constrains all slices of the Jacobian.

However, it is also worth noting that lower norms do not always lead to better adversarial performance. In both MNIST and CIFAR-10 experiments, JR produces lower Jacobian norms than IR, but is consistently outperformed across all attacks. This indicates there are other factors at play.

To further disentangle the effects of Jacobian norms and the interpretability of the Jacobians, we examine how degrees of random permutation affect adversarial robustness in IR (shown in Figure 3). As the proportion of permuted pixels increases, the network gradually becomes less capable of withstanding ℓ_∞ attacks. Nevertheless, the reduction in robustness happens while the gradient magnitude (Jacobian's norm) decreases. This behavior cannot be explained from the perspective of Jacobian regularization or the minimization of $\|\boldsymbol{s}\|_2$. With the other hyperparameters and training losses kept equal, we attribute the decrease in performance to the decline in quality of target interpretations.

The quality of the interpretation matters. We now examine how the interpretability of the Jacobian contribute to adversarial robustness. In the analysis in the introduction, we inferred that it is important to allocate the available budget of $\|\boldsymbol{s}\|_2$ carefully in order to maximize predictive performance. Inspired by [28], we conjecture that an interpretable Jacobian, which selects features that humans regard as important to the prediction, should provide adversarial robustness.

Empirical evidences from the MNIST and CIFAR-10 experiments strongly corroborate this argument. First, models trained with target interpretations from robust models consistently outperform target interpretations from non-robust models. Second, random permutation of the interpretations causes significant performance drop. The final and the most compelling observation is that, in most cases, SmoothGrad interpretations perform better than simple gradient maps from both robust and non-robust models. This is especially pronounced when the attack uses a large ℓ_∞ perturbation, which delivers severe attacks for robust models trained with ℓ_2 perturbations. In the MNIST experiments with ℓ_∞ radius of 0.25, IR(R,SMG) beats the AT network trained with ℓ_2 radius of 1.5, from which the target interpretations for IR are extracted. Moreover, at ℓ_∞ radius of 0.30, IR(R,SMG) obtains the best robustness, surpassing even the AT network trained with ℓ_2 radius of 2.5. In CIFAR-10, under ℓ_∞ perturbations, the performance of IR(R,SMG) always exceeds that of IR(R,simp).

We ascribe the strength of SmoothGrad to the fact that it removes noise-like patterns in the interpretation and creates more human-like interpretations than simple gradient. A qualitative observation of Figure 1 suggests that SmoothGrad interpretations on MNIST are consistent with human intuition. For example, the black spots (negative gradient values) for the digit 3 indicate key differences between 3 and the digits 6 or 8 and thus supply important features for classification. Similarly, the black spots around the top of the digit 4 highlight the differences with the digit 9. The strong SmoothGrad performance shows that interpretability is directly correlated with adversarial robustness and Interpretation Regularization attains more than just the distillation of adversarially trained models.

5 Conclusion

The abundance of adversarial attacks and the lack of interpretation of how a deep neural network makes its predictions are two issues that render some applications of artificial intelligence untrustworthy in the eye of the general public. The literature suggests that these two issues may be closely related, as works have indicated qualitatively that adversarial defenses techniques, such as adversarial training [65], Jacobian regularization [49], and Lipschitz constraints [18] produce models that have salience maps that agree with human interpretations.

These findings naturally lead to the question if the converse is true. If we force a neural network to have interpretable gradients, will it then become robust? We devise a technique called Interpretation Regularization, which regularizes the gradient of a model to match the target interpretation extracted from an adversarially trained robust model. The new model performs better than Jacobian regularization, which applies more constraints than Interpretation Regularization. Most importantly, applying the network interpretation technique SmoothGrad [57] improves robustness over simple gradients, and in few cases, over the AT networks from which the target interpretations are extracted. These results suggest Interpretation Regularization accomplishes more than distilling existing robust models.

In the discussion, we carefully disentangle two factors that contribute to the effectiveness of Interpretation Regularization: the suppression of the gradient and the selective use of features guided by high-quality interpretations. With the two factors, we manage to explain model behaviors under various settings of regularization and target interpretation. We believe this study provides useful insights into the research of adversarial defenses and interpretation methods. The joint investigation of these two issues will continue to foster our understanding of deep neural networks.

Compliance with Ethical Standards

Conflicts of interest: The work done by Adam Noack was funded by the NSF Center for Big Learning (CBL) and a grant from the Air Force Research Laboratory and Defense Advanced Research Projects Agency, under agreement number FA8750-16-C-0166, subcontract K001892-00-S05. Isaac Ahern's work was funded by the NSF CBL. Dejing Dou was originally funded by the NSF CBL and now works at Baidu. Boyang Li originally worked with Baidu, but now works at Nanyang Technological University.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Funding This work was funded by the NSF Center for Big Learning and a grant from the Air Force Research Laboratory and Defense Advanced Research Projects Agency, under agreement number FA8750-16-C-0166, subcontract K001892-00-S05.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I.J., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
2. Ahern, I., Noack, A., Guzman-Nateras, L., Dou, D., Li, B., Huan, J.: Normlime: A new feature importance metric for explaining deep neural networks. arXiv Preprint (2019)
3. Anil, C., Lucas, J., Grosse, R.B.: Sorting out Lipschitz function approximation. In: Proceedings of the International Conference on Machine Learning (ICML) (2018)
4. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. Journal of Machine Learning Research **11**, 1803–1831 (2010)
5. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3319–3327 (2017)
6. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint (2017)
7. Bubeck, S., Price, E., Razenshteyn, I.: Adversarial examples from computational constraints. Proceedings of the International Conference on Machine Learning (2019)
8. Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. arXiv Preprint (arXiv 1902.06705) (2019)
9. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (2017)
10. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: The IEEE Symposium on Security and Privacy (2017)
11. Chalasani, P., Jha, S., Sadagopan, A., Wu, X.: Adversarial learning and explainability in structured datasets. arXiv Preprint (arXiv 1810.06583) (2018)
12. Chan, A., Tay, Y., Ong, Y.S., Fu, J.: Jacobian adversarially regularized networks for robustness. In: International Conference on Learning Representations (2020)
13. Chen, P.Y., Zhang, H., Sharma, Y., Yi, J., Hsieh, C.J.: 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 Intelligenceand Security, p. 15âĂŞ26 (2017)
14. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., Usunier, N.: Parseval networks: Improving robustness to adversarial examples. In: Proceedings of the International Conference on Machine Learning, pp. 854–863 (2017)
15. Dhurandhar, A., Chen, P.Y., Luss, R., Tu, C.C., Ting, P., Shanmugam, K., Das, P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
16. Dong, Y., Su, H., Zhu, J., Bao, F.: Towards interpretable deep neural networks by leveraging adversarial examples. In: AAAI-19 Workshop on Network Interpretability for Deep Learning (2017)
17. Drucker, H., LeCun, Y.: Double backpropagation increasing generalization performance. In: Proceedings of the International Joint Conference on Neural Networks, pp. 145–150 (1992)

18. Etmann, C., Lunz, S., Maass, P., SchÃűnlieb, C.B.: On the connection between adversarial robustness and saliency map interpretability. In: Proceedings of the International Conference on Machine Learning (2019)
19. Fong, R., Vedaldi, A.: Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. arXiv preprint arXiv:1801.03454 (2018)
20. Ghorbani, A., Abid, A., Zou, J.Y.: Interpretation of neural networks is fragile. In: AAAI (2017)
21. Gilmer, J., Metz, L., Faghri, F., Schoenholz, S.S., Raghu, M., Wattenberg, M., Goodfellow, I.J.: Adversarial spheres. In: Workshop of International Conference on Learning Representations (ICLR) (2018)
22. Gong, Z., Wang, W., Ku, W.S.: Adversarial and clean data are not twins. arXiv Preprint (2017)
23. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015)
24. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.: On the (statistical) detection of adversarial examples. arXiv Preprint (2017)
25. Hein, M., Andriushchenko, M.: Formal guarantees on the robustness of a classifier against adversarial manipulation. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
26. Hoffman, J., Roberts, D.A., Yaida, S.: Robust learning with jacobian regularization. In: arxiv Preprint (2019)
27. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attackswith limited queries and information. In: Proceedings of the International Conference on Machine Learning (ICML) (2018)
28. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
29. Jakubovitz, D., Giryes, R.: Improving DNN robustness to adversarial attacks using jacobian regularization. In: ECCV (2018)
30. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (Un)reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, vol. 11700. Springer (2019)
31. Kindermans, P.J., SchÃijtt, K.T., Alber, M., MÃijller, K.R., Erhan, D., Kim, B., DÃď’hne, S.: Learning how to explain neural networks: PatternNet and PatternAttribution (2017)
32. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: International Conference on Machine Learning (2017)
33. Lamb, A., Verma, V., Kannala, J., Bengio, Y.: Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. arXiv Preprint (2019)
34. Lampinen, A.K., Ganguli, S.: An analytic theory of generalization dynamics and transfer learning in deep linear networks. In: International Conference on Learning Representations (ICLR) (2019)
35. Li, B., Chen, C., Wang, W., Carin, L.: Second-order adversarial attack and certifiable robustness (2018)
36. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. In: Proceedings of the International Conference on Learning Representation (ICLR) (2017)
37. LÃl’cuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., Jana, S.: Certified robustness to adversarial examples with differential privacy. pp. 656–672 (2019). DOI 10.1109/SP.2019.00044
38. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2017)
39. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv Preprint (2017)
40. Montavon, G., Samek, W., Müller, K.: Methods for interpreting and understanding deep neural networks. arXiv Preprint (2017)
41. Moosavi-Dezfooli, S., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: CVPR (2016)
42. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
43. Nakkiran, P.: Adversarial robustness may be at odds with simplicity. ArXiv (2019)
44. Oberman, A.M., Calder, J.: Lipschitz regularized deep neural networks converge and generalize. arXiv Preprint (arxiv 1808.09540) (2018)
45. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519. ACM, New York, NY, USA (2017)
46. Papernot, N., McDaniel, P.D., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. IEEE European Symposium on Security and Privacy (2016)

47. Papernot, N., McDaniel, P.D., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy (2016)
48. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Pytorch torchvision (2017)
49. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: AAAI (2018)
50. Roth, K., Kilcher, Y., Hofmann, T.: The odds are odd: A statistical test for detecting adversarial examples. In: Proceedings of the International Conference on Machine Learning (ICML) (2019)
51. Saxe, A.M., McClelland, J.L., Ganguli, S.: A mathematical theory of semantic development in deep neural networks. Proceedings of the National Academy of Sciences **116**(23), 11537–11546 (2019). DOI 10.1073/pnas.1820226116
52. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
53. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. ICCV (2017)
54. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Advances in Neural Information Processing Systems, pp. 3358–3369 (2019)
55. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the International Conference on Machine Learning (2017)
56. Simon-Gabriel, C.J., Ollivier, Y., Bottou, L., Schölkopf, B., Lopez-Paz, D.: First-order adversarial vulnerability of neural networks and input dimension. In: K. Chaudhuri, R. Salakhutdinov (eds.) Proceedings of the 36th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 97, pp. 5809–5817. PMLR, Long Beach, California, USA (2019)
57. Smilkov, D., Thorat, N., Kim, B., Viégas, F.B., Wattenberg, M.: Smoothgrad: removing noise by adding noise. In: Proceedings of the International Conference on Machine Learning (2017)
58. Sokolic, J., Giryes, R., Sapiro, G., Rodrigues, M.R.D.: Robust large margin deep neural networks. In: IEEE Transactions on Signal Processing, vol. 65, pp. 4265–4280 (2016)
59. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. In: ICLR Workshop (2014)
60. Stutz, D., Hein, M., Schiele, B.: Confidence-calibrated adversarial training: Generalizing to unseen attacks. arXiv Preprint (2019)
61. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the International Conference on Machine Learning (2017)
62. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Proceedings of the International Conference on Learning Representation (ICLR) (2014)
63. TensorFlow: TensorFlow models repository (2017). URL `https://github.com/tensorflow/models/blob/master/tutorials/image/mnist/convolutional.py`
64. TramÃÍr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv Preprint (2017)
65. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness May Be at Odds with Accuracy. In: Proceedings of the International Conference on Learning Representation (ICLR) (2019)
66. Uesato, J., Alayrac, J.B., Huang, P.S., Stanforth, R., Fawzi, A., Kohli, P.: Are labels required for improving adversarial robustness? In: Advances in Neural Information Processing Systems (2019)
67. Uesato, J., OâĂŹDonoghue, B., van den Oord, A., Kohli, P.: Adversarial risk and the dangers of evaluating against weak attacks. arXiv preprint (2018)
68. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.: Improving transferability of adversarial examples with input diversity. arXiv Preprint (2018)
69. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC) (2016)
70. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 818–833 (2014)
71. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. In: Proceedings of International Conference on Learning Representations (ICLR) (2015)