Deeply Supervised Discriminative Learning for Adversarial Defense

Aamir Mustafa, Salman H. Khan, Munawar Hayat, Roland Goecke, Jianbing Shen and Ling Shao

Abstract—Deep neural networks can easily be fooled by an adversary with minuscule perturbations added to an input image. The existing defense techniques suffer greatly under *white-box* attack settings, where an adversary has full knowledge of the network and can iterate several times to find strong perturbations. We observe that the main reason for the existence of such vulnerabilities is the close proximity of different class samples in the learned feature space of deep models. This allows the model decisions to be completely changed by adding an imperceptible perturbation to the inputs. To counter this, we propose to class-wise disentangle the intermediate feature representations of deep networks, specifically forcing the features for each class to lie inside a convex polytope that is maximally separated from the polytopes of other classes. In this manner, the network is forced to learn distinct and distant decision regions for each class. We observe that this simple constraint on the features greatly enhances the robustness of learned models, even against the strongest *white-box* attacks, without degrading the classification performance on clean images. We report extensive evaluations in both *black-box* and *white-box* attack scenarios and show significant gains in comparison to state-of-the-art defenses.

Index Terms—Adversarial defense, adversarial robustness, white-box attack, distance metric learning, deep supervision.

1 Introduction

Deep Convolutional Neural Network (CNN) models can easily be fooled by adversarial examples containing small, human-imperceptible perturbations specifically designed by an adversary [1], [2], [3]. Such adversarial examples pose a serious threat to security critical applications, e.g. autonomous cars [4], bio-metric identification [5] and surveillance systems [6]. Furthermore, if a slight perturbation added to a benign input drastically changes the deep network's output with high-confidence, this implies that our current models are not distinctively learning the underlying fundamental visual concepts. Therefore, designing robust deep networks goes a long way towards developing reliable and trustworthy artificial intelligence systems.

Numerous defense methods have recently been proposed in the literature to mitigate adversarial attacks. These can be broadly classified into two categories: (a) *Reactive defenses* that modify the inputs during testing time, using image transformations to counter the effect of adversarial perturbations [7], [8], [9], [10], and (b) *Proactive defenses* that alter the underlying model's architecture or learning procedure, e.g., by adding more layers, using ensemble/adversarial training or changing the loss/activation functions [11], [12], [13], [14], [15], [16], [17], [18]. Proactive defenses are generally more valued, as they provide relatively better robustness against *white-box* attacks. However, iterative *white-box* adversaries pose a challenge for both proactive and reactive defenses [19].

This paper introduces a novel distance based training procedure, *'Prototype Conformity Loss'*, as a proactive defense against adversarial attacks, which seeks to maximally separate the learned feature representations at multiple depth levels of the deep model (from hereon referred to as *'Deep Supervision'*). We note that the addition of perturbations in the input domain leads to a corresponding polytope in the high-dimensional manifold of the intermediate features and the output classification space. Based upon this observation, we propose to maximally separate the polytopes for different class samples, such that there is a minimal overlap between any two classes in the decision and intermediate feature space. This ensures that an adversary can no longer fool the network within a restricted perturbation budget. In other words, we build on the intuition that two different class samples, which are visually dissimilar in the input domain, must be mapped to different regions in the output space. Therefore, we must also enforce that their feature representations are well separated along the hierarchy of network layers. This is achieved by improving within-class proximities and enhancing between-class differences of the activation maps, along multiple levels of the deep model. As illustrated in Fig. 1, the penultimate layer features learnt by the proposed scheme are well separated and hard to penetrate compared with the easily attacked features learnt using the standard softmax loss without any deep supervision.

Our approach provides strong evidence towards the notion that the adversarial perturbations exist not only due to the properties of data (e.g. high-dimensionality) and network architectures (e.g. non-linearity functions) but are also greatly influenced by the choice of objective functions used for optimization. Our primary claim is that the distance-based objectives are better suited for learning robust models compared to the commonly used cross-entropy loss. Among such objectives, we demonstrate that our proposed approach provides the best performance, while being highly

- *A. Mustafa is with University of Cambridge, UK.*
- *S. H. Khan, M. Hayat, J. Shen and L. Shao are with Inception Institute of Artificial Intelligence, Abu Dhabi, UAE.*
- *R. Goecke is with University of Canberra, Australia.*

. Codes are made public at https://github.com/aamir-mustafa/pcl-adversarial-defense

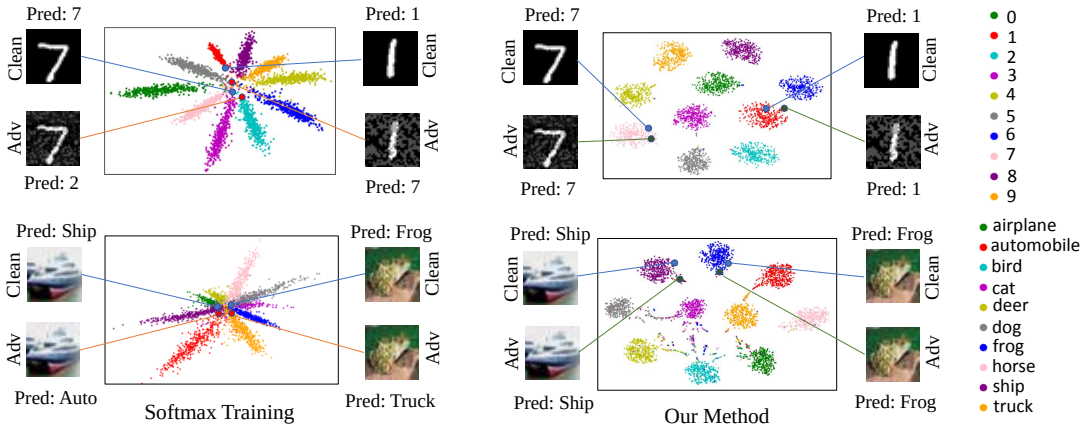


Fig. 1: 2D penultimate layer activations of a clean image and its adversarial counterpart (PGD attack) for standard softmax trained model and our method on MNIST (top row) and CIFAR-10 (bottom row) datasets. Note that our method correctly maps the attacked image to its true-class feature space.

efficient. To this end, we propose a deeply supervised multi-layered loss based defense which provides a significant boost in robustness under the strictest attack conditions, where the balance is shifted heavily towards the adversary. These include *white-box* attacks and *iterative adversaries* including the strongest first-order attack (Projected Gradient Descent). We evaluate the robustness of our proposed defense through extensive evaluations on five publicly available datasets and achieve a robustness of 46.7% and 36.1% against the strongest PGD attack ($\epsilon = 0.03$) for the CIFAR-10 and CIFAR-100 datasets, respectively. To the best of our knowledge, these are significantly higher levels of robustness against a broad range of adversarial attacks.

Our empirical evaluations (Sec. 6) demonstrate that the proposed method provides an effective and robust defense, significantly outperforming current state-of-the-art defenses under both *white-box* and *black-box* settings. Adversarial examples have been shown to exhibit the surprising property of being highly transferable [2], [20], i.e. given two classification models trained on either the same or different datasets, the adversarial samples generated using one model often succeed in fooling the other model (without having any information about the latter). Motivated by this observation, we perform a transferability test to show the robustness of our technique under extreme *black-box* settings (i.e., a transfer attack). Specifically, using adversarial examples generated from various models trained via diverse schemes as source models, we demonstrate that our approach defends against transfer attacks to a great extent. In addition, we experimentally show that our method does not suffer from the gradient masking effect [19], which makes defenses vulnerable to *black-box* attacks.

A preliminary version of this work appeared in [21]. In addition, the current paper includes: (1) a systematic study and comparison of the proposed defense framework with other popular distance-based loss functions (center [22], contrastive [23] and triplet loss [24]), (2) additional experimental and architecture details, (3) new visualizations to illustrate the learning process, (4) experiments with the more challenging '*adaptive white-box*' setting and (5) timing and efficiency comparisons between several distance-based objective functions.

2 Related Work

Generating adversarial examples to fool a deep network and developing defenses against such examples have recently gained significant research attention. Adversarial perturbations were first proposed by Szegedy *et al.* [1] using an L-BFGS based optimization scheme, followed by the Fast Gradient Sign Method (FGSM) [2] and its iterative variant [12]. Moosavi-Dezfooli *et al.* [25] then proposed DeepFool, which iteratively projects an image across the decision boundary (in the form of a polyhydron) until it crosses the boundary and is misclassified. One of the strongest attacks proposed recently is the Projected Gradient Descent (PGD) [17], which takes maximum loss increments allowed within a specified l_∞ norm-ball. Other popular attacks include the Carlini and Wagner Attack [26], Jacobian-based Saliency Map Approach [27], Momentum Iterative Attack [28] and Diverse Input Iterative Attack [29].

Two main lines of defense mechanisms have been proposed in the literature to counter adversarial attacks. The first involves applying different pre-processing steps and transformations on the input image at inference time [9], [30]. The second category of defenses improve a network's training regime to counter adversarial attacks. An effective scheme in this regards is *adversarial training*, where the model is jointly trained with clean images and their adversarial counterparts [2], [31]. Ensemble adversarial training was used in [11] to soften the classifier's decision boundaries. Virtual Adversarial Training [32] smoothes the model distribution using a regularization term. Papernot *et al.* [13] used distillation to improve the model's robustness by retraining it with soft labels. Parsevel Networks [14] restrict the Lipschitz constant of each layer of the model. Input Gradient Regularizer [33] penalizes the change in a model's prediction w.r.t. input perturbations by regularizing the gradient of the cross-entropy loss. The Frobenius norm of the Jacobian of the network was shown to improve a model's stability in [34]. [35] proposed a defensive quantization method to control the Lipschitz constant of the network and mitigate the adversarial noise during inference. [36] proposed Stochastic Activation Pruning as a defense against adversarial attacks. Min-Max Optimization [17] is one of the strongest defense methods, which augments the training data with first order attacked samples. Despite significant research activity in devising defenses against adversarial attacks, it was recently shown in [19] that the

current defenses are successfully circumvented under *white-box* settings. Further, many of the existing defenses, e.g. thermometer encoding [37], stochastic activation pruning [36] mitigating through randomization [9] and DefenseGAN [38], obfuscate gradients. Only Min-Max optimization [17] and Cascade adversarial machine learning [39] retained 47% and 15% accuracy respectively on the CIFAR-10 dataset, and withstood attacks under *white-box* settings. In our experiments (see Sec. 6), we extensively compare our results with [17] and make a compelling case by achieving significant improvements.

At the core of our defense are the proposed objective function and multi-level deep supervision, which ensure feature space discrimination between classes. A classification model trained with the standard Softmax loss only ensures that the features of a sample align well with its respective class prototype, without enforcing any explicit inter-class separation or margin constraints. In pursuit of making models robust against adversarial attacks, we propose to maximally separate the intermediate feature representations of different class samples. To this end, our training objective is inspired from center loss [22], which clusters penultimate layer features. We propose multiple novel constraints (Sec. 3) to enhance between-class distances, and ensure maximal separation of a sample from its non-true classes (see Fig. 2). Our method is therefore fundamentally different from [22], since the proposed multi-layered hierarchical loss formulation and the notion of maximal separation has not been previously explored for adversarial robustness. Our experiments (Sec. 6.3) further show that the proposed method significantly outperforms [22].

3 Prototype Conformity Loss

Below, we first introduce the notations used, then provide a brief overview of the conventional cross entropy loss, followed by a detailed description of our proposed method.

Notations: Let $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y}$ denote an input-label pair and $\mathbf{1}_y$ be the one-hot encoding of $\boldsymbol{y}$. We denote a deep neural network (DNN) as a function $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $\boldsymbol{\theta}$ are the trainable parameters. The DNN outputs a feature representation $\boldsymbol{f} \in \mathbb{R}^d$, which is then used by a classification layer to perform multi-class classification. Let k be the number of classes; the parameters of the classifier can then be represented as $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k] \in \mathbb{R}^{d \times k}$. To train the model, we find the optimal $\boldsymbol{\theta}$ and $\boldsymbol{W}$ that minimize a given objective function. Next, we introduce a popular loss function for deep CNNs.

Cross-entropy Objective: The cross-entropy objective function maximizes the dot product between an input feature $\boldsymbol{f}_i$ and its true class representative vector $\boldsymbol{w}_y$, such that $\boldsymbol{w}_y \in \boldsymbol{W}$. In other words, cross-entropy forces the classifier to learn a mapping from feature to output space such that the projection onto the correct class vector is maximized:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{r} -\log \frac{\exp(\boldsymbol{w}_{y_i}^T \boldsymbol{f}_i + \boldsymbol{b}_{y_i})}{\sum_{j=1}^{k} \exp(\boldsymbol{w}_j^T \boldsymbol{f}_i + \boldsymbol{b}_j)}, \tag{1}$$

where, r is the number of images, and $\boldsymbol{f}_i$ is the feature of an i^{th} image $\boldsymbol{x}_i$ with the class $\boldsymbol{y}_i$. $\boldsymbol{W}$ and $\boldsymbol{b}$ are, respectively, the weights and the bias terms for the classification layer.

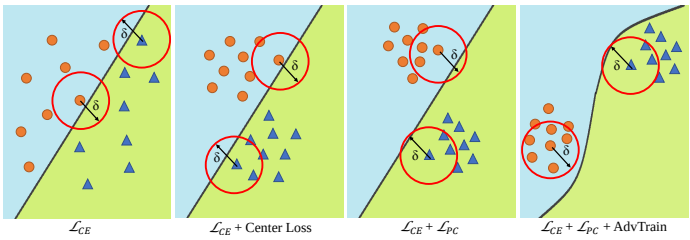


Fig. 2: Comparison between different training methods. The red circle encompasses the adversarial sample space within a perturbation budget $\|\boldsymbol{\delta}\|_p < \epsilon$.

Adversarial Perspective: The main goal of an attack algorithm is to force a trained DNN $\mathcal{F}_\theta$ to make wrong predictions. Attack algorithms seek to achieve this goal within a minimal perturbation budget. The attacker's objective can be represented by:

$$\underset{\boldsymbol{\delta}}{\text{argmax}}\, \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}), \quad s.t., \|\boldsymbol{\delta}\|_p < \epsilon, \tag{2}$$

where $\boldsymbol{y}$ is the ground-truth label for an input sample $\boldsymbol{x}$, $\boldsymbol{\delta}$ denotes the adversarial perturbation, $\mathcal{L}(\cdot)$ denotes the error function, $\|\cdot\|_p$ denotes the p-norm, which is generally considered to be an ℓ_∞-ball centered at $\boldsymbol{x}$, and ϵ is the available perturbation budget.

In order to create a robust model, the learning algorithm must consider the allowed perturbations in the input domain and learn a function that maps the perturbed images to the correct class. This can be achieved through the following min-max (saddle point) objective that minimizes the empirical risk in the presence of perturbations:

$$\min_{\theta} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}} \big[\max_{\delta} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}; \boldsymbol{\theta}) \big], \; s.t., \|\boldsymbol{\delta}\|_p < \epsilon, \tag{3}$$

where $\mathcal{D}$ is the data distribution.

CE Loss in Adversarial Settings: The CE loss is the default choice for conventional classification tasks. However, it simply assigns an input sample to one of the pre-defined classes. It therefore does not allow one to distinguish between normal and abnormal inputs (adversarial perturbations in our case). Further, it does not explicitly enforce any margin constraints amongst the learned classification regions. It can be seen from Eq. 3 that an adversary's job is to maximize $\mathcal{L}(\cdot)$ within a small perturbation budget ϵ. Suppose, the adversarial polytope in the output space[1] with respect to an input sample $\boldsymbol{x}$ is given by:

$$\mathcal{P}_\epsilon(\boldsymbol{x}; \theta) = \{\mathcal{F}_\theta(\boldsymbol{x} + \boldsymbol{\delta}) \; s.t., \; \|\boldsymbol{\delta}\|_p \leq \epsilon\}. \tag{4}$$

An adversary's task is easier if there is an overlap between the adversarial polytopes for different input samples belonging to different classes.

Definition 1: *The overlap $\mathcal{O}_\epsilon^{i,j}$ between polytopes for each data sample pair (i, j) can be defined as the volume of intersection between the respective polytopes:*

$$\mathcal{O}_\epsilon^{i,j} = \mathcal{P}_\epsilon(\boldsymbol{x}_{y_i}^i; \theta) \cap \mathcal{P}_\epsilon(\boldsymbol{x}_{y_j}^j; \theta).$$

1. Note that the output space in our case is not the final prediction space, but the intermediate feature space.

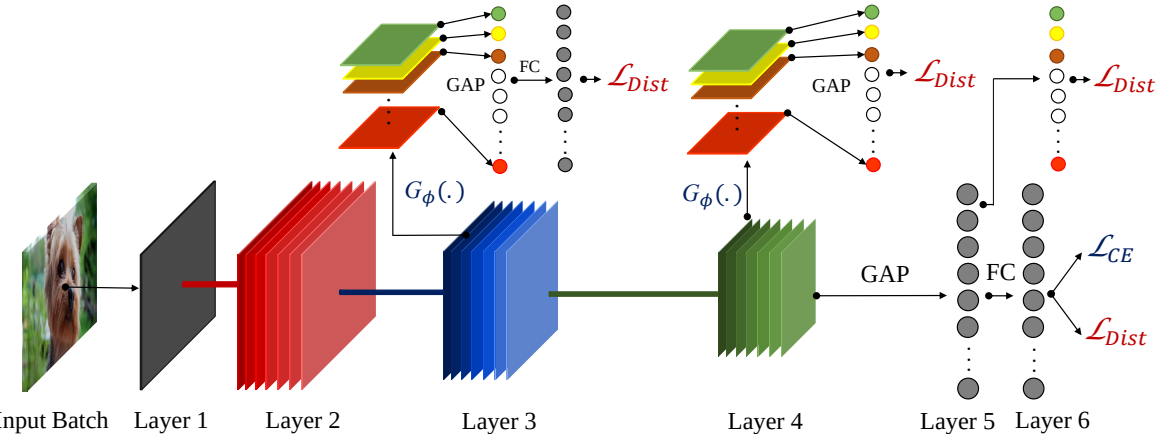


Fig. 3: An illustration of our training with deep supervision of distance-based loss functions $\mathcal{L}_{\text{Dist}}$. $G_\phi(\cdot)$ is an auxiliary branch to map features to a low-dimensional output, which is then used as the loss in Eq. 9.

Note that the considered polytopes can be non-convex as well. However, the overlap computation can be simplified for convex polytopes [40].

Proposition 1: *For an i^{th} input sample $\boldsymbol{x}^i_{y_i}$ with class label $\boldsymbol{y}_i$, reducing the overlap $\mathcal{O}^{i,j}_{\epsilon}$ between its polytope $\mathcal{P}_\epsilon(\boldsymbol{x}^i_{y_i};\theta)$ and the polytopes of other class samples $\mathcal{P}_\epsilon(\boldsymbol{x}^j_{y_j};\theta)$, s.t., $\boldsymbol{y}_j \neq \boldsymbol{y}_i$ will result in lower adversary success for a bounded perturbation $\|\boldsymbol{\delta}\|_p \leq \epsilon$.*

Proposition 2: *For a given adversarial strength ϵ, assume λ is the maximum distance from the center of the convex polytope to its outer boundary. Then, a classifier maintaining a margin $m > 2\lambda$ between two closest samples belonging to different classes will result in a decision boundary with guaranteed robustness against perturbation within the budget ϵ.*

In other words, if the adversarial polytopes for samples belonging to different classes are non-overlapping, the adversary cannot find a viable perturbation within the allowed budget. We propose that an adversary's task can be made difficult by including a simple maximal separation constraint in the objective of deep networks. The conventional CE loss does not impose any such constraint, which makes the resulting models weaker against adversaries. A more principled approach is to define convex category-specific classification regions for each class, where any sample outside all of such regions is considered an adversarial perturbation. Consequently, we propose the Prototype Conformity Loss (PCL) function, described below.

Proposed Objective: We represent each class with its prototype vector, which represents the training examples of that class. Each class is assigned a fixed and non-overlapping p-norm ball and the training samples belonging to a class i are encouraged to be mapped close to its hyper-ball center:

$$\mathcal{L}_{\text{PC}}(\boldsymbol{x},\boldsymbol{y}) = \sum_i \Big\{ \|\boldsymbol{f}_i - \boldsymbol{w}^c_{y_i}\|_2^2 - \frac{1}{k-1}\sum_{j\neq y_i}\Big(\|\boldsymbol{f}_i - \boldsymbol{w}^c_j\|_2^2 + \|\boldsymbol{w}^c_{y_i} - \boldsymbol{w}^c_j\|_2^2\Big)\Big\}. \tag{5}$$

The gradients for the PCL can be computed as:

$$\frac{\partial \mathcal{L}_{\text{PC}}}{\partial \boldsymbol{f}_i} = 2\Big(\boldsymbol{f}_i - \boldsymbol{w}^c_{y_i} - \frac{1}{k-1}\sum_{j\neq y_i}\boldsymbol{f}_i - \boldsymbol{w}^c_j\Big). \tag{6}$$

During model inference, a feature's similarity is computed with all the class prototypes and it is assigned the closest class label if and only if the sample lies within its decision region:

$$\hat{\boldsymbol{y}}_i = \underset{j}{\text{argmin}}\, \|\boldsymbol{f}_i - \boldsymbol{w}^c_j\|. \tag{7}$$

Here, $\boldsymbol{w}^c$ denotes the trainable class centroids. Note that the classification rule is similar to the Nearest Class Mean (NCM) classifier [41], but we differ in some important aspects: (a) the centroids for each class are not fixed as the mean of training samples, but are instead learned automatically during representation learning, (b) class samples are explicitly forced to lie within respective class norm-balls, (c) feature representations are appropriately tuned to learn discriminant mappings in an end-to-end manner, and (d) to avoid inter-class confusions, disjoint classification regions are considered by maintaining a large distance between each pair of prototypes. We also experiment with the standard softmax classifier and retain the same as the nearest prototype rule mentioned above.

Deeply Supervised Learning: The overall loss function used for training our model is given by:

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{L}_{\text{CE}}(\boldsymbol{x},\boldsymbol{y}) + \mathcal{L}_{\text{PC}}(\boldsymbol{x},\boldsymbol{y}). \tag{8}$$

The above loss enforces intra-class compactness and inter-class separation using learned prototypes in the output space. In order to achieve a similar effect in the intermediate feature representations, we include other auxiliary loss functions $\{\mathcal{L}^n\}$ along the depth of our deep networks, which act as companion objective functions for the final loss. This is achieved by adding an auxiliary branch $\mathcal{G}_\phi(\cdot)$ after the defined network depth, which maps the features to a lower dimension output, and is then used in the loss definition. For illustration, see Fig. 3.

$$\mathcal{L}^n(\boldsymbol{x},\boldsymbol{y}) = \mathcal{L}_{\text{CE}}(\boldsymbol{f}^l,\boldsymbol{y}) + \mathcal{L}_{\text{PC}}(\boldsymbol{f}^l,\boldsymbol{y}) \tag{9}$$

$$s.t.,\ \boldsymbol{f}^l = \mathcal{G}^l_\phi(\mathcal{F}^l_\theta(\boldsymbol{x})). \tag{10}$$

These functions reinforce the desired intra-class separation in the feature space at several intermediate layers of the deep network.

4 Studying other Distance-based Losses

The separation in the feature space can be enforced by any distance-based loss function in our proposed deeply-supervised framework. In addition to our proposed Prototype Conformity Loss (PCL), in this section, we also study other distance-based loss functions, such as center loss [22], contrastive loss [23] and triplet loss [24]. As we will demonstrate through quantitative results in Sec. 6, other distance-based metrics also deliver strong improvements compared to the traditional cross-entropy loss; however, the best performance is achieved with the proposed PCL.

Center Loss: Wen *et al.* [22] proposed center loss ($\mathcal{L}_{\text{CL}}$), which discriminates the feature activations ($\boldsymbol{f}_i$) of the penultimate layer of a deep model, by learning class-wise centers:

$$\mathcal{L}_{\text{CL}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{f}_i - \boldsymbol{c}_{y_i}\|_2^2, \tag{11}$$

where c_{y_i} denotes the $\boldsymbol{y}_i$th class center of deep features. During the course of training, the class-wise centers are updated as follows:

$$\boldsymbol{c}_j^{t+1} = \boldsymbol{c}_j^t - \alpha \cdot \nabla \boldsymbol{c}_j^t \tag{12}$$

$$\text{where,} \quad \nabla \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(\boldsymbol{y}_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{f}_i)}{1 + \sum_{i=1}^{m} \delta(\boldsymbol{y}_i = j)}, \tag{13}$$

where $\delta(\cdot)$ is the delta function, and α is the learning rate for updating the centers. The deep model is trained under the joint supervision of the softmax loss and center loss, with a hyper-parameter to balance the two supervision signals. Intuitively, the softmax loss forces the deep features of different classes to be apart. The center loss pulls the deep features of the same class to their respective centers. This joint supervision tends to maximize the inter-class feature differences and reduce the intra-class variations.

Contrastive Loss: Hadsell *et al.* [23] proposed the contrastive loss ($\mathcal{L}_{\text{Co}}$) to learn discriminative features for images. During training, an image pair is fed into the model with its ground truth defined as 1 if both images belong to the same class and 0 otherwise. In our experiments, we train a Siamese network that takes a pair of images and trains the embeddings at different layers of the network so that the distance between them is minimized if they are from the same class and is greater than some specified margin value if they represent different classes. The loss is given by:

$$\mathcal{L}_{\text{Co}}(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}\|f_0 - f_1\|_2^2 + \frac{1}{2}(1-\boldsymbol{y})\{\max(0, \boldsymbol{m} - \|f_0 - f_1\|_2)\}^2. \tag{14}$$

Here, $\boldsymbol{m}$ denotes the margin whose violation results in a penalty.

Triplet Loss: Triplet loss was introduced by Schroff *et al.* [24] and is commonly used in image retrieval. During the training process, an image triplet $(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n)$ is fed into the model as a data sample, where $\boldsymbol{x}_a, \boldsymbol{x}_p$ and $\boldsymbol{x}_n$ are the anchor, positive and negative images, respectively. The objective is to learn embeddings such that the anchor is closer to the positive example than the negative one. Formally, the triplet loss is defined as:

$$\mathcal{L}_{\text{T}}(\boldsymbol{x}_a, \boldsymbol{x}_p, \boldsymbol{x}_n) = \max(0, \boldsymbol{m} + \|f_a - f_p\|_2^2 - \|f_a - f_n\|_2^2. \tag{15}$$

Here, $\boldsymbol{m}$ is a margin term used to stretch the gap between similar and dissimilar pairs in the triplet and f_a, f_p and f_n are the intermediate feature embeddings for the anchor, positive and negative images, respectively.

Next, we outline the adversarial attacks considered in this work to defy our proposed defense mechanism, as well as different existing state-of-the-art defenses.

5 Adversarial Attacks

We evaluate our defense model against five recently proposed state-of-the-art attacks, which are summarized below, for completeness.

Fast Gradient Sign Method (FGSM) [2] generates an adversarial sample $\boldsymbol{x}_{adv}$ from a clean sample $\boldsymbol{x}$ by maximizing the loss in Eq. 2. It finds $\boldsymbol{x}_{adv}$ by moving a single step in the opposite direction to the gradient of the loss function, as:

$$\boldsymbol{x}_{adv} = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})). \tag{16}$$

Here, ϵ is the allowed perturbation budget.

Basic Iterative Method (BIM) [12] is an iterative variant of FGSM and generates an adversarial sample as:

$$\boldsymbol{x}_m = \text{clip}_\epsilon(\boldsymbol{x}_{m-1} + \frac{\epsilon}{i} \cdot \text{sign}(\nabla_{x_{m-1}}(\mathcal{L}(\boldsymbol{x}_{m-1}, \boldsymbol{y}))), \tag{17}$$

where $\boldsymbol{x}_0$ is clean image $\boldsymbol{x}$ and i is the iteration number.

Momentum Iterative Method (MIM) [28] introduces an additional momentum term to BIM to stabilize the direction of gradient. Eq. 17 is modified as:

$$g_m = \mu \cdot g_{m-1} + \frac{\nabla_{x_{m-1}} \mathcal{L}(\boldsymbol{x}_{m-1}, \boldsymbol{y})}{\| \nabla_{x_{m-1}}(\mathcal{L}(\boldsymbol{x}_{m-1}, \boldsymbol{y})) \|_1} \tag{18}$$

$$\boldsymbol{x}_m = \text{clip}_\epsilon(\boldsymbol{x}_{m-1} + \frac{\epsilon}{i} \cdot \text{sign}(g_m)), \tag{19}$$

where μ is the decay factor.

Carlini & Wagner Attack [26] defines an auxiliary variable ζ and minimizes the objective function:

$$\min_{\zeta} \| \frac{1}{2}(\tanh(\zeta) + 1) - \boldsymbol{x} \| + c \cdot f(\frac{1}{2}(\tanh\zeta + 1)), \tag{20}$$

where $\frac{1}{2}(\tanh(\zeta) + 1) - \boldsymbol{x}$ is the perturbation $\boldsymbol{\delta}$, c is the constant chosen and $f(.)$ is defined as:

$$f(\boldsymbol{x}_{adv}) = \max(\mathcal{Z}(\boldsymbol{x}_{adv})_{\boldsymbol{y}} - \max\{\mathcal{Z}(\boldsymbol{x}_{adv})_k : k \neq \boldsymbol{y}\}, -\kappa). \tag{21}$$

Here, κ controls the adversarial sample's confidence and $\mathcal{Z}(\boldsymbol{x}_{adv})_k$ are the logits values corresponding to a class k.

Projected Gradient Descent (PGD) [17] is similar to BIM, and starts from a random position in the clean image neighborhood $\mathcal{U}(\boldsymbol{x}, \epsilon)$. This method applies FGSM for m iterations with a step size of γ as:

$$\boldsymbol{x}_m = \boldsymbol{x}_{m-1} + \gamma \cdot \text{sign}(\nabla_{x_{m-1}} \mathcal{L}(\boldsymbol{x}_{m-1}, \boldsymbol{y})). \quad (22)$$

$$\boldsymbol{x}_m = \text{clip}(\boldsymbol{x}_m, \boldsymbol{x}_m - \epsilon, \boldsymbol{x}_m + \epsilon). \quad (23)$$

It proves to be a strong iterative attack, relying on the first order information of the target model.

6 Experiments

Datasets and Models: We extensively evaluate the proposed method on five datasets: MNIST, Fashion-MNIST (F-MNIST), CIFAR-10, CIFAR-100 and Street-View House Numbers (SVHN). For the MNIST and F-MNIST datasets, the CNN model chosen has six layers, as in [22]. For the CIFAR-10, CIFAR-100 and SVHN datasets, we use a ResNet-110 model [42] (see Table 1). The deep features for the prototype conformity loss are extracted from different intermediate layers using an auxiliary branch, which maps the features to a lower dimension output (see Fig. 3). We first train for $T^{'}$ epochs ($T^{'} = 50$ for F/MNIST, $T^{'} = 200$ for CIFAR-10/100 and SVHN) with $\mathcal{L}_{\text{CE}}$ and then use the loss in Eq. 9 for 300 epochs. A batch size of 256 and a learning rate of 0.1 ($\times 0.1$ at T=200, 250) are used. Further training details are summarized in Algorithm 1.

Algorithm 1: Model training with Prototype Conformity Loss.

Input: Classifier $\mathcal{F}_\theta(\boldsymbol{x})$, training data $\{\boldsymbol{x}\}$, ground truth labels $\{\boldsymbol{y}\}$, trainable parameters $\boldsymbol{\theta}$, trainable class centroids $\{\boldsymbol{w}_j^c : j \in [1, k]\}$, perturbation budget ϵ, epochs T, number of auxiliary branches L.
Output: Updated parameters $\boldsymbol{\theta}$

Initialize $\boldsymbol{\theta}$ in convolutional layers.
for $t = 0$ to T:
  if $t < T^{'}$:
    Converge softmax objective, $\boldsymbol{\theta} := \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{CE}}$.
  else:
    Compute joint loss $\mathcal{L} = \mathcal{L}_{\text{CE}} + \tau \sum_l^L \mathcal{L}_{\text{PC}}$
    Compute gradients w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{x}$, $\nabla_\theta \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ and $\nabla_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ respectively.
    Update model weights, $\boldsymbol{\theta} := \arg\min_{\boldsymbol{\theta}} \mathcal{L}$.
    Update class centroids $\boldsymbol{w}_j^c \; \forall \; l$
    Generate adversarial examples as:
      if FGSM: **then** $\boldsymbol{x}_{adv} = \boldsymbol{x} + \epsilon \cdot \text{sign}\,(\nabla_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}))$
      elif PGD: **then** $\boldsymbol{x}_{adv} = \text{clip}\,(\boldsymbol{x}, \boldsymbol{x} - \epsilon, \boldsymbol{x} + \epsilon)$
    Augment $\boldsymbol{x}$ with $\boldsymbol{x}_{adv}$
return $\boldsymbol{\theta}$

Hyper-parameters for Adversaries: For each dataset, the details on the hyper-parameters of various adversarial attacks used in our implementations are given in Table 2.

6.1 Results and Analysis

***White-Box* vs *Black-Box* Settings:** In an adversarial setting, there are two main threat models: *white-box* attacks, where the adversary possesses complete knowledge of the target model, including its parameters, architecture and the training method, and *black-box attacks* (transfer attack in our case), where the adversary feeds perturbed images at test time (which are generated without any knowledge of the target model). We evaluate the robustness of our proposed defense against both *white-box* and *black-box* settings. Table 3 shows our results for the different attacks described in Sec. 5. The number of iterations for BIM, MIM and PGD are set to 10 with a step size of $\epsilon/10$ for BIM and MIM and $\epsilon/4$ for PGD. The iteration steps for C&W are $1,000$, with a learning rate of 0.01. We report our model's robustness with and without adversarial training for standard perturbation size, i.e., $\epsilon = 0.3$ for F/MNIST and and $\epsilon = 0.03$ for the CIFAR-10/100 and SVHN datasets.

Recent literature has shown the transferability of adverserial attacks amongst deep models [2], [31], [43], where adversarial images are effective even for the models they were never generated on. An adversary can therefore exploit this characteristic of deep models and generate generic adversarial samples to attack unseen models. Defense against *black-box* attacks is therefore highly desirable for secure deployment of machine learning models [27]. To demonstrate the effectiveness of our proposed defense under *black-box* settings (specifically in a transfer attack case), we generate adversarial samples using a standard VGG-19 model (trained on clean images only), and feed them to the model trained using our proposed strategy. Results in Table 3 show that *black-box* settings have negligible attack potential against our model. For example, on the CIFAR-10 dataset, where our model's accuracy on clean images is 91.89%, even the strongest iterative attack (PGD-0.03) fails, and our defense retains an accuracy of 88.8%.

Adaptive White-box Attack: To be consistent with existing defenses, our model performs conventional softmax predic-

TABLE 1: Two network architectures: CNN-6 (MNIST, FMNIST) and ResNet-110 (CIFAR-10,100 and SVHN). Features are extracted in CNN-6 (after Layer 3 and two FC layers) and ResNet-110 (after Layer 3, 4 and FC layer) to impose the proposed $\mathcal{L}_{\text{PC}}$. Auxiliary branches are shown in green color.

Layer #	6-Conv Model	ResNet-110
1	[Conv(32, 5 × 5) PReLu(2 × 2)] ×2	Conv(16, 3 × 3) + BN ReLU(2 × 2)
2	[Conv(64, 5 × 5) PReLu(2 × 2)] ×2	[Conv(16, 1 × 1) + BN Conv(16, 3 × 3) + BN Conv(64, 1 × 1) + BN] ×12
3	[Conv(128, 5 × 5) PReLu(2 × 2)] ×2	[Conv(32, 1 × 1) + BN Conv(32, 3 × 3) + BN Conv(128, 1 × 1) + BN] ×12
	GAP → $\mathcal{L}_{\text{PC}}$	(GAP→FC(512) → $\mathcal{L}_{\text{PC}}$)
4	FC(512) → $\mathcal{L}_{\text{PC}}$	[Conv(64, 1 × 1) + BN Conv(64, 3 × 3) + BN Conv(256, 1 × 1)+ BN] ×12 GAP → $\mathcal{L}_{\text{PC}}$
5	FC(64) → $\mathcal{L}_{\text{PC}}$	FC(1024) → $\mathcal{L}_{\text{PC}}$
6	FC(10) → $\mathcal{L}_{\text{CE}}$	FC(100/10) → $\mathcal{L}_{\text{CE}}$

TABLE 2: Adversarial settings of our experiments. α, μ, i respectively denote the step-size, the decay-factor and the number of attack steps for a perturbation budge ϵ.

Attack	Parameters	ℓ_p Norm
FGSM	$i = 1$	ℓ_∞
BIM	$\alpha = \epsilon/10, i = 10$	ℓ_∞
MIM	$\alpha = \epsilon/10, i = 10, \mu = 1.0$	ℓ_∞
C&W	learning rate = 0.01, $i = 1000$	ℓ_2
PGD	$\alpha = \epsilon/4, i = 10$	ℓ_∞

TABLE 3: Robustness of our model in *white-box* and *black-box* settings. Adversarial samples generated in the *black-box* settings (i.e., transfer attack) show negligible attack potential against our models. Here ϵ is the perturbation size and c is the initial constant for C&W attack. It can be seen that AdvTrain further complements the robustness of our models.

		White-Box Setting					Black-Box Setting				
Training	**No Attack**	**FGSM**	**BIM**	**C&W**	**MIM**	**PGD**	**FGSM**	**BIM**	**C&W**	**MIM**	**PGD**
MNIST ($\epsilon = 0.3, c = 10$)											
Softmax	98.71	4.9	0.0	0.2	0.01	0.0	23.0	17.8	20.9	14.8	11.9
Ours	**99.53**	31.1	23.3	29.1	24.7	19.9	78.3	72.7	77.2	74.5	69.5
Ours + AdvTrain$_{\text{FGSM}}$	99.44	**53.1**	36.6	40.9	37.0	34.5	**85.6**	81.0	82.3	81.4	78.2
Ours + AdvTrain$_{\text{PGD}}$	99.28	49.8	**40.3**	**46.0**	**41.4**	**39.8**	85.2	**81.9**	**83.5**	**82.8**	**80.8**
CIFAR-10 ($\epsilon = 0.03, c = 0.1$)											
Softmax	90.80	21.4	0.0	0.6	0.0	0.01	39.0	30.1	31.8	30.9	29.1
Ours	90.45	67.7	32.6	37.3	33.2	27.2	85.5	83.7	83.3	81.9	76.4
Ours + AdvTrain$_{\text{FGSM}}$	91.28	**75.8**	45.9	5.7	44.7	42.5	**88.9**	87.6	87.4	88.2	84.5
Ours + AdvTrain$_{\text{PGD}}$	**91.89**	74.9	**46.0**	**51.8**	**49.3**	**46.7**	88.5	**88.3**	**88.2**	**88.5**	**88.8**
CIFAR-100 ($\epsilon = 0.03, c = 0.1$)											
Softmax	**72.65**	20.0	4.2	1.1	3.52	0.17	40.9	34.3	37.1	35.5	30.7
Ours	71.90	56.9	28.0	31.1	28.7	25.9	65.3	64.5	64.1	64.8	62.8
Ours + AdvTrain$_{\text{FGSM}}$	69.11	**61.3**	32.3	35.2	33.3	31.4	**66.1**	65.2	65.7	65.5	63.4
Ours + AdvTrain$_{\text{PGD}}$	68.32	60.9	**34.1**	**36.7**	**33.7**	**36.1**	65.9	**66.1**	**66.7**	**66.1**	**66.7**
F-MNIST ($\epsilon = 0.3, c = 10$)											
Softmax	**91.51**	8.7	0.1	0.2	0.0	0.0	46.7	29.3	30.8	29.5	26.0
Ours	91.32	29.0	22.0	23.9	21.8	20.3	84.8	79.0	79.2	78.4	76.3
Ours + AdvTrain$_{\text{FGSM}}$	91.03	**55.1**	37.5	41.7	40.6	35.3	**89.1**	87.0	87.7	87.9	85.2
Ours + AdvTrain$_{\text{PGD}}$	91.30	47.2	**40.1**	**44.6**	**41.3**	**40.7**	88.2	**88.0**	**88.2**	**88.3**	**89.7**
SVHN ($\epsilon = 0.03, c = 0.1$)											
Softmax	93.45	30.6	6.2	7.1	7.3	9.6	48.1	30.3	31.4	33.5	21.5
Ours	**94.36**	69.3	37.1	39.2	41.0	33.7	77.4	73.1	76.4	74.0	70.1
Ours + AdvTrain$_{\text{FGSM}}$	94.18	**80.1**	47.4	51.9	45.6	40.5	**90.1**	87.4	88.0	87.6	84.4
Ours + AdvTrain$_{\text{PGD}}$	**94.36**	76.5	**48.8**	**54.8**	**47.1**	**47.7**	88.7	**88.2**	**89.2**	**88.6**	**89.3**

tion at inference time. The reported results in Table 3 are therefore for robustness against standard *white-box* attack settings (where the adversarial samples are generated by maximizing the softmax loss only). Here, we also experiment in an *adaptive white-box* setting, where the attack is performed on the joint PC+CE loss as per Eq. 8. This means that the adversary now has access to the learned class prototypes that are specific to our proposed training regime. Results in Table 4 indicate a negligible performance drop in under *adaptive white-box* settings.

TABLE 4: Robustness in *adaptive white-box* attack settings. The performances for conventional attacks (where CE is the adversarial loss) are shown in blue. * indicates adversarially trained models.

Training	**No Attack**	**FGSM**	**BIM**	**MIM**	**PGD**
CIFAR-10 ($\epsilon = 0.03$)					
Ours	90.45	66.90 (67.7)	31.29 (32.6)	32.84 (33.2)	27.09 (27.2)
Ours$^*_{\text{FGSM}}$	91.28	74.24 (75.8)	44.05 (45.9)	43.77 (44.7)	41.32 (42.5)
Ours$^*_{\text{PGD}}$	91.89	74.31 (74.9)	44.85 (46.0)	47.31 (49.3)	44.75 (46.7)
F-MNIST ($\epsilon = 0.3$)					
Ours	91.32	28.1 (29.0)	21.7 (22.0)	20.3 (21.8)	19.5 (20.3)
Ours$^*_{\text{FGSM}}$	91.03	53.3 (55.1)	36.0 (37.5)	39.3 (40.6)	34.7 (35.3)
Ours$^*_{\text{PGD}}$	91.30	46.0 (47.2)	40.1 (40.1)	40.7 (41.3)	39.7 (40.7)

Adversarial Training (AdvTrain) has been shown to enhance many recently proposed defense methods [44]. We also evaluate the impact of AdvTrain in conjunction with our proposed defense. For this, we jointly train our model on clean and attacked samples, which are generated using FGSM [2] and PGD [17] by uniformly sampling ϵ from an interval of [0.1, 0.5] for MNIST and F-MNIST and [0.01, 0.05] for CIFAR and SVHN. Results in Table 3 indicate that AdvTrain further complements our method and provides an enhanced robustness under both *black-box* and *white-box* attack settings.

6.2 Comparison with Existing Defenses

We compare our method with recently proposed state-of-the-art proactive defense mechanisms, which alter the network or use modified training loss functions. To this end, we compare with [31], which injects adversarial examples into the training set and generates new samples at each iteration. We also compare with [16], which introduces an Adaptive Diversity Promoting (ADP) regularizer to improve adversarial robustness. Further, we compare with an input gradient regularizer mechanism [33] that penalizes the degree to which input perturbations can change a model's predictions by regularizing the gradient of the cross-entropy loss. Finally, we compare with the current state-of-the-art Min-Max optimization based defense [17], which augments the training data with adversarial examples, causing the maximum gradient increments to the loss within a specified l_∞ norm. The results in Tables 5 , 6 and 7 in terms of retained classification accuracy on different datasets show that our method significantly outperforms all existing defense schemes by a large margin. The performance gain is more pronounced for the strongest iterative attacks (e.g. C&W and PGD) with large perturbation budget ϵ. For example, our method achieves a relative gain of 20.6% (AdvTrain models) and 41.4% (without AdvTrain) compared to the 2$^{\text{nd}}$ best methods on the CIFAR-10 and MNIST datasets, respectively, for the PGD attack. On the CIFAR-100 dataset, for the strongest PGD attack with $\epsilon = 0.01$, the proposed method achieves 38.9% compared with 18.3% by ADP [16], which, to the best of our knowledge, is the only method in the literature evaluated on the CIFAR-100 dataset. Our results further indicate that adversarial training consistently

TABLE 5: Comparison on **CIFAR-10** dataset for *white-box* adversarial attacks (numbers shows robustness, higher is better). * sign denotes adversarially trained models. For our model, we report results without adversarial training (Ours) and with adversarially generated images from FGSM (Ours$^*_f$) and PGD (Ours$^*_p$) attacks.

Attacks	Params.	Baseline	AdvTrain [31]*	Yu *et al.* [45]*	Ross *et al.* [33]*	Pang *et al.* [16]*	Madry *et al.* [17]*	Ours	Ours$^*_f$	Ours$^*_p$
No Attack	-	90.8	84.5	83.1	86.2	90.6	87.3	90.5	91.3	**91.9**
FGSM	$\epsilon = 0.02$	36.5	44.3	48.5	39.5	61.7	71.6	72.5	**80.8**	78.5
	$\epsilon = 0.04$	19.4	31.0	38.2	20.8	46.2	47.4	56.3	**70.5**	69.9
BIM	$\epsilon = 0.01$	18.5	22.6	62.7	19.0	46.6	64.3	62.9	67.9	**74.5**
	$\epsilon = 0.02$	6.1	7.8	39.3	6.9	31.0	49.3	40.1	51.2	**57.3**
MIM	$\epsilon = 0.01$	23.8	23.9	-	24.6	52.1	61.5	64.3	68.8	**74.9**
	$\epsilon = 0.02$	7.4	9.3	-	9.5	35.9	46.7	42.3	53.8	**60.0**
C&W	$c = 0.001$	61.3	67.7	82.5	72.2	80.6	84.5	84.3	91.0	**91.3**
	$c = 0.01$	35.2	40.9	62.9	47.8	54.9	65.7	63.5	72.9	**73.7**
	$c = 0.1$	0.6	25.4	40.7	19.9	25.6	47.9	41.1	55.7	**60.5**
PGD	$\epsilon = 0.01$	23.4	24.3	-	24.5	48.4	67.7	60.1	68.3	**75.7**
	$\epsilon = 0.02$	6.6	7.8	-	8.5	30.4	48.5	39.3	50.6	**58.5**

TABLE 6: Comparison on **MNIST** dataset for *white-box* adversarial attacks (numbers shows robustness, higher is better). * sign denotes adversarially trained models. For our model, we report results without adversarial training (Ours) and with adversarially generated images from FGSM (Ours$^*_f$) and PGD (Ours$^*_p$) attacks.

Attacks	Params.	Baseline	AdvTrain [31]*	Yu *et al.* [45]	Ross *et al.* [33]	Pang *et al.* [16]	Madry *et al.* [17]*	Ours	Ours$^*_f$	Ours$^*_p$
No Attack	-	98.7	99.1	98.4	99.2	**99.5**	98.8	**99.5**	99.4	99.3
FGSM	$\epsilon = 0.1$	58.3	73.0	91.6	91.6	96.3	**97.3**	97.1	97.2	96.5
	$\epsilon = 0.2$	12.9	52.7	70.3	60.4	52.8	**96.4**	70.6	80.0	77.9
BIM	$\epsilon = 0.1$	22.5	62.0	88.1	87.9	88.5	-	90.2	92.0	**92.1**
	$\epsilon = 0.15$	12.2	18.7	77.1	32.1	73.6	-	76.3	76.5	**77.3**
MIM	$\epsilon = 0.1$	58.3	64.5	-	83.7	92.0	-	92.1	92.7	**93.0**
	$\epsilon = 0.15$	16.1	28.8	-	29.3	77.5	-	77.7	80.2	**82.0**
C&W	$c = 0.1$	61.6	71.1	89.2	88.1	97.3	**97.7**	**97.7**	97.1	97.6
	$c = 1.0$	30.6	39.2	79.1	75.3	78.1	**93.4**	80.4	87.3	91.2
	$c = 10.0$	0.2	17.0	37.6	20.0	23.8	-	29.1	39.7	**46.0**
PGD	$\epsilon = 0.1$	50.7	62.7	-	77.0	82.8	**95.4**	83.6	93.7	93.9
	$\epsilon = 0.15$	6.3	31.9	-	44.2	41.0	**93.7**	62.5	78.8	80.2

TABLE 7: Comparison on **CIFAR-100** dataset for *white-box* adversarial attacks (numbers shows robustness, higher is better). * sign denotes adversarially trained models. For our model, we report results without adversarial training (Ours) and with adversarially generated images from FGSM (Ours$^*_f$) and PGD (Ours$^*_p$) attacks.

Attacks	Params.	Baseline	ADP [16]	Ours	Ours$^*_f$	Ours$^*_p$
No Attack	-	**72.6**	70.2	71.9	69.1	68.3
BIM	$\epsilon = 0.005$	21.6	26.2	44.8	55.1	**55.7**
	$\epsilon = 0.01$	10.1	14.8	39.8	46.2	**46.9**
MIM	$\epsilon = 0.005$	24.2	29.4	46.1	56.7	**57.1**
	$\epsilon = 0.01$	11.2	17.2	40.6	43.8	**45.9**
PGD	$\epsilon = 0.005$	26.6	32.1	42.2	53.6	**55.0**
	$\epsilon = 0.01$	11.7	18.3	38.9	40.1	**44.0**

complements our method and augments its performance across all evaluated datasets.

Additionally, we compare our model's performance with a similar method proposed by Song *et al.* [46] in Table 8. The results indicate that our proposed approach outperforms [46] by a significant margin. Besides, a clear improvement in the performance, there are several other distinguishing features from [46] as follows: **(a)** Our approach is based on a "deeply-supervised" loss that prevents changes to the outputs within the limited perturbation budget. This supervision paradigm is the main contributing factor behind our improved results (see Table 11). **(b)** [46] focuses on domain adaption between adversarial and natural samples without any constraint on the intermediate feature representations. In contrast, we explicitly enforce the hidden layer activations to be maximally separated in our network design. **(c)** [46] only considers adversarially trained models, while we demonstrate clear improvements both with and without adversarial training (a more challenging setting). For a fair comparison, we have followed the exact model settings used in [46] for the results reported in Table 8.

TABLE 8: Comparison of our approach with [46] on 4 datasets.

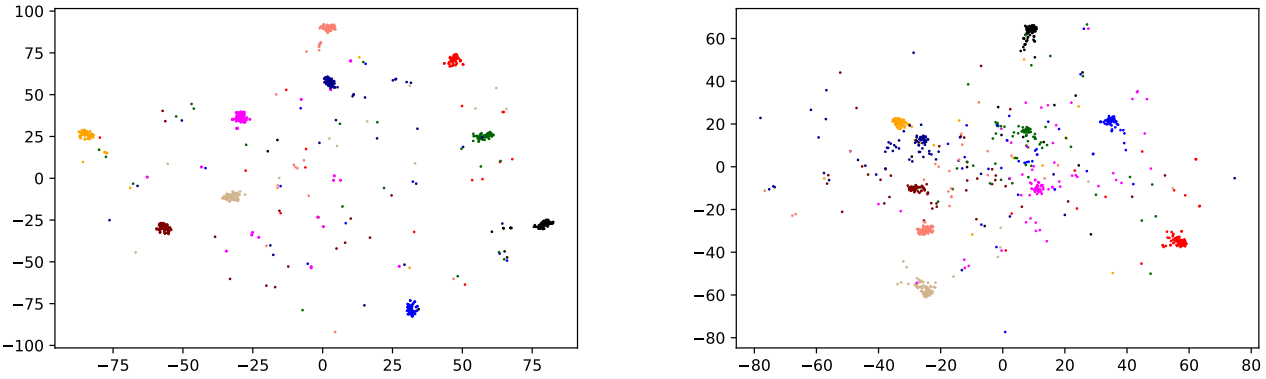
Dataset	Method	Clean	FGSM	MIM	PGD
F-MNIST ($\epsilon = 0.1$)	[46]	85.5	78.2	68.8	68.6
	Ours	**91.3**	**86.6**	**80.1**	**79.4**
SVHN ($\epsilon = 0.02$)	[46]	82.9	57.2	53.9	53.2
	Ours	**94.4**	**87.1**	**82.2**	**80.7**
CIFAR-10 ($\epsilon = 4/255$)	[46]	84.8	60.7	59.0	58.1
	Ours	**91.9**	**85.3**	**70.1**	**69.4**
CIFAR-100 ($\epsilon = 4/255$)	[46]	61.6	29.3	27.3	26.2
	Ours	**71.9**	**49.1**	**40.7**	**38.6**

6.3 Comparison with Other Distance-based Losses

Here, we report performances when other distance-based loss functions are used within our proposed deeply-supervised framework. In this case, the proposed PCL is replaced with popular loss functions that directly incorporate distance metrics, e.g. center loss, contrastive loss and triplet loss. For a fair comparison, we train the same model using cross-entropy ($\mathcal{L}_{\text{CE}}$), center loss ($\mathcal{L}_{\text{CL}}$) [22], contrastive loss ($\mathcal{L}_{\text{Co}}$) [23] and triplet loss ($\mathcal{L}_{\text{T}}$) [24] applied after different layers (similar to our $\mathcal{L}_{\text{PC}}$ loss), and evaluate robustness against different *white-box* attack settings. For $\mathcal{L}_{\text{CL}}$ [22] and $\mathcal{L}_{\text{PC}}$, the models are trained jointly with $\mathcal{L}_{\text{CE}}$ in the last layer. Table 10 provides architectural details of the model used for results reported in Table 9. For the deeply supervised Softmax loss, the cross-entropy loss $\mathcal{L}_{\text{CE}}$ is deployed after layers 3, 4 and 5 in conjunction with additional fully connected and global average pool (GAP) layers, as shown in Table 10. Similarly, for the deeply supervised center loss, the center loss $\mathcal{L}_{\text{CL}}$ is applied after layers 3, 4 and 5. Note that $\mathcal{L}_{\text{CL}}$ is

TABLE 9: Comparison of adversarial robustness for various loss functions (center loss, contrastive loss and triplet loss) on the CIFAR-10 & 100 datasets. It can be seen that deep supervision of distance based loss functions consistently achieve superior performance against various types of adversarial attacks. The proposed prototype conformity loss achieves the best performance overall.

Attack	Softmax	Ours-Center [22]	Ours-Contrastive [23]	Ours-Triplet [24]	Ours-PCL
CIFAR-10					
Clean	**90.80**	86.29	86.9	89.75	90.45
FGSM ($\epsilon = 8/255$)	21.4	27.38	59.3	63.90	**67.7**
BIM ($\epsilon = 8/255$)	0.0	9.4	28.4	31.04	**32.6**
MIM ($\epsilon = 8/255$)	0.0	9.6	29.0	31.80	**33.2**
PGD ($\epsilon = 8/255$)	0.0	6.93	24.5	25.51	**27.2**
CIFAR-100					
Clean	**72.6**	67.8	67.1	68.1	71.9
FGSM ($\epsilon = 8/255$)	20.0	17.8	21.9	24.3	**56.9**
BIM ($\epsilon = 8/255$)	4.2	9.8	13.0	15.3	**28.0**
MIM ($\epsilon = 8/255$)	3.5	8.9	13.5	15.9	**28.7**
PGD ($\epsilon = 8/255$)	0.2	6.1	9.0	9.1	**25.9**



(a) PC Loss (b) Triplet Loss

Fig. 4: t-SNE visualization of 10 classes for the last layer feature embeddings from model trained using Prototype Conformity Loss (*left*) and Triplet Loss (*right*) for CIFAR-100 dataset. The class-wise feature space embeddings for our loss are better separated than for other distance-based metrics for hard datasets like CIFAR-100.

applied jointly with $\mathcal{L}_{\text{CE}}$, as recommended in the original paper. For the deeply supervised contrastive loss $\mathcal{L}_{\text{Co}}$ and triplet loss $\mathcal{L}_{\text{T}}$, the loss function is applied after layers 3, 4 and 5 as shown in Table 10. Note that there is no final fully connected (FC-10) layer at the end in $\mathcal{L}_{\text{Co}}$ and $\mathcal{L}_{\text{T}}$ training.

Results Comparison: Table 9 provides a comparison of our loss function with various other deeply supervised loss functions. The results indicate a clear advantage of our proposed $\mathcal{L}_{\text{PC}}$ and other distance-based loss functions, compared with the standard cross-entropy loss. For CIFAR-10, distance based losses, namely center, contrastive and triplet losses, have a comparable performance to PCL; however, as the number of classes in the training dataset increases (CIFAR-100), robustness against adversarial attacks for the former drastically drops. Among the distance-based loss functions, the proposed PCL shows a consistently better performance, even for harder datasets like CIFAR-100. We attribute this to the inclusion of contrastive constraints within our proposed loss formulation, which jointly consider the samples in a batch to enforce separation (see Fig. 6 for visualization of our training process with PCL).

PCL vs Triplet Loss: To better understand the effectiveness of our proposed PCL compared to the deeply supervised triplet loss, we visualize the last layer test features of a sample of 10 classes (randomly picked) in Fig. 4 for the CIFAR-100 dataset. Although both loss functions achieve a comparable performance on clean test images (68.1% for Triplet Loss and 71.9% for PCL), our method shows enhanced robustness against strong iterative *white-box* attacks where other distance-based loss functions fail. This is achieved by the explicitly included contrastive constraints in PC Loss.

6.4 Ablation Analysis

$\mathcal{L}_{\text{PC}}$ at Different Layers: We investigate the impact of our proposed prototype conformity loss ($\mathcal{L}_{\text{PC}}$) at different depths of the network. Specifically, as shown in Table 11, we apply $\mathcal{L}_{\text{PC}}$ individually after each layer (see Table 1 for architectures) and in different combinations. We report the achieved results on the CIFAR-10 dataset for clean and perturbed samples (using FGSM and PGD attacks) in Table 11. The network without any $\mathcal{L}_{\text{PC}}$ loss is equivalent to a standard softmax trained model. It achieves good performance on clean images, but fails under both *white-box* and *black-box* attacks (see Table 3). The models with $\mathcal{L}_{\text{PC}}$ loss in early layers are unable to separate deep features class-wise (also visualized in Fig. 5) , thereby resulting in inferior performance. Our proposed $\mathcal{L}_{\text{PC}}$ loss has maximum impact in the deeper layers of the network. This justifies our

TABLE 10: ResNet-110 network architecture used for deep Softmax supervision ($\mathcal{L}_{\text{CE}}$) and deep center loss ($\mathcal{L}_{\text{CL}}$), contrastive loss ($\mathcal{L}_{\text{Co}}$) or triplet loss ($\mathcal{L}_{\text{T}}$) supervision for CIFAR-10 dataset. All these distance-based loss functions are denoted with ($\mathcal{L}_{\text{Dist}}$) below. Auxiliary branches are shown in green color.

	ResNet-110 Softmax Supervision	**ResNet-110** Center/Contrastive/Triplet Loss Supervision
1	Conv(16, 3 × 3) + BN ReLU(2 × 2)	Conv(16, 3 × 3) + BN ReLU(2 × 2)
2	[Conv(16, 1 × 1) + BN Conv(16, 3 × 3) + BN Conv(64, 1 × 1) + BN] ×12	[Conv(16, 1 × 1) + BN Conv(16, 3 × 3) + BN Conv(64, 1 × 1) + BN] ×12
3	[Conv(32, 1 × 1) + BN Conv(32, 3 × 3) + BN Conv(128, 1 × 1) + BN] ×12 (GAP→FC(512)→FC(10) → $\mathcal{L}_{\text{CE}}$)	[Conv(32, 1 × 1) + BN Conv(32, 3 × 3) + BN Conv(128, 1 × 1) + BN] ×12 (GAP→FC(512) → $\mathcal{L}_{\text{Dist}}$)
4	[Conv(64, 1 × 1) + BN Conv(64, 3 × 3) + BN Conv(256, 1 × 1)+ BN] ×12 (GAP→ FC(10)→ $\mathcal{L}_{\text{CE}}$)	[Conv(64, 1 × 1) + BN Conv(64, 3 × 3) + BN Conv(256, 1 × 1)+ BN] ×12 GAP → $\mathcal{L}_{\text{Dist}}$
5	GAP→FC(1024)→FC(10) → $\mathcal{L}_{\text{CE}}$	FC(1024) → $\mathcal{L}_{\text{Dist}}$
6	-	FC(10) → $\mathcal{L}_{\text{CE}}$ (only for Center Loss)

TABLE 11: **Ablation Analysis** with $\mathcal{L}_{\text{PC}}$ applied at different layers of ResNet-110 (Table 1) for CIFAR-10 dataset.

Layer #	No Attack $\epsilon = 0$	FGSM $\epsilon = 0.03$	PGD $\epsilon = 0.03$
None	90.80	21.40	0.01
Layer 1	74.30	23.71	0.01
Layer 2	81.92	30.96	8.04
Layer 3	88.75	33.74	10.47
Layer 4	90.51	39.90	11.90
Layer 5	**91.11**	47.02	13.56
Layer 4+5	90.63	55.36	20.70
Layer 3+4+5	90.45	**67.71**	**27.23**

choice of different layers for $\mathcal{L}_{\text{PC}}$ loss, indicated in Table 1.

Feature Space Visualizations: We visualize the 2-D t-SNE plots of features obtained from Layer 1, 3, 4 and 5 of our best model for the MNIST, CIFAR-10 and SVHN datasets in Fig. 5. We can see that, for deeper layers, feature clusters are more separable, with enhanced intra-class compactness, thereby reducing the polytope overlap of different classes, leading to a lower adversary success for a bounded perturbation $\|\boldsymbol{\delta}\|_p \leq \epsilon$.

Illustration of Learning Process: Fig. 6 illustrates the gradual progression of our learning process, by visualizing the class-wise 2-D deep features after different epochs. These features are plotted by reducing the dimensions of the penultimate layer to two neurons, and training the model for 300 epochs, with a learning rate of 0.1 (×0.1 at epochs 122 and 250). We note the following: *(i)* our proposed loss gradually causes the learnt features to become more discriminative and well seperated, *(ii)* the inter-class distances gradually increase over the course of training, making it harder for an adversary to cross the decision boundary within the specified perturbation budget ϵ. These observations provide evidence that introducing the proposed $\mathcal{L}_{\text{PC}}$ at multiple layers enhances discrimination and improves model performance.

Timing Comparison for Various Training Losses: Comparing the time taken per epoch during training for various distance-based loss functions can give us further insights into the effectiveness of the method. In the cases of the contrastive and triplet losss, the training time is higher due to the formation of pairs and triplets, respectively (see Table 12). Our Prototype Conformity Loss not only achieves higher levels of robustness compared to other methods but also requires a shorter training time. In Table 12, we also show the number of epochs required when training a model to achieve the robustness mentioned in Table 9. It can be seen that, in the case of the contrastive and the triplet losses, a greater number of epochs is required to effectively separate the intermediate feature embeddings in order to achieve adversarial robustness.

TABLE 12: Comparison of various loss functions (cross-entropy, center, contrastive, triplet and prototype conformity loss) on the basis of training time taken per epoch (seconds) and the number of epochs required during training to achieve the robustness stated in Table 9 for the CIFAR-10 dataset.

Parameter	$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{CL}}$ [22]	$\mathcal{L}_{\text{Co}}$ [23]	$\mathcal{L}_{\text{T}}$ [24]	$\mathcal{L}_{\text{PC}}$
Time per epoch	25.9	102.0	133.5	138.1	102.9
Number of epochs	164	300	400	400	300

TABLE 13: **Transferability Test** on CIFAR-10: PGD adversaries are generated with $\epsilon = 0.03$, using the source network, and then evaluated on target model. Underline denotes robustness against *white-box* attack. Note that adversarial samples generated on our model are highly transferable to other models as *black-box* (transfer) attacked images.

Source \ Target	VGG-19	AdvTrain [31]	Madry *et al.* [17]	Ours
VGG-19	0.00	16.20	52.71	88.80
AdvTrain [31]	12.43	0.00	49.80	72.53
Madry *et al.* [17]	58.91	67.32	43.70	71.72
Ours	50.31	61.02	66.70	49.10

6.5 Transferability Test

We investigate the transferability of PGD adversaries on the CIFAR-10 dataset between a standard VGG-19 model, adversarially trained VGG-19 [31], Madry *et al.*'s [17] and our model. We report the accuracy of target models (columns) on adversarial samples generated from source models (rows) in Table 13. Our results yield the following findings:

Improved *black-box* robustness: As noted in [19], a model that gives a false sense of security due to obfuscated gradients can be identified if the *black-box* attacks are stronger than the *white-box* ones. In other words, the robustness of such a model under *white-box* settings is higher than under *black-box* settings. It was shown in [19] that most of the existing defenses suffer from obfuscated gradients. Madry *et al.*'s approach [17] was endorsed by [19] to not cause obfuscated gradients. The comparison in Table 13 shows that our method outperforms [17].

Similar architectures increase transferability: Changing the source and target network architectures decreases the transferability of an attack. The same architectures (e.g. VGG-19 and its AdvTrain counterpart, as in Table 13) show increased robustness against *black-box* attacks generated from each other.

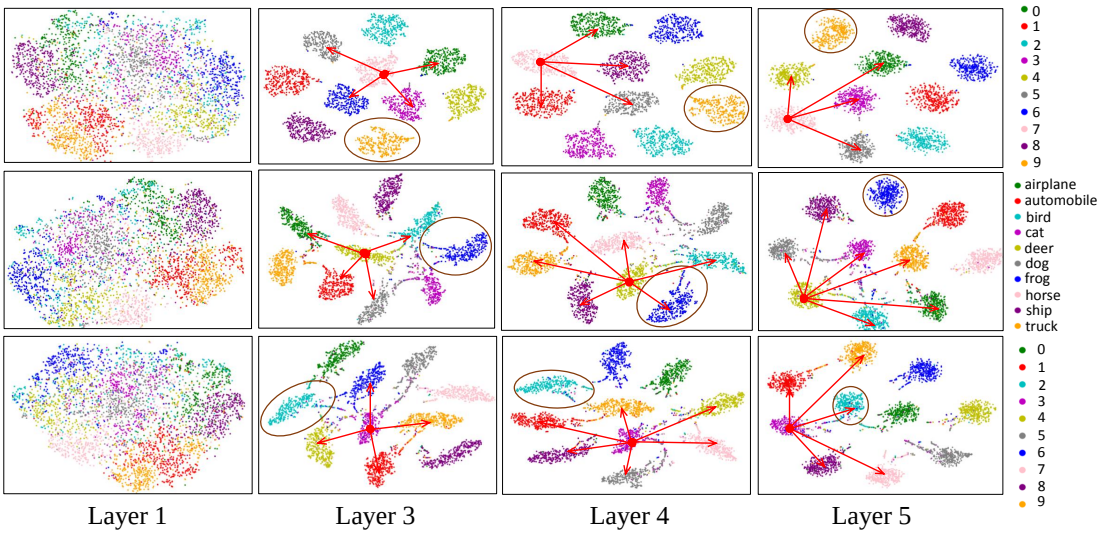


Fig. 5: t-SNE plots of features at different layers, where $\mathcal{L}_{\text{PC}}$ is applied, on MNIST (top), CIFAR-10 (middle) and SVHN (bottom) datasets. The inter-class distances and intra-class compactness (denoted by red arrows and brown circles, respectively) increase along the depth of the network. Best viewed in color.

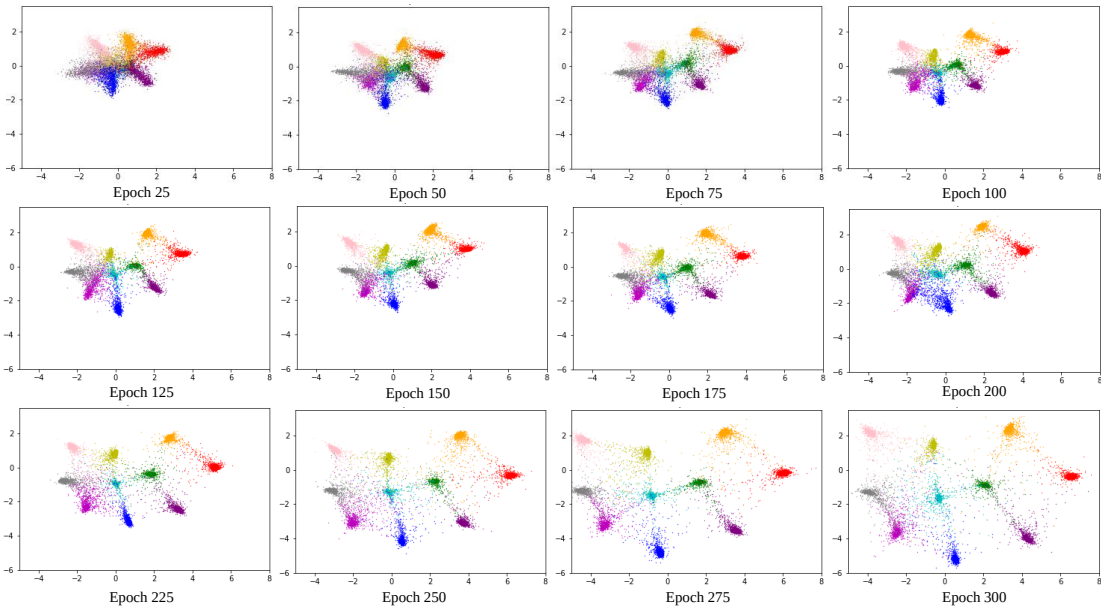


Fig. 6: Class-wise distribution of penultimate layer features (2 neurons only) on CIFAR-10 dataset. As the training progresses, the class prototypes move farther away from each other, resulting in a guaranteed robustness against adversarial attacks. Best viewed in color.

6.6 Identifying Obfuscated Gradients

Recently, Athalye *et al.* [19] were successful in breaking several defense mechanisms in the *white-box* settings by identifying that they exhibit a false sense of security. They call this phenomenon *gradient masking*. Below, we discuss how our defense mechanism does not cause gradient masking on the basis of the characteristics defined in [19], [47].

Iterative attacks perform better than one-step attacks: Our evaluations in Fig. 7 indicate that stronger iterative attacks (e.g. BIM, MIM, PGD) in the *white-box* settings are more successful at attacking the defense models than single-step attacks (FGSM in our case).

Robustness against *black-box* settings is higher than *white-box* settings: In *white-box* settings, the adversary has complete knowledge of the model, so attacks should be more successful. In other words, if a defense does not suffer from obfuscated gradients, robustness of the model against *white-box* settings should be inferior to that in the *black-box* settings. Our extensive evaluations in Table 3 show that the proposed defense follows this trend and therefore does not obfuscate gradients.

Increasing the distortion bound (ϵ) decreases the robustness of defense: On increasing the perturbation size, the success rate of the attack method should significantly

increase monotonically. For an unbounded distortion, the classifier should exhibit 0% robustness to the attack, which again is true in our case (see Fig. 7).

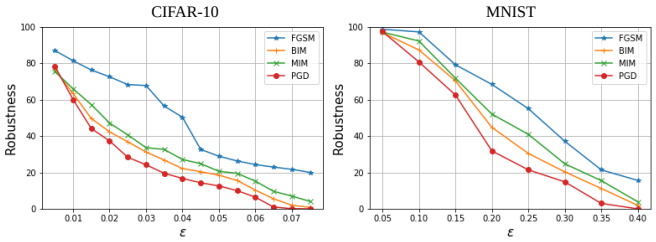


Fig. 7: Robustness of our model (without adversarial training) against *white-box* attacks for various perturbation budgets.

7 Conclusion

Our findings provide evidence that the adversary's task can be made difficult by incorporating a maximal separation constraint in the objective function of DNNs, which conventional cross-entropy loss fails to impose. Our theory and the experiments indicate that if the adversarial polytopes for samples belonging to different classes are non-overlapping, the adversary cannot find a viable perturbation within the allowed budget. We extensively evaluate the proposed model against a diverse set of attacks (both single-step and iterative) in *black-box* and *white-box* settings and show that the proposed model maintains its high robustness in all cases. Through empirical evaluations, we further demonstrate that the achieved performance is not due to obfuscated gradients, thus the proposed model can provide significant security against adversarial vulnerabilities in deep networks.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[3] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2018.

[4] E. Ackerman, "How drive. ai is mastering autonomous driving with deep learning," *IEEE Spectrum Magazine*, vol. 1, 2017.

[5] M. Hayat, S. H. Khan, N. Werghi, and R. Goecke, "Joint registration and representation learning for unconstrained face identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2767–2776.

[6] M. Hayat, M. Bennamoun, and S. An, "Deep reconstruction models for image set classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 4, pp. 713–727, 2014.

[7] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, "Foveation-based mechanisms alleviate adversarial examples," *arXiv preprint arXiv:1511.06292*, 2015.

[8] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *arXiv preprint arXiv:1705.02900*, 2017.

[9] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=Sk9yuql0Z

[10] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image super-resolution as a defense against adversarial attacks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1711–1724, 2019.

[11] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rkZvSe-RZ

[12] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016. [Online]. Available: http://arxiv.org/abs/1607.02533

[13] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[14] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 854–863.

[15] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1778–1787.

[16] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97. PMLR, 09–15 Jun 2019, pp. 4970–4979. [Online]. Available: http://proceedings.mlr.press/v97/pang19a.html

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[18] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial logit pairing," *arXiv preprint arXiv:1803.06373*, 2018.

[19] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *ICML*, 2018, pp. 274–283. [Online]. Available: http://proceedings.mlr.press/v80/athalye18a.html

[20] M. M. Naseer, S. H. Khan, M. H. Khan, F. S. Khan, and F. Porikli, "Cross-domain transferability of adversarial perturbations," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 885–12 895.

[21] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, and L. Shao, "Adversarial defense by restricting the hidden space of deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3385–3394.

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

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

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

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

[28] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9185–9193.

[29] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.

[30] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SyJ7ClWCb

[31] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[32] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in *International Conference on Learning Representations*, 2016.

[33] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] D. Jakubovitz and R. Giryes, "Improving dnn robustness to adversarial attacks using jacobian regularization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 514–529.

[35] J. Lin, C. Gan, and S. Han, "Defensive quantization: When efficiency meets robustness," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=ryetZ20ctX

[36] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar, "Stochastic activation pruning for robust adversarial defense," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=H1uR4GZRZ

[37] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=S18Su--CW

[38] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=BkJ3ibb0-

[39] T. Na, J. H. Ko, and S. Mukhopadhyay, "Cascade adversarial machine learning regularized with a unified embedding," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=HyRVBzap-

[40] M. De Berg, O. Cheong, O. Devillers, M. Van Kreveld, and M. Teillaud, "Computing the maximum overlap of two convex polygons under translations," *Theory of computing systems*, vol. 31, no. 5, pp. 613–628, 1998.

[41] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2624–2637, 2013.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

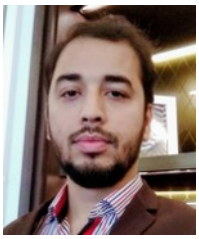
[44] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie *et al.*, "Adversarial attacks and defences competition," in *The NIPS'17 Competition: Building Intelligent Systems*. Springer, 2018, pp. 195–231.

[45] F. Yu, C. Liu, Y. Wang, and X. Chen, "Interpreting adversarial robustness: A view from decision surface in input space," 2019. [Online]. Available: https://openreview.net/forum?id=rylV6i09tX

[46] C. Song, K. He, L. Wang, and J. E. Hopcroft, "Improving the generalization of adversarial training with domain adaptation," in *International Conference on Learning Representations*, 2019.

[47] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," *arXiv preprint arXiv:1807.06732*, 2018.

Aamir Mustafa is currently a Ph.D. student at Computer Laboratory, University of Cambridge. He received B.Tech from National Institute of Technology, Srinagar, India in 2018. His current research interests include computer vision, computer graphics, pattern recognition, and machine learning. His research interests in particular include building robust and reliable machine learning systems for real world problems.

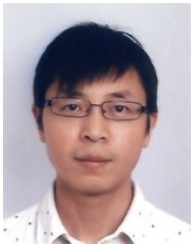


Salman H. Khan received the Ph.D. degree from The University of Western Australia, in 2016. His Ph.D. thesis received an honorable mention on the Deans List Award. From 2016 to 2018, he was a Research Scientist with Data61, CSIRO. He has been a Senior Scientist with Inception Institute of Artificial Intelligence, since 2018, and an Adjunct Lecturer with Australian National University, since 2016. He has served as a program committee member for several premier conferences, including CVPR, ICCV, and ECCV. In 2019, he was awarded the outstanding reviewer award at CVPR.

Munawar Hayat received his PhD from The University of Western Australia (UWA). His PhD thesis received multiple awards, including the Deans List Honorable Mention Award and the Robert Street Prize. After his PhD, he joined IBM Research as a postdoc and then moved to the University of Canberra as an Assistant Professor. He is currently a Senior Scientist at Inception Institute of Artificial Intelligence, UAE. Munawar was granted two US patents, and has published over 30 papers at leading venues in his field, including TPAMI, IJCV, CVPR, ECCV and ICCV.

Roland Goecke is Professor of Affective Computing at the University of Canberra, where he is the Director of the Human-Centred Technology Research Centre and leads the Vision and Sensing Group. He received his Masters degree in Computer Science from the University of Rostock, Germany, in 1998 and his PhD in Computer Science from the Australian National University, Canberra, Australia, in 2004. His research interests are in affective computing, pattern recognition, computer vision, human-computer interaction and multimodal signal processing.

Jianbing Shen is currently acting as the Lead Scientist with the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE. He is also an Adjunct Professor with the School of Computer Science, Beijing Institute of Technology. He has published more than 120 top journal and conference papers, and eleven papers are selected as the ESI Hightly Cited Papers. His current research interests include computer vision, deep learning, autonomous driving, medical image analysis, and intelligent systems. He is an Associate Editor of *IEEE Trans. on Image Processing*, *IEEE Trans. on Neural Networks and Learning Systems* & *Pattern Recognition*.



Ling Shao is the CEO of the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. He was Chair Professor and Director of the Artificial Intelligence Laboratory at the University of East Anglia, Norwich, UK. He received the B.Eng. degree in Electronic and Information Engineering from the University of Science and Technology of China (USTC), the M.Sc degree in Medical Image Analysis and the PhD degree in Computer Vision at the Robotics Research Group from the University of Oxford. His research interests include Computer Vision, Deep Learning/Machine Learning, Multimedia, and Image/Video Processing. He has published over 300 papers at top venues such as TPAMI, TIP, IJCV, ICCV, CVPR, ECCV, etc.