Journal Pre-proof

Jacobian norm with selective input gradient regularization for interpretable adversarial defense

Deyin Liu, Lin Yuanbo Wu, Bo Li, Farid Boussaid, Mohammed Bennamoun, Xianghua Xie, Chengwu Liang

PII: S0031-3203(23)00600-3
DOI: https://doi.org/10.1016/j.patcog.2023.109902
Reference: PR 109902

To appear in: *Pattern Recognition*

Received date : 5 May 2023
Revised date : 29 July 2023
Accepted date : 21 August 2023

Please cite this article as: D. Liu, L.Y. Wu, B. Li et al., Jacobian norm with selective input gradient regularization for interpretable adversarial defense, *Pattern Recognition* (2023), doi: https://doi.org/10.1016/j.patcog.2023.109902.

Jacobian Norm with Selective Input Gradient Regularization for Interpretable Adversarial Defense

Deyin Liu[a], Lin Yuanbo Wu[b], Bo Li[c], Farid Boussaid[d], Mohammed Bennamoun[d], Xianghua Xie[b], Chengwu Liang[e]

[a]*Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, School of Artificial Intelligence, Anhui University, Hefei, 230039, Anhui, China*
[b]*Department of Computer Science, Swansea University, Swansea, SA1 8EN, Wales, United Kingdom*
[c]*Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, China*
[d]*The University of Western Australia, Perth, 6009, WA, Australia*
[e]*Henan University of Urban Construction, Pingdingshan, 467036, Henan, China*

Abstract

Deep neural networks (DNNs) can be easily deceived by imperceptible alterations known as adversarial examples. These examples can lead to misclassification, posing a significant threat to the reliability of deep learning systems in real-world applications. Adversarial training (AT) is a popular technique used to enhance robustness by training models on a combination of corrupted and clean data. However, existing AT-based methods often struggle to handle transferred adversarial examples that can fool multiple defense models, thereby falling short of meeting the generalization requirements for real-world scenarios. Furthermore, AT typically fails to provide interpretable predictions, which are crucial for domain experts seeking to understand the behavior of DNNs. To overcome these challenges, we present a novel approach called Jacobian norm and Selective Input Gradient Regularization (J-SIGR). Our method leverages Jacobian normalization to improve robustness and introduces regularization of perturbation-based saliency maps, en-

abling interpretable predictions. By adopting J-SIGR, we achieve enhanced defense capabilities and promote high interpretability of DNNs. We evaluate the effectiveness of J-SIGR across various architectures by subjecting it to powerful adversarial attacks. Our experimental evaluations provide compelling evidence of the efficacy of J-SIGR against transferred adversarial attacks, while preserving interpretability. The project code can be found at https://github.com/Lywu-github/jJ-SIGR.git.

Keywords: Selective Input Gradient Regularization, Jacobian Normalisation, Adversarial Robustness

1. Introduction

In recent years, deep neural networks (DNNs) have demonstrated impressive performance in various image recognition tasks with real-world benchmarks. However, DNNs are vulnerable to imperceptible perturbations that can easily deceive the network into making incorrect predictions. This susceptibility poses significant obstacles in the deployment of deep learning systems for real-world applications. Since this vulnerability was originally identified by Szegedy et al. [1], there have been a pyramid of techniques for generating malicious examples using either white-box attacks [2, 1, 3] or transferable white-box attacks [4, 5, 6]. The transferable adversarial examples crafted by black-box attacks, which can attack DNNs without any knowledge of the model parameters, poses the grave threat to deep learning systems, implying that examples generated to fool one model can fool *all* models trained on the same dataset. Realising that these vulnerabilities can have serious security implications, researchers have focused on developing techniques to

mitigate these vulnerabilities, such as adversarial training and defense mechanisms. Adversarial training involves training the network on a mixture of corrupted and clean data to improve robustness, while defense mechanisms aim to detect and reject adversarial examples. However, these techniques do not guarantee complete robustness, and the transferability of adversarial examples has made this problem even more difficult to solve.

Currently, the most effective defense mechanisms are based on adversarial training. By training models on a mixture of clean and adversarial examples, it improves the ability to correctly classify potentially adversarial examples during testing [7]. However, many of these defenses rely on computationally expensive brute-force solutions to generate potent adversarial examples [8], which limits their practicality. Moreover, recent studies have shown that adversarial training based models are vulnerable to adversarial examples that are generated by randomizing or transferring perturbations from other models [9]. This raises concerns about the generalization ability of adversarial training based models to attacks from adversarial examples that are not generated by gradient-based techniques. For instance, adversarially train a neural network to be robust against gradient-based adversarial examples may still be vulnerable to adversarial examples created by adding Gaussian noise to the feature dimensions of the original examples [10]. Therefore, it is necessary to develop defense mechanisms that are both effective and generalizable against a wide range of adversarial attacks.

In addition to robustness, the interpretability of a network's predictions is also a concern for domain experts [11]. This is particularly important in domains with safety requirements, where it is necessary to understand how

a model is trained and used. For example, medical specialists need to know how the model responds to training data from different hospitals. To enhance interpretability, some methods have been proposed that regularize the input gradients [12, 13], which can highlight the regions of confidence predicted by the DNNs. However, these previous endeavors have simply smoothed out all the gradients without capturing the most appropriate interpretability, i.e., their response to small input variations. Other techniques, such as integrated gradients [14] and SmoothGrad [15], generate smoother and more interpretable prediction confidence, but do not provide insight into the response of a deep neural network to input variations. In fact, local behavior can simulate how the network responds to small perturbations in its inputs.

In this paper, we address the issue of improving both robustness and prediction interpretability of DNNs when dealing with small input perturbations. Our approach involves simultaneously minimizing the Jacobian norm of the entire network and regularizing the input gradients. To this end, we introduce the concept of **linearized robustness**, which provides a measure of the network's response to a perturbed input. We demonstrate that the Jacobian norm can serve as a reliable approximation of the linearized robustness, as it captures the gradients of the network prediction logits with respect to the input. However, using raw input gradients can be noisy and difficult to interpret. Inspired by the approach presented in [12], which uses gradient suppression and selected features to explain model robustness, we propose to regularize the input gradients to achieve a better input space. Our method distinguishes itself from [12] by training the network using the Jacobian's Frobenius norm (while [12] uses the reproducing kernel Hilbert space norm).

This distinction allows us to establish boundaries on the response of network layers to input perturbations, contributing to improved performance.

To interpret the network's response to input perturbations, we propose to use this saliency map to visualize and highlight the important regions of an input image that are responsible for a network prediction. The map is computed by taking the absolute values of the gradients of the output logits with respect to the input image, and then multiplying them element-wise with the input image itself. This results in a saliency map that shows the most important regions of the input image for making the given prediction. By analyzing the saliency map for a given perturbation, we can understand which regions of the input image are most sensitive to small perturbations and how the network reacts to them. This helps to improve the interpretability of the network's predictions and provides insights into its internal workings. Furthermore, we show that the saliency map can also be used to assess the robustness of a model by analyzing its response to different types of perturbations. In this way, we can obtain a more complete understanding of the network's behavior and improve its performance in real-world scenarios.

The contributions of this paper can be summarized as follows: 1) We propose a novel approach to achieve both improved robustness and high interpretability of DNNs under adversarial attacks. The proposed approach effectively leverages the Jacobian norm and selective input gradient regularization, which explicitly describes the network responses to input perturbations. 2) We investigate the relationship between a Jacobian norm and

the linearized robustness [1] of neural networks. Based on perturbation-based saliency maps, our results give insights into the prediction confidence, which relates to the adversarial effects. 3) Our method improves the robustness of DNNs towards transferred adversarial examples across multiple architectures and different attacks. The rest of this paper is organized as follows. Section 2 reviews the recent related works. Section 3 and Section 4 detail the proposed method. Section 5 presents extensive experiments to evaluate our method. Finally, we conclude the paper in Section 6.

2. Related work

2.1. Adversarial training based methods

There is a sizable body of work proposing various attack and defense mechanisms for the adversarial setting. Among them, the current unbroken defenses [2, 16] are based on adversarial training, which uses adversarial examples as training data to protect DNNs against a range of adversarial attacks. For example, projected gradient descent (PGD) [2] is one such strong defense that is able to generate universal adversarial examples using a first-order approach. And [16] encourages the decision boundary to be smooth by adding a regularization term to reduce the difference between the predictions from natural and adversarial examples. Qin et al. [17] smoothened the loss landscape through local linearization by minimizing the prediction difference between the clean and adversarial examples. While the various aforementioned approaches can improve the adversarial training, they require the generation of sufficient adversarial examples for training. This results in a prohibitive computational cost, which is proportional to the number of

[1] (The distance from the input image to the decision boundary)

steps needed to generate the adversarial examples. In addition, it requires a back-propagation for each iteration. To strengthen DNNs under adversarial attacks, a biologically-inspired approach [18] was introduced to learn flat, compressed representations that are sensitive to a minimal number of input dimensions. Unlike [18], this paper introduces a simpler yet effective approach for model regularization that is based on input gradient regularization. A concurrent method is [19], which can improve robustness by imposing the input gradient regularization. However, performing such gradient with respect to a high dimensional input from back-propagation is quite time-consuming. In contrast, the proposed method approximates the linearized robustness of neural networks via the penalization of a classifier's Jacobian norm. Such a Jacobian norm derives salient gradient maps to selectively activate the most discriminative gradients.

2.2. Regularization for robustness

To defend against adversarial examples, provable defenses promote the concept of improving model robustness through regularization. A well-known strategy includes noise injection, which is a variant of dropout weights [20] or activations [21]. Several works have investigated the benefits of using a regularization term on top of the standard training objective to reduce the Jacobian's Frobenius norm. Such a term aims to reduce the adversarial effect on the model predictions caused by input perturbation. For instance, Hoffman et al. [22] proposed an efficient method to approximate the input-class probability through the output Jacobians of a classifier so as to minimize the computational cost associated to these Jacobian norms. Tsipras et al. [23] observed that adversarially trained models produce salient Jacobian matrices

that loosely resemble the input images whilst less robust models have noisier Jacobians. Etmann et al. [24] interpreted linearized robustness as the alignment between the Jacobian and the input image, and trained a robust model by using double back-propagation. In comparison to these methods [23, 24], our work offers the following merits: 1) it focuses on the local linearized robustness of neural networks to provide an interpretation of the network's response to the inputs via Jacobian norm; 2) the proposed selective input gradient regularization explicitly measures the degree of robustness, which improves the interpretability of the network prediction; and 3) our method is computationally more efficient in calculating the input gradients and also highly interpretable to understand the network's prediction.

3. Preliminary

3.1. Definition

Let a classification model $F_\theta(\mathbf{X}) : \mathbf{X} \mapsto \hat{\mathbf{Y}} \in \mathbb{R}^{N\times K}$ map the inputs $\mathbf{X} \in \mathbb{R}^{N\times D}$ to the output probabilities for K classes, where θ represents the classifier's parameters and $\hat{\mathbf{Y}} \in \mathbb{R}^{N\times K}$ returns the predictions of F_θ. To train the model F_θ, we aim to find a set of parameter θ^* that minimizes the total distance between the predictions $\hat{\mathbf{Y}}$ and the one-hot encoded true labels $\mathbf{Y} \in \mathbb{R}^{N\times K}$ on a training set: $\theta^* = \arg\min_\theta \sum_{n=1}^{N}\sum_{k=1}^{K} -\mathbf{Y}_{n,k} \log F_\theta(\mathbf{X}_{n,k})$, which can also be written as $\arg\min_\theta H(\mathbf{Y}, \hat{\mathbf{Y}})$, where H is the sum of the cross entropies between the predictions and the true labels.

Given an input $\mathbf{x} \in \mathbb{R}^{h\times w\times c}$ to a DNN, one can define the Jacobian matrix J with respect to $\mathbf{x}$ as

$$J(\mathbf{x}) := \nabla_{\mathbf{x}} F_\theta(\mathbf{x}) = \left[\frac{\partial F_\theta(\mathbf{x})}{\partial \mathbf{x}_1}, \ldots, \frac{\partial F_\theta(\mathbf{x})}{\partial \mathbf{x}_D}\right], \tag{1}$$

where $D = h \times w \times c$ is the dimensionality of $\mathbf{x}$. While a DNN can be trained empirically to perform well on the training data, the accuracy degrades sharply in the presence of adversarial examples. When a small perturbation $\mathbf{z}$ is applied to the input $\mathbf{x}$, a model is still deemed robust against this attack if it satisfies

$$\arg\max_{k \in K} F_\theta^k(\mathbf{x}) = \arg\max_{k \in K} F_\theta^k(\mathbf{x} + \epsilon\mathbf{z}), \forall \epsilon \in B_p(\varepsilon) = \epsilon : ||\epsilon||_p \leq \varepsilon, \tag{2}$$

where ε is the scaling factor, and $p = \infty$. To improve the model's robustness, adversarial training [1] tries to find a distribution match between the training data and the adversarial test data. Specifically, adversarial training attempts to minimize the loss function as:

$$\min_\theta \rho(\theta), \text{where} \quad \rho(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{y})} \left[\max_{\epsilon \in B(\varepsilon)} H(F_\theta(\mathbf{x} + \epsilon\mathbf{z}), \mathbf{y}) \right], \tag{3}$$

where the inner maximization terms are usually obtained by performing an iterative gradient-based optimization, such as PGD [2].

3.2. Attacks

Fast Gradient Sign Method (FGSM) [1]. FGSM generates adversarial examples by perturbing the inputs that increase the local linear approximation of the loss function: $\mathbf{x}_{FGSM} = \mathbf{x} + \epsilon \cdot \text{sign}\nabla_{\mathbf{x}} H(\mathbf{y}, \hat{\mathbf{y}})$. To perform this attack, one can iteratively use a small ϵ (e.g., $\epsilon = 0.01$) to induce misclassifications by following the non-linear loss function in a series of small linear steps.

Projected Gradient Descent (PGD) [2]. PGD generates the adversarial examples by first uniforming the random perturbation as the initialization, and then iteratively performs the form $\mathbf{x}_{PGD}^{t+1} = \prod_{\mathbf{x}+S}[\mathbf{x}_{PGD}^t + \epsilon \cdot \text{sign}\nabla_{\mathbf{x}} H(\mathbf{y}, \hat{\mathbf{y}})]$, where $\prod$ is the projection operator that clips the input at positions around

the predefined perturbation range. $\mathbf{x} + S$ represents the perturbation set, and ϵ is the gradient step.

Jacobian-based Saliency Map Attack (JSMA) [25]. JSMA iteratively searches for input pixels to be changed such that the probability of the target label is increased and the probability of all other labels are decreased. It can produce examples that have only been modified in a fraction of feature dimensions, which are hard for humans to detect.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [4]. MI-FGSM integrates the momentum term into iterative attack to stabilize the update directions. Thus, it can improve the transferability of adversarial examples.

AdaBelief Iterative Fast Gradient Method (ABI-FGM) [5]. This method improves the transferability of adversarial examples by applying the Adabelief optimizer to generate adversarial examples yet similar to training examples.

Nesterov Iterative Fast Gradient Sign Method (NI-FGSM) [6]. This method adapts Nesterov accelerated gradient into the iterative attacks so as to effectively look ahead and improve the transferability of adversarial examples.

3.3. Defenses

Adversarial Training [7]. It can enhance robustness by injecting adversarial examples into training process. Following the implementation in [26], we augment the network to run FGSM [1] on the training batches, and compute the average loss on both the normal and adversarial examples as the loss function of the model. To inhibit the FGSM attack [1], gradients are not

allowed to propagate and FGSM perturbations are computed with respect to the predicted labels (instead of the true labels) to prevent label leaking.

