Adversarial Defence by Diversified Simultaneous Training of Deep Ensembles

Bo Huang[1 *], **Zhiwei Ke**[1,2 *], **Yi Wang**[1 †], **Wei Wang**[3], **Linlin Shen**[2,4], **Feng Liu**[2]

[1]Dongguan University of Technology, Dongguan, China
[2]Computer Vision Institute, Shenzhen University, Shenzhen, China
[3]The University of New South Wales, Sydney, Australia
[4]Shenzhen Institute of Artificial Intelligence & Robotics for Society
huangbo1024@gmail.com, kezhiwei2018@email.szu.edu.cn,
wangyi@dgut.edu.cn, weiw@unsw.edu.au, {llshen, fengliu}@szu.edu.cn

Abstract

Learning-based classifiers are susceptible to adversarial examples. Existing defence methods are mostly devised on individual classifiers. Recent studies showed that it is viable to increase adversarial robustness by promoting diversity over an ensemble of models. In this paper, we propose adversarial defence by encouraging ensemble diversity on learning high-level feature representations and gradient dispersion in simultaneous training of deep ensemble networks. We perform extensive evaluations under white-box and black-box attacks including transferred examples and adaptive attacks. Our approach achieves a significant gain of up to 52% in adversarial robustness, compared with the baseline and the state-of-the-art method on image benchmarks with complex data scenes. The proposed approach complements the defence paradigm of adversarial training, and can further boost the performance. The source code is available at https://github.com/ALIS-Lab/AAAI2021-PDD.

Introduction

In many security applications, learning-based classifiers are posed in an adversarial environment and susceptible to intelligent attackers (Biggio et al. 2013; Dalvi et al. 2004; Kurakin et al. 2018). In particular, *adversarial examples* can be generated by adversarial learning of image perturbations that are imperceptible to human eyes (Carlini and Wagner 2017b; Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2017; Kurakin et al. 2018). Such adversarial examples can induce wrong decisions by systems and often not easily detected (Carlini and Wagner 2017a) nor prevented (Athalye, Carlini, and Wagner 2018).

Depending on adversarial knowledge, the attack scenarios can be roughly classified into two categories of white-box and black-box accesses. In the *white-box* setting, an attacker can access target model details, typically the loss function, to build adversarial examples via gradient descent (Carlini and Wagner 2017b; Goodfellow, Shlens, and Szegedy 2015). In the *black-box* setting, an attacker cannot directly access the target information but attempting attacks by either transferrable examples built on a surrogate (Demontis et al. 2019) (a.k.a. *transfer-based* attacks) or probing decision boundaries via numeral classification queries (Brendel, Rauber, and Bethge 2018). Thus, a main paradigm of defence is to prevent the gathering of useful information (e.g., the loss gradients) from generating adversarial examples (Athalye, Carlini, and Wagner 2018; Kurakin, Goodfellow, and Bengio 2017).

Existing defence methods mainly focus on improving the performance of **individual models** where there is often an inherent trade-off between classification accuracy and adversarial robustness (Su et al. 2018). Defences deployed on a single model are often circumvented by *adaptive attacks* in the white-box setting. In such circumstances, the defence mechanism itself can be exploited to launch more intelligent attacks under only restrictions of the threat model. For instance, (Athalye, Carlini, and Wagner 2018) proposed to overcome the defence of randomization by introducing Expectation Over Transformation (EOT) and that of gradient shattering by Backward Pass Differentiable Approximation.

By contrast, it is intuitively more difficult to compromise an **ensemble of models** rather than a single one. Ensemble models are widely used to improve model generalizability over the prediction accuracy. In many cases, DNNs are no longer weak classifiers and thus the conventional ensemble methods are no longer effective. Recently, deep ensembles are studied for predictive uncertainties (Lakshminarayanan, Pritzel, and Blundell 2017) and against network deceptions (Liu et al. 2019; Pang et al. 2019; Zhang, Cheng, and Hsieh 2019). In particular, (Pang et al. 2019) proposed a so-called ADP training to improve adversarial robustness of deep ensembles by facilitating the output diversity on *non-maximal* predictive scores of the base models over those less-likely class labels. The gain of adversarial robustness, however, diminishes quickly when the dataset has increased class labels or more complex data scenes. The other limitation of this approach is that the resulting ensemble model is still fairly vulnerable to strong attacks [1].

In this paper, we promote ensemble diversity on high-level feature representation learning. It was shown that fea-

*Equal Contribution
†Corresponding Author (OrcID: 0000-0002-8448-8570)
Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]c.f., Tables 2 and 3

ture maps with similar activation patterns have close semantic implications (Ke et al. 2020; Kim et al. 2017). We are motivated to encourage base models to learn different feature representations at the fully-connected (FC) layer. This is enabled by designing a diversified dropout strategy coupled with a gradient regularization term in simultaneous training of ensemble networks. Our method complements exiting defence paradigms acting on individual models such as adversarial training (Goodfellow, Shlens, and Szegedy 2015) and other data augmentation schemes (Zhang et al. 2019).

Our main contributions are:

1. We propose a novel strategy of diversified learning of high-level feature representations by ensemble networks.
2. We design two regularization schemes in simultaneous training to facilitate the proposed diversified learning.
3. We study ensemble diversity promoted by the proposed approach for adversarial defence from three aspects.

