

Ensemble Generative Cleaning with Feedback Loops for Defending Adversarial Attacks

Jianhe Yuan and Zhihai He
 University of Missouri, Columbia MO
 {yuanjia, hezhi}@missouri.edu

Abstract

Effective defense of deep neural networks against adversarial attacks remains a challenging problem, especially under powerful white-box attacks. In this paper, we develop a new method called ensemble generative cleaning with feedback loops (EGC-FL) for effective defense of deep neural networks. The proposed EGC-FL method is based on two central ideas. First, we introduce a transformed deadzone layer into the defense network, which consists of an orthonormal transform and a deadzone-based activation function, to destroy the sophisticated noise pattern of adversarial attacks. Second, by constructing a generative cleaning network with a feedback loop, we are able to generate an ensemble of diverse estimations of the original clean image. We then learn a network to fuse this set of diverse estimations together to restore the original image. Our extensive experimental results demonstrate that our approach improves the state-of-art by large margins in both white-box and black-box attacks. It significantly improves the classification accuracy for white-box PGD attacks upon the second best method by more than 29% on the SVHN dataset and more than 39% on the challenging CIFAR-10 dataset.

1. Introduction

Researchers have recognized that deep neural networks are sensitive to adversarial attacks [32]. Very small changes of the input image can fool the state-of-art classifier with very high success probabilities. The attackers often generate noise patterns by exploiting the specific network architecture of the target deep neural network so that small noise at the input layer can accumulate along the network inference layers, finally exceed the decision threshold at the output layer, and result in false decision. On the other hand, we know a well-trained deep neural networks are robust to random noise [1], such as Gaussian noise. Therefore, the key issue in network defense is to destroy the sophisticated pattern or accumulative process of the attack noise while

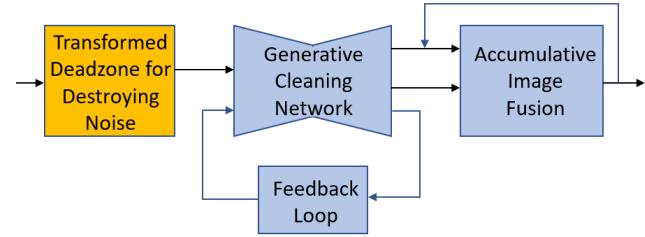


Figure 1. Illustration of the proposed ensemble generative cleaning with feedback loops for defending adversarial attacks.

preserving the original image content or network classification performance.

During the past few years, a number of methods have been proposed to construct adversarial samples to attack the deep neural networks, including fast gradient sign (FGS) method [10], Jacobian-based saliency map attack (J-BSPA) [26], and projected gradient descent (PGD) attack [18, 20]. Different classifiers can be failed by the same adversarial attack method [32]. The fragility of deep neural networks and the availability of these powerful attacking methods present an urgent need for developing effective defense methods. Meanwhile, deep neural network defense methods have also been developed, including adversarial training [18, 32], defensive distillation [27, 4, 24], Magnet [21], and featuring squeezing [13, 41]. It has been recognized that these methods suffer from significant performance degradation under strong attacks, especially white-box attacks with large magnitude and iterations [29].

In this work, we explore a new approach, called *ensemble generative cleaning with feedback loop* (EGC-FL), to defend deep neural network against powerful adversarial attacks. Our approach is motivated by the following observation: (1) the adversarial attack has sophisticated noise patterns which should be disturbed or destroyed during the defense process. (2) The attack noise, especially those powerful white-box attacks, such as the PGD and BPDA attacks [2], are often generated with an iterative process. To clean them, we also need an iterative process with multiple rounds

of cleaning to achieve effective defense.

Motivated by these observations, our proposed EGC-FL approach first introduces a transformed deadzone (TDZ) layer into the defense network, which consists of an orthonormal transform and a deadzone-based activation function, to destroy the sophisticated noise pattern of adversarial attacks. Second, it introduces a new network structure with feedback loops, as illustrated in Figure 1, into the generative cleaning network. This feedback loop network allows us to remove the residual attack noise and recover the original image content in an iterative fashion. Specifically, over multiple feedback iterations, the EGC-FL network generates an ensemble of cleaned estimations of the original image. Accordingly, we also learn an accumulative image fusion network which is able to fuse the new estimation with existing result in an iterative fashion. According to our experiments, this feedback and iterative process converges very fast, often within 2 to 4 iterations. Our extensive experimental results on benchmark datasets demonstrate that our EGC-FL approach improves the state-of-art by large margins in both white-box and black-box attacks. It significantly improves the classification accuracy for white-box attacks upon the second best method by more than 29% on the SVHN dataset and more than 39% on the challenging CIFAR-10 dataset with PGD attacks.

The **major contributions** of this work can be summarized as follows. (1) We have introduced a transform deadzone layer into the defense network to effectively destroy the noise pattern of adversarial attacks. (2) We have developed a new network structure with feedback loops to remove adversarial attack noise and recover original image content in an iterative manner. (3) We have successfully learned an accumulative image fusion network which is able to fuse the incoming sequence of cleaned estimations and recover the original image in an iterative manner. (4) Our new method has significantly improved the performance of the state-of-the-art methods in the literature under a wide variety of attacks.

The rest of this paper is organized as follows. Section 2 reviews related work. The proposed EGC-FL method is presented in Section 3. Experimental results, performance comparisons with existing methods, and ablation studies are provided in Section 4. Section 5 concludes the paper.

2. Related work

In this section, we review related work on adversarial attack and network defense methods which are two tightly coupled research topics. The goal of attack algorithm design is to fail all existing network defense methods, while the goal of defense algorithms is to defend the deep neural networks against all existing adversarial attack methods.

(A) Attack methods. Attack methods can be divided into two threat models: *white-box* attacks and *black-box* at-

tacks. The white-box attacker has full access to the classifier network parameters, network architecture, and weights. The black-box attacker has no knowledge of or access to the target network. For white-box attack, a simple and fast approach called *Fast Gradient Sign (FGS)* method has been developed by Goodfellow *et al.* [10] using error back propagation to directly modify the original image. Kurakin *et al.* [18] apply FGS iteratively and propose BIM. Carlini *et al.* [4] designed an optimization-based attack method, called *Carlini-Wagner (C&W) attack*, which is able to fool the target network with the smallest perturbation. Xiao *et al.* [37] trained a generative adversarial network (GAN) [9] to generate perturbations. Kannan *et al.* [17] found that the *Projected Gradient Descent (PGD)* is the strongest among all attack methods. It can be viewed as a multi-step variant of FGS^k [20]. Athalye *et al.* [2] introduced a method, called Backward Pass Differentiable Approximation (BPDA), to attack networks where gradients are not available. It iteratively computes the adversarial gradient on the defense results. It is able to successfully attack all existing state-of-the-arts defense methods. For black-box attacks, the attacker has no knowledge about the target classifier. Papernot *et al.* [25] introduced the first approach for black-box attack using a substitute model. Dong *et al.* [8] proposed a momentum-based iterative algorithms to improve the transferability of adversarial examples. Xie *et al.* [40] boosted the transferability of adversarial examples by creating diverse input patterns.

(B) Defense methods. Several approaches have recently been proposed for defending both white-box attacks and black-box attacks. Adversarial training trains the target model using adversarial examples [32, 10]. Madry *et al.* [20] suggested that training with adversarial examples generated by PGD improves the robustness. [21] proposed a method, called MagNet, which detects the perturbations and then reshape them according to the difference between clean and adversarial examples. Recently, there are several defense methods based on GANs have been developed. Samangouei *et al.* [29] projected the adversarial examples into a trained generative adversarial network (GAN) to approximate the input using generated clean image. Recently, some defense methods have been developed based on input transformations. Guo *et al.* [11] proposed several input transformations to defend the adversarial examples, including image cropping and re-scaling, bit-depth reduction, and JPEG compression. Xie *et al.* [38] proposed to defend against adversarial attacks by adding a randomization layer, which randomly re-scales the image and then randomly zero-pads the image. Jia *et al.* [15] proposed an image compression framework to defend adversarial examples, called *ComDefend*. Xie *et al.* [39] introduced a feature denoising method for defending PGD white-box attacks.

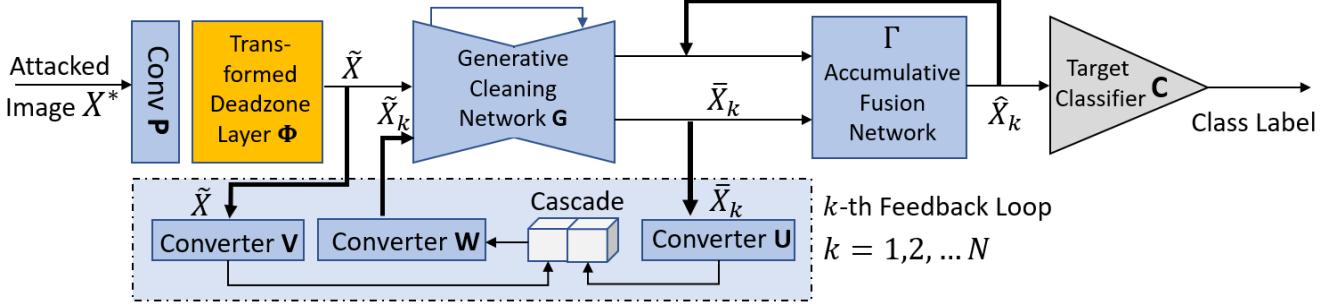


Figure 2. Framework of the proposed ensemble generative cleaning network for defending adversarial attacks.

3. The Proposed Method

In this section, we present our method of ensemble generative cleaning with feedback loops for defending adversarial attacks.

3.1. Overview

As illustrated in Figure 1, our proposed method of ensemble generative cleaning with feedback loops (EGC-FL) for defending adversarial attacks is based on **two main ideas**: (1) we introduce a transformed deadzone layer into the cleaning network to destroy the sophisticated noise patterns of adversarial attacks. (2) We introduce a generative cleaning network with a feedback loop to generate a sequence of diverse estimations of the original image, which will be fused in an accumulative fashion to restore the original image.

Figure 2 shows a more detailed framework of the proposed EGC-FL method. The attacked image X^* is first pre-processed by a convolutional layer P and then passed to the transformed deadzone layer Φ , which aims to destroy the sophisticated noise patterns of the adversarial attacks. To remove the residual attack noise in \tilde{X} and recover the original image content X , the generative cleaning network G generates a series of estimations of the original image using a feedback loop. The feedback network consists of three converter networks, U , V , and W , which are fully convolutional layers. These three converter networks are used to normalize the output features from different networks before they are concatenated or fused together. At the k -th feedback loop, let \bar{X}_k be the output of the generative cleaning network G . We concatenate the output \bar{X}_k and the original \tilde{X} after being normalized by converter networks U and V , respectively. The concatenated feature map is then normalized by converter W before feeding back to the generative cleaning network G to produce the output \bar{X}_{k+1} . This feedback loop is summarized by the following formula:

$$\bar{X}_{k+1} = G\{\mathbf{W}[\mathbf{V}(\tilde{X}) \uplus \mathbf{U}(\bar{X}_k)]\}, \quad (1)$$

where \uplus represents the cascade operation. This ensemble

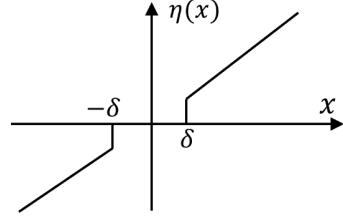


Figure 3. Activation function for the TDZ layer.

generative cleaning network with feedback will generate a series of cleaned versions $\{\bar{X}_k | k = 1, 2, \dots\}$, representing a diverse set of estimations of the original image X . To recover the original image X , we introduce an accumulative image fusion network Γ , which operates as follows

$$\hat{X}_{k+1} = \Gamma(\hat{X}_k, \bar{X}_k). \quad (2)$$

Specifically, the input to the fusion network Γ are two images: \bar{X}_k which is the current output from the generative cleaning network G , and \hat{X}_k which is the current fused image produced by Γ . The generative cleaning network G is separated from the accumulative fusion network Γ so that the generative network can generate multiple estimations of the original image. The fusion network can then fuse them together. In other words, the output of Γ is fed back to itself as the input for the next round of fusion. All networks, including the convolution pre-processing, the generative cleaning network, converter networks, and the accumulative fusion network are learned from our training data, which will be explained in more detail in the following sections.

3.2. Transformed Deadzone Layer

The goal of the transformed deadzone layer in our defense network is to destroy the noise pattern and perform the first round of removal of the adversarial attack noise. Let X be the original image and $x(i, j)$ be its pixel at location (i, j) . The attacked image is given by $X^* = X + \alpha_\epsilon$ where α_ϵ is adversarial attack with magnitude ϵ and $\alpha_\epsilon(i, j)$

is the attack noise at pixel location (i, j) , which is a random variable with maximum magnitude of ϵ . We have $x^*(i, j) = x(i, j) + \alpha_\epsilon(i, j)$. In the spatial domain, it is very challenging to separate the attack noise from the original image content since the attacked image X^* and the original image X are visually very similar to each other perceptually.

To address this issue, we propose to first transform the image using a de-correlation or energy compaction orthonormal transform matrix \mathbf{T} . One choice of this transform is the blockwise discrete cosine transform (DCT) [34]. After this transform, the energy of the original image will be aggregated onto a small fraction of transform coefficients with the remaining coefficients being very close to zeros. We then pass this transformed image through a deadzone activation function $\eta(x)$ shown in Figure 3. Here, $\eta(x) = 0$ if $x \in [-\delta, \delta]$. Otherwise $\eta(x) = x$. Since the transform is linear, the transformed image after the deadzone activation is given by

$$\eta(T \cdot X^*) = \eta(T \cdot X + T \cdot \alpha_\epsilon), \quad (3)$$

$$\eta(x_t^*(i, j)) = \eta(x_t(i, j)) + \eta(\alpha_t(i, j)) \quad (4)$$

$$\approx \eta(x_t(i, j)). \quad (5)$$

Statistically, the attack noise is white noise. After transform, $\alpha_t(i, j)$ remains white noise. Notice that a vast majority of transform coefficients $x_t(i, j)$ in the transformed image $T \cdot X$ will be very small. In this case, the deadzone activation function $\eta(x)$ will largely remove the transformed attack noise $\alpha_t(i, j)$. Meanwhile, since the major image content or energy has been aggregated onto a smaller number large-valued coefficients, which remain unchanged by the deadzone function. In this way, the energy-compaction transform is able to help protecting the original image content from being damaged by the deadzone activation function during removal of attack noise. Certainly, it will still cause some damage to the original image content since the small transform coefficients $x_t(i, j)$ are forced to zeros. Figure 4 shows the energy of the attack noise before and after the TDZ, namely, $\|\alpha_\epsilon\|_2 = \|X - X^*\|_2$ and $\|\Gamma(X) - \Gamma(X^*)\|_2$, for 860 test images organized in 215 batches. Here $\Gamma(\cdot)$ represents the transformed deadzone operation. We can see that the energy of attack noise has been significantly reduced. Certainly, some parts of the original image content, especially those high-frequency details, are also removed, which need to be recovered by the subsequent generative cleaning network.

3.3. Learning the Ensemble Generative Cleaning Network

In our defense method design, the generative cleaning network \mathbf{G} , the feedback loop $\mathbf{U}, \mathbf{V}, \mathbf{W}$ and accumulative fusion network $\mathbf{\Gamma}$ are jointly trained. The goal of our

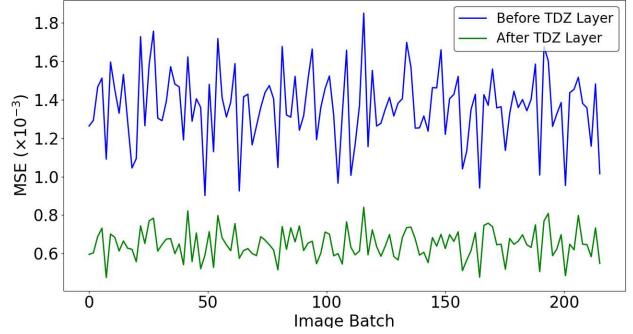


Figure 4. The energy of attack noise before and after the transformed TDZ for 860 test images from 215 batches.

method is three-fold: (1) first, the generative cleaning network \mathbf{G} needs to make sure that the original image content is largely recovered. (2) Second, the feedback loop needs to successfully remove the residual attack noise. (3) Third, the accumulative fusion network $\mathbf{\Gamma}$ needs to iteratively recover the original image content. To achieve the above three goals, we formulate the following generative loss function for training the networks

$$\mathbf{L} = \lambda_1 \mathbf{L}_P + \lambda_2 \mathbf{L}_A + \lambda_3 \mathbf{L}_C, \quad (6)$$

where \mathbf{L}_P is perceptual loss, \mathbf{L}_A is the adversarial loss and \mathbf{L}_C is the cross-entropy loss. λ_i is a weighting parameter. In our experiments, we set it to be 1/3. To define the perceptual loss, the L_2 -norm between the recovered image \hat{X}_k and the original image X is used [16]. In this work, we observe that the small adversarial perturbation often leads to very substantial noise in the feature map of the network [39]. Motivated by this, we use a pre-trained VGG-19 network, denoted by \mathbf{F}_β to generate visual features for the recovered image \hat{X}_k and the original image X , and use their feature difference as the perceptual loss \mathbf{L}_P . Specifically,

$$\mathbf{L}_P = \|\mathbf{F}_\beta(X) - \mathbf{F}_\beta(\hat{X}_k)\|_2^2. \quad (7)$$

The adversarial loss \mathbf{L}_A aims to train generative cleaning network \mathbf{G} and the feedback loop $\mathbf{U}, \mathbf{V}, \mathbf{W}$ so that the recovered images will be correctly classified by the target network. It is formulated as

$$\mathbf{L}_A = \|\mathbf{G}\{\mathbf{W}[\mathbf{V}(\tilde{X}) \uplus \mathbf{U}(\bar{X}_k)]\} - X\|_2^2. \quad (8)$$

[U+FF0C] We train our accumulative fusion network $\mathbf{\Gamma}$, along with the generative cleaning network \mathbf{G} , to optimize the following loss function:

$$\mathbf{L}_C = \mathbb{E}_{X \in \Omega} \Phi[\mathbf{\Gamma}(\hat{X}_k, \bar{X}_k), \mathbf{I}_{clean}]. \quad (9)$$

Here, $\Phi[\cdot, \cdot]$ represents the cross-entropy between the output generated by the generative network and the target label

Table 1. Performance of our method (classification accuracy after defense) against white-box attacks on CIFAR-10 dataset ($\epsilon = 8/256$). Some methods did not provide results on specific attack methods, which were left blank (marked with (NA)).

Defense Methods	Clean	FGS	PGD	BIM	C&W
<i>No Defense</i>	94.38%	31.89%	0.00%	0.00%	0.99%
Label Smoothing [36]	92.00%	54.00%	(NA)	8.00%	2.00%
Feature Squeezing [41]	84.00%	20.00%	(NA)	0.00%	78.00%
PixelDefend [30]	85.00%	70.00%	(NA)	70.00%	80.00%
Adv. Network [35]	91.08%	72.81%	44.28%	(NA)	(NA)
Parametric Noise Injection (PNI) [14]	85.17%	56.51%	49.07%	(NA)	(NA)
Sparse Transformation Layer (STL) [31]	90.11%	87.15%	(NA)	88.03%	89.04%
Our Method	91.65%	88.51%	88.61%	88.75%	90.03%
Gain		+1.36%	+39.54%	+0.72%	+0.99%

\mathbf{I}_{clean} for clean images. With the above loss functions, our ensemble generative cleaning network learns to iteratively recover adversarial images.

The accumulative fusion network $\mathbf{\Gamma}$ acts as a multi-image restoration network for original image reconstruction. Cascaded with the generative cleaning network \mathbf{G} , it will guide the training of \mathbf{G} and feedback loop network using back propagation of gradients from its own network, aiming to minimize the above loss function. In our design, during the adversarial learning process, the target classifier \mathbf{C} is called to determine if the recovered image \hat{X}_k is clean or not, as illustrated in Figure 2. The output of $\mathbf{\Gamma}$ is fed back to itself as the input to enhance the next round of fusion.

4. Experimental Results

In this section, we implement and evaluate our EGC-FL defense method and compare its performance with state-of-the-art defense methods under a wide variety of attacks, with both white-box and black-box attack modes.

4.1. Experimental Settings

Our experiments are implemented on the Pytorch platform [28]. Our proposed method is implemented on the AdverTorch [7] in both white and black-box attack modes, including the BPDA attack [2]. We choose the CIFAR-10 and SVHN (Street View House Number) datasets for performance evaluations and comparisons since most recent papers reported results on these two datasets. The CIFAR-10 dataset consists of 60,000 images in 10 classes of size 32×32 . The Street View House Numbers (SVHN) dataset [23] has about 200K images of street numbers. For each of these two datasets, a classifier is independently trained on its training set, and the test set is used for evaluations.

4.2. Results on the CIFAR-10 Dataset

We compare the performance of our defense method with state-of-the-art methods developed in the literature under five different white-box attacks: (1) FGS attack [10],

Table 2. BPDA attack results on CIFAR-10 dataset. Results with * are achieved with additional adversarial training.

Defense Methods	Accuracy
Thermometer Encodings (TE) [3]	0.00%*
Stochastic Activation Pruning (SAP) [6]	0.00%
Local Intrinsic Dimensionality (LID) [19]	5.00%
PixelDefend [30]	9.00%*
Cascade Adv. Training ($L_\infty=0.015$) [22]	15.00%
PGD Adv. Training [20]	47.00%*
Sparse Transformation Layer (STL) [31]	42.00%*
Our Method	85.77%*
Gain	+38.77%

Table 3. Performance of our method against black-box attacks on CIFAR-10 ($\epsilon = 8/256$).

Defense Methods	No Attack	FGS	PGD
<i>No Defense</i>	94.38%	63.21%	38.71%
Adv. PGD [33]	83.50%	57.73%	55.72%
Adv. Network [35]	91.32%	77.23%	74.04%
Our Method	91.65%	79.09%	82.78%
Gain		+1.86%	+8.74%

(2) PGD attack [20], (3) BIM attack [18], (4) C&W attack [5], and (5) BPDA attack [2]. Following [17] and [35], the white-box attackers generate adversarial perturbations within a range of $\epsilon = 8/255$. In addition, we set the step size of attackers to be $\epsilon = 1/255$ with 10 attack iterations as the baseline setting.

We generate the perturbed images for training using PGD attacks and tested for all attack methods. During training, we set the iteration number $K = 3$. The perturbed images are used as the input, passing through our EGC-FL network for 3 iterations. But, during test, K is flexible. In our white-box attack experiments, we unfold the feedback loops so that the attacker has full access to the end-to-end defense network, including the number of iterations.

Table 4. Performance of our method against white-box attacks on SVHN ($\epsilon = 12/256$).

Defense Methods	No Attack	FGS	PGD
<i>No Defense</i>	96.21%	50.36%	0.15%
M-PGD [20]	96.21%	(NA)	44.40%
ALP [17]	96.20%	(NA)	46.90%
Adv. PGD [33]	87.45%	55.94%	42.96%
Adv. Network [35]	96.21%	91.51%	37.97%
Our Method	94.00%	94.10%	76.67%
Gain		+2.59%	+29.77%

(1) Defending against white-box attacks. Table 1 shows image classification accuracy with 6 defense methods: (1) Label Smoothing [36], (2) Feature Squeezing [41], (3) PixelDefend [30], (4) Adversarial Network [35], (5) the PNI (Parametric Noise Injection) method [14], and (6) the STL (Sparse Transformation Layer) method [31]. The second column shows the classification accuracy when the input images are all clean. We can see that some methods, such as the PixelDefend [30], Feature Squeezing [41], and PNI [14], degrade the classification accuracy of clean images. This implies that their defense methods have caused significant damages to the original images, or they cannot accurately tell if the input image is clean or being attacked. Since our method has a strong reconstruction capacity, the ensemble of reconstructed images still preserve the useful information. The rest four columns list the final image classification accuracy with different defense methods. For all of these four attacks, our methods significantly outperforms existing methods. For example, for the powerful PGD attack, our method outperforms the Adv. Network and the PNI method by more than 39%.

(2) Defending against the BPDA attack. The Backward Pass Differentiable Approximation (BPDA) [2] attack is very challenging to defend since it can iteratively strengthen the adversarial examples using gradient approximation according to the defense mechanism. BPDA also targets defenses in which the gradient does not optimize the loss. This is the case for our method since the transformed deadzone layer is non-differentiable. Table 2 summarizes the defense results of our algorithm in comparison with other seven methods: (1) Thermometer Encodings (TE) [3], (2) Stochastic Activation Pruning (SAP) [6], (3) Local Intrinsic Dimensionality (LID) [19], (4) PixelDefend [30], (5) Cascade Adversarial Training [22], (6) PGD Adversarial Training [20], and (7) Sparse Transformation Layer (STL) [31]. We choose these methods for comparison since the original BPDA paper [2] has reported results of these methods. We can see that our EGC-FL network is much more robust than other defense methods on the CIFAR-10 dataset, outperforming the second best by more than 38%.

(3) Defending against black-box attacks. We generate the black-box adversarial examples using FGS and PGD attacks with a substitute model [25]. The substitute model is trained in the same way as the target classifier with a ResNet-34 network [12] structure. Table 3 shows the performance of our defense mechanism under back-box attacks on the CIFAR-10 dataset. The adversarial examples are constructed with $\epsilon = 8/256$ under the substitute model. We observe that the target classifier is much less sensitive to adversarial examples generated by FGS and PGD black-box attacks than the white-box ones. But the powerful PGD attack is still able to decrease the overall classification accuracy to a very low level, 38.71%. We compare our method with the Adversarial PGD [20] and Adversarial Network [35] methods. We include these two because they are the only ones that provide performance results on CIFAR-10 with black-box attacks. From the Table 3, we can see our method improves the accuracy by 8.74% over the state-of-the-art Adversarial Network method for the PGD attack.

4.3. Results on the SVHN Dataset.

We evaluate our EGC-FL method on the SVHN dataset with comparison with four state-of-the-art defense methods: (1) M-PGD (Mixed-minibatch PGD) [20], (2) ALP (Adversarial Logit Pairing) [17], (3) Adversarial PGD [33], and (4) Adversarial Network [35]. For the SVHN dataset, as in the existing methods [17, 35], we used the Resnet-18 [12] for the target classifier. The average classification accuracy is 96.21%. We use the same parameters as in [17] for the PGD attack with a total magnitude of $\epsilon = 0.05$ (12/255). Within each single step, the perturbation magnitude is set to be $\epsilon = 0.01$ (3/255) and 10 iterative steps are used.

(1) Defending against white-box attacks. Table 4 summarizes the experimental results and performance comparisons with those four existing defense methods. We can see that on this dataset the PGD attack is able to decrease the overall classification accuracy to an extremely low level, 0.15%. Our algorithm outperforms existing methods by a very large margin. For example, for the PGD attack, our algorithm outperforms the second best ALP [17] algorithm by more than 29%. With the FGS attacks, the iterative cleaning process will produce image versions with more diversity than the clean image without attack noise. This helps reconstruct the original image.

(2) Defending against black-box attacks. We also perform experiments of defending black-box attacks on the SVHN dataset. Table 5 summarizes our experimental results with the powerful PGD attack and provides the comparison with those four methods. We can see that our approach outperforms other methods by 2.25% for the FGS attacks and 5.37% for the PGD attacks. From the above results, we can see that our proposed method is particularly effective for defense against the strong attacks, for example,

Table 5. Performance of our method against black-box attacks on SVHN ($\epsilon = 12/256$).

Defense Methods	No Attack	FGS	PGD
<i>No Defense</i>	96.21%	69.91%	67.66%
M-PGD [20]	96.21%	(NA)	55.40%
ALP [17]	96.20%	(NA)	56.20%
Adv. PGD [33]	87.45%	87.41%	83.23%
Adv. Network [35]	96.21%	91.48%	81.68%
Our Method	94.00%	94.03%	88.60%
Gain		+2.55%	+5.37%

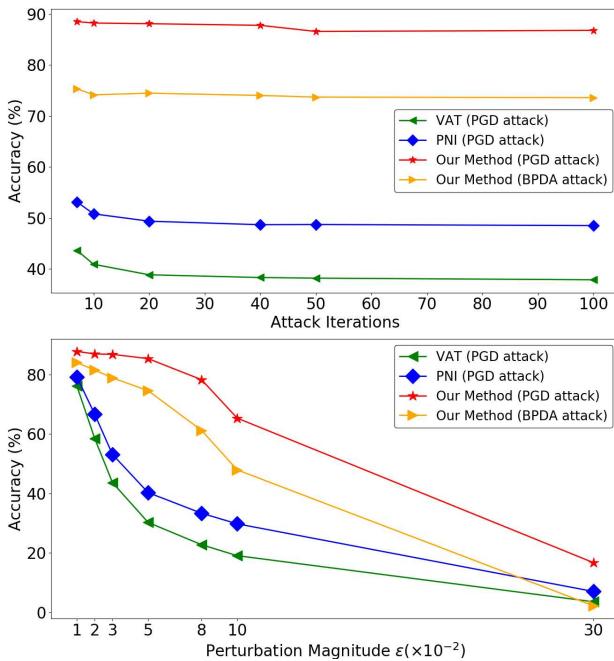


Figure 5. The perturbed-data accuracy of ResNet-18 under adversarial attack (Top) versus number of attack iteration, and (Bottom) versus perturbation magnitude (under L_∞) on CIFAR-10 dataset.

the PGD attacks with large iteration steps and noise magnitude.

4.4. Ablation Studies and Algorithm Analysis

In this section, we provide in-depth ablation study results of our algorithm to further understand its capability.

(1) Defense against large-iteration and large-epsilon attacks. Figure 5 (Top) shows the performance results under large-iteration PGD and BPDA attacks. We can see that the large-iteration PGD attack significantly degrades the accuracy of the Vanilla Adversary Training method (VAT) [20] and the PNI (Parametric Noise Injection) method [14], as well as our method. But, our method significantly outperforms the other two. In both cases, the perturbed-data accuracy starts saturating without further drop when

Table 6. Performance of our method with feedback loops under adversarial attacks on CIFAR-10 dataset.

Attack Method	Gen1	Gen2	Gen3	Gen4
FGS	57.64%	78.04%	78.15%	78.31%
PGD	78.46%	85.36%	86.25%	86.55%
BPDA	19.40%	79.12%	79.28%	79.79%

$N_{step} \geq 50$. In Figure 5 (Top), we also include the performance results of our method under large-iteration BPDA attacks. We set the adversarial perturbations within a range of $\epsilon = 12/255$ with 10 attack iterations as the baseline setting. This result is not reported by other methods so we could not include them for comparison. We can see that the BPDA attack is much more powerful. But, our algorithm can still survive large-iteration BPDA attacks and largely maintain the defense performance.

Figure 5 (bottom) shows comparison results against attacks with large perturbation magnitude. We can see that our method significantly outperforms the VAT and PNI defense methods even when the magnitude of adversarial noise is increased to $\epsilon = 0.3$ under the PGD attack. We also include the performance of our method under large- ϵ BPDA attacks. We can see that our method is robust under very powerful attacks of large magnitudes.

(2) Analyze the impact of feedback loops. We notice that the feedback loop network plays an important role in the defense. In our method, the key parameter controlling the image quality is the number of feedback loop k . We gradually increase k and explore classification accuracy of the fused image. Table 6 shows the performance (classification accuracy after defense) of our method on the CIFAR-10 dataset with various attacks. We denote Gen_k as the number of feedback loops. We can see that the feedback loops within the range of 3 or 4 yields the best performance. One feedback loop does not provide efficient defense since the EGC-FL network is not able to fully destroy the attack noise pattern and restore useful information. Once the key features in the original image have been reconstructed, the classification accuracy will be stable and maintain the highest performance, although the image quality may get even better with accumulative fusion. In Figure 6, we show sample images from the CIFAR-10 when our method is applied. The first column is the clean image without attacks. The second column is attacked image. The third to last columns are reconstructed images of 4 generations by our EGC-FL method. We can see that our algorithm is able to remove the attack noise and largely recover the original image content.

(3) In-depth analysis of major algorithm components. In the following ablation studies, we perform in-depth analysis of major components of our EGC-FL algorithm, which includes the transform, deadzone, and the EGC network with feedback loops. In Table 7, the first row shows the

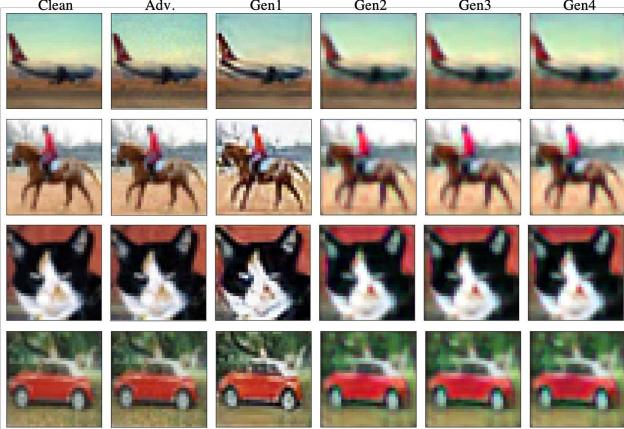


Figure 6. Adversarial images and their fused image produced by our method.

Table 7. Performance analysis of algorithm components.

Defense Methods	FGS	PGD	BPDA
Our Method (Full Alg.)	88.51%	88.04%	85.77%
- Without Transform	79.32%	79.35%	79.62%
- Without Feedback	77.12%	78.46%	19.37%

classification accuracy of images after defense with our proposed EGC-FL method (full algorithm) on the CIFAR-10 dataset with FGS, PGD, and BPDA attacks. The second row shows results without the transform. We can see that the accuracy drops about 7-9%. The transform module is important because it can help protecting the original content from being damaged by the deadzone activation function $\delta(x)$ by aggregating the energy of the original image into a small number of large transform coefficients. The third row shows the results without the feedback loop. We can see that it drops the accuracy by 10-11% under the FGS and PGD attacks. For the powerful BPDA attack, the drop is very dramatic, about 66%. With multiple feedback loops for progressive attack noise removal and original image reconstruction, it can significantly improve the defense performance, especially under powerful BPDA attacks.

(4) Visualizing the defense process. Network defense is essentially a denosing process of the feature maps. To further understand how the the proposed EGC-FL method works, we visualize the feature maps of original, attacked, and EGC-FL cleaned images. We use the feature map from the activation layer, the third from the last layer in the network. Figure 7 shows two examples. In the first example, the first row is the original image (classified into terrapin), its gradient-weighted class activation heatmap, and the heatmap overlaid on the original image. The heatmap shows which parts of the original image the classification network is paying attention to. The second row shows the

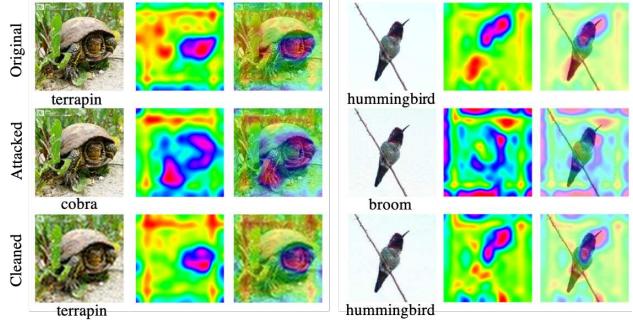


Figure 7. Each pair of examples are feature maps corresponding to clean images (top), to their adversarial perturbed images (middle) and to their reconstructed images (bottom).

attacked image (being classified into cobra), heatmap, and the heatmap overlaid on the attacked image. We can see that the feature map is very noisy and the heatmap is distorted. The third row shows the EGC-cleaned images. We can see that both the feature map and heatmaps have been largely restored.

5. Conclusion

We have developed a new method for defending deep neural networks against adversarial attacks based on the EGC-FL network. This network is able to recover the original image while cleaning up the residual attack noise. We introduced a transformed deadzone layer into the defense network, which consists of an orthonormal transform and a deadzone-based activation function, to destroy the sophisticated noise pattern of adversarial attacks. By constructing a generative cleaning network with a feedback loop, we are able to generate an ensemble of diverse estimations of the original clean image. We then learned a network to fuse this set of diverse estimation images together to restore the original image. Our extensive experimental results demonstrated that our approach outperforms the state-of-art methods by large margins in both white-box and black-box attacks. Our ablation studies have demonstrated that the major components of our method, the transformed deadzone layer and the ensemble generative cleaning network with feedback loops, are both critical, contributing significantly to the overall performance.

Acknowledgement

This work was supported in part by National Science Foundation under grants 1647213 and 1646065. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, pages 274–283, 2018.

[3] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[4] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017.

[6] Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.

[7] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

[11] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

[14] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019.

[15] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. *CoRR*, abs/1811.12673, 2019.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[17] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing, 2018.

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[19] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality, 2018.

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

[21] Dongyu Meng and Hao 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.

[22] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. In *International Conference on Learning Representations*, 2018.

[23] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

[24] Nicolas Papernot and Patrick McDaniel. On the effectiveness of defensive distillation. *arXiv preprint arXiv:1607.05113*, 2016.

[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[26] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. pages 372–387, 2016.

[27] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[29] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[30] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging genera-

tive models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

- [31] Bo Sun, Nian-Hsuan Tsai, Fangchen Liu, Ronald Yu, and Hao Su. Adversarial defense by stratified convolutional sparse coding. *CVPR*, Jan 2019.
- [32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- [33] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.
- [34] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
- [35] Huaxia Wang and Chun-Nam Yu. A direct approach to robust deep learning using adversarial networks. In *International Conference on Learning Representations*, 2019.
- [36] David Warde-Farley. 11 adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, 311, 2016.
- [37] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.
- [38] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.
- [39] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018.
- [40] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity, 2018.
- [41] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.