Defensive distillation [7]. Distillation can be used as a defense technique by first using the one-hot ground truth labels to train an initial model and subsequently utilizing the initial model's softmax probability outputs. Since distillation extracts class knowledge from these probability vectors, this knowledge can be transferred into a different DNN architecture by annotating the inputs in the training dataset of the second DNN using classification predictions from the first DNN. This idea is formulated to improve the resilience of DNNs in the presence of perturbations [27]. Within a softmax layer, we divide all of the logit network output (which we call $\hat{z}_k$) by a temperature T: $F_{T,\theta}(\mathbf{X}_{n,k}) = \frac{e^{\hat{z}_k(\mathbf{X}_{n,k})/T}}{\sum_{i=1}^{K} e^{\hat{z}_i(\mathbf{X}_{n,k})/T}}$, where $F_{T,\theta}$ denotes a network output in the form of a softmax vector with temperature T. The predictions will converge to $1/K$ as $T \to \infty$. The distillation based defense can be formulated as

$$\theta^0 = \arg\min_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} -\mathbf{Y}_{n,k} \log F_{T,\theta}(\mathbf{X}_{n,k}), \theta^* = \arg\min_{\theta} \sum_{n=1}^{N} \sum_{k=1}^{K} -F_{T,\theta^0}(\mathbf{X}_{n,k}) \log F_{T,\theta}(\mathbf{X}_{n,k}). \quad (4)$$

4. Proposed approach

4.1. Jacobian norm

In the following, we study the relationship between the Jacobian norm based regularization term and the notion of linearized robustness. Since adversarial perturbations are small variations that change the predicted result of a neural network classifier, it is sensible to define the robustness towards adversarial perturbations via the distance of the clean image to the nearest perturbed image which may cause the incorrect classification. When such

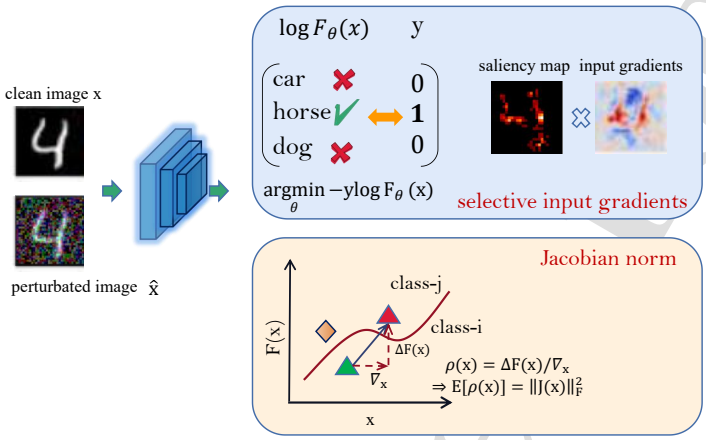

Figure 1: The proposed scheme for adversarial robustness based on Jacobian normalization and selective input gradient regularization (J-SIGR). The Jacobian normalization sets the linear robustness bounds for perturbations. The selective input gradient regularization is based on perturbation-based saliency map to not only encourage the insensitivity of the input space but also improves the interpretability.

distance gets smaller, the perturbed and its clean counterpart are more indistinguishable for a neural network, and thus the prediction of the neural network will be correct.

Linearized adversarial robustness bound: Let $i^* = \arg\max_i F_\theta^i(\mathbf{x})$ and $j^* = \arg\max_{j \neq i^*} F_\theta^j(\mathbf{x} + \epsilon\mathbf{z})$ be the top prediction of $\mathbf{x}$ and its corrupted sample $\hat{\mathbf{x}} = \mathbf{x} + \epsilon\mathbf{z}$, respectively. Here, $\hat{\mathbf{x}}$ is formed by small additive perturbations with Gaussian distribution $\mathbf{z} \sim \mathcal{N}(0, \sigma^2)$. The linearized adversarial robustness is upper-bounded by the Jacobian norm $||J(\mathbf{x})||_F^2$ w.r.t $\mathbf{x}$.

Proof. Denoting $F_\theta^i(\mathbf{x})$ as the logits value of class i in a classifier F [2] for

[2]For notation simplicity, we omit θ in the following.

$\mathbf{x}$ with $\epsilon \ll 1$, then its linearized robustness can be expressed as

$$\rho(\mathbf{x}) := \min_{j^* \neq i^*} \frac{F^{i*}(\mathbf{x}) - F^{j*}(\mathbf{x})}{||\nabla_{\mathbf{x}} F^{i*}(\mathbf{x}) - \nabla_{\mathbf{x}} F^{j*}(\mathbf{x})||}. \tag{5}$$

Denoting $g := \nabla_{\mathbf{x}}(F^{i*} - F^{j*})(\mathbf{x})$ as the Jacobian w.r.t the difference of two logits and $\alpha(\mathbf{x}, g) =< \mathbf{x}, g >$ as the alignment between the Jacobian and the input, then we have the decision boundary $\rho(\mathbf{x}) \leq \frac{\alpha(\mathbf{x},g)+C}{||g||}$, where C is a positive constant. Therefore, $\rho(\mathbf{x}) \leq \frac{J(\mathbf{x})+g+C}{||g||}$, where $J(\mathbf{x})$ is the Jacobian of the network output w.r.t the input $\mathbf{x}$. We can now treat the term $\hat{\rho} = \mathbf{z}^T J(\mathbf{x})^T J(\mathbf{x})\mathbf{z}$ as one sample stochastic trace estimator for $Tr(J(\mathbf{x})^T J(\mathbf{x}))$ with a Gaussian variable $\mathbf{z}$:

$$\mathbb{E}_{\mathbf{z}}[\rho] = \frac{Tr(J(\mathbf{x})^T J(\mathbf{x})\mathbb{E}[\mathbf{z}\mathbf{z}^T])}{||g||} = \frac{||J(\mathbf{x})||_F^2}{||g||}. \tag{6}$$

Taking expectation over m samples of a mini-batch X, we have $\mathbb{E}[\rho] = \mathbb{E}_{\mathbf{x}}[||J_X||]_F^2$, where $||\cdot||_F^2$ represents the Frobenius norm. We remark that the above assumption holds true given that the neural network can be locally approximated by a linear model [24]. Since adversarial perturbations refer to small perturbations that can cause a neural network to predict a different class, it is sensible to define the robustness towards adversarial perturbations via the distance of the unperturbed image to its nearest perturbed image, such that the classification is changed. In other words, the robustness of a classifier at a given point can be defined as the distance to its nearest decision boundary. In general, it is intractable to compute the distance to the decision boundary $\rho(x)$. Nonetheless, for classifiers built on locally affine score functions, as in the case of most neural networks using ReLU or leaky ReLU activations, the decision boundary can be computed, provided that the locally affine region around the point $\mathbf{x}$ is sufficiently large. As proved in

Algorithm 1 The proposed J-SIGR.

Initialize θ by using a pre-trained network architecture;
Set $\lambda_j = \lambda_m = 0.5; \epsilon = 0.3$.
while each iteration or a condition is met **do**
    Sample $(\mathbf{x}, \mathbf{y}) \sim D_{train}$; /*Input to the DNN*/
    Generate the noise perturbation $\mathbf{z} \sim \mathcal{N}(0, \sigma^2)$;
    $\hat{\mathbf{x}} = \mathbf{x} + \epsilon\mathbf{z}$; /*Generate a perturbed sample*/
    $\nabla_{\mathbf{x}} F_\theta(\hat{\mathbf{x}})$; /*Compute the perturbation-based saliency map*/
    Compute the Jacobian norm $||J(\mathbf{x})||_F^2$;
    Train the network using Eq. (10) and update θ.
end while
return θ.

[24], for a classifier defined with a locally affine score function, the decision boundary between the clean and the perturbed data is close in the Euclidean space when their respective input signals are also close enough in the Euclidean space. Thus, the linearized robustness holds approximately as long as the linear approximation to the network's score function is sufficiently plausible in the relevant neighborhood of $\mathbf{x}$.