Related Work

Opposing adversarial examples, there are an increasing number of defence methods proposed for hardening DNN models (Kurakin et al. 2018). However, many of them are later defeated by stronger iterative attacks (Carlini and Wagner 2017b) or adaptive adversaries (Athalye, Carlini, and Wagner 2018). So far, one of the most effective defence paradigm is randomization (Carlini and Wagner 2017b), including randomness added to the input (Xie et al. 2018) and randomness added to the model (Dhillon et al. 2018; Feinman et al. 2017; Liu et al. 2018). In particular, (Dhillon et al. 2018) proposed Stochastic Activation Pruning (SAP) of a random subset of activations in DNN's. The scheme is similar to randomized *dropout* (Feinman et al. 2017) in terms of sampling activations but differs in that SAP is applied *post-hoc* to a pre-trained model rather than involved in the iterative training process. (Liu et al. 2018) propose to add random noise layers to a DNN in both the training and testing phases and then ensemble the predictions. This is equivalent to training the original network with an extra regularization of Lipschitz constant. The random self-ensemble approach demonstrates significant improvement of model robustness over strong gradient-based attacks.

The randomization techniques are often used together with an adversarially trained model as in (Dhillon et al. 2018; Meng et al. 2020; Xie et al. 2018). In fact, adversarial training (Goodfellow, Shlens, and Szegedy 2015) has been considered as a standard method for defending against adversarial examples (Madry et al. 2018; Kurakin et al. 2018). It works by mixing normal and adversarially generated examples in the training set to improve the model robustness on small perturbations. Ensemble adversarial training was also proposed for decoupling adversarial examples from the parameters of a trained model to increase the diversity of perturbations seen during training (Tramèr et al. 2018).

Another paradigm of adversarial defence is training with regularizations (Cisse et al. 2017; Ros and Doshi-Velez 2018; Yan, Guo, and Zhang 2018). In particular, DNN trained with input gradient regularization exhibit remarkable robustness against transferred examples that are generated to fool all of the other models (Demontis et al. 2019; Ros and Doshi-Velez 2018). The gradient regularization therein intends to optimize a DNN to have smooth input gradients with respect to its predictions during training. It is able to alter the shape of decision boundaries for interpretable and qualitatively different reasons. One disadvantage of gradient regularization is its computation cost for including input gradients in parameter gradient descent requires taking second derivatives in each mini-batch, which is generally expensive.

The above defence methods are devised on individual models. Recent work are emerged to study the adversarial robustness of ensemble classifiers (Liu et al. 2019; Pang et al. 2019; Zhang, Cheng, and Hsieh 2019). Diversity has been recognized as a very important characteristic in classifier combination. (Liu et al. 2019) defines three types of ensemble diversity by 1) the difference in network structures, 2) disagreements on negative examples, and 3) the posterior distribution, and reports that the type 2 diversity performs better under L_2 attacks by promoting failure independence of the ensemble classifiers.

(Pang et al. 2019) promotes ensemble diversity on the model outputs by proposing an Adaptive Diversity Promoting (ADP) scheme on training the ensemble networks. In the ADP training, two regularization terms are amended to the cross-entropy (CE) loss for simultaneous training. The first term is a Shannon entropy measure of the ensemble predictions. When it is removed, ADP degenerates to *independent* training on the conventional CE loss. The second term is a defined measure of ensemble diversity to encourage the non-maximal predictions of base networks to become mutually orthogonal. When it is removed, ADP training effectively performs label smoothing with a constant smoothing factor (Pang et al. 2019).

On the other hand, (Ilyas et al. 2019) showed that there are robust and non-robust features that have different vulnerabilities with respect to adversarial perturbations. Therefore, it is intuitive to diversify the risk of attack over ensemble models at the feature representation level. Therefore, we are inspired to encourage diversified learning on the feature level for improving adversarial defence in this paper.

Proposed Method

We first introduce necessary notations for describing simultaneous training of deep ensembles. Suppose that the ensemble model $\mathcal{F}$ is composed of K base networks denoted by $F(\mathbf{x};\theta_k)$ for $k = 1, 2, ..., K$. A common strategy for modelling $\mathcal{F}$ is the simple average over all individual predictors, i.e., $\hat{\mathbf{y}}_{\mathcal{F}} = \frac{1}{K}\sum_{k=1}^{K} F(\mathbf{x};\theta_k)$.

In simultaneous training, all the classifiers are trained on the same mini-batch of data in each training iteration. Conventionally, the objective function is simply the ensemble cross-entropy (ECE) loss summed over individual CE losses (Zhou 2012):

$$\mathcal{L}_{\texttt{ECE}} = \sum_{k=1}^{K} \mathcal{L}_{\texttt{CE}}(\hat{\mathbf{y}}_k, \mathbf{y}) \,, \tag{1}$$

where $\hat{\mathbf{y}}_k = F(\mathbf{x};\theta_k)$ contains predictive scores by the k-th network and $\mathbf{y}$ is one-hot encoding of the truth label for

$\mathbf{x}$. Hereafter, we refer (1) as the *baseline* approach by training each classifier independently without any interaction. In (Pang et al. 2019), the ADP regularizer was defined on the predictive scores $\{\hat{\mathbf{y}}_k\}$ for all k to encourage non-maximal predictions of each network to be mutually orthogonal.

In this paper, we propose a diversified learning directly on the feature level for simultaneous training. This is enabled by two novel regularization schemes: 1) Priority Diversified Dropouts (PDD), and 2) Dispersed Ensemble Gradients (DEG). The dropout regularization aims to encourage each member to learn diversified feature representations of the input, while the gradient regularization is amended to the ECE loss as a penalty term for gradient descents of classifiers in similar directions of the learning space. The two parts work to enhance each other: Members in the ensemble are able to have more dispersed gradients when learning more diversified features, and vice versa.

Priority Diversified Dropouts (PDD)

Dropout is a stochastic regularization technique commonly used for neural nets. Conventional dropout and variants are applied to training *individual* networks by setting a random subset of activations to zero, i.e., "dropping" FC units, with a certain probability $(1-p)$ where p is the *keep rate*. The process is known to have the effect of helping model generalization, making node activations more robust to the input noise (Wang, Zhou, and Bilmes 2019). Recent studies have shown that feature nodes act interactively in a way that is related to latent semantic features (Du et al. 2018; Kim et al. 2017). In particular, the ReLU activation strength was found to play an important part in the analysis of dropout features and has been exploited to improve the prediction performance (Ke et al. 2020; Keshari, Singh, and Vatsa 2019; Wang, Zhou, and Bilmes 2019).

Therefore, we are inspired to design an adaptive dropout in simultaneous training to enforce diversified learning of deep feature representations amongst the ensemble networks. This can be viewed as a *feature selection* by each base network. Because the dropout induces sparsity in feature representation by disregarding some high-level features, resulting in different activation patterns between networks. The ensemble models as a whole span the latent semantic feature space.

Algorithm 1 outlines the main procedures. Specifically, we divide the ensemble range of activation strength into $M > K$ intervals and count the number of neurons of K base networks that fall in the intervals $G_1, G_2, ..., G_M$, respectively. We leave discussion about the influence of M later to Ensemble Diversity Analysis for PDD. The intervals with the largest counts are considered having *priority* for activating the neurons therein with higher probability. The top-K such intervals each is assigned to one of the K base networks for diversified learning with different priority of activation strength.

Without loss of generality, let the k-th network have $N_m^{(k)}$ neurons in the m-th interval G_m for $m = 1, 2, ..., M$. The total number of neurons in the k-th network is $C_k = \sum_{m=1}^{M} N_m^{(k)}$. Let the k-th network have activation priority within the interval G_{t_k} for $t_1 \neq t_2 \neq ... \neq t_K$. Then, the keep rate for FC units of the k-th network with activation length in the interval G_m is

$$p_m^{(k)} = \begin{cases} \alpha\,, & m = t_k \\ \beta \cdot (1 - N_m^{(k)}/C_k), & m \neq t_k \end{cases} \tag{2}$$

where α and β are coefficient parameters between $[0, 1]$. We choose a large α to activate priority neurons with high probability and a small β for capping the total number of neuron activations in all other intervals. For the latter, the keep rate is negatively proportional to neuron density in the m-th interval for the k-th network.

Algorithm 1 Priority Diversified Dropouts (PDD)

Require: The ensemble C_k units in the last FC layer of the k-th network $F(\theta_k)$ for $k \in [K]$ in a training period.

Find the spanning range of activation strength $[u, v]$ for $\sum C_k$ activation units ensembled from all K networks;
Divide $[u, v]$ into M intervals and count the ensemble FC units in each interval $G_1, G_2, ..., G_M$;
Sort $G_1, G_2, ..., G_M$ in descending order by the count;
Find $G_{t_1}, G_{t_2}, ..., G_{t_K}$ with the largest counts;
for k=1 **to** K **do**
  Assign G_{t_k} as activation priority to $F(\theta_k)$
  for m=1 **to** M **do**
    Compute the keep rate $p_m^{(k)}$ as in (2);
    Keep the $F(\theta_k)$ units in G_m with probability $p_m^{(k)}$;
  end for
end for
return The last FC layer of $F(\theta_k)$ for $k \in [K]$.

Dispersed Ensemble Gradients (DEG)

Gradient regularization can significantly change model decision boundaries by incorporating some interpretation of explanation (Ros and Doshi-Velez 2018). In this work, we incorporate a gradient regularization term in the ensemble loss to encourage more dispersed gradient descents between individual base models. The goal is to make adversarial examples generated on one member network less transferable to another, and thus improve the global adversarial robustness of deep ensembles.

Algorithm 2 outlines the main procedure of our gradient regularization. In each training iteration, we first calculate the conventional CE losses of $\mathcal{L}_{\texttt{CE}}(\hat{\mathbf{y}}_k, \mathbf{y})$ as usual as well as the corresponding CE loss gradients $g_k = \partial \mathcal{L}_{\texttt{CE}}(\hat{\mathbf{y}}_k, \mathbf{y})/\partial \mathbf{x}$ for $k = 1, 2, ..., K$. Then, the penalty term for dispersed gradients can be computed as

$$\mathcal{L}_g = \sum_{1 \leq i < j \leq K} \frac{\langle g_i, g_j \rangle}{\|g_i\| \cdot \|g_j\|}, \tag{3}$$

where $\|\cdot\|$ is the gradient magnitude. It can be seen that $\mathcal{L}_g$ is effectively the sum of cosine values between pairwise input gradients. In this way, we encourage gradient dispersion by amending the regularization term (3) to the CE loss in (1):

$$\mathcal{L}_{\texttt{ours}} = \mathcal{L}_{\texttt{ECE}} + \lambda \cdot \mathcal{L}_g\,, \tag{4}$$

Algorithm 2 Simultaneous Training with DEG

Require: The training dataset $\mathcal{D} := \{(\mathbf{x}, \mathbf{y})_i\}_{i\in[N]}$; the ensemble of K base networks $\mathcal{F} := \{F(\theta_k)\}_{k\in[K]}$.

1: **for** k=1 **to** K **do**
2: Compute the CE loss $\mathcal{L}_{\texttt{CE}}(\hat{\mathbf{y}}_k, \mathbf{y})$;
3: Compute $g_k = \partial\mathcal{L}_{\texttt{CE}}(\hat{\mathbf{y}}_k, \mathbf{y})/\partial\mathbf{x}$;
4: **end for**
5: Compute the regularization term $\mathcal{L}_g$ as in (3);
6: Compute the overall loss $\mathcal{L}_{\texttt{ours}}$ as in (4);
7: Update $\{\theta_k\}_{k\in[K]}$ with $\nabla_{\theta_k}\mathcal{L}_{\texttt{ours}}$ until convergence.
8: **return** The optimal $\{\theta_k^*\}_{k\in[K]}$ s.t. $\mathcal{L}_{\texttt{ours}}$ is minimized.

Ensemble Methods	CIFAR-100		Tiny-ImageNet	
	$K=3$	$K=5$	$K=3$	$K=5$
Baseline	0.1905	0.1874	0.2553	0.2624
ADP	0.2155	0.2240	0.3082	0.3046
PDD	**0.2429**	0.2372	0.3903	0.3439
PDD+DEG	0.2277	**0.2462**	**0.4117**	**0.4126**

Table 1: The entropy measure E computed by (5) on the ensemble learning methods.

where λ controls the penalty strength. The objective is thus to find an optimal set of $\{\theta_k\}$ so that (4) is minimized.

Ensemble Diversity Analysis

In this section, we study ensemble diversity of the proposed method in terms of three perspectives, namely the ambiguity of member outputs, the discrimination of feature selection, and the dispersion of ensemble gradients. Specifically, we exploit entropy to summarize the *ambiguity* level of the member outputs as in (Kuncheva and Whitaker 2003)

$$E = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K - \lceil K/2\rceil}\min\left\{l(\mathbf{x}_i), K - l(\mathbf{x}_i)\right\}\ , \quad (5)$$

where N denotes the size of test dataset, $l(\mathbf{x}_i)$ denotes the number of member classifiers that correctly recognize an input $\mathbf{x}_i$. The highest diversity among all K members in an ensemble is manifested by $\lfloor K/2\rfloor$ votes with the same value (0 or 1), while the lowest diversity is when there is no disagreement with all 0's or 1's. Table 1 shows the entropy measures on CIFAR-100 and Tiny-ImageNet, where the maximal disagreement is always achieved by PDD or PDD+DEG. The output ambiguity becomes larger for PDD+DEG when the number of ensemble size K increases.

At training time, the PDD regularization induces sparsity in the FC activation patterns of each base network by setting a subset of relevant weights to zero according to the activation strength. In this way, the PDD learning can be viewed as a stochastic feature selection by a base network in the ensemble. The PDD method facilitates ensemble diversity on high-level feature representation by learning different network activation patterns. Together, the ensemble networks span the latent semantic feature space.

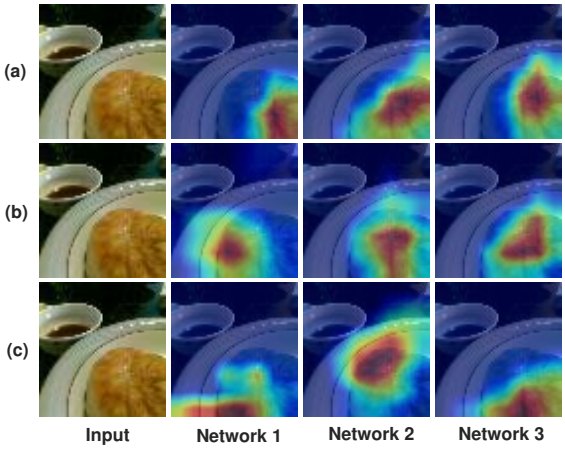


Figure 1: Grad-CAM visualization of the last FC layer by each base network in the deep ensemble: (a) Baseline, (b) ADP, and (c) PDD. The input image is labelled with class `meatloaf` from the testing dataset of Tiny ImageNet.

Figure 1 illustrates the PDD effect on a test example taken from Tiny ImageNet[2]. We adopt Grad-CAM (Selvaraju et al. 2017) to invert the FC features learned by each base network in an ensemble and superimposed them onto the input image, respectively. The visualization tool localizes class-discriminative regions where **red** highlights evidence. Darker color indicates a higher score for the predictive class. In Figure 1, all heat maps have allocated the darkest red region on the object to predict `meatloaf`. Comparing the heat maps, PDD in Figure 1 (c) has learned to look at visual and textual explanations at different parts of the object. To quantify the discrimination of feature learning between base networks of an ensemble method, we further estimate the geometric center of red regions with the darkest color (i.e., positive value of the neuron importance higher than 0.9) for each heatmap in Figure 1, and then compute the Euclidean distance. The proposed PDD method has the largest Euclidean distance of 24.68, comparing with 8.8 for the baseline and 14.34 for ADP, indicating more diversified learning by the proposed ensemble networks.

Inspired by Fisher score (Duda, Hart, and Stork 2012), we develop a measure to quantify the discrimination of feature selection at test time. Suppose that an ensemble of K networks is pretrained with PDD as outlined in Algorithm 1. Let the variable set $\mathcal{A}_k$ contain activation values of activated units in the last FC layer of the k-th network for $k \in [K]$. Denote μ_k and σ_k^2 as the mean and variance of $\mathcal{A}_k$, respectively. We measure the total *discrimination score* between $\mathcal{A}_i$ and $\mathcal{A}_j$ of every two base networks in the ensemble as

$$\mathcal{L}_F = \sum_{1\le i<j\le K}\frac{S_b(\mathcal{A}_i, \mathcal{A}_j)}{S_w(\mathcal{A}_i, \mathcal{A}_j)} = \sum_{1\le i<j\le K}\frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}, \quad (6)$$

where S_b and S_w are between- and within-network scatter measures of $\mathcal{A}_i$ and $\mathcal{A}_j$, respectively.

[2]https://tiny-imagenet.herokuapp.com/

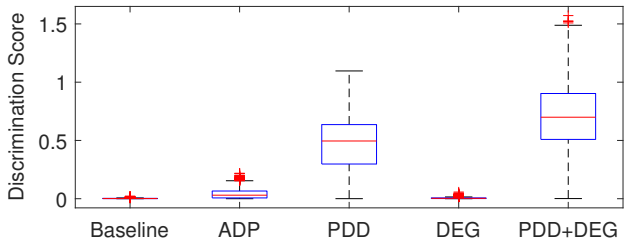


Figure 2: Boxplots of the average discrimination score computed from (6) over 2000 test images from CIFAR-100. A higher value indicates a larger discrimination of FC features between base networks of an ensemble model ($K = 3$).

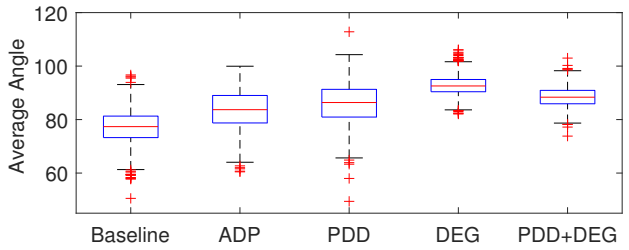


Figure 3: Boxplots of average angles between loss gradients of the base networks on 2000 test images from CIFAR-100. A higher value indicates larger gradient dispersions between base networks of an ensemble model ($K = 3$).

We observe that the number of intervals M has an influence on both the S_b and S_w scatter measures for the PDD training. We choose the M value empirically with the ensemble size K as specified in **Target Model** of Evaluations.

Figure 2 displays box-plots of the average discrimination score computed from (6) over 2000 **test images** from CIFAR-100 that have been classified correctly by all five ensemble methods. The proposed PDD enforces significantly larger discrimination of FC activation patterns between base networks of the ensemble. It is interesting to see that DEG by itself does not improve the high-level feature selection comparing with the baseline. However, it is able to boost the discrimination scores further when used in conjunction with PDD. This observation is in line with the results of ensemble recognition accuracy shown in Figure 4.

The DEG regularization is to encourage *dispersion* of ensemble gradients so as to expand and explore the learning space of normal features. Figure 3 presents the box-plots of average angles between the loss gradients of base networks in the ensembles trained on the same 2000 test images from CIFAR-100. We see less variability and outliers of average angles by PDD+DEG as well as an increased mean by adding the proposed gradient regularization. This indicates that DEG can complement PDD, especially when it is underperformed as shown in our later experiments.

Performance Evaluations

Datasets. We evaluate our method on three image benchmarks with increasing complexity and cluttered scenes, namely Fashion-MNIST, CIFAR-100, and Tiny-ImageNet. In particular, Fashion-MNIST [3] has 10-class ($L = 10$) labels and consists of 60k training samples and 10k testing samples each with 28×28 resolution; CIFAR-100 has 100-class ($L = 100$) labels and contains 50k training samples and 10k testing samples each with $32 \times 32 \times 3$ resolution; Tiny-ImageNet has 200 classes ($L = 200$), containing 100k and 10k samples each of $64 \times 64 \times 3$ resolution for training and validation testing, respectively. In all cases, the image intensity is normalized to 1 in our experiments.

Target Models. On each dataset, we implement deep ensembles comprising K ResNet-18 networks (He et al. 2016). The PDD method is applied to the last FC layer of 512 neurons before the softmax layer. Two cases of $K = 3$ and $K = 5$ are tested with model parameters set as described in the PDD section. Note that our methods do not require any specification on L. For the PDD regularization, we set $\alpha = 0.9$ and $\beta = 0.1$ for computing the keep rate in (2) in all our experiments. Unless otherwise specified, we choose $M = 10$ for $K = 3$ and $M = 20$ for $K = 5$ empirically. For the DEG regularization, we set $\lambda = 0.01$ to control the penalty strength in (3). The proposed training method is evaluated in comparison with the baseline, i.e., independent training each network with (1), and the ADP training with best performing parameters set as described in (Pang et al. 2019). Following common practice, we evaluate the recognition accuracy under adversarial perturbations to measure the adversarial robustness of an ensemble model. A higher recognition accuracy corresponds to a *lower* success rate by adversarial attacks.

Threat Models. Denote the target model by F and the defence measure by D. Based on the adversary's knowledge, we consider the following threat models:

- *White-box Attack* assumes a full access and knowledge of both F and D (if deployed).
- *Type I (Oblivious) Black-box Attack* has no knowledge about the model F or the defence method D if there is any.
- *Type II (Adaptive) Black-box Attack* assumes knowledge about the type of F and D in use but no implementation details such as the model and defence parameters.

Attack Methods. Under the above threat models, we conduct untargeted attacks with five representative methods, namely FGSM (Goodfellow, Shlens, and Szegedy 2015), BIM (Kurakin, Goodfellow, and Bengio 2017), PGD (Madry et al. 2018), JSMA (Papernot et al. 2016), and the C&W attack (Carlini and Wagner 2017b). The control parameter(s) for each attack method varies. Unless otherwise specified, the attack parameter (Para.) in all tables is referred as the attack strength ε for FGSM, BIM and PGD, γ with the control

[3] https://github.com/zalandoresearch/fashion-mnist/

Attacks	Fashion-MNIST				CIFAR-100				Tiny-ImageNet			
	Para.	Base.	ADP	PDD+DEG	Para.	Base.	ADP	PDD+DEG	Para.	Base.	ADP	PDD+DEG
No Attack	-	94.37	93.91	94.39	-	80.32	80.36	79.81	-	67.36	66.94	64.68
FGSM	0.01	82.36	86.86	**89.85**	0.02	23.47	35.70	**44.74**	0.02	18.87	12.46	**45.63**
	0.02	67.38	79.83	**80.90**	0.04	12.13	**22.81**	21.81	0.04	7.44	3.79	**24.92**
BIM	0.01	77.74	82.96	**84.58**	0.01	13.13	23.94	**45.74**	0.01	16.87	12.41	**34.22**
	0.02	58.40	**70.84**	66.54	0.02	2.38	12.28	**33.96**	0.02	3.30	2.32	**16.35**
PGD	0.01	82.13	85.78	**87.19**	0.01	18.38	26.57	**48.37**	0.01	22.85	19.85	**39.16**
	0.02	63.40	**72.80**	69.92	0.02	2.88	10.90	**29.88**	0.02	5.30	3.81	**17.49**
JSMA	0.1	62.17	**80.26**	77.37	0.05	32.52	52.94	**65.14**	0.05	43.64	47.74	**65.78**
	0.2	29.89	**50.53**	46.84	0.1	13.50	26.69	**45.74**	0.1	26.18	28.95	**52.09**
C&W	0.1	34.10	34.25	**89.60**	0.01	0.38	1.88	**53.63**	0.01	3.57	1.02	**27.76**

Table 2: Recognition accuracy (%) under white-box attacks with control parameter (Para.) as ε in L_∞ for FGSM, BIM and PGD, L_0 for JSMA, and L_2 norm for C&W. The ensemble size is $K = 3$. Higher ensemble accuracy indicates better recognition robustness under adversarial attacks. The best performance is marked in bold.

Attacks	CIFAR-100			
	Para.	Base.	ADP	PDD+DEG
FGSM	0.02	23.96	39.82	**49.85**
	0.04	13.45	25.34	**28.43**
BIM	0.01	13.23	31.61	**47.01**
	0.02	2.82	22.71	**32.16**
PGD	0.01	18.04	32.54	**47.66**
	0.02	2.83	19.09	**29.82**
JSMA	0.05	37.16	52.42	**67.51**
	0.1	15.71	26.67	**46.33**
C&W	0.01	0	6.67	**53.50**

Table 3: Recognition accuracy (%) under white-box attacks with $K = 5$ on CIFAR-100.

Black-Box	Attacks	Para.	Base.	ADP	PDD+DEG
Type I	FGSM	0.02	42.93	**46.25**	45.62
		0.04	26.08	27.00	**27.08**
	PGD	0.01	66.95	68.50	**69.14**
		0.02	34.07	37.35	**42.84**
	JSMA	0.05	38.06	55.46	**65.62**
		0.1	12.69	29.70	**43.22**
Type II	FGSM	0.02	31.73	45.02	**58.59**
		0.04	17.22	14.02	**38.38**
	PGD	0.01	14.02	57.75	**87.37**
		0.02	3.94	27.75	**77.60**
	JSMA	0.05	37.76	54.85	**70.66**
		0.1	16.92	28.18	**51.97**

Table 4: Recognition accuracy (%) under black-box attacks by surrogates of the ensembles with $K = 5$ on CIFAR-100.

of L_0 perturbation intensity $\theta = 0.1$ for JSMA. The hyper-parameter c in the C&W objective function affects the classification error and is chosen by a modified binary search. We use the default setting of C&W for controlling the predictive confidence of L_2 perturbations. Our implementation is based on Pytorch and the Adversarial Robustness 360 Toolbox (ART) v1.1 library [4].

White-Box Attacks

Table 2 and 3 display the recognition accuracy of different ensemble methods for $K = 3$ and $K = 5$, respectively. For fair comparisons, we follow the experimental settings in (Pang et al. 2019) to set up the ε values for FGSM, BIM and PGD. For example, each pixel is allowed to perturbed up to $5/255$ when $\varepsilon = 0.02$ of which the image artefacts are already visible as shown in Figure 5 (b). For BIM and PGD, the attack iterations is 10. The learning rate is set to 0.001 for C&W with 1000 iteration steps.

Table 2 shows that 20 out of 27 baseline results (i.e., without defence) have a recognition accuracy less than 50%, which corresponds to a high attack success rate. For example, the CIFAR-100 baseline has a recognition accuracy of only 2.88% and 0.38%, and the Tiny-ImageNet model has only 5.30% and 3.57% under PGD (0.02) and C&W (0.1), respectively, indicating these attacks are fairly **strong**. In these cases, the proposed PDD+DEG is able to achieve a more significant gain of adversarial robustness by up to 52%. Similar gain can be also observed with $K = 5$ in Table 3. In general, the proposed method performs better on CIFAR-100 and Tiny-ImageNet where the data has more label classes and complex scenes.

Black-Box Attacks

Table 4 reports under the two types of black-box attacks defined in **Threat Models**. We simulate the black-box attacks by building an ensemble of surrogate models with $K = 5$ on CIFAR-100. For Type I (oblivious) attacks, the surrogates are trained with baseline VGG-16 that have a different architecture from the target model. For Type II (adaptive) attacks, the surrogates are trained with the same defence method (if

[4] https://github.com/IBM/adversarial-robustness-toolbox

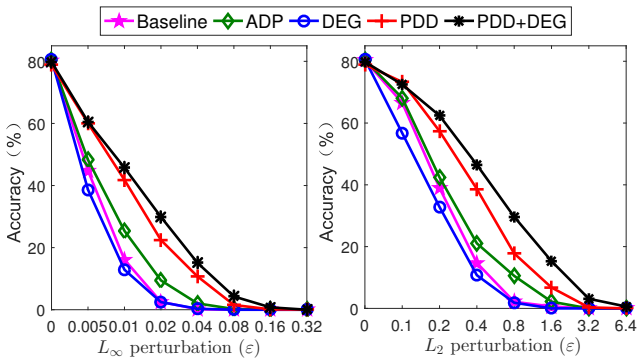
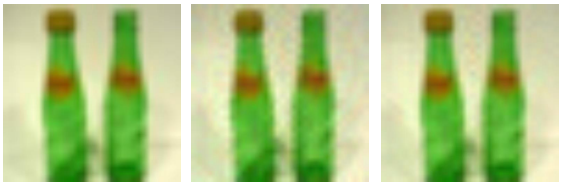


Figure 4: Recognition accuracy (%) under the PGD attack with an increasing value (in log_2 scale) of the normalized perturbation intensity ε on CIFAR-100.



(a) (b) (c)

Figure 5: (a) Original image from CIFAR-100, (b) PGD (L_∞) with $\varepsilon = 0.02$, (c) PGD (L_2) with $\varepsilon = 0.4$. Artefacts can be visible between bottles on the adversarial examples.

there is any) as the target but deployed on ResNet-34 instead of the ResNet-18 networks.

In Table 4, adversarial examples generated on the surrogate baseline and ADP models in general have higher success rates when attacking the target models (reporting lower recognition accuracies), especially under adaptive attacks when the defence method D is known to the adversary. On the other hand, the proposed training of PDD+DEG is able to restrain the transferrability of adversarial examples generated on the surrogates, making them less effective to attack the target model. Therefore, the gain of adversarial robustness by our method is significantly higher, especially under the Type II adaptive attacks.

Increasing Attack Strength

Figure 4 evaluates the adversarial robustness of deep ensembles with $K = 3$ by increasing the perturbation intensity of PGD on CIFAR-100 in a white-box setting. Note that the image pixels are normalized to be within 1. The plots are on a semi-log scale with the ε value doubly increased over the x-axis. In general, the proposed method of PDD+DEG is more robust as the recognition accuracy of all methods declines with an increasing intensity of ε constrained in L_2 and L_∞ norms. For example, it remains a recognition accuracy of 50% at $\varepsilon = 0.4$ (L_2) while the baseline performance drops below 20%. The robustness limit of PDD+DEG is 0.32 (i.e., 82/255 pixels) in terms of L_∞ perturbations before reaching a zero recognition accuracy. This is about 8 times larger than

Attack Methods	CIFAR-100, $K = 3$		
	PDD	DEG	PDD+DEG
BIM ($\varepsilon = 0.02$)	21.76	2.45	**30.90**
PGD ($\varepsilon = 0.02$)	22.43	2.45	**29.88**
JSMA ($\gamma = 0.1$)	34.73	18.41	**45.74**

Table 5: Ablation tests under white-box attacks with the best performances are marked in bold.

Attacks	CIFAR-100			
	FGSM		PGD	
L_∞ perturbation (ε)	0.04	0.08	0.02	0.04
$\text{AdvT}_{\text{FGSM}}$	41.06	20.8	20.15	3.62
$\text{AdvT}_{\text{FGSM}}$ + PDD	**59.64**	**27.09**	**38.17**	**19.5**
AdvT_{PGD}	44.14	21.67	44.11	15.69
AdvT_{PGD} + PDD	**55.56**	**34.68**	**51.02**	**26.13**

Table 6: Recognition accuracy (%) under white-box attacks by incorporating adversarial training to the ensemble model with $K = 3$ on CIFAR-100.

that of the baseline on CIFAR-100. Similarly, the L_2 limit is improved by about 4 times by the proposed PDD+DEG.

Figure 5 gives PGD-generated attack examples with the original image labelled `bottle` taken from the test set of CIFAR-100. Image artefacts are clearly visible between the two bottles on adversarial examples generated with $\varepsilon = 0.02$ (L_∞) and $\varepsilon = 0.4$ (L_2), respectively. For such perturbation intensities, the baseline performance drops quickly to 2.54% and 14.72% while the PDD+DEG performance still has 29.88% and 46.48%. We have also conducted experiments to empower white-box attacks by increasing the number of attack iterations from 10 to 30 for BIM, PGD, 1000 to 2000 for CW. The proposed method of PDD+DEG improves the robustness of baseline by 14-50% and that of ADP by 13-48% under different attacks.

Ablation Tests

In Figure 4, we see that the DEG-only performance is close to the baseline. Similar (sometimes better) results can be observed under other attacks such as FGSM. This indicates that promoting dispersion of ensemble gradients by itself is not as effective. However, it is able to enhance the PDD regularization to promote the diversified learning by exploring and expanding the normal feature space. Our ablation tests verify this. In Table 5, PDD contributes more to the gain of robustness while PDD+DEG improves the recognition performance by more than 7% under different white-box attacks.

Adversarial Training

The proposed regularization techniques are designed to promote diversified learning of simultaneous training for deep ensembles. It can work in conjunction with other defences acting on individual models. To show this, we combine the PDD strategy with adversarial training (Goodfellow, Shlens, and Szegedy 2015) that augments the training data with ad-

versarial examples in each mini-batch. We denote the one injected with FGSM examples by $\mathrm{AdvT}_{\mathrm{FGSM}}$ and the one with PGD examples by $\mathrm{AdvT}_{\mathrm{PGD}}$. Table 6 demonstrates results under white-box attacks for the ensemble methods with $K = 3$ on CIFAR-100. It can be seen that our method is able to complement adversarial training to boost the ensemble robustness to a new level, especially under attacks of large perturbation intensities. The improvement is also significant for adversarial training with different attack examples, e.g. $\mathrm{AdvT}_{\mathrm{FGSM}}$ against PGD and $\mathrm{AdvT}_{\mathrm{PGD}}$ against FGSM.

Complexity Analysis

The PDD method involves counting in $\mathcal{O}(C_k)$ and sorting in $\mathcal{O}(M \log M)$, where C_k is the number of FC units in the k-th network and M is the number of intervals. DEG involves gradient regularization which is a second-order method that can increase training time per batch by a factor of two (Ros and Doshi-Velez 2018). Given the input gradients, computing the penalty form in (3) requires $\mathcal{O}(K^2)$ for pair-wise operations over K networks. The cosine similarity conventionally requires $\mathcal{O}(n^2)$ where n is the gradient dimension. We test the training time per epoch with a mini-batch size of 64 on CIFAR-100. When $K = 3$, for example, it takes 54s/epoch for baseline, 64s/epoch for ADP, 93s/epoch for PDD, and 703s/epoch for DEG on Tesla V100. We note that PDD contributes most to the gain of robustness in ablation tests, while DEG improves the diversified learning by exploring and expanding the normal feature space. A future work may consider improving the efficiency of the proposed training strategy, e.g., by avoiding the expensive second-order methods.

Conclusion

In this paper, we have proposed to improve adversarial defence by promoting ensemble diversity of high-level feature representations between base networks. To this end, we have devised a novel diversified dropout with gradient dispersion to regularize the simultaneous training of deep ensembles. The proposed approach is effective under different attacks and reduces the transferrability of adversarial examples in black-box settings, especially against adaptive attacks. It also complements the defence paradigm of adversarial training, and can further boost the ensemble performance under more intensive adversarial perturbations.

Acknowledgements

The work was supported in part by Natural Science Foundation of China (grant no. 61876038, 62076163, 91959108), in part by Guangdong Regular Institutions of Higher Education Key Laboratory of Robotics and Intelligent Equipment (grant no. 2017KSYS009), in part by Dongguan Social Science and Technology Development Key Project (grant no. 2020507140146), and Dongguan University of Technology under project no. KCYKYQD2017003.

References

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Int. Conf. Mach. Learn.*, volume 80, 274–283.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion Attacks against Machine Learning at Test Time. In *Mach. Learn. Knowl. Discov. Databases*, 387–402.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *Int. Conf. Learn. Represent.*, 1–12.

Carlini, N.; and Wagner, D. 2017a. Adversarial Examples Are Not Easily Detected. In *ACM Work. Artif. Intell. Secur.*, 3–14.

Carlini, N.; and Wagner, D. 2017b. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symp. Secur. Priv.*, 39–57.

Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval Networks: Improving Robustness to Adversarial Examples. In *Int. Conf. Mach. Learn.*, 854–863.

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 99–108.

Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; Nita-Rotaru, C.; and Roli, F. 2019. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. In *USENIX Secur. Symp.*, 321–338.

Dhillon, G. S.; Azizzadenesheli, K.; Lipton, Z. C.; Bernstein, J.; Kossaifi, J.; Khanna, A.; and Anandkumar, A. 2018. Stochastic Activation Pruning for Robust Adversarial Defense. In *Int. Conf. Mach. Learn.*, 1–13.

Du, Y.; Yuan, C.; Li, B.; Zhao, L.; Li, Y.; and Hu, W. 2018. Interaction-Aware Spatio-Temporal Pyramid Attention Networks for Action Classification. In *Eur. Conf. Comput. Vis.*, 388–404.

Duda, R.; Hart, P.; and Stork, D. 2012. *Pattern Classification*. New Jersey: John Wiley & Sons.

Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting Adversarial Samples from Artifacts. *arXiv Prepr.* 1–9.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *Int. Conf. Learn. Represent.*, 1–11.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 770–778.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples Are Not Bugs, They Are Features. In *Advances in Neural Information Processing Systems*, volume 32, 125–136.

Ke, Z.; Wen, Z.; Xie, W.; Wang, Y.; and Shen, L. 2020. Group-Wise Dynamic Dropout Based on Latent Semantic Variations. *Assoc. Adv. Artif. Intell.* 11229–11236.

Keshari, R.; Singh, R.; and Vatsa, M. 2019. Guided Dropout. In *Assoc. Adv. Artif. Intell.*, 4065–4072.

Kim, S.; Min, D.; Ham, B.; Jeon, S.; Lin, S.; and Sohn, K. 2017. FCSS : Fully Convolutional Self-Similarity for Dense Semantic Correspondence. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, 616–625.

Kuncheva, L. I.; and Whitaker, C. J. 2003. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Mach. Learn.* 51: 73–82.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial Machine Learning at Scale. In *Int. Conf. Learn. Represent.*, 1–17.

Kurakin, A.; Goodfellow, I.; Bengio, S.; Dong, Y.; Liao, F.; Liang, M.; Pang, T.; Zhu, J.; Hu, X.; Xie, C.; and Others. 2018. Adversarial Attacks and Defences Competition. In *NeurIPS'17 Compet. Build. Intell. Syst.*, 195–231.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Adv. Neural Inf. Process. Syst.*, 6402–6413.

Liu, L.; Wei, W.; Chow, K.-H.; Loper, M.; Gursoy, E.; Truex, S.; and Wu, Y. 2019. Deep Neural Network Ensembles against Deception: Ensemble Diversity, Accuracy and Robustness. In *IEEE Int. Conf. Mobile Ad-Hoc and Sensor Syst.*, 274–282.

Liu, X.; Cheng, M.; Zhang, H.; and Hsieh, C.-J. 2018. Towards Robust Neural Networks via Random Self-ensemble. In *Eur. Conf. Comput. Vis.*, 369–385.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Int. Conf. Learn. Represent.*, 1–28.

Meng, Y.; Su, J.; O'Kane, J.; and Jamshidi, P. 2020. Ensembles of Many Diverse Weak Defenses can be Strong: Defending Deep Neural Networks Against Adversarial Attacks. *arXiv Prepr.* 1–18.

Pang, T.; Xu, K.; Du, C.; Chen, N.; and Zhu, J. 2019. Improving Adversarial Robustness via Promoting Ensemble Diversity. In *Int. Conf. Mach. Learn.*, 4970–4979.

Papernot, N.; Mcdaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *IEEE Eur. Symp. Secur. Priv.*, 372–387.

Ros, A. S.; and Doshi-Velez, F. 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *Assoc. Adv. Artif. Intell.* 1660–1669.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *IEEE Int. Conf. Comput. Vis.*, 618–626.

Su, D.; Zhang, H.; Chen, H.; Yi, J.; Chen, P.-Y.; and Gao, Y. 2018. Is Robustness the Cost of Accuracy? - A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In *Eur. Conf. Comput. Vis.*, 644–661.

Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; and McDaniel, P. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *Int. Conf. Learn. Represent.*, 1–22.

Wang, S.; Zhou, T.; and Bilmes, J. 2019. Jumpout : Improved Dropout for Deep Neural Networks with ReLUs. In *Int. Conf. Mach. Learn.*, 6668–6676.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2018. Mitigating Adversarial Effects Through Randