ShieldNets: Defending Against Adversarial Attacks Using Probabilistic Adversarial Robustness

[1]Rajkumar Theagarajan*†, [2]Ming Chen*, [1]Bir Bhanu and [3]Jing Zhang‡
[1]Center for Research in Intelligent Systems, University of California, Riverside, CA 92521
[2]Lawrence Berkeley National Laboratory, Berkeley, CA 94720
[3] KLA Corporation, Milpitas, CA 95035
rthea001@ucr.edu, mingchen.chem@lbl.gov, bhanu@cris.ucr.edu, jing.zhang@kla-tencor.com

Abstract

Defending adversarial attack is a critical step towards reliable deployment of deep learning empowered solutions for industrial applications. Probabilistic adversarial robustness (PAR), as a theoretical framework, is introduced to neutralize adversarial attacks by concentrating sample probability to adversarial-free zones. Distinct to most of the existing defense mechanisms that require modifying the architecture/training of the target classifier which is not feasible in the real-world scenario, e.g., when a model has already been deployed, PAR is designed in the first place to provide proactive protection to an existing fixed model. ShieldNet is implemented as a demonstration of PAR in this work by using PixelCNN. Experimental results show that this approach is generalizable, robust against adversarial transferability and resistant to a wide variety of attacks on the Fashion-MNIST and CIFAR10 datasets, respectively.

1. Introduction

Deep learning has demonstrated impressive performance on many important practical problems such as image [11], video [8], audio [9] and text classification [2]. Despite their outstanding performance, it has been recently shown that deep learning models are vulnerable to adversarial manipulation of their input which is intended to cause a misclassification [12, 25, 3]. These adversarial manipulations are carefully crafted perturbations that are so subtle that a human observer does not even notice the modification at all, but can cause deep learning models to mis-classify the input. Fig. 1(a) shows examples of original images from the CIFAR10 [10] and Fashion-MNIST [30] testing datasets that were correctly classified by a VGG [23] Convolutional Neural Network (CNN), (b) shows the adversarial perturbations crafted using attacks discussed in this literature and (c) shows the adversarially perturbed images being misclassified using the same CNN.

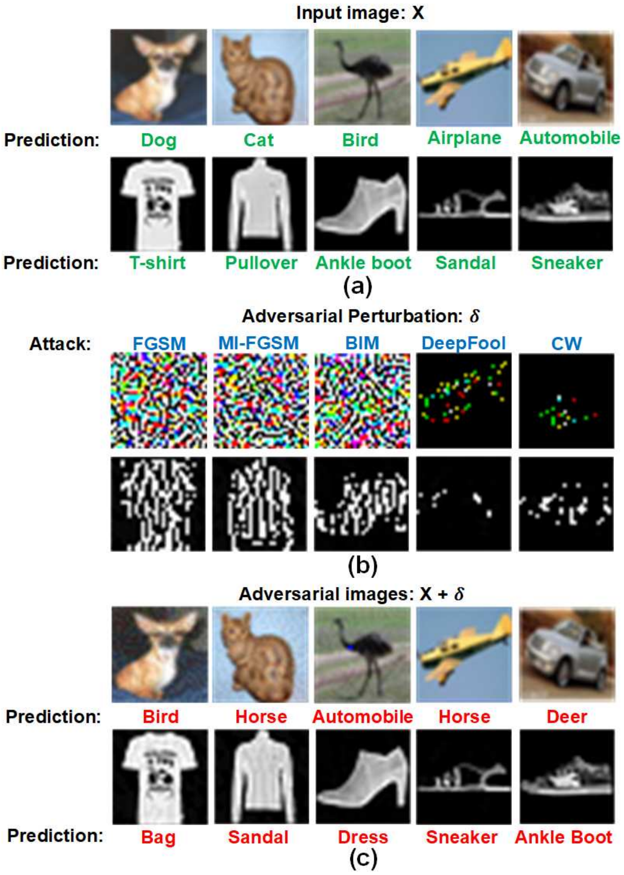

Figure 1. (a) Examples of original images from the CIFAR10 and fashion-MNIST testing datasets correctly classified by VGG, (b) generated perturbation for the corresponding images, (c) corresponding adversarial examples mis-classified by VGG.

[*] Both authors contributed equally.
[†] This work was done in part as an internship at KLA Corporation.
[‡] Corresponding author.

Adversarial attacks can be achieved through black-box attacks and white-box attacks. In the black-box attack model [19], the attacker does not have any access to the parameters or architecture of the classification model, whereas in the white-box attack [5], the attacker has complete access to all the parameters and architecture of the classification model. Szegedy *et al.* [25] showed that an adversarial example that was designed to be mis-classified by a model *M1* can also be used to mis-classify a different model *M2*. This adversarial transferability helps bridge the gap between white-box attacks and black-box attacks. Furthermore, Kurakin *et al.* [12] showed that adversarial examples can also exist in the physical world: https://www.youtube.com/watch?v=zQ_uMenoBCk&feature=youtu.be. The authors of [12] created an adversarial perturbation, printed the perturbed image, photographed the printed image and fed it back to the classifier. Their results show that the classifier mis-classified the photographed image, indicating that physical sensors are also prone to adversarial examples. These kind of attacks can provide disastrous results in safety-critical applications such as self-driving cars [1].

Existing defense approaches can be grouped into three categories: (1) augmenting the training with adversarial examples, e.g., adversarial training [7, 26, 27], (2) modifying the training and architecture of the classifier, e.g., label smoothing [20, 29] and (3) detecting and modifying the adversarial example, e.g., PixelDefend [24], DefenseGAN [22], MagNet [15]. In terms of real world applications, preemptive detection and modification the adversarial example is the most feasible approach to defend classification systems because it does not depend on the architecture of the classifier or the attacking method making it model and attack agnostic.

To this end we propose Probabilistic Adversarial Robustness (PAR) as a fundamental approach to neutralize adversarial attacks. The underlying concept of PAR is to utilize the application loss function to guide a probabilistic model for projecting adversarial examples to the adversarial-free zones. In this paper, we present the theory of PAR and its theoretical possibility to reliably prevent adversarial attacks in a compact region. Moreover, as a demonstration, we select PixelCNN [18] as a specific implementation of PAR's probabilistic model for its state-of-the-art performance in modeling image distributions and tractability of evaluating the data likelihood [21, 28]. The resulting defense network is named as ShieldNet. We train ShieldNet to learn the adversarial-free zones around the input data distribution of the target CNN, and numerically show that the transformed image does belong closer to the training/testing manifold using statistical p-value tests. The rest of this paper is organized as follows. We introduce the necessary related works and background information for adversarial attacks and adversarial defenses in Section 2. The theory of PAR and the construction of ShieldNet are introduced in Section 3. Finally experimental results under different attacks, as well as comparison with other defense methods are presented in Section 4.

2. Related Works and Our Contributions

In this section we explain in detail the related works in two parts. First, we introduce and discuss different adversarial attack strategies employed in this literature. Second, we introduce and discuss existing defense mechanisms.

2.1. Adversarial Attacks

For any given test image $\mathbf{X}$, adversarial attacks try to find a small perturbation δ with $\parallel \delta \parallel_\infty \leq \epsilon_{attack}$ such that a classifier f gives a mis-classification for $\mathbf{X}^{\text{adv}} = \mathbf{X} + \delta$. ϵ_{attack} is a parameter that sets the perturbation limit for each pixel in $\mathbf{X}$ on the color scale.

Fast Gradient Sign Method (FGSM): This attack was proposed by Goodfellow *et al.* [7]. The authors generate a malicious perturbation given by

$$\delta = \epsilon_{attack}\, sign(\bigtriangledown_{\mathbf{X}} L(\mathbf{X}, y)) \tag{1}$$

where $\bigtriangledown_{\mathbf{X}} L(\mathbf{X}, y)$ is the loss function used to train the model and y is the class label. This approach uses the sign of the gradients at every pixel to determine the direction with which to change the corresponding pixel value.

Basic Iterative Method (BIM): This attack was proposed by Kurakin *et al.* [12]. The authors implemented a variant of the FGSM attack by applying it multiple times with a smaller step size. The adversarial examples are formally computed as:

$$\begin{gathered}\mathbf{X}_0^{adv} = \mathbf{X}, \\ \mathbf{X}_{n+1}^{adv} = Clip_{\mathbf{X}}^{\epsilon_{attack}}(\mathbf{X}_n^{adv} + \alpha sign(\bigtriangledown_{\mathbf{X}} L(\mathbf{X}_n^{adv}, y)))\end{gathered} \tag{2}$$

$Clip_{\mathbf{X}}^{\epsilon_{attack}}$ clips the resulting image to be within the ϵ_{attack}-ball of $\mathbf{X}$. Similar to Kurakin *et al.* [12], we set α = 1 and limit the number of iterations to be [min(ϵ_{attack} + 4, 1.25ϵ_{attack})]

DeepFool: This attack was proposed by Moosavi-Dezfooli *et al.* [16]. The authors construct DeepFool by assuming that Neural Networks are linear, with a hyperplane separating each class. Based on this, they iteratively linearize the decision boundary and find the closest adversarial example. We clip the resulting image so that its perturbation is not larger than ϵ_{attack}.

Carlini-Wagner (CW): Carlini and Wagner [5] proposed an efficient optimization objective for iteratively finding the adversarial examples with the smallest perturbation leading to high probability of mis-classification. We clip the resulting image so that its perturbation is not larger than ϵ_{attack}.

Momentum Iterative-FGSM (MI-FGSM): Dong *et al*. [6] proposed integrating a momentum term into an existing iterative attack helps improve the success rate of the attack. The attack won first place in the NIPS 2017 adversarial attack competition for both Targeted and Non-Targeted attacks. The adversarial examples are formally computed as:

$$g_0 = 0, X_0^{adv} = X,$$
$$g_{t+1} = \mu \cdot g_t + \frac{\bigtriangledown_X \ L(X_t^{adv}, y)}{\| \bigtriangledown_X \ L(X_t^{adv}, y) \|_1},$$
$$X_{t+1}^{adv} = X_t^{adv} + \alpha \ sign(g_{t+1}) \tag{3}$$

where, $\bigtriangledown_X L(X, y)$ is the loss function used to train the model, μ is the momentum and $\alpha = \epsilon_{attack}/T$.

2.2. Adversarial Defense

As mentioned in Section 1, current adversarial defense strategies fall into three categories (1) augmenting the dataset with adversarial examples, (2) modifying the training procedure and architecture of the classifier, and (3) detecting and modifying the adversarial example.

2.2.1 Augmenting the Training Dataset

Adversarial Training: This approach proposed by Goodfellow *et al*. [7] works by generating adversarial examples on-the-fly during training and augmenting it to the training dataset. A drawback of this approach was that it was not scalable as shown by [12, 27].

Ensemble Adversarial Training: In this approach Tramèr *et al*. [26] augment their training data with perturbations generated from many different models. This approach tries to decouple the generation of adversarial examples from the model being trained, while simultaneously drawing an explicit connection with robustness to adversarial attacks. The authors mention that this approach makes the classifier more robust but at a high performance cost.

2.2.2 Modifying the Training and Architecture

Label Smoothing: This approach proposed by Warde-Farley and Goodfellow [29] converts one-hot labels to soft targets, where the correct class has a value 1 - *T* while the wrong classes have T/(N - 1). Here *T* is a small constant and *N* is the number of classes. When the classifier is retrained on these soft targets rather than one-hot labels it is more robust to adversarial examples. This method is similar to defensive distillation proposed by [20] but is shown to be computationally inexpensive.

Feature Squeezing: Feature squeezing proposed by Xu *et al*. [31] is both attack and model agnostic. For any given image, its color space is reduced from [0, 255] to a smaller value and then smoothened using a median filter. The resulting image is then passed to the classifier.

2.2.3 Detecting and Modifying Adversarial Examples

MagNet: This approach proposed by Meng *et al*. [15] uses an auto encoder to learn the distribution of the training data. During testing if the input image is from the real dataset the reconstruction loss will be minimum but if the input is an adversarial example, then the loss will be higher.

Defense-GAN: This approach proposed by Samagouei *et al*. [22] uses a trained generative adversarial network to distinguish between real and adversarial examples. During test time given an adversarial image $\mathbf{X} + \delta$, the authors try to project $\mathbf{X} + \delta$ onto the range of the generator by minimizing the reconstruction error $\| \ G(z) - (\mathbf{X} + \delta) \ \|_2^2$. The resulting construction G(z) is then passed to the classifier.

Thermometer Encoding: This approach proposed by Buckman *et al*. [4] discretizes the input as a defense against adversarial examples. The authors replace the pixel values with a binary vector using a thermometer encoding process. The idea here is that by using this kind of encoding, the threshold effects of discretization makes it harder to find adversarial examples that only make small alterations of the image. A drawback of this approach is that it scales the input space dimension linearly with the number of discretization steps, leading to a significant increase in the number of parameters for the model.

PixelDefend: This approach proposed by Song *et al*. [24] leverages pre-trained probabilistic generative networks to purify an adversarial example to resemble the distribution of the training dataset. Although their approach is model and attack agnostic, the performance of their approach decreases as the strength of the attack increases.

2.3. Contributions of This Paper

In light of the state-of-the-art, our work is significantly different from the existing approaches.

- We introduce the theoretic framework of Probabilistic Adversarial Robustness (PAR) to neutralize adversarial attacks by concentrating sample probability to adversarial-free zones.
- We theoretically demonstrate the connection between PAR loss and the SGD loss, and prove the existence of an optimal distribution for the probabilistic transformation to reach a theoretical lower bound.
- We empirically show that our approach is generalizable and robust to adversarial transferability of attacks.
- Our approach is model and attack agnostic, and can be combined with other existing approaches which results in even more improved performance.

3. Probabilistic Adversarial Robustness (PAR)

In this paper, we introduce the PAR to provide a theoretical foundation of possibly neutralizing adversarial attack (AA) samples in the compact regions near the good samples. The approach of PAR is to seek a random function via a probabilistic model to transform the AA samples to the adversarial-free regions. In the following subsections, we establish the theory of PAR, and provide a demonstration of PAR implementation via PixelCNN.

3.1. Theory of PAR

For any given image $\mathbf{X} \in \mathbb{R}^{M \times N}$ where $M \times N$ is the number of pixels in the image, an ϵ-bounded adversarial sample is denoted as $\mathbf{X} + \delta$, where δ belongs to the l_p-bounded neighbourhood $\Delta = \{\delta \in \mathbb{R}^{M \times N} \mid \|\delta\|_p \leq \epsilon\}$ to $\mathbf{X}$. The probabilistic generative model $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta)$ in PAR is expected to map the AA samples from adversarial regions back to a safer space in Δ. Adversarial attacks on any classification task with a loss function of $\mathscr{L}(\mathbf{X}', \mathbf{Y}; \theta)$, where X' is sampled from $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta)$ transformation, can be achieved by optimizing,

$$\arg\max_{\delta \in \Delta} \int_{\Delta} \pi_\omega(\mathbf{X}'|\mathbf{X}+\delta)\mathscr{L}(\mathbf{X}', \mathbf{Y}; \theta)\mathrm{d}\mathbf{X}'. \tag{4}$$

The loss function of PAR can be expressed as the marginalized expectation,

$$\mathscr{L}_{PAR} = \mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})\sim D} \int_{\Delta} \mathbb{E}_{\mathbf{X}'\sim\pi_\omega(\cdot|\mathbf{X}+\delta)}\big[\mathscr{L}(\mathbf{X}', \mathbf{Y}; \theta)\big]p(\delta)\mathrm{d}\delta \tag{5}$$

where $p(\delta)$ represents the distribution of adversarial samples in Δ. The theoretical possibility of PAR to neutralize AA samples is supported by the following three theorems (for proofs, see Appendix 1).

Theorem 1 *if $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta) = \delta_{Dirac}(\mathbf{X}'-\mathbf{X})$ for $\forall \mathbf{X} \sim D$, where $\delta_{Dirac}(\cdot)$ is the Dirac delta function, then $\mathscr{L}_{PAR}$ reduces to SGD loss.*

Theorem 2 *Assume $\mathscr{L}(\mathbf{X}', \mathbf{Y}; \theta)$ is continuous in $\mathbf{X} + \Delta$ and $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta)$ is supported on $\mathbf{X} + \Delta$, there exists a lower bound for $\mathscr{L}_{PAR}$ in space Δ. If $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta) = \delta_{Dirac}(\mathbf{X}' - \mathbf{X} - \beta_0)$, $\mathscr{L}_{PAR}$ reaches the lower bound, where $\beta_0 = \arg\min_{\beta\in\Delta} \mathscr{L}(\mathbf{X}+\beta, \mathbf{Y}; \theta)$.*

Corollary 1 *If $\mathscr{L}_{PAR}$ reaches the lower bound, adversarial perturbation existis only if $\delta \notin \Delta$.*

In practice, there is no guarantee that the lower bound of $\mathscr{L}_{PAR}$ can be realized. Although adversarial attacks through Eq.(4) are possible, the optimization requires SGD and the convergence rate is in the order of $O(1/\lambda)$ [17], which is exponentially slower than the deterministic optimization, where λ is the convergence error.

3.2. PAR via PixelCNN

In this paper we use PixelCNN as the probabilistic model for PAR. $\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta)$ is a joint probability among all pixels, i.e.

$$Pcnn(\mathbf{X}) = \prod_{i}^{M\times N} Pcnn(x_i|x_{1:(i-1)}) \ , \tag{6}$$

where x_i is the i-th pixel of the image. By adopting PixelCNN, it can be factorized into a product of conditional distributions.

$$\pi_\omega(\mathbf{X}'|\mathbf{X}+\delta) = \prod_{i=1}^{M\times N} p(x_i|[x_1, ..., x_{i-1}], \mathbf{X}+\delta) \ . \tag{7}$$

Solving Eq. (5) requires a proper definition of the space Δ. As a practical solution, we introduce a regularization term $\alpha\mathrm{Reg}(\mathbf{X}', \mathbf{X})$ to constraint how far $\mathbf{X}'$ can deviate from $\mathbf{X}$, which implicitly confines the space Δ. Besides, as illustrated by Theorem 1, this regularization also acts as a restraint on $\mathscr{L}_{PAR}$ to be close to the SGD loss without adversarial perturbation. In this work, we use the PixelCNN loss as $\mathrm{Reg}(\mathbf{X}', \mathbf{X})$. The combined loss function is given by:

$$\begin{aligned}\mathscr{L}_{imp} = \mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})\sim D} \int_{\Delta} \mathbb{E}_{\mathbf{X}'\sim\pi_\omega(\cdot|\mathbf{X}+\delta)}\big[&\mathscr{L}(\mathbf{X}', \mathbf{Y}; \theta) \\ &+ \alpha\mathrm{Reg}(\mathbf{X}', \mathbf{X})\big]p(\delta)\mathrm{d}\delta \ ,\end{aligned} \tag{8}$$

where ω is the learnable parameters in the probabilistic model of PAR. It should be noted that θ is the parameters of the protected CNN model, and it is fixed during the learning of PAR. In all of our approaches for a given input image to the probabilistic model, we sample $n = 10$ number of transformations of $\mathbf{X}'$.

For white-box scenario, the optimization of Eq.(8) requires the estimation of $\partial\mathscr{L}_{imp}/\partial\omega$. We utilize PixelCNN++ [21] implementation, which employs mixture models of logistic distributions to represent pixel-wise conditional probability. Through variable transformation of $\mathbf{X}' = \mathbf{X}'(\omega)$, the gradients can be directly evaluated by chain rule.

3.3. ShieldNet Implementation

Fig. 2 shows the overview of one implementation of PAR named as ShieldNet. The ShieldNet consist of three major components: the probabilistic transformation model via PixelCNN , the target CNN classifier, and the optional averager for logits. The inputs to ShieldNet are samples potentially with adversarial perturbations $\mathbf{X} + \delta$. The PixelCNN

In this work, the mean and standard deviation of the logistic distribution with respect to ω are optimized.

(denoted as green box in Fig. 2) is a probabilistic model that generates n different neutralized samples ($\mathbf{X'^{n}_{i=1}}$) for the provided AA sample. The neutralized samples are then given as input to the original target CNN classifier and the average of the n logits is taken for deciding the final prediction Y. The detailed network topologies can be found in Appendix 3.

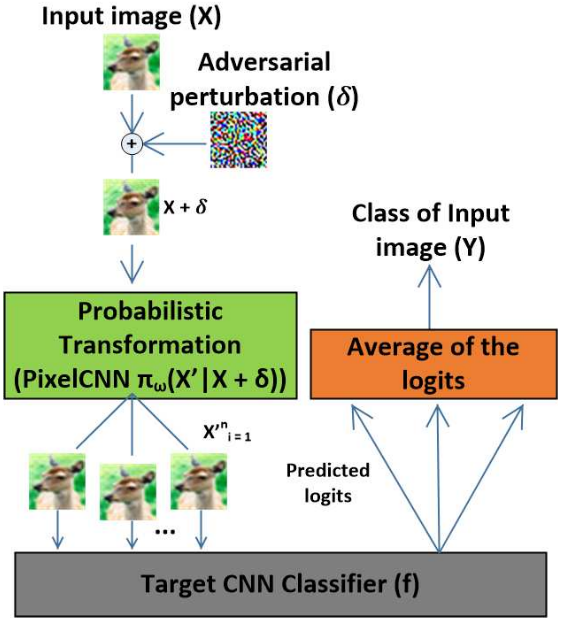
Input image (X)
Adversarial perturbation (δ)
X + δ
Class of Input image (Y)
Probabilistic Transformation (PixelCNN π_ω(X' | X + δ))
Average of the logits
X'^n_{i=1}
Predicted logits
...
Target CNN Classifier (f)

Figure 2. Implementation of PAR: ShieldNet.

4. Experimental Results

4.1. Datasets and Target CNN Models

We evaluated our approach on two publicly available datasets namely: Fashion MNIST [30] and CIFAR10 [10]. Fashion MNIST was designed to be a much more difficult and drop-in replacement for the MNIST dataset [13]. The dataset consists of 60,000 training and 10,000 testing grayscale images of size 28x28 distributed evenly into 10 different classes. CIFAR10 is another widely used dataset that consists of 50,000 training and 10,000 testing RGB images of size 32x32 distributed evenly into 10 different classes.

We evaluated our proposed approach on two state-of-the-art classifiers: ResNet and VGG. For fair comparison we use the same architectures used by [24]. The architectures of the individual CNNs are described in Appendix 3 of this paper. Before training the agent, the two CNNs: ResNet and VGG, were pre-trained on the CIFAR10 and Fashion MNIST datasets, and after pre-training the parameters are fixed and not updated. In principle, we could train both the agent and the CNN jointly but, this is not our desired task, as our aim is to train an agent that can defend a CNN without changing the architecture or re-training the CNN. Table 1 shows the classification results of ResNet and VGG on the original Fashion MNIST and CIFAR10 testing datasets.

Network	**Fashion MNIST**	**CIFAR10**
ResNet	93.51%	95.31%
VGG	93.05%	92.53%

Table 1. Classification accuracy of ResNet and VGG on the Fashion MNIST and CIFAR10 testing datasets

4.2. Neutralizing Adversarial Examples

It has been shown by [24] that adversarial examples have lower probability densities compared to the original training/testing images. Most classifiers suffer from a covariate shift due to the lack of adversarial instances for training leading to mis-classifications.

Similar to [24], we empirically verify this hypothesis by training the PixelCNN model on the CIFAR10 dataset and then use its log-likelihood estimate combined with a p-value test to detect if an input image is from the original training/testing distribution or from the low probability density adversarial space. Let us assume the adversarial input to the PixelCNN model $\mathbf{X}+\delta$ belongs to a distribution $q(\mathbf{X})$ while the original images $\mathbf{X}$ belong to the distribution $p(\mathbf{X})$. The pseudo code for estimating the p-value is given below.

Pseudo code for *p*-value estimation

Assumptions: If the adversarial distribution is same as the training/testing distribution, then the null hypothesis H_0 is given by $q(X) = p(X)$. The alternate hypothesis H_1 is given by $q(X) \neq p(X)$

Input to PixelCNN: perturbed image $\mathbf{X}+\delta$
Output: p-value of perturbed image

- Compute the output probability of the perturbed image as Pcnn($\mathbf{X}+\delta$)
- Compute the output probabilities of the original images in the dataset as {Pcnn($\mathbf{X_1}$), ..., Pcnn($\mathbf{X_N}$)}
- Compute the p-value P given by:
$P = \frac{1}{N+1}\sum_{i=1}^{N}\mathbb{I}[Pcnn(\mathbf{X_i}) \leq Pcnn(\mathbf{X}+\delta)] + 1$
where, $\mathbb{I}$[.] = 1, if the condition in the bracket is true, otherwise it is 0.

Fig. 3(a) shows the p-values of the original testing dataset of CIFAR10 and p-values of state-of-the-art adversarial attacks with $\epsilon_{attack} = 8$. It can be observed that the original testing images have a more uniform p-value distribution compared to the adversarial attacks which significantly deviate from a uniform distribution. This proves that the distribution space of the original testing images is different from the adversarial distribution space proving the alternate hypothesis H_1. Fig. 3(b) shows the p-values of the

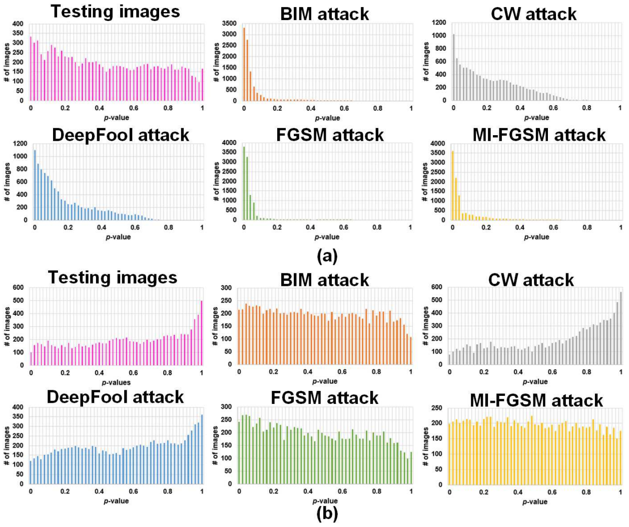

Figure 3. (a) p-values of the original testing dataset of CIFAR10 and the state-of-the-art attacks on the testing dataset of CIFAR10 using $\epsilon_{attack} = 8$ (b) p-values of the corresponding neutralized images after transformation using ShieldNet

corresponding images in Fig. 3(a) after being transformed by our approach. It can be observed that the transformed adversarial images have a much more uniform p-value distribution similar to the p-value distribution of the original testing dataset. In Fig. 3(b) after the neutralization, DeepFool and CW attacks have distributions very similar to the original testing images compared to FGSM, BIM and MI-FGSM. The reason for this is that DeepFool and CW attacks are designed to linearize the decision boundaries between classes which result in perturbations that are small enough just to fool the classifier compared to FGSM, BIM and MI-FGSM that create larger perturbations as shown in Fig. 1.

4.3. ShieldNet Defending Intra-Attack

In this sub-section we evaluate the performance of individual ShieldNet defending against the same attacking schemes as the ones used in training. The evaluations cover the state-of-the-art attacking algorithms including FGSM, BIM, DeepFool, CW and MI-FGSM for ResNet and VGG on both Fashion MNIST and CIFAR10 datasets, as shown in Table 2 and 3. For fair comparison on both datasets, we use the same $\epsilon_{\mathtt{attack}}$ used by [24] for evaluating and comparing our approach. In Table 2 the evaluation on Fashion MNIST dataset utilizes $\epsilon_{\mathtt{attack}}$ = 8 and 25, where the CIFAR10 experiments in Table 3 apply $\epsilon_{\mathtt{attack}}$ = 2 and 16. The cells in Table 2 and 3, as well as all following tables, are formatted as *x/y* where x and y are the accuracies for smaller/larger $\epsilon_{\mathtt{attack}}$.

Table 2 and 3 show that our approach in general outperforms other defending algorithms listed in the tables. For example, on Fashion MNIST dataset, our approach outperforms PixelDefend in FGSM attack and achieves an accuracy of **89.04%** and **88.59%** against the strongest attack for ResNet and VGG respectively whereas, PixelDefend achieves 74% and 82% respectively. Although there is a drop in performance as the strength of the attack increases it does not significantly drop as compared to PixelDefend. On CIFAR10 dataset, PixelDefend slightly outperforms our approach in defending VGG against FGSM and BIM when $\epsilon_{\mathtt{attack}}$ = 2 but, as $\epsilon_{\mathtt{attack}}$ increases from 2 to 16, the performance of PixelDefend drops down drastically. When protecting ResNet against FGSM and BIM attacks with $\epsilon_{\mathtt{attack}}$ = 16, PixelDefend only achieves 24% and 25% while our approach achieves an accuracy of **70.52%** and **68.86%** respectively. Moreover, by combining our approach with adversarial training using FGSM we observe overall increases in accuracies as shown in the bottom rows in Table 2 and 3.

It should be noted that in general as the strength of the attack increases the performance of defense algorithms tends to decrease, but these perturbations become more clearly visible even to a human observer and can easily be detected and filtered out using statistical p-value tests as described in Section 4.2.

4.4. Generalization Across Different Attacks

In this sub-section we demonstrate the generalizability of ShieldNet against different attack schemes. It has been shown that adversarial training does not generalize across different attacking schemes. As an example, from Table 2 and 3, it can be observed that adversarial training with FGSM examples is able to defend against the basic FGSM attacks, but fails to defend against other attacks. This finding is consistent with the results obtained by [14] and [24].

Table 4 and 5 demonstrate that ShieldNet is able to generalize for both the training and other attacking schemes. As an example, ShieldNet trained with FGSM samples achieves an accuracy of 89.04%, 88.09%, 84.39%, 83.05%, 82.01% accuracy against FGSM, BIM, DeepFool, CW and MI-FGSM attacks respectively. We believe the reason that ShieldNet generalizes across different attacks is that, by using a PixelCNN model in PAR, the model learns to make small changes on the individual pixels that can move the perturbed image back to an adversarial-free zone around the training/testing data distribution. In Table 4 and 5 training our approach using the BIM attack had the best overall accuracy and generalization across different attacks. From Table 5 it should be noted that training on DeepFool is only able to successfully defend against CW attacks and *vice-versa* and falls short across the other attacks compared to BIM. This is because, DeepFool and CW attacks are designed to create higher order perturbations by directly linearizing the decision boundaries of the CNN, whereas, iterative attacks such as BIM create perturbations based on the sign of the gradient at every pixel irrespective of the deci-

Fashion MNIST $\epsilon_{\mathtt{attack}}$ = 8, 25						
Network	**Training technique**	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
ResNet	Label Smoothing	64.23/36.81	9.76/0.00	22.42/3.37	20.77/4.61	4.25/0.00
	Adversarial FGSM	82.49/78.43	44.34/6.46	57.28/11.92	51.03/15.70	39.72/0.00
	PixelDefend	85.00/74.00	83.00/76.00	87.00/87.00	87.00/87.00	NA
	Our Approach	**91.59/89.04**	**91.17/89.74**	**92.62/90.28**	**92.66/90.78**	**90.63/90.47**
	Our Approach + Adversarial FGSM	***92.46/90.35***	***91.93/90.68***	***92.88/91.36***	***93.47/91.61***	***91.45/90.59***
VGG	Label Smoothing	58.92/44.11	12.24/5.47	31.37/9.73	35.65/11.06	13.17/5.40
	Adversarial FGSM	84.55/76.21	56.39/22.74	37.48/18.71	30.69/12.52	28.72/10.11
	PixelDefend	87.00/82.00	85.00/83.00	88.00/88.00	88.00/88.00	NA
	Our Approach	**89.04/88.59**	**90.78/87.59**	**90.11/90.29**	**90.56/90.33**	**90.49/89.81**
	Our Approach + Adversarial FGSM	***91.55/88.72***	***91.37/90.15***	***91.02/90.77***	***91.27/90.76***	***90.95/90.56***

Table 2. Performance comparison of ShieldNet and other defense algorithms on the Fashion MNIST testing dataset. The highest accuracy is indicated in bold + italic and the second highest accuracy is indicated in bold.

CIFAR10 $\epsilon_{\mathtt{attack}}$ = 2, 16						
Network	**Training technique**	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
ResNet	Label Smoothing	64.57/14.78	43.28/2.92	53.45/20.56	50.78/14.37	32.91/6.73
	Adversarial FGSM	***83.47/79.13***	34.58/6.73	39.22/8.76	28.47/5.38	26.94/2.33
	Adversarial BIM	71.46/45.92	67.49/12.57	70.59/34.28	75.31/22.89	72.09/27.85
	PixelDefend	73.00/24.00	71.00/25.00	80.00/80.00	78.00/78.00	NA
	Our Approach	76.57/70.52	**73.13/68.86**	**83.47/82.34**	**80.71/80.43**	**75.81/70.42**
	Our Approach + Adversarial FGSM	**81.29/72.61**	***75.59/69.84***	***84.73/84.08***	***82.91/80.86***	***78.44/71.27***
VGG	Label Smoothing	43.26/7.22	28.94/0.00	36.15/3.47	31.65/4.83	20.72/0.00
	Adversarial FGSM	79.28/71.59	39.88/2.96	28.49/2.60	34.27/5.39	30.06/3.18
	Adversarial BIM	76.24/37.38	40.52/11.86	77.14/54.95	71.44/36.91	69.81/20.41
	PixelDefend	**80.00**/52.00	***80.00***/48.00	81.00/76.00	81.00/79.00	NA
	Our Approach	78.61/**68.25**	75.32/**67.34**	**83.19/76.20**	**83.26/79.11**	**73.92/70.43**
	Our Approach + Adversarial FGSM	***81.34/70.61***	**77.58/70.13**	***88.42/79.35***	***83.82/80.79***	***75.69/71.98***

Table 3. Performance comparison of ShieldNet and other defense algorithms on the CIFAR10 testing dataset

sion boundary of the CNN. This means that DeepFool and CW attacks are highly dependent on the individual CNN indicating that they have less adversarial transferability.

4.5. Robustness against Adversarial Transferability

From a security perspective, an important property of adversarial examples is that they tend to transfer from one model to another, enabling an attacker to create adversarial examples from a source model $M1$ and then deploy those adversarial examples to fool a target model $M2$. To evaluate our approach against this property, we created adversarial examples that fooled the source CNN (ResNet/VGG) and used those adversarial examples to evaluate the performance on the target CNN (VGG/ResNet). Table 6 and 7 shows the robustness of our approach against adversarial transferability. In Table 6 and 7, lower classification accuracy values indicate higher adversarial transferbility. For example, from the attackers perspective, one can transfer AA samples optimized for ResNet with FGSM method and $\epsilon_{attack} = 16$ to fool the VGG model since the VGG model can only provide 20.77% accuracy with these AA samples.

From Table 6 and 7, it can be observed that the adversarial transferability property of FGSM and MI-FGSM was the highest followed by BIM. DeepFool and CW attacks had the least adversarial transferability which is evident from the fact that, these attacks are designed to linearize decision boundaries of the CNN. Since ResNet and VGG were trained independent to each other, they do not have the same decision boundaries thus making the adversarial examples from DeepFool and CW less transferable. As the adversarial transferability of DeepFool and CW are relatively low, the accuracy of ShieldNet does not vary too much. However, for FGSM, BIM and MI-FGSM attacks which have relatively high adversarial transferability, as shown in Ta-

Generalization of the ShieldNet + ResNet on Fashion MNIST with $\epsilon_{\mathtt{attack}}$ = 8, 25					
Attack used for training ⇓	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
FGSM	***91.59/89.04***	**88.58/88.09**	86.43/**84.39**	84.22/83.05	83.79/**82.01**
BIM	**88.57/86.18**	***91.17/89.74***	**88.46**/83.99	**86.67/85.43**	**84.25**/81.22
DeepFool	81.39/78.21	83.47/81.26	***92.62/90.28***	85.91/84.04	83.19/81.48
CW	80.11/78.56	86.39/82.11	87.48/83.72	***92.66/90.78***	82.44/79.49
MI-FGSM	82.83/79.34	83.41/81.93	86.90/81.38	84.31/82.01	***90.63/90.47***

Table 4. Cross evaluation of adversarial attacks on the Fashion MNIST dataset using ResNet.

Generalization of ShieldNet + ResNet on CIFAR10 with ϵ_{attack} = 2, 16					
Attack used for training ⇓	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
FGSM	***76.57/70.52***	**71.56**/66.82	68.96/63.51	65.23/61.40	66.02/59.24
BIM	**70.44/68.52**	***73.13/68.86***	70.38/68.44	68.87/67.49	**71.37/68.94**
DeepFool	66.97/63.21	68.55/60.13	***83.47/82.34***	**78.37/77.40**	66.54/61.10
CW	70.91/64.11	65.22/60.98	**73.50/71.25**	***80.71/80.43***	63.17/61.45
MI-FGSM	66.10/61.32	68.90/**68.27**	71.44/66.71	71.77/70.84	***75.81/70.42***

Table 5. Cross evaluation of adversarial attacks on the CIFAR10 dataset using ResNet.

Adversarial transferability on CIFAR10 with $\epsilon_{\mathtt{attack}}$ = 2, 16						
Source CNN	**Target CNN**	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
ResNet	**VGG**	51.61/20.77	**72.45**/41.56	83.24/**82.56**	**81.02/77.29**	44.37/19.86
	ShieldNet + VGG	**78.62/72.60**	71.43/**66.03**	**83.51**/79.95	79.91/76.53	**73.29/70.17**
VGG	**ResNet**	53.81/29.14	58.44/27.67	79.41/**82.46**	**81.93**/80.85	51.45/18.26
	ShieldNet + ResNet	**79.14/70.82**	**70.43/68.18**	**80.94**/79.81	81.34/**81.37**	**71.66/68.54**

Table 6. Evaluation of our approach against adversarial transferability on the CIFAR10 testing dataset.

Adversarial transferability on the Fashion MNIST dataset with $\epsilon_{\mathtt{attack}}$ = 8, 25						
Source CNN	**Target CNN**	**FGSM**	**BIM**	**DeepFool**	**CW**	**MI-FGSM**
ResNet	**VGG**	66.41/26.37	70.39/37.25	86.41/87.26	81.92/82.43	45.22/22.19
	ShieldNet + VGG	**85.95/82.40**	**81.33/76.38**	**90.29/89.26**	**87.31/84.53**	**82.60/80.41**
VGG	**ResNet**	59.27/35.91	72.83/40.90	83.27/80.18	80.66/78.51	64.92/38.36
	ShieldNet + ResNet	**84.32/80.17**	**83.50/84.26**	**88.74/86.71**	**85.70/86.65**	**88.67/81.51**

Table 7. Evaluation of our approach against adversarial transferability on the Fashion MNIST testing dataset.

ble 6 and 7, our approach is able to defend against the adversarial transferability property by improving the accuracy from 20.77% to **72.60%** for ResNet and from 29.14% to **70.82%** for VGG against the strongest FGSM attack on the CIFAR10 testing dataset. Similarly, the performance is improved from 26.37% to **82.40%** for ResNet and from 35.91% to **80.17%** for VGG against the strongest FGSM attack on the Fashion MNIST testing dataset.

5. Conclusions

Probabilistic Adversarial Robustness (PAR) is proposed and implemented via adopting PixelCNN as the probabilistic transformation model to defend target CNNs against adversarial attacks. We theoretically derived the connection between PAR loss and the SGD loss, and the existence of a theoretical lower bound of PAR loss representing the optimal mapping of the adversarial examples to the adversarial-free zones. We numerically demonstrated that ShieldNet can greatly improve the defending accuracy for intra-attack and generalize well across different attacking methods. Moreover, experimental results demonstrated the generality of our approach to adversarial transferability with respect to different CNN models and its resistance to existing attacks.

Acknowledgements

This work was supported in part by an internship at KLA Corporation and by the Bourns Endowment funds at the University of California, Riverside (UCR).

References

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016. 2

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013. 1

[4] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018. 3

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *arXiv preprint arXiv:1608.04644*, 2016. 2

[6] Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, and J. Zhu. Boosting adversarial attacks with momentum. *arXiv preprint arXiv:1710.06081*, 2017. 3

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples (2014). *arXiv preprint arXiv:1412.6572*. 2, 3

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012. 1

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[10] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 1, 5

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[12] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2, 3

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 6

[15] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 2, 3

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 2

[17] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. 4

[18] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[19] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017. 2

[20] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *arXiv preprint arXiv:1511.04508*, 2015. 2, 3

[21] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 2, 4

[22] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018. 2, 3

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[24] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 2, 3, 5, 6

[25] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[26] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 2, 3

[27] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 2, 3

[28] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016. 2

[29] D. Warde-Farley and I. Goodfellow. 11 adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, page 311, 2016. 2, 3

[30] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv cs.LG/1708.07747*, 2017. 1, 5

[31] W. Xu, D. Evans, and Y. Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017. 3