Improved robustness using the Jacobian norm: In the presence of perturbed examples, the expected response of a DNN should stay similar to the correct prediction, which can be mathematically described as $\Omega = F_\theta(\mathbf{x}) - F_\theta(\hat{\mathbf{x}})$. Suppose $\mathbf{x}_i$ is the i-th component of noise-free signal $\mathbf{x}$, and $\hat{\mathbf{x}}_i = \mathbf{x}_i + \epsilon\mathbf{z}_i$ is the noise-crafted tensor variation of $\mathbf{x}$. Note that the term Ω measures the difference of the predictions in the case of clean data and its perturbed counterpart. $\mathbf{z}$ is the noise term, which is sampled from a Gaussian distribution with zero mean and variance σ^2 for each inference. Note that the noise term has an identical variance to $\mathbf{x}$ so that the additive noise only relies on the distribution of $\mathbf{x}$ to dynamically perturb the input. According to the

above, the linearized robustness can be approximated and upper-bounded by the Jacobian of the classifier's prediction w.r.t the input $\mathbf{x}$. Thus, with the gradient back-propagation, we can determine the gradient calculation w.r.t the difference of the two predictions Ω via:

$$\nabla_{\mathbf{x}}\Omega = \nabla_{\mathbf{x}}(F_\theta(\mathbf{x}) - F_\theta(\hat{\mathbf{x}})) = \nabla_{\mathbf{x}}F_\theta(\mathbf{x}) - \nabla_{\mathbf{x}}F_\theta(\hat{\mathbf{x}}) \Rightarrow ||\nabla_{\mathbf{x}}F_\theta(\mathbf{x}) - \nabla_{\mathbf{x}}F_\theta(\hat{\mathbf{x}})||_F^2 \leq ||J(\mathbf{x})||_F^2. \quad (7)$$

The above Jacobian norm can approximate the linear robustness towards the input, which simulates how the network will respond to those small variations of the input. To compute the Jacobian norm, one needs to take the model's gradient with respect to its inputs, which provides a local linear approximation of the model's behavior. However, directly using the raw input gradients is known to be ineffective since these gradients are quite noisy and hard to interpret. To combat this challenge, in the following we propose to train the classification model under an input gradient regularization with fewer extreme values and a minimized Jacobian norm.

4.2. Selective input gradient regularization

Input gradient regularization. The idea of input gradient regularization was first introduced by Drucker et al [28] to train neural networks by minimizing not just the "energy" of the network but also the rate of the change of the energy with respect to the input features. The energy can be formulated using the cross-entropy as follows:

$$\theta^* = \arg\min_\theta \sum_{n=1}^{N}\sum_{k=1}^{K} -\mathbf{Y}_{n,k}\log F_\theta(\mathbf{X}_{n,k}) + \lambda_m \sum_{n=1}^{N}\sum_{d=1}^{D}\left(\frac{\partial}{\partial \mathbf{x}_d}\sum_{k=1}^{K} -\mathbf{Y}_{n,k}\log F_\theta(\mathbf{X}_{n,k})\right)^2, \quad (8)$$

where λ_m is a hyperparameter modulating the penalty strength. Such input gradient regularization ensures that even if the input change slightly, the KL divergence between the predictions and the true labels will not be changed

significantly. This double back-propagation can provide a constraint on the sensitivity caused by perturbations. Intuitively, the gradient penalty term encourages the predictions not be sensitive to small perturbations in the input space because it regularizes the input gradients to be smoother with fewer extreme values. We combine the Jacobian norm and the input gradient regularization, which can be formulated as follows:

$$\theta^* = \arg\min_{\theta} \sum_{n=1}^{N}\sum_{k=1}^{K} -\mathbf{Y}_{n,k} \log F_\theta(\mathbf{X}_{n,k}) + \lambda_m \sum_{n=1}^{N}\sum_{d=1}^{D} \left(\frac{\partial}{\partial \mathbf{x}_d} \sum_{k=1}^{K} -\mathbf{Y}_{n,k} \log F_\theta(\mathbf{X}_{n,k}) \right)^2 + \lambda_j ||J(\mathbf{x})||_F^2, \tag{9}$$

where λ_m and λ_j denote the weights for the selective gradient regularization and the Jacobian normalization, respectively. As formulated in Eq. (9), the joint optimisation over gradient regularisation and Jacobian normalisation plays a crucial role in maintaining the local prediction capabilities of the deep classifier. When confronted with transferable adversarial examples that are specifically crafted to evade local maxima by accumulating stable gradient directions, our proposed method effectively can regularise the smoother gradients, thereby stabilizing the network's predictions. However, in Eq. (9), the combination of Jacobian norm and input gradient regularization only provides constraints for very near training examples. Thus, it does not solve the adversarial perturbation problem. It is also expensive to make derivatives smaller to limit the sensitivity to infinitesimal perturbations. With this regard, in the following, we propose a *perturbation based saliency map* to select the most discriminative features, which are not only robust to perturbations but more interpretable to the network behaviour.

Perturbation based saliency for selective input gradient regularization. Saliency map in deep learning is a technique used to interpret input features that are

determined to be important for the neural network output [29, 15]. As domain experts are more concerned with the interpretability of a DNN, some methods have been proposed to generate saliency maps to explain the decision making of DNN. One may directly use gradients to estimate the influence of input features on the output. However, the quality of the gradient-based saliency maps is generally poor as gradient-based saliency map methods tend to overly smooth the gradients [19].

In the spirit of saliency map in highlighting the importance of input features, we propose to use the perturbation based saliency map, denoted as $\mathcal{M}_d = f(\nabla_{\mathbf{x}} F_\theta(\hat{\mathbf{x}}))$, which is derived from the gradient of a perturbed input. The $f(\cdot)$ is a mapping function to be detailed later. Such a perturbation-based saliency map can be computed by perturbing the input and observing the change in output, and thus shows high interpretability in a DNN's behaviour. Specifically, we compute this saliency map by resembling the input images and highlighting the most salient parts. Following [30], we use a Generative Adversarial Network (GAN) to generate a saliency map to be visually similar to the real image. Given $J(\hat{\mathbf{x}}) = \nabla_{\mathbf{x}} F_\theta(\hat{\mathbf{x}})$, we use an aligning network (f) to map the Jacobian into the domain of the input image: $J' = f(J(\hat{\mathbf{x}}))$. In our implementation, f (parameterized by Φ) is implemented by a single 1×1 convolutional layer with a `tanh` activation function. Hence, the generator $G(\mathbf{x}, \mathbf{y})$ can be framed by using both the classifier and the aligning network: $G_{\theta,\Phi}(\mathbf{x}, \mathbf{y}) = f(\nabla_{\mathbf{x}} F_\theta(\hat{\mathbf{x}}))$. As a result, the generator can learn to model the distribution of $p_{J'}$ to resemble that of $p_{\mathbf{x}}$.

Once $\mathcal{M}_d$ is obtained, we incorporate the salient map into input gradient computation such that the most robust gradients can be selected during

back-propagation. Mathematically, the objective with Jacobian norm and selective input gradient regularization is defined as

$$\begin{aligned}\theta^* = \arg\min_{\theta} \sum_{n=1}^{N}\sum_{k=1}^{K} -\mathbf{Y}_{n,k}\log F_\theta(\mathbf{X}_{n,k}) \\ + \lambda_m \sum_{n=1}^{N}\sum_{d=1}^{D}\left(\frac{\partial}{\partial \mathbf{x}_d}\mathbb{1}(\beta - \mathcal{M}_d)\sum_{k=1}^{K} -\mathbf{Y}_{n,k}\log F_\theta(\mathbf{X}_{n,k})\right)^2 + \lambda_j ||J(\mathbf{x})||_F^2.\end{aligned} \tag{10}$$

We suggest two terms have equal importance to the overall optimization, and thus set $\lambda_m = \lambda_j = 0.5$ as default configuration except for otherwise specified. The term $||J(\mathbf{x})||_F^2$, i.e., the Jacobian Frobenius norm acts as a regularizer clipping the values of inputs such that the gradients of classification logits with respect to the inputs can ensure the linearized robustness in the presence of perturbed data, and thus we can minimize the number of mispredictions of the learned classifier. However, this is ineffective by directly evaluating each raw input features, which could be very noisy. Then, the input gradient regularization, as being modulated by λ_m, can smooth the gradients to be less noisy with fewer extreme values. Finally, in Eq.(10) we embed the perturbation-based saliency map ($\mathcal{M}_d$) to improve the interpretability of a neural network's prediction. The indicator function $\mathbb{1}$ is to determine whether the saliency for an input feature is below a threshold β, and thus it returns 1 if $\beta - \mathcal{M}_d \geq 0$ and 0 otherwise. The whole training procedure of the proposed method is illustrated in Algorithm 1.

5. Experiments

To evaluate the robustness of the proposed J-SIGR, we conducted experiments on three image datasets: MNIST [31], CIFAR-10 [32] and ImageNet [33]. Below, we first describe the experimental settings and then report the

experimental results under a range of attacks. Finally, we performed ablation studies to provide a more insightful analysis of the proposed method.

5.1. Experimental settings

5.1.1. Datasets

MNIST dataset [31] consists of handwritten digit 28×28 gray-scale images divided into 60K training and 10K test images. We trained a CNN, composed of 3 convolutional layers and one final soft-max layer, to suit the small-sized MNIST. All convolutional blocks have a stride of 5 while each layer has an increasing number of output channels (i.e., c= 64-128-256). CIFAR-10 [32] dataset contains a collection of $32 \times 32 \times 3$ colored images that are categorized into 10 classes with 50K training and 10K test images. We use the ResNet-20 architecture [34] with 20 convolutional layers to train the images from CIFAR-10. Throughout the network, the kernel size is set to 9×9 in all convolutional layers and the number of channels is increased from 9, 18 to 36 for the three building blocks, respectively. For ImageNet, we randomly choose 1000 images belonging to the 1000 categories from ILSVRC 2012 validation set, which are almost correctly classified by all the testing models.

5.1.2. Implementations and evaluation metrics

We used ResNet-20 architecture [34] as the backbone for most of the comparative experiments and ablation studies. Thus, the parameter θ is instantiated by ResNet-20 configuration. We adopted the data augmentation [34, 10] but without the input normalization. Alternatively, we placed a non-trainable data normalization layer preceding the network to perform the identical function so that the attack tactics can directly add perturbations

into the natural images. We set $\lambda_j = \lambda_m = 0.5$ and the step size $\epsilon = 0.3$ in Eq. (10). To generate the saliency map as required, we use a discriminator network of 5 CNN layers (32-64-128-256-512 output channels) and update it for every 20 $\log F_\theta$ training iterations. Then, we computed the class-entropy loss and use the gradients to compute the Jacobian matrix. Since our method involves randomness, we reported the accuracy in the format of mean± std with 10 trails to compute the statistical values. To measure the robustness of multiple attacks, we tested all models against adversarial examples generated for each model and reported the accuracy. In contrast to the JSMA setting [25], where the generated adversarial examples would resemble the targets rather than their original labels, we opted to a human subject experiment presented in [19]. This experiment aims to evaluate the legitimacy of misclassifications caused by adversarial examples.

Attack implementations. We considered multiple powerful attacks: while-box attacks, JSMA [25] and white-box attacks transferred attacks. The white-box attacks include FGSM [1] and PGD [2]. FGSM [1] is an efficient single-step adversarial attack scheme and PGD attack [2] is a multi-step variant of FGSM [1]. The iterative update of crafted data $\hat{\mathbf{x}}$ in the $t+1$-th step can be expressed as $\hat{\mathbf{x}}^{t+1} = \prod_{\mathbf{x}+S}(\hat{\mathbf{x}}^t + \epsilon \cdot \text{sgn}(\nabla_{\mathbf{x}}(F_\theta(\hat{\mathbf{x}}^t), \mathbf{y})))$, where $\prod$ is the projection space bounded by $\mathbf{x} \pm S$, and ϵ is the step size. For the PGD attack [2] on three datasets, S is set to 0.3/1, 8/255, 16/255, and the number of iterations N_{step} is set to 40, 7 and 10, respectively. FGSM [1] adopts the same ϵ setup as PGD [2]. To generate adversarial examples for JSMA [25], we used the Cleverhans adversarial example library [26]. For the black-box transferred attacks, i.e., MI-FGSM [4], ABI-FGM [5], NI-FGSM [6], we follow

their own configuration of each work.

Defense implementations. To evaluate the improved robustness of our method, we compared it with state-of-the-art defense models: adversarial training [7], distillation [27] and a gradient regularization based model [19]. More specifically, for adversarial training, we trained FGSM [1] with perturbations at $\epsilon = 0.3$. For distillation based defense, we used a soft-max of temperature $T = 50$. For the gradient regularization based model as shown in Eq. (8), we used the double back-propagation to train the classification model.

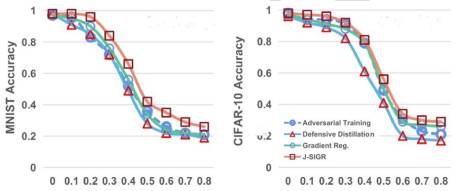

Figure 2: Defensive model accuracy against PGD attack when applying gradient regularization as the fool target.

5.2. Robustness under white-box attacks

In Fig. 3, we show the robustness results of our method as well as the performance of other defensive models under the FGSM attack [1] on two datasets. It can be observed that the gradient-regularized model [19] exhibits strong robustness to transferred FGSM [1] attack (examples produced by attacking other models). For example, on the MNIST dataset when FGSM attacking [1] is to fool the defensive distillation, i.e., first row and second column, the adversarial examples produced by attacking the defensive distillation can successfully fool the model based on adversarial training. In contrast, the gradient regularization based methods (including J-SIGR) can still maintain the accuracy. We evaluate the robustness of the gradient regularization models under a different attack, i.e., PGD [2], and report the results

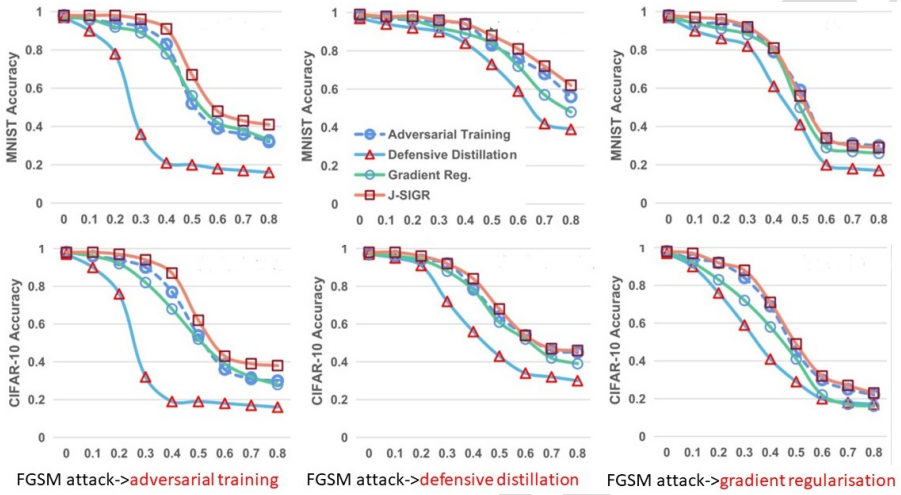

Figure 3: Comparison of different methods' accuracy on two datasets. For each column, we used FGSM attacking to generate adversarial examples to fool different defenses: adversarial training, defensive distillation, and gradient regularization.

in Fig. 2 on two datasets. Under this attack, the adversarial examples are generated to fool a gradient regularized model, while the results of the two models show that gradient regularization is still effectively robust against a white-box attack [2]. Interestingly, gradient-regularized models seem to be vulnerable to white-box attacks, but can still fool all other models. In this respect, in the presence of adversarially transferred examples, we hypothesize that gradient regularization is particular not only for defense but also attack.

Since our model consists of two robustness mechanisms, we investigated the impact of Jacobian norm (JN) by disabling double back-propagation and examining the output response of each layer with respect to two different attacks. More specifically, a Jacobian-norm based variant of our method was implemented by adding layer-wise Jacobian norm into the DNN, together

Table 1: Convergence of gradient regularization with layer-wise Jacobian norm (LW-JN) on the CIFAR-10 dataset with ResNet-20 as backbone. The Jacobian norm is given for lower convolutional layers (Conv1 to Conv3_5). Test accuracy for perturbed data is computed for the PGD and FGSM attacks.

Layer	Vanilla Train	JN+AT	Grad.Reg.+LW-JN
Conv1	0.004	0.157	0.155
Conv2_0	0.003	0.089	0.091
Conv2_1	0.001	0.067	0.061
Conv2_2	0.002	0.055	0.057
Conv2_3	0.002	0.099	0.098
Conv2_4	0.004	0.782	0.580
Conv2_5	0.004	0.422	0.333
Conv3_0	0.002	0.087	0.079
Conv3_1	0.000	0.064	0.064
Conv3_2	0.003	0.072	0.069
Conv3_3	0.001	0.062	0.060
Conv3_4	0.001	0.046	0.046
Conv3_5	0.000	0.022	0.021
FC	0.001	0.013	0.012
PGD	0.00	0.54 ± 0.11	0.57±0.10
FGSM	0.14	0.62 ± 0.10	0.66±0.09

with the input gradient regularization. This variant is called *Grad.Reg.+LW-JN*. As shown in Table 1, only performing vanilla training using momentum SGD optimizer can lead to the failure of adversarial defense with the values of Jacobian norm converging towards negligible values. After applying the JN as a regularization of the network (i.e., JN+AT), the lower convolutional layers attain a relatively large JN. The variant of our method with layer-wise Jacobian norm (i.e., *Grad.Reg.+LW-JN*) achieves the highest performance with respect to the two attacks. This demonstrates robustness improvement by leveraging the proposed network architecture, which is parameterized to resist perturbations via gradient back-propagation. Since JN can reflect the

robustness of the network, we plotted the evolution curves of the JN values for the lower convolutional layers to illustrate the robustness of the network connections (Fig. 4).

