Adversarial Defense by Restricting In-variance and Co-variance of Representations

Jeevithan Alagurajah
University of Louisiana at Lafayette
Lafayette, Louisiana
C00289092@louisiana.edu

Chee-Hung Henry Chu
University of Louisiana at Lafayette
Lafayette, Louisiana
chu@louisiana.edu

ABSTRACT

Despite high accuracies achieved by deep neural networks (DNNs) in image classification, DNNs have been shown to be highly vulnerable to structured and unstructured perturbations to the input images. Robustness of many existing defense methods for these models suffers greatly when an attacker has full knowledge of the model and can iterate over the model to craft stronger attacks, which is known as white box attacks. We conduct empirical analysis on the representation of DNN under state-of-the-art attacks to find this causes instances to move closer to a false class in representation space when such perturbation is added to the input. This causes the model to make incorrect decisions even when the adversary and clean images are indistinguishable to human perception. Motivated by this observation, we propose a class-wise disentanglement on intermediate representations of DNN. Specifically, we force DNNs to learn same-class representations to be closer and different-class representations to be maximally farther apart. Moreover, we force the representations of clean and noisy data to be closer if it comes from the same class by restricting its variance in representation. In this approach, a DNN is forced to learn decision boundaries that are distinct for each class with clear separation. We observe that this constraint on representations enhance the robustness of learned models even against strongest white-box attacks. Further we evaluate extensively on both white-box and black-box settings and show significant gains in comparison to state-of-the art defenses. (Implementation:https://github.com/Jeevi10/AICR)

CCS CONCEPTS

• **Computing methodologies** → **Hierarchical representations**; **Object recognition**; **Supervised learning by classification**;

KEYWORDS

Deep neural networks(DNN), Representation, Disentanglement

ACM Reference Format:
Jeevithan Alagurajah and Chee-Hung Henry Chu. 2022. Adversarial Defense by Restricting In-variance and Co-variance of Representations. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event,* . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3477314.3507067

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/10.1145/3477314.3507067

1 INTRODUCTION

In recent years Deep Neural Networks (DNNs) have achieved impressive performance in such a wide range of domains as computer vision [13, 14, 16, 30], natural language processing [8, 20], speech recognition [7, 10], and reinforcement learning [4, 34] tasks. However, their performance in image classification degrades drastically under adversarial attacks, where carefully crafted noises can perturb the natural samples such that while they are not distinguishable by a human they are enough to manipulate DNN models to produce incorrect results. This clearly indicates our current learning algorithms do not learn salient visual concepts in a distinctive manner. This raises serious concerns about the accountability of the DNN models, especially when we deploy DNN models in autonomous cars [1], surveillance systems [27], and biometric identifications. Therefore, designing Artificial Intelligence (AI) systems that are robust and generalizable against adversarial attacks is imperative in the current world and remains an open question.

In order to defend against the attacks, numerous methods have been proposed in recent years. In the literature, they can be broadly categorized into reactive defense and proactive defense. Reactive defense methods include modifying inputs, transforming inputs to take counter measurements of attackers' strategy [6, 21, 25, 36]. However, they each have their limitation in applications. For example, counter strategies can only be immune to currently known counter attacks. Proactive defense includes modifying current model parameters, modifying network architecture, training model with enhanced objective functions and training with adversarial examples. Proactive methods are widely used in defense as they provide better robustness against white-box attacks.

We propose a novel and enhanced adversarial training framework with two additional objectives in the model. First, we use an attractive and repulsive mechanism at the hierarchical levels of representation to build a more robust classifier. The proposed objective enforces maximum separations between different class samples. In other words, this helps to attract samples from the same class together and repulse samples from different class farther away. Second, to ensure that there is little variance between adversarial and clean images from the same class examples, we encourage maximum correlation and minimum redundancy in intermediate representations of adversarial and clean images that belong to the same class.

The objective of maximizing separations has two clear merits. First, a maximum separation between classes would create a naturally robust model that is immune to noise present in the data and it helps to achieve better generalization. Second, during the adversary

attacks, the adverser cannot fool the model within a given restricted budget. In other words, we ensure that two same class samples, which are visually similar, must be projected to the same region in the multidimensional space such that samples cannot be shifted without notable changes in them. Therefore, we must ensure their representations are separated maximally across multi-layers of the network and also these representations are highly correlated with-in the same class with less redundancy.

To achieve this, we propose a novel Adversarial In-variance and Covariance Restriction (AICR) loss to enforce the model to have intra-class compactness and inter-class maximum separability on multiple levels of representations in DNNs. In order to classify a sample, convolutional layers are used to extract discriminative features similar to traditional CNN networks. On top of that, we use the centroid to represent each class in an intermediate layer in the dataset as illustrated in Fig. 1. The classification is done by assigning the nearest centroid in the feature space. To enforce high correlation between adversary and clean images from the same class, its intermediate representation is projected through a generative network; subsequently the output of the network is used to maximize the correlation and minimize the redundancy. On one hand, this acts like a regularizer, which can prevent overfitting; on the other hand, it improves intra-class compactness in the feature representation.

2 RELATED WORK

Finding an effective attack to fool the DNNs and developing defense methods for such attacks have recently gained significant attention among researchers. Deep learning algorithms are vulnerable to the presence of carefully crafted adversarial perturbations. Several defense algorithms have been proposed in the literature to counter such adversarial attacks. These attacks methods can be categorized into two main strategies. First is the so-called "input transformation" in which an input image is transformed or pre-processed prior to the inference in order to mitigate the effect of adversarial perturbation [11, 36]. A second approach is to modify the model in order to defend against forthcoming adversarial examples [2, 24, 38]. Adversarial training is an intuitive method in this regard by including both clean and adversarial samples to jointly train the model [17, 31]. Madry et al. [22] claimed that PGD adversarial training prevents first order adversarial attack as PGD attack is a universal first order attack, the model trained with PGD adversarial samples are robust to first order attacks. However, an adversarial trained model is still vulnerable to black-box attacks. To overcome this [33] introduced diversity in adversarial training by incorporating multiple retrained models. Kannan et al. [15] proposed to encourage similarity between a learned logits pair between clean samples and its counter adversarial samples in the logits space. Chengzhi et al. [23] proposed a learning objective metric which forced the model to map clean and its counter adversarial parts to be closer and its negative pair to be farther apart. Similarly, Mustafa at el. [25] proposed a proximity objective to force the clean and its counter samples to be closer and the negative samples to be further apart; unlike others they proposed this objective in multi-layers to ensure feature discrimination at multiple levels. However, this did not lead to local compactness of the set of the image set and corresponding adversarial images.

Redundancy reduction has become the standard operation in the machine learning approach [32, 39]. Recent literature reported redundancy information mitigation in self-supervised learning tasks by minimizing the variance across all the identical samples from the same class. Inspired by [29], it has been used recently in a self-supervised learning task for preventing the dimensions of the representations from encoding the same information. To provide adversarial robustness [32] maximizes the mutual information of representations and input to have concise information of input. Ref. [9] introduces a whitening operation via Cholesky decomposition and scatters the representation on the unit sphere; nevertheless it requires inverting the covariance matrix of the embeddings and is therefore computationally expensive and often unstable.

Our method is fundamentally different from [25] in that the proposed multi-layered architecture reduces redundant information in hierarchical layers while enforcing maximum separation. Information reduction in a hierarchical architecture has not been explored in the adversarial robustness literature.

3 METHODOLOGY

3.1 Notations

Let K be the number of classes of a given data set distribution $\mathcal{D}$ and N be the number of examples in the data set. For an image classification task, we formulate a deep neural network as $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $\boldsymbol{\theta}$ is the trainable parameters and $\boldsymbol{x}$ is the input image. The DNN outputs a feature representation $\boldsymbol{h}_x \in \mathbb{R}^d$ for input $\boldsymbol{x}$ which is then used for classification in a multiclass classifier $\boldsymbol{Z} = [\boldsymbol{z}_{\boldsymbol{k}}] \in \mathbb{R}^{d\times K}$. Here $k = (1, \cdots, K)$. To train the model we minimize $\boldsymbol{\theta}$ and $\boldsymbol{Z}$ in the given objective function. The output of the model is

$$f_k(\boldsymbol{x}) = \text{softmax}(\boldsymbol{z}_{\boldsymbol{k}} \cdot \boldsymbol{h}_{\boldsymbol{x}} + \boldsymbol{b}_{\boldsymbol{k}}), \tag{1}$$

where $f_k(\boldsymbol{x})$ denotes the probability of $\boldsymbol{x}$ being in class k and b_k denotes the bias term of the particular class. For a single input-label pair $(\boldsymbol{x}, \boldsymbol{y})$ the softmax Cross-Entropy (CE) loss is defined by

$$\boldsymbol{\mathcal{L}}_{CE}(\boldsymbol{x}, \boldsymbol{y}) = -\log\left[f_{k=y}(\boldsymbol{x})\right] \tag{2}$$

We define the function $\text{softmax}(\boldsymbol{v}) : \mathbb{R}^K \implies \mathbb{R}^K$ as

$$\text{softmax}(\boldsymbol{v}_i) = \exp(\boldsymbol{v}_i)/\sum_{l=1}^{K}\exp(\boldsymbol{v}_l),$$

$i \in [\boldsymbol{k}]$, where $\boldsymbol{v}$, is referred to as logits.

Adversarial objective: The goal of the adversary is to fool the trained DNN $\boldsymbol{\mathcal{F}_{\theta}}$ to make wrong predictions. Adversaries seek to attain this goal by adding visually imperceptible noise to the input image. Therefore, the adversarial objective is

$$\max_{\delta} \boldsymbol{\mathcal{L}}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}), \text{ s.t., } ||\boldsymbol{\delta}||_p \leq \boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{x}$ is a given input sample and $\boldsymbol{y}$ is the ground truth label for the given sample. Here, $\mathcal{L}(\cdot)$ is the loss function that the target DNN is trained on, $||\cdot||_p$ is the p-norm such as $\ell_1, \ell_2, \ell_\infty$. Surprisingly, it has been shown that one can choose a $\boldsymbol{\delta}$ with the very small norm with an available perturbation budget $\boldsymbol{\epsilon}$ to completely change the prediction of DNN.

Defense objective: Defense algorithm's aim is to prevent the model from producing a wrong prediction when the input is perturbed with

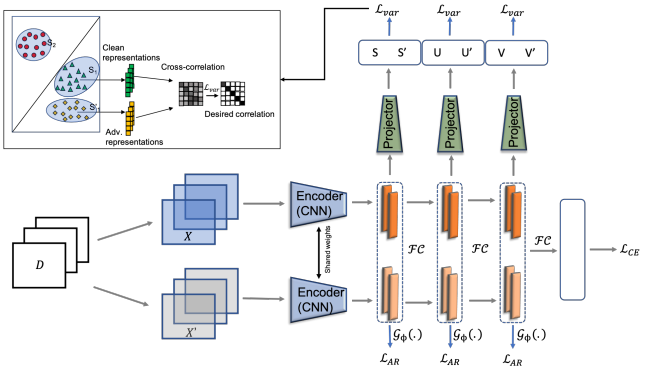


Figure 1: The architecture of the proposed supervised training method to learn jointly from $\mathcal{L}_{AR}$, $\mathcal{L}_{CE}$, and $\mathcal{L}_{var}$. Projector and $\mathcal{G}_\phi$ are auxiliary mapping functions for $\mathcal{L}_{var}$ and $\mathcal{L}_{AR}$, respectively, to create low dimension features from convolutional layers.

visually imperceptible noise. This can be done by modifying an existing DNN model to be more resistant to attacks. Therefore, defense algorithms minimize the empirical risk present during adversarial influence. This can be formulated as

$$\min_{\theta} \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} \max_{\delta} \boldsymbol{\mathcal{L}}(\boldsymbol{x}+\boldsymbol{\delta}, \boldsymbol{y}), \text{ s.t., } ||\boldsymbol{\delta}||_p \leq \boldsymbol{\epsilon} \tag{4}$$

This is a min-max optimization of adversarial training.

3.1.1 Motivation: Predictive behaviour of CE loss in adversarial settings. Previous studies show that naturally trained DNN models are susceptible to adversarial attacks, even when samples are only slightly different from the clean samples. This vulnerability in DNN models motivates researchers to develop numerous methods for attacks. Here, we conduct an empirical analysis of the predictive behaviour of CE loss using a very common loss function that has been in practice for many years. We consider a clean test sample $(\boldsymbol{x}, \boldsymbol{y})$ and seek $\boldsymbol{\delta}$ to misclassify $\boldsymbol{x}+\boldsymbol{\delta}$. A sample is misclassified when

$$\hat{y} = \arg\max_{k} (f_k(\boldsymbol{x}+\boldsymbol{\delta})), \text{ s.t., } ||\boldsymbol{\delta}||_p \leq \boldsymbol{\epsilon}, \tag{5}$$

where $\hat{y} \neq y$; i.e., the attack succeeds when Eq. 5 holds. Note that, $y = \arg\max_k (\boldsymbol{z_k} \cdot \boldsymbol{h_x})$ This suggests $\hat{y}$ is the second most probable class for the given x. This is equivalent to

$$\arg\max_{k}{}_{\#2}(f_k(\boldsymbol{x})) = \arg\max_{k} (f_k(\boldsymbol{x}+\boldsymbol{\delta})), \text{ s.t., } ||\boldsymbol{\delta}||_p \leq \boldsymbol{\epsilon} \tag{6}$$

Here $\arg\max_{\#2}$ denotes representing the second largest value. Assuming that the DNN model is Lipschitz continuous [12], then we have an inequality,

$$||f(\boldsymbol{x}+\boldsymbol{\delta}) - f(\boldsymbol{x}))||_p \leq l||(\boldsymbol{x}+\boldsymbol{\delta}) - \boldsymbol{x}||_p = l||\boldsymbol{\delta}||_p \tag{7}$$

where l is the Lipschitz constant and $f(\cdot) = (f_1(\cdot), ..., f_K(\cdot))$. The Lipschitz continuity implies the change in output is bounded by the change in inputs that is the small perturbation in adversarial attacks. To map $\boldsymbol{x}+\boldsymbol{\delta}$ and $\boldsymbol{x}$ to the same class, the minimum distance between class centers should be at least $l||\boldsymbol{\delta}||_p$. However, this margin constraint is not naturally imposed in the CE loss. Therefore finding visually imperceptible perturbation with an allowed budget is feasible. Inspired by this margin constraint, we propose a defense strategy of the maximal separation objective for deep neural networks.

3.2 Proposed Objectives

Attract-Repulsive (AR): We represent each class with its trainable centroid. Namely, we define centers as $\boldsymbol{c}_i$ where $i \in \{1, ..., K\}$ represents the index of classes. For a training example $(\boldsymbol{x}, \boldsymbol{y})$, a corresponding adversarial example $\boldsymbol{x'}$ exists close proximity in the data space. Let $\boldsymbol{c}_y$ and $\boldsymbol{c}_j$ denote the estimated center of the correct class and the competitive class center, respectively. The probability of $\boldsymbol{x}$ belonging to $\boldsymbol{c}_y$ can be measured from the distance between them as follows,

$$\mathcal{L}_{AR}(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}) = ||\boldsymbol{h_x} - \boldsymbol{c}_y||_2 + ||\boldsymbol{h_{x'}} - \boldsymbol{c}_y||_2 - \frac{1}{K-1} \sum_{j\neq y} \left(||\boldsymbol{h_x} - \boldsymbol{c}_j||_2 + ||\boldsymbol{h_{x'}} - \boldsymbol{c}_j||_2 \right) \tag{8}$$

During the test time, the feature similarity distance is measured across all the centroids and given test samples and it is assigned to the closest centroid as its class label:

$$\hat{y} = \arg\min_{i} ||\boldsymbol{h_x} - \boldsymbol{c}_j||_2 \tag{9}$$

This learning rule is similar to the methods in [37], but it is different in some important aspects: (1) the centroids are not fixed as the mean of training examples; rather, they are learned in a generative approach; (2) the class samples and their adversarial counters are explicitly attracted towards to respective class centers.

Variance loss (var): The loss function defined above is used for the measurement of classification accuracy. By minimizing this loss, we can train a robust model to classify instances correctly. In other words, this loss function enforces intra-class compactness and inter-class separability with maximum separation while learning class centers. However, an adversarial example and its clean example in an intermediate layer have significantly divergent representations. Specifically, visually similar classes in data space tend to have similar salient features, therefore they are more likely to have overlapping representations in an immediate layer. On the other hand, the divergence in the representation between clean and adversarial examples narrows the separation between visually similar classes. This will become a hindrance for finding maximum separation between these two classes, hence making the model susceptible to adversarial attacks.

To address this above mentioned issue we provide an objective function that enforces local compactness between images and their adversarial counterpart by maximizing the intermediate representation similarity and minimizing the redundancy; i.e. it removes redundant information in the input data by decorrelating the clean and its adversarial variables and making the variance of all variables identical. Therefore the clean and its counter adversarial example will have minimal and sufficient representation for the classification task. The following generator network G_Ψ maps the intermediate layer into a different feature space where we enforce the local compactness as follows,

$$\mathcal{L}_{var} = \sum_{i} (1 - \boldsymbol{Q}_{ii})^2 + \lambda \sum_{i} \sum_{j\neq i} \boldsymbol{Q}_{ij}^2 \tag{10}$$

where λ is the trade-off parameter of invariant (diagonal) and redundancy (off-diagonal) terms of the matrix and where

$$\boldsymbol{Q} = \frac{\boldsymbol{G_\Psi}(\boldsymbol{h_x}) \times \boldsymbol{G_\Psi}(\boldsymbol{h_{x'}})}{\sqrt{\boldsymbol{G_\Psi}(\boldsymbol{h_x})}\sqrt{\boldsymbol{G_\Psi}(\boldsymbol{h_{x'}})}} \tag{11}$$

Here, $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$, where d is the dimension of the output of $\boldsymbol{G_\Psi}$. Minimizing this objective correlates clean and adversary counterparts and encourages to have non-redundant information while lies closer in intermediate layers. This allows the centers to have contrasting representations and promotes maximum separation between classes. Here, the $\boldsymbol{G_\Psi}$ mapping is learned in an end-to-end manner.

Adversarial Invariant and Covariance Restriction loss (AICR): By combining all the loss functions above, we define our final loss function to train our model as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y}) = \sum_{i}^{N} (\mathcal{L}_{CE}(\boldsymbol{x}_i, \boldsymbol{x}'_i, \boldsymbol{y}_i) + \mathcal{L}^{'}(\boldsymbol{x}_i, \boldsymbol{x}'_i, \boldsymbol{y}_i)). \tag{12}$$

Here

$$\mathcal{L}^{'}(\boldsymbol{x}_i, \boldsymbol{x}'_i, \boldsymbol{y}_i) = \sum_{l}^{n} (\mathcal{L}_{AR}(\boldsymbol{h}^l_i, \boldsymbol{h'}^l_i, \boldsymbol{y}_i) + \alpha \times \mathcal{L}_{var}(\boldsymbol{h}^l_i, \boldsymbol{h'}^l_i, \boldsymbol{y}_i))$$

such that

$$\boldsymbol{h}^l = \mathcal{G}^l_{\boldsymbol{\phi}}(\mathcal{F}^l_{\boldsymbol{\theta}}(\boldsymbol{x})) \quad \text{and} \quad \boldsymbol{h'}^l = \mathcal{G}^l_{\boldsymbol{\phi}}(\mathcal{F}^l_{\boldsymbol{\theta}}(\boldsymbol{x'}))$$

where α is the regularizing term for contrastive centroid loss, $\mathcal{G}_\phi$ is an auxiliary function that maps intermediate layers to lower dimension output, and n denotes the number of layers. This loss function encourages the same classes to be mapped closer and different classes to be mapped farther from each other by a large margin.

4 EXPERIMENTS

In this section, we present an evaluation of our proposed method to defend state-of-the-art attacks including attacks in Ref. [26].

Datasets, Models and Hyperparameters: We evaluate our proposed method using five data sets: MNIST, Fashion-MNIST (F-MNIST), CIFAR-10, CIFAR-100, and Street-View House Numbers (SVHN). We evaluate MNIST and F-MNIST in the variant of CNN6-net [35] variant model; for CIFAR-10 and CIFAR-100 we use the Resnet-110 [14] model (see Table 1). We scale all pixel values to $[0, 1]$ following the preprocessing procedure in [28]. All the models are trained with batch size 256, using the Adam optimizer with an initial learning rate of 0.01 while SGD optimizer is used to update the center of the classes, and tradeoff parameters $\alpha = 1$ and $\lambda = 0.005$ are used. In training, we first trained the model T epochs, where $T = 50$ for F-MNIST and $T = 150$ for other data sets, using the CE loss to ensure we have good initial representations of classes to train our proposed objective. Then, we train the model using Eq. 12 to another $T_p = 300$ epochs. We use the learning rate of 0.01($\times$0.1 at $T_p = 200, 250$). We use a 1024-unit single linear network as the projector. Further training details are summarized in Algorithm 1.

Algorithm 1: Adversarial Invariant and Covariance Restriction (AICR)

Input: Classifier $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x'})$, training data $\{\boldsymbol{x}, \boldsymbol{x'}\}$, ground truth labels $\{\boldsymbol{y}\}$, trainable parameters $\boldsymbol{\theta}$, trainable class centroids $\{c_j : j \in [1, K]\}$, trainable projectors $\boldsymbol{G}_\Psi$, pertubation budget ϵ, epochs T, T_p, number of auxiliary layers n.
Output: Trained parameters $\boldsymbol{\theta}$
Initialize $\boldsymbol{\theta}$ in DNN and Ψ projectors
for $t = 0$ *to* T **do**
  Converge Cross-entropy objective, $\boldsymbol{\theta}$: $\arg\min_{\boldsymbol{\theta}} \mathcal{L}_{CE}$
end
for $t = 0$ *to* T_p **do**
  Computes loss $\mathcal{L}$ in Eq. 12
  Compute gradient w.r.t $\boldsymbol{\theta}$,$\boldsymbol{\Psi}$ and $\boldsymbol{x}$ as $\nabla_\theta \mathcal{L}(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y})$,$\nabla_\Psi \mathcal{L}(\boldsymbol{x}, \boldsymbol{x'}, \boldsymbol{y})$ and $\nabla_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$, respectively.
  Update the model parameters, $\theta := \arg\min_\theta \mathcal{L}$
  Update the projector parameters, $\Psi := \arg\min_\Psi \mathcal{L}$
  Update class centers $c_j \forall n$
  if *Adversarial Training:* **then**
    $\boldsymbol{x'} = \boldsymbol{x} + \epsilon \cdot sign(\nabla_x L(x, y))$
  else
    $\boldsymbol{x'} = \boldsymbol{x} + \epsilon \cdot sign(\mathcal{N}(0, 1))$
end
return $\boldsymbol{\theta}$

Table 1: Network architecture: CNN-6 (MNIST/F-MNIST) and Resnet-110 (CIFAR-10,100 and SVHN), here # means number of layer.

#	CNN6-net	Resnet-110
1	[Conv (32, 5, 5), Relu, Pool (2, 2)] $\times 2$	Conv (16, 3, 3) + BN, Relu
2	[Conv (64, 5, 5), Relu, Pool (2, 2)] $\times 2$	[Conv (16, 1, 1) + BN, Conv (16, 3, 3) + BN, Conv (64, 1, 1) + BN] $\times 12$
3	[Conv (128, 5, 5), Relu, Pool (2, 2)] $\times 2$, Flattent $\rightarrow \mathcal{G}_\phi \rightarrow \mathcal{L}_{AICR}$	[Conv (32, 1, 1) + BN, Conv (32, 3, 3) + BN, Conv (128, 1, 1) + BN] $\times 12$, Flatten $\rightarrow \mathcal{G}_\phi \rightarrow \mathcal{L}_{AICR}$
4	FC (512) $\rightarrow \mathcal{L}_{AICR}$	[Conv (128, 1, 1) + *BN*, Conv (128, 3, 3) + BN, Conv (256, 1, 1) + BN] $\times 12$, Flatten $\rightarrow \mathcal{G}_\phi \rightarrow \mathcal{L}_{AICR}$
5	FC (64) $\rightarrow \mathcal{L}_{AICR}$	FC (1024) $\rightarrow \mathcal{L}_{AICR}$
6	FC (10) $\rightarrow \mathcal{L}_{CE}$	FC (10/100) $\rightarrow \mathcal{L}_{CE}$

4.1 Results and Analysis

Performance on white-box vs black-box attacks: Following the attack settings in [5], we crafted adversarial examples in a non-targeted way with respect to allowed perturbation ϵ for gradient based attacks, i.e., FGSM, BIM, PGD, MIM. The number of iterations for BIM, MIM, PGD are set to 10 with a step size of $\epsilon/10$. For parameters of optimization-based attack C&W, the maximum iteration steps is set to 100, with a learning rate of 0.001, and confidence set to 0. Results in Table 2 shows the performance of the proposed framework for the different attacks described in Mustafa et al [26]. We report model robustness under random noise training and adversarial training for

standard perturbation *i.e.*$\epsilon = 0.3$ for F-MNIST and $\epsilon = 0.03$ for CIFAR-10/100 and SVHN datasets.

To demonstrate the effectiveness of our proposed defense under *black-box* settings, we generate adversarial samples using Lenet [18] for F-MNIST and VGG11 for CIFAR-10/100 and SVHN and feed them to the model trained using our framework. Results in Table 2 show *black-box* settings have negligible performance degradation on our model. For example, on the MNIST data set, our model's accuracy dropped $\sim 2\%$ from clean images to adversarial images even under the strongest PGD ($\epsilon = 0.3$) attack. Likewise, the CIFAR-10 dataset has 92.4% accuracy for clean image and under PDG ($\epsilon = 0.03$) attack our model defense retains 82.8% accuracy.

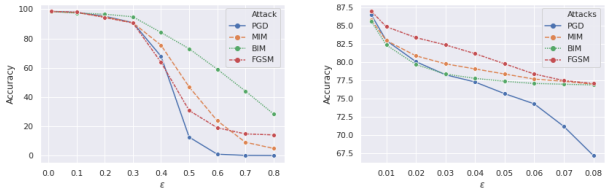
Training: Adversarial training has been successfully shown to enhance performance in many recently proposed methods [26, 28]. We evaluate the effectiveness of adversarial training (AT) with our proposed defense. We train our base model on clean and noisy data which are generated from $\mathcal{N}(0, 1)$. To be consistent with an adversarial attack perturbation budget, we uniformly sample an interval of $[0.1, 0.3]$ for MNIST and F-FMNIST and $[0.01, 0.03]$ for CIFAR-10/100 and SVHN. During adversarial training(AT), we generate samples using FGSM and used the above mentioned intervals and ϵ. Results provided in Table 2 shows our model has significantly enhanced robustness under both *white-box* and *black-box* attacks.

A model can give a false sense of security due to the obfuscated gradients ref [3]. This can be identified if the *black-box* attacks are stronger than *white-box* attacks. In Table 2, the surpassing *black-box* robustness shows that our model does not suffer from obfuscated gradients. We also experimented with imposing this proposed method across different layers and we found Table 1 gives the best results.

Increasing attack budget ϵ decrease the robustness of the defense: On increasing perturbation budget rate of attack success should significantly increase. In other words, increasing distortion in the model should decrease the model robustness. However, we train our base model with Gaussian noise with interval of $[0.1, 0.3]$ for MNIST and $[0.01, 0.03]$ for CIFAR10. Therefore, the our model retained performance without significant loss in accuracy until distortion reach 0.3 and 0.03 for MNIST and CIFAR10, respectively, then model performance decreases monotonically (see Fig. 2)

4.2 Effectiveness of Variance Loss

We first show how the proposed method contributes to adversarial robustness. Recall that our model consists of two components on top of the cross-entropy loss. The term $\mathcal{L}_{var}$ is the redundancy and invariant loss while $\mathcal{L}_{AR}$ forms the Attract-Repulsive loss. We investigate the effectiveness of the proposed $\mathcal{L}_{var}$ loss under black-box settings. Results in Table 3 shows the model trained with $\mathcal{L}_{var}$ retained the highest accuracy than the model that does not consist of $\mathcal{L}_{var}$ for the strongest attacks like PGD. For example, among adversarially trained models 20.2% accuracy is retained when model trained (with adversarial examples) jointly with $\mathcal{L}_{var}$ for PDG attack with $\epsilon = 0.4$ on the MNIST dataset. Likewise, on CIFAR-10 there is a 13.4% relative gain when the model trained (without adversarial examples) with $\mathcal{L}_{var}$ for PGD attack with $\epsilon = 0.4$. A model trained with *variance loss* retains performance even when attacks are much stronger, yet the model does not necessarily have to be trained on adversarial examples. Further, we observe that models perform very similarly when there is a weaker attack. This shows that *variance loss* mitigates the effect of adversarial noise by reducing the redundant information present in representation.



(a) MNIST **(b) CIFAR-10**

Figure 2: Accuracy of our model (without adversarial training) against white-box attacks for various perturbation budgets

4.3 Analysis of Projector Network

Depth: We compare the effectiveness of depth of projector network with 1-layer, 2-layer 3-layer and direct representation. Results in Table 4 shows that increasing the depth of linear layers increases the performance of the model and using direct feature representation yields the best performance. However, this phenomenon only lasted until the crafted attacks come from the interval it used to train the model. When model sees a test sample that comes from different strength than training samples, the model fails and only a single layer projector retain most of the accuracy of 52.5 for FGSM and 44.6 for PGD attacks.

Wide: The accuracy of white-box attacks decreases as the wideness of the projected network increases, interestingly accuracy of black-box attack increases as the wideness of projector network increases. Looking at Figure 3, it shows that model suffers from obfuscated gradients in red area therefore, it is better to select projector parameter with around optimal point. However, the model yields better defense against black-box attacks when wideness is much higher.

4.4 Comparison with Existing Defenses

We compare our method with recently proposed state-of-the-art defense mechanisms, which includes altering the network or using modified training loss and adversarial training methods. We compare with Ref. [17] which crafts adversarial examples into the training set, trains the model and crafts examples at each iteration with the knowledge of the model. We compare with the triplet loss regularizer model [19], which forces the model to have a margin between positive and negative samples. We also compare with Adaptive diversity promoting (ADP) [28]. We compare with Min-max optimization training with adversarial samples which takes samples that causes maximum gradient increment to the loss and augment those data to training data. Finally, we compare with [26] which is closely related to our method.

Results in Tables 5, 6 in terms of classification accuracy on different datasets show that our method outperforms existing methods by a large margin. The performance gain is more prominent even for strongest attacks (*e.g.* C&W and PGD) with large perturbation size ϵ. For example, our model achieves a relative gain of 13.3% with

Table 2: Evaluation of our model robustness in *white-box* and *black-box* settings. Adversarial samples generated in the *white-box* settings shows insignificant effectiveness of attacks against our model. Here ϵ is perturbation budget and c is initial constant for C & W attack.

Objective	clean	White-Box Attacks					Black-Box Attacks				
		FGSM	BIM	CW	MIM	PGD	FGSM	BIM	CW	MIM	PGD
Mnist ($\epsilon = 0.3, c = 10$)											
$\mathcal{L}_{CE}$	**99.21**	7.1	0.8	4.3	.1	0.0	53.7	37.5	34.6	33.1	36.3
$\mathcal{L}_{AICR}$	99.18	94.8	**90.6**	**98.8**	**90.7**	**90.8**	95.0	95.5	**99.0**	94.5	96.8
$\mathcal{L}_{AICR}$ + AT_{FGSM}	98.99	**98.4**	84.4	98.6	87.4	70.3	**97.4**	**97.0**	98.6	**97.1**	**97.8**
FashionMnist ($\epsilon = 0.3, c = 10$)											
$\mathcal{L}_{CE}$	**91.51**	7.9	0.1	0.2	0.01	0.0	42.6	21.3	29.6	32.1	27.7
$\mathcal{L}_{AICR}$	90.86	**67.2**	**56.9**	**57.8**	**55.8**	**46.6**	**82.6**	**84.2**	**88.6**	**81.8**	**85.8**
$\mathcal{L}_{AICR}$ + AT_{FGSM}	91.43	59.6	48.7	23.9	49.0	29.9	74.3	71.3	87.1	68.5	74.7
CIFAR10 ($\epsilon = 0.03, c = 0.1$)											
$\mathcal{L}_{CE}$	90.70	20.4	0.0	0.6	0.0	0.0	38.4	29.6	30.3	28.5	27.6
$\mathcal{L}_{AICR}$	92.42	82.4	78.4	81.2	**79.8**	**78.6**	85.4	84.2	86.3	85.4	82.8
$\mathcal{L}_{AICR}$ + AT_{FGSM}	**92.99**	**87.0**	**78.6**	**83.4**	79.0	72.3	**88.0**	**86.4**	**87.2**	**85.7**	**83.6**
CIFAR100 ($\epsilon = 0.03, c = 0.1$)											
$\mathcal{L}_{CE}$	**72.53**	19.5	4.1	1.6	3.4	0.17	39.5	32.8	37.2	34.6	28.9
$\mathcal{L}_{AICR}$	69.9	**40.2**	**26.8**	**31.2**	26.3	**24.2**	**57.6**	36.4	**41.7**	44.9	**47.2**
$\mathcal{L}_{AICR}$ + AT_{FGSM}	70.2	43.2	23.4	26.4	**27.4**	23.1	53.5	**37.8**	38.9	**46.7**	42.5
SVHN ($\epsilon = 0.03, c = 0.1$)											
$\mathcal{L}_{CE}$	93.75	29.9	5.7	7.1	8.3	9.4	54.3	39.3	33.4	31.4	29.4
$\mathcal{L}_{AICR}$	**94.46**	78.9	47.4	51.7	**53.4**	42.1	83.2	78.9	**87.7**	**76.5**	**86.4**
$\mathcal{L}_{AICR}$ + AT_{FGSM}	92.32	**82.1**	**51.1**	**57.8**	52.0	**56.7**	**83.4**	**79.8**	82.3	73.2	82.6

Table 3: Evaluation of effectiveness of $\mathcal{L}_{var}$ under *black-box* settings, here $(^*)$ denote model trained on adversarial samples.

Attacks	param	MNIST				param	CIFAR10			
		$\mathcal{L}_{AR}$	$\mathcal{L}_{AICR}$	$\mathcal{L}^*_{AR}$	$\mathcal{L}^*_{AICR}$		$\mathcal{L}_{AR}$	$\mathcal{L}_{AICR}$	$\mathcal{L}^*_{AR}$	$\mathcal{L}^*_{AICR}$
Noattack	–	**99.5**	99.2	**99.5**	99.0	–	92.6	92.4	**93.2**	93.0
FGSM	$\epsilon = 0.1$	**98.8**	98.7	99.1	98.7	$\epsilon = 0.04$	79.5	85.4	84.3	**86.2**
	$\epsilon = 0.2$	96.3	97.7	98.1	**98.4**	$\epsilon = 0.02$	88.5	87.9	89.1	90.3
	$\epsilon = 0.4$	**49.1**	40.3	44.2	41.72	$\epsilon = 0.1$	28.7	**42.3**	31.2	40.7
BIM	$\epsilon = 0.1$	98.8	98.7	**99.1**	98.7	$\epsilon = 0.04$	76.3	83.2	79.7	**84.3**
	$\epsilon = 0.2$	96.4	97.8	**98.2**	**98.2**	$\epsilon = 0.02$	86.4	86.3	83.1	**86.7**
	$\epsilon = 0.4$	45.5	62.9	56.6	**64.9**	$\epsilon = 0.1$	17.4	**37.2**	21.6	36.4
MIM	$\epsilon = 0.1$	98.7	98.7	**99.1**	98.7	$\epsilon = 0.04$	78.9	82.7	80.3	**83.4**
	$\epsilon = 0.2$	96.0	97.7	98.1	**98.3**	$\epsilon = 0.02$	86.3	**87.2**	81.3	86.7
	$\epsilon = 0.4$	28.5	**42.7**	36.0	**42.7**	$\epsilon = 0.1$	18.4	**39.3**	22.3	36.2
PGD	$\epsilon = 0.1$	99.0	98.8	**99.1**	98.8	$\epsilon = 0.04$	72.3	80.3	77.1	**81.2**
	$\epsilon = 0.2$	97.4	98.3	**98.6**	98.5	$\epsilon = 0.02$	**84.7**	83.8	81.4	84.2
	$\epsilon = 0.4$	45.4	59.4	41.1	**61.3**	$\epsilon = 0.1$	21.4	**34.7**	23.4	31.7

Adversarial training and 24.1% compared to [26] methods on MNIST and CIFAR-10, respectively, for PGD attacks. Further, our method retains the highest accuracy when stronger attacks are deployed. Results further indicate our methods consistent with performance across all evaluated datasets.

5 CONCLUSION

Previous studies have shown that adversarial training has been one of the stronger models to defend against various attacks types against deep neural networks. In this paper, we show that the adversary's task can be made more difficult by enforcing a large margin between classes along with adversarial training. Our theory and experiments indicate that if adversarial samples belonging to different classes are

Table 4: Accuracy comparison of projector network with different depths on MNIST dataset for whitebox attacks, here #-layer means number of linear layers used in the projector network and None means and direct feature representation is used in variance loss

Attacks	ϵ	1-layers	2-layer	3-layer	None
No-attack	-	99.2	99.4	**99.6**	**99.6**
FGSM	0.3	94.8	96.2	96.3	**96.6**
	0.5	**52.5**	23.3	45.6	22.8
PDG	0.3	90.8	92.8	93.4	**94.7**
	0.5	**44.6**	36.8	30.5	22.6

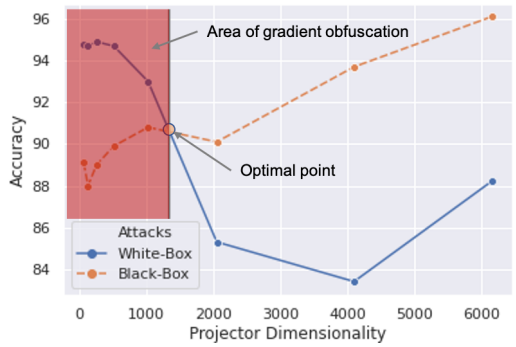


Figure 3: Accuracy of our model (without adversarial training) against white-box and black-box attacks with wideness of projector network under $PGD(\epsilon = 0.3)$ for MNIST dataset.

non-overlapping, the adversary cannot find visually imperceptible attacks with the allowed budget. We extensively evaluate the proposed model under diverse threat settings. Empirical evaluations (Sec.4.1) verify that the proposed method provides an effective and robust defence against state-of-the-art *white-box* attacks and *black-box* settings. Our approach corroborates strongly towards the notion that the adversarial training is influenced by the choice of the objective function used in optimization, but is not restricted to the properties of the data and network architecture. We provide evidence to show the robustness of the proposed method under white-box attacks which includes the strongest first order attacks (viz. Projected Gradient Descent). Towards this end, we conduct experiments on five publicly available datasets and we achieved retained accuracy of 90.8% and 78.6% on MNIST and CIFAR-10 datasets against Strongest PGD attack ($\epsilon = 0.3/0.03$), respectively. To the best of our knowledge, these are the highest robustness achieved against wide range of strong adversarial attacks. Further, our evaluation of the effectiveness of *variance loss* under *black-box* settings demonstrates that the proposed models retain the highest robust performance in all cases.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers whose suggestions helped to clarify and improve our paper. This work was supported in part by the National Science Foundation under grant number OIA-1946231 and the Louisiana Board of Regents for the Louisiana Materials Design Alliance (LAMDA).

REFERENCES

[1] Evan Ackerman. 2017. How drive. ai is mastering autonomous driving with deep learning. *IEEE Spectrum Magazine* 1 (2017).
[2] Naveed Akhtar, Ajmal Mian, Navid Kardan, and Mubarak Shah. 2021. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access* (2021).
[3] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*. PMLR, 274–283.
[4] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. 2010. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1* (2010), 183–221.
[5] Hao-Yun Chen, Jhao-Hong Liang, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. 2019. Improving adversarial robustness via guided complement entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4881–4889.
[6] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Li Chen, Michael E Kounavis, and Duen Horng Chau. 2017. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900* (2017).
[7] Li Deng and Xiao Li. 2013. Machine learning paradigms for speech recognition: An overview. *IEEE Transactions on Audio, Speech, and Language Processing* 21, 5 (2013), 1060–1089.
[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[9] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. 2020. Whitening for self-supervised representation learning. *arXiv preprint arXiv:2007.06346* (2020).
[10] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100* (2020).
[11] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117* (2017).
[12] Pierre Hansen and Brigitte Jaumard. 1995. Lipschitz optimization. In *Handbook of global optimization*. Springer, 407–493.
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*. 1026–1034.
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[15] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373* (2018).
[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.
[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236* (2016).
[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[19] Pengcheng Li, Jinfeng Yi, Bowen Zhou, and Lijun Zhang. 2019. Improving the robustness of deep neural networks via adversarial training with triplet loss. *arXiv preprint arXiv:1905.11713* (2019).
[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
[21] Yan Luo, Xavier Boix, Gemma Roig, Tomaso Poggio, and Qi Zhao. 2015. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292* (2015).
[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).
[23] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. 2019. Metric learning for adversarial robustness. *arXiv preprint arXiv:1909.00900* (2019).
[24] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. 2015. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677* (2015).

Table 5: Comparison on CIFAR10 dataset for white-box adversarial attacks (accuracy is reported as evaluation metric). * sign denotes adversarial trained models. For our model, we report results with gaussian noise (Ours*) and adversarially generated images from FGSM (Ours*_f) attack.

Attacks	params	Baseline	AT*	Li et al.*	Pang et al.*	Mandry et al. *	Mustafa et al.*	Ours*	Ours*$_f$
No attack	-	90.8	84.5	90.5	90.6	87.3	91.3	92.4	**93.0**
FGSM	ϵ = 0.02	36.5	44.3	80.5	61.7	71.6	80.8	83.4	**87.4**
	ϵ = 0.04	19.4	31.0	64.4	46.2	47.4	70.5	81.2	**86.2**
BIM	ϵ = 0.01	18.5	22.6	42.1	46.6	64.3	67.9	**82.3**	80.4
	ϵ = 0.02	6.1	7.8	36.4	31.0	49.3	51.2	**79.7**	79.1
MIM	ϵ = 0.01	23.8	23.9	36.4	52.1	61.5	68.8	**83.0**	81.2
	ϵ = 0.02	7.4	9.3	27.8	35.9	46.7	53.8	**80.9**	79.7
PGD	ϵ = 0.01	23.4	24.3	38.6	48.4	67.7	68.3	**83.1**	79.8
	ϵ = 0.02	6.6	7.8	17.8	30.4	48.5	50.6	**80.0**	74.7
C & W	c = 0.001	61.3	67.7	83.6	80.6	84.5	91.0	**91.5**	91.0
	c = 0.01	35.2	40.9	67.7	54.9	65.7	72.9	85.3	**89.4**
	c = 0.1	0.6	25.4	51.8	25.6	47.9	55.7	81.2	**83.4**

Table 6: Comparison on MNIST dataset for white-box adversarial attacks (accuracy is reported as evaluation metric). * sign denotes adversarial trained models. For our model, we report results with gaussian noise (Ours*) and adversarially generated images from FGSM (Ours*$_f$) attack.

Attacks	params	Baseline	AT [17]*	Li et al. [19]*	Pang et al. [28]*	Mustafa et al. [25]*	Ours*	Ours*$_f$
No attack	-	98.7	99.1	99.2	**99.5**	**99.5**	99.2	99.0
FGSM	ϵ = 0.1	58.3	73.0	98.2	96.3	97.2	98.0	**98.8**
	ϵ = 0.2	52.7	70.3	97.3	52.8	80.0	96.6	**98.6**
BIM	ϵ = 0.1	22.5	88.1	89.7	88.5	92.0	**97.4**	95.5
	ϵ = 0.15	12.2	77.1	82.7	73.6	77.3	**95.9**	93.4
MIM	ϵ = 0.1	58.3	64.5	88.2	92.0	92.7	**97.6**	96.4
	ϵ = 0.15	16.1	28.8	82.7	77.5	80.2	**96.2**	94.8
PGD	ϵ = 0.1	50.7	62.7	68.4	82.8	93.7	**97.9**	94.9
	ϵ = 0.15	6.3	31.9	66.2	41.0	78.8	**96.6**	92.1
C & W	c = 0.1	66.6	71.1	88.1	97.3	97.7	**99.1**	98.6
	c = 1.0	30.6	39.2	81.4	78.1	87.3	**98.8**	98.6
	c = 10.0	0.2	17.0	79.3	23.8	39.7	**98.8**	98.6

[25] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. 2019. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3385–3394.

[26] Aamir Mustafa, Salman H Khan, Munawar Hayat, Jianbing Shen, and Ling Shao. 2019. Image super-resolution as a defense against adversarial attacks. *IEEE Transactions on Image Processing* 29 (2019), 1711–1724.

[27] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. 2015. Deep learning applications and challenges in big data analytics. *Journal of big data* 2, 1 (2015), 1–21.

[28] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. 2019. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*. PMLR, 4970–4979.

[29] Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. 2018. Whitening and coloring batch transform for gans. *arXiv preprint arXiv:1806.00420* (2018).

[30] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[31] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[32] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[33] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* (2017).

[34] Hado Van Hasselt. 2012. Reinforcement learning in continuous state and action spaces. In *Reinforcement learning*. Springer, 207–251.

[35] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. 2016. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*. Springer, 499–515.

[36] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991* (2017).

[37] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. 2018. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3474–3482.

[38] Fuxun Yu, Chenchen Liu, Yanzhi Wang, Liang Zhao, and Xiang Chen. 2018. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144* (2018).

[39] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230* (2021).