5.3. Robustness under black-box transferable attacks

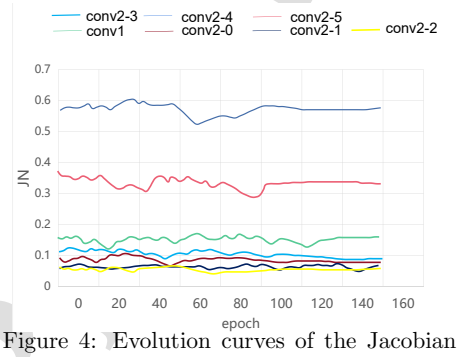

Figure 4: Evolution curves of the Jacobian norm computed at each convolutional layer.

In this experiment, we evaluate our scheme against the following black-box transferable attacks: MI-FGSM [4], ABI-FGM [5], NI-FGM [6]. More specifically, two neural networks were trained with their individual architectures with one network chosen as the source model and the other chosen as the target model. An adversarial example $\hat{\mathbf{x}}$ generated from the source model was then used to attack the target model without access to the parameters of the target model. This is denoted as Source→Target. We trained two ResNet-18 models (i.e., M_1 and M_2) on ImageNet dataset to attack each other with M_1 optimized through PGD adversarial training [2] and M_2 optimized through our proposed method. Experimental results are given in Table 2. Our proposed method achieves higher accuracy under two transferable attacks and is seen similar perturbed-data accuracy for both transferable scenarios. This indicates that our method provides robustness against transferable black-box attack. This also shows that the presence of J-SIGR has negligible effect on inference under a strong PGD attack [2]. For the powerful black-box transferable attacks, we evaluated our defense ability on 200 randomly selected test examples for an untargeted at-

Table 2: The proposed method against transferred attacks on ImageNet test sets. Model M_1 and M_2 are trained by PGD adversarial training and J-SIGR based on ResNet-18.

Transferable attack		MI-FGSM [4]	ABI-FGM [5]	NI-FGSM [6]
$M_1 \rightarrow M_2$	$M2 \rightarrow M_1$	Success rate =90%	Success rate=77%	Success rate= 80%
78.14 ± 0.26	76.82 ± 0.19	49.00	48.80	41.20

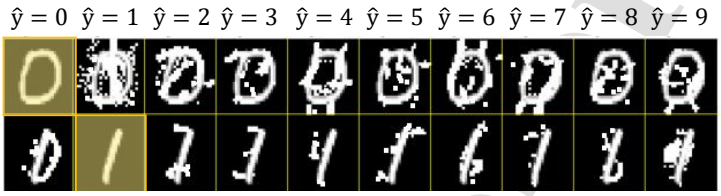

Figure 5: Perturbations by applying JSMA to digits 0 and 1 with maximum distortion parameter $\gamma = 0.25$ for a gradient regularization model. The highlighted images in each row are modified until the model predicts the digit corresponding to their column or the maximum distortion is reached.

tack. The success rate refers here to the percentage of test samples which are wrongly classified under the attack. For example, the MI-FGSM [4] attack success rate for vanilla ResNet-18 with adversarial training is close to 90%. The results in Table 2 suggest that our method is robust as it resists the three attacks by noticeably dropping the success rate from 90% to 49%, 77% to 48.8% and 80% to 41% under MI-FGSM [4], ABI-FGM [5] and NI-FGSM [6], respectively (Lower success rates means higher robustness).

5.4. Evaluation on human subject study under JSMA

Unlike other attacks that stop generating adversarial examples when the maximum distortion is met, JSMA [25] constantly generates adversarial examples until the model predicts the target. Thus, evaluating the robustness under JSMA [25] using accuracy numbers is inappropriate. This is also because the perturbations created by JSMA [25] alter the adversar-

ial examples so they resemble the target labels instead of the original labels. As shown in Fig. 5, for a gradient regularized model, we applied JSMA [25] on each image 0 or 1 to generate perturbations until the model predicts the digit corresponding to their column target label or the maximum distortion is reached (We set the maximum distortion $r = 0.25$). Then, we tested these different robustness scenarios using 11 human subjects who were invited to evaluate whether examples generated by different methods are more or less plausible instances of their targets. Specifically, the subjects were shown 30 images of JSMA-crafted examples, with each of these 30 images corresponding to one original digit (from 0 to 9) and one model (defensive distillation, gradient regularization and J-SIGR). Images were randomly and uniformly sampled from a larger set of 45 examples corresponding to the first 5 images of the original digit in the test set. Images in the test set were transformed by using JSMA [25] to resemble each of the other 9 digits. Subjects were not provided the original labels and were asked to identify the most two plausible predictions for the image they believed a classifier would produce (they entered N/A if they found no label was a good choice).

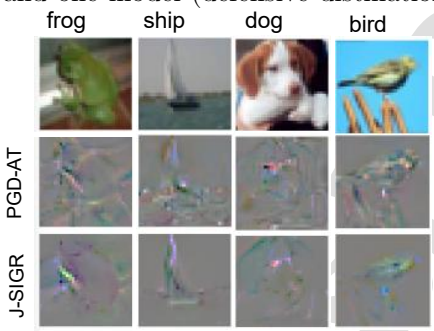

Figure 6: Visualization of Jacobian matrice of PGD-AT and J-SIGR on CIFAR-10.

Table 4 shows the quantitative results from the human subject experiment. The measure "human fooled" records the percentage of examples which are classified by human subjects as the most plausibly adversarial targets (the

higher, the better). "Mistake reasonable" measures the percentage of examples which are classified as either the target plausible or unrecognizable as any label (the higher, the better). Overall, human subjects found that gradient-regularized models can generate the most convincing examples to fool humans. More specifically, humans mostly believe gradient-regularized adversarial examples (both the input gradient regularization [19] and our method) are favourably classified as their target labels instead of the original digits. For example, the values of "human fooled" column in Table 4 show that the mispredictions of gradient regularized models are very "reasonable" in comparison to adversarial training and defensive distillation.

Table 3: Comparison with SOTA defense methods using clean and perturbed data on CIFAR-10 under PGD attack.

Defense method	Clean	PGD
PGD-AT [2]	87.0 ± 0.1	46.1 ± 0.1
DP [35]	87.0	25.1
Adv-BNN [36]	79.7	45.4
JARN [30]	84.8	51.8
J-SIGR (Ours)	90.1 ± 0.2	57.6 ± 0.4

5.5. Comparison with other defensive models for both clean and crafted data

In this experiment, we compared our method with state-of-the-art methods in dealing with both clean and crafted data. Apart from PGD [2] and randomness-based methods [35, 36], we also compared against JARN [30], which utilizes the Jacobians to generate images resembling to the original images. The compared performance results are shown in Table 3. Note that some of previous defense methods often achieve improved accuracy on contaminated data at the expense of lowering clean data accuracy. In contrast, we introduced a notion of linearized robustness which performs well in both clean and perturbed data. As shown in Table 3, our method outperforms all methods for both clean and perturbed data accuracy under the white-box attack. For example, differ-

ential privacy (DP) [35], which introduces noises at various locations of the network so as to guarantee a certified defense, does not perform well against L_∞-norm based attacks, e.g., PGD [2] and FGSM [1]. Moreover, to pursue a higher level of certified defense, DP [35] dramatically reduces the clean data accuracy down to 25.1%. As a noise injection method, Adv-BNN [36] combines the adversarial training and noise injection into the inputs/weights of the network. However, this method manually sets the noise configurations, making it very ad-hoc, and thus not generalizable to different datasets. In contrast, our method regularizes the Jacobian norm and the input gradients, such that the network parameters can be dynamically trained to perform better adversarial defense.

5.6. Connection to network interpretability

Understanding Jacobian matrices. An adversarially trained model can gain robustness and also produce salient Jacobian matrices as byproduct. It has been shown that the saliency in Jacobians is a result of robustness [22]. Thus, it is interesting to use the Jacobian saliency to evaluate how robust a model is. In this study, we visualize the Jacobian matrices of the proposed model and an adversarial-trained PGD [2] to show how salient the Jacobian map is. As shown in Fig. 6, the proposed method can better visually resemble the corresponding images than PGD-AT. This demonstrates the improved robustness of the proposed method.

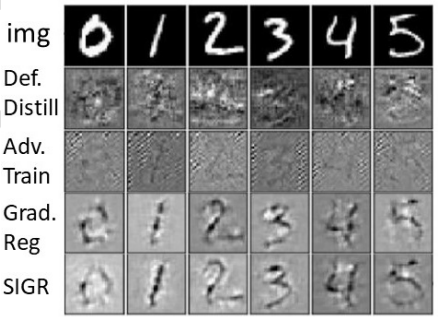

Figure 7: Visualization of input gradients.

Table 5: The proposed method (J-SIGR) with various architectures on CIFAR-10 test set.

	No defense			Vanilla Adv. Train			J-SIGR		
Model	Clean	PGD	FGSM	Clean	PGD	FGSM	Clean	PGD	FGSM
Net20	92.1	0.0±0.0	14.1	83.8	39.1±0.1	46.4	90.1±0.2	53.7±0.3	57.6±0.1
Net32	92.8	0.0±0.0	17.8	85.6	42.1±0.0	50.3	91.1±0.2	52.8±0.1	54.2±0.1
Net44	93.1	0.0±0.0	23.9	85.9	40.8±0.1	48.2	90.0±0.1	**55.4**±0.1	**58.6**±0.2
Net56	93.3	0.0±0.0	24.2	86.5	40.1±0.1	48.8	92.1±0.2	54.9±0.2	55.8±0.1
Net20 (1.5×)	93.5	0.0±0.0	15.9	85.8	42.1±0.0	49.6	91.4±0.1	55.2±0.3	55.8±0.1
Net20 (2×)	94.0	0.0±0.0	13.0	86.3	43.1±0.1	52.6	91.5±0.1	55.1±0.2	55.0 ±0.1

Table 4: Quantitative results form human subject experiment on MNIST. Best results are in bold.

	MNIST (JSMA)	
Model	human fooled	mistake reasonable
Def. Distill	0.0%	23.5%
Grad. Reg.	16.4%	41.8%
SIGR	**20.2**%	**45.1**%

Understanding input gradients. Fig. 7 visualizes the input gradients across different defensive models on the MNIST dataset. This qualitative visualization shows the different interpretability of the input gradients derived from models based on defensive distillation, adversarial training, gradient regularization and the proposed selective gradient regularization. The adversarially trained model can provide more interpretable gradients than defensive distillation, but not as highly interpretable as gradient regularized models. The proposed method presents the most interpretable gradients, and thus can provide an explanation for adversarial attacks.

5.7. Ablation studies

Our proposed technique introduces tight bounds to the response of the output layer to adversarial perturbation added to the input. Herein, we raise two concerns in regards to our proposed regulariza-

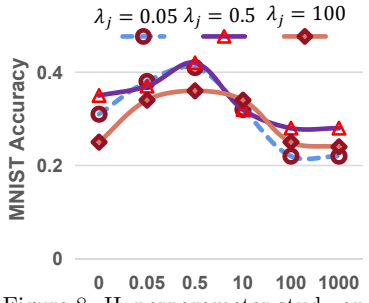

Figure 8: Hyperparameter study on MNIST.

tion term: 1) whether the robustness improvement introduced by our method is not relying on the stochastic gradient; 2) how the scale of the network architecture (i.e., width and depth) affects the network robustness. Our first evaluation aims to show that our method is free of gradient obfuscation by increasing the PGD [2] attack steps and the attack bound ε.

Influence of the network capacity. In order to investigate the links between the network capacity (i.e., width, depth and number of trainable parameters) and the robustness improvement of J-SIGR, we analyzed various network architectures in terms of depth and width. For varied depth, we considered ResNet20/32/44/56 and conducted experiments under vanilla training [2] and our technique. For varied width, we employed the original ResNet-20 as the baseline and expanded the input/output channel of each layer by 1.5× and 2× scale, respectively. We report both clean and perturbed data accuracy using the network trained with Jacobian term. The results in Table 5 suggest that increasing the model's capacity positively improves the network robustness against white-box attacks, and our proposed method outperforms vanilla training in both clean and perturbed data accuracy for powerful PGD and FGSM attacks. The other observation is that the noticeable robustness improvement provided by our method is indeed provided by the effective training with the proposed

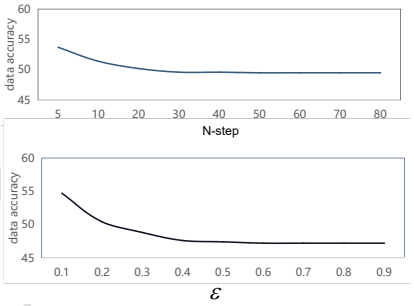

Figure 9: Accuracy of CIFAR-10 test set under PGD attack v.s. number of attack steps (N-step) and attack bound (ε).

J-SIGR. Our method updates the network parameters without introducing randomness in the test phase.

Effect of hyperparameters. In this experiment, we study the effect of two hyperparameters, i.e., λ_m and λ_j on the MNIST accuracy under FGSM. To de-correlate the impact of two parameters, we fix $\lambda_m = 0.5$ and examine the model accuracy with varied value of λ_j. The results are reported in Fig. 8, and it shows that the highest accuracy of DNN under an attack is achieved when we set $\lambda_j = 0.5$. Thus, we empirically $\lambda_m = \lambda_j = 0.5$ in all experiments.

Non-dependence on stochastic gradients. To prove the robustness improvement of our method is not due to stochastic gradients, we examine the perturbed data accuracy by evaluating the PGD attack steps [2], i.e., N-step, and the attack bound ε. As shown in Fig. 9, increasing the attack steps or attack bound can boost the attack strength, which inevitably leads to accuracy degradation. However, the accuracy does not degrade further when N-step=40 or $\varepsilon \geq 0.5$. If the stochastic gradient was improving robustness, then increasing the attack strength would have broken the defense of our method. This is not observed in reported experiments.

6. Conclusion

In this paper, we propose an approach called J-SIGR (Jacobian normalization and selective input gradient regularization) to improve both the robustness and interpretability of deep neural networks (DNNs). The proposed approach leverages Jacobian matrices to generate gradient-based salient maps, which select informative input gradients to improve DNN robustness against

multiple attacks and enhance the interpretability of adversarial perturbations. We believe that our approach can contribute to the development of trustworthy real-world systems and facilitate the practical deployment of deep learning. Although the proposed method is effective in countering adversarial attacks, the generation of saliency map relies on a plug-in network (e.g., GAN), which can lead to inefficiencies when dealing with large datasets. Future work will explore the use of dense semantic networks for extracting saliency maps, which can enhance distillation and further improve the generalisation of DNNs [37, 38].

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: ICLR, 2014.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models using resistant to adversarial attacks, in: ICLR, 2018.

[3] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: IEEE Symposium on Security and Privacy (SP), 2017, pp. 39–57.

[4] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: CVPR, 2018, pp. 9185–9193.

[5] B. Yang, H. Zhang, Z. Li, Y. Zhang, K. Xu, J. Wang, Adversarial example generation with adabelief optimizer and crop invariance, Applied Intelligence 53 (2023) 2332–2347.

[6] J. Lin, C. Song, K. He, L. Wang, J. E. Hopcrof, Nesterov accelerated gradient and scale invariance for adversarial attacks, in: ICLR, 2020, pp. –.

[7] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, in: NeurIPS, 2019, pp. 125–136.

[8] H. Kannan, A. Kurakin, I. Goodfellow, Adversarial logit pairing, in: arXiv:1803.06373, 2018.

[9] F. TramÃ¨r, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses, in: ICLR, 2018.

[10] Z. He, A. S. Rakin, D. Fan, Parametric noise injection: trainable randomness to improve deep neural network robustness against adversarial attack, in: CVPR, 2020, pp. 588–597.

[11] H. Huang, Y. Wang, S. Erfani, Q. Gu, J. Bailey, X. Ma, Exploring architectural ingredients of adversarially robust deep neural networks, in: NeurIPS, 2021.

[12] A. Bietti, G. Mialon, J. Mairal, On regularization and robustness of deep neural networks, in: arXiv:1810.00363, 2018.

[13] A. S. Ross, F. Doshi-Velez, Improving the adversarial robustness and interpretability of deep neural networks by regularizing input gradients, in: AAAI, 2018.

[14] Avanti Shrikumar and Peyton Greenside and Anshul Kundaje , Learning important features through propagating activation differences, in: ICML, 2017, pp. 3145–3153.

[15] D. Smilkov, N. Thorat, B. Kim, F. Viegas, M. Wattenberg, Smoothgrad: removing noise by adding noise, in: ICML Workshop on Visualization for Deep Learning, 2017.

[16] Hongyang Zhang and Yaodong Yu and Jiantao Jiao and Eric P Xing and Laurent El Ghaoui and Michael I Jordan , Theoretically principled trade-off between robustness and accuracy , in: ICML, 2019.

[17] Chongli Qin and James Martens and Sven Gowal and Dilip Krishnan and Alhussein Fawzi and Soham De and Robert Stanforth and Pushmeet Kohli , Adversarial robustness through local linearization , in: NeurIPS, 2019.

[18] Sameerah Talafha and Banafsheh Rekabdar and Christos Mousas and Chinwe Ekenna, Biologically Inspired Variational Auto-Encoders for Adversarial Robustness, in: International Conference on Deep Learning, Big Data and Blockchain, 2022.

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

[20] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, R. Fergus, Regularization of neural networks using dropconnect, in: ICML, 2013, pp. 1058–1066.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.

[22] J. Hoffman, D. A. Roberts, S. Yaida, Robust learning with jacobian regularization, in: arXiv:1908.02729, 2019.

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

[24] C. Etmann, S. Lunz, P. Maass, C.-B. Schonlieb, On the connection between adversarial robustness and saliency map interpretability, in: ICML, 2019.

[25] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: IEEE European Symposium on Security and Privacy, 2016, pp. 372–387.

[26] N. Papernot, I. Goodfellow, R. Sheatsley, R. Feinman, P. McDaniel, Cleverhans v1.0.0: an adversarial machine learning library, in: arXiv:1610.00768, 2016, pp. –.

[27] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: IEEE Symposium on Security and privacy, 2016, pp. 582–597.

[28] H. Drucker, Y. L. Cun, Improving generalization performance using double back-propagation, IEEE Transactions on Neural Networks 3 (1992) 991–997.

[29] J. Xing, T. Nagata, X. Zou, E. Neftci, J. L. Krichmar, Achieving efficient interpretability of reinforcement learning via policy distillation and selective input gradient regularization, Neural Networks 161 (2023) 228–241.

[30] A. Chan, Y. Tay, Y.-S. Ong, J. Fu, Jacobian adversarially regularized networks for robustness, in: ICLR, 2020.

[31] Y. L. Cun, L. Bottou, Y. Bengio, P. Haffner, Gradient based learning applied to document recognition, Proceeding of IEEE 86 (1998) 2278–2324.

[32] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, in: Technique report, Citeseer, 2009.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (2015) 211–252.

[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[35] Mathias Lecuyer and Vaggelis Atlidakis and Roxana Geambasu and Daniel Hsu and Suman Jana , Certified robustness to adversarial examples with differential privacy, in: IEEE Symposium on Security and Privacy, 2019.

[36] X. Liu, Y. Li, C. Wu, C.-J. Hsieh, Adv-bnn: Improved adversarial defense through robust bayesian neural network, in: ICLR, 2019.

[37] A. Ghosh, S. S. Mullick, S. Datta, S. Das, A. K. Das, R. Mallipeddi, A black-box adversarial attack strategy with adjustable sparsity and generalizability for deep image classifiers, Pattern Recognition 122 (2022) –.

[38] A. E. Cinà, A. Torcinovich, M. Pelillo, A black-box adversarial attack for poisoning clustering, Pattern Recognition 122 (2022) –.

Research highlights

1. An innovative approach based on Jacobian norm and selective input gradient regularization is presented to improve both robustness and interpretability of DNNs under powerful adversarial attacks.

2. Insightful investigation into the prediction confidence of DNNs are delivered to reveal the relationship between Jacobian norm and linear robustness.

3. Extensive experiments are conducted on a variety of attacks to prove the effectiveness of the proposed method.

Dr Deyin Liu received the B.E. degree from Zhengzhou University, China, in 2010. He is currently working with the School of Artificial Intelligence, Anhui University, China. His main research interests include optimization in computer vision, unsupervised learning, and sparse representation learning.

Dr Lin Yuanbo Wu (IEEE senior member) received the Ph.D. degree from the University of New South Wales, Sydney, Australia. She is currently working as a senior lecturer with the Department of Computer Science, Swansea University, United Kingdom. She was previously working as a Research Fellow with The University of Adelaide, The University of Queensland, and The University of Western Australia. She has intensively published more than 70 peer-reviewed academic articles (including two book chapters) in premier journals and proceedings. She is serving as Associate Editor with IEEE Trans on Neural Networks and Learning Systems, IEEE Trans on Multimedia, IEEE Trans on Big Data, and Pattern Recognition Letters. She was a recipient of the Award for Growth of 2021 The 4th Eureka International Innovation and Entrepreneurship Competition (Eureka IIEC 2021). She has served as the Area Chair for ACM Multimedia 2023 and 2022.

Dr Bo Li is currently an Associate Professor with School of Electronics and Information, Northwestern Polytechnical University, China. He has authored more than 20 articles in Pattern Recognition (PR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Transactions on Geoscience and Remote Sensing (TGRS), and IEEE Transactions on Multimedia (TMM). His research interests include deep learning, deep reinforcement learning, and computer vision.

Farid Boussaid received the M.S. and Ph.D. degrees in microelectronics from the National Institute of Applied Science and Technology (INSA), Toulouse, France, in 1996 and 1999, respectively. He joined Edith Cowan University, Perth, Australia, as a Postdoctoral Research Fellow; and a member of the Visual Information Processing Research Group in 2000. He joined The University of Western Australia, Crawley, Australia, in 2005, where he is currently a professor. His current research interests include neuromorphic engineering, smart sensors, and machine learning.

Mohammed Bennamoun (Senior Member, IEEE) is currently a Winthrop Professor with the Department of Computer Science and Software Engineering, UWA; and a Researcher in computer vision, machine/deep learning, robotics, and signal/speech processing. He has published four books. His H-index is 54 and his number of citations is more than 23,000 (Google Scholar). He was awarded more than 65 competitive research grants from the Australian Research Council and numerous other government UWA and industry research grants. He successfully supervised more than 26 Ph.D. students to completion. He has won the Best Supervisor of the Year Award at QUT in 1998 and received award for research supervision at UWA in 2008 and 2016 and the Vice-Chancellor Award for mentorship in 2016. He has delivered conference tutorials at major conferences, including IEEE Computer Vision and Pattern Recognition (CVPR 2016), Interspeech 2014, the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), and European Conference on Computer Vision (ECCV).

Professor Xianghua Xie is currently leading a research team on Computer Vision and Machine Learning (http://csvision.swan.ac.uk) in the Department of Computer Science, Swansea University. He was a recipient of an RCUK Academic Fellowship (tenure-track research focused lectureship) between September 2007 and March 2012. He was appointed as a Senior Lecturer from October 2012, then an Associate Professor in April 2013, and a full Professor from March 2019. Prior to his position at Swansea, He was a Research Associate at the Computer Vision Group, Department of Computer Science, University of Bristol, where he completed both his PhD (2006) and MSc (2002) degrees. By 2020, he has published over 150 fully refereed research publications and (co-)edited several conference proceedings. He is an associate editor of IET Computer Vision and an editorial

member of a number of other international journals and has chaired and co-chaired several international conferences, e.g. BMVC2015 and BMVC2019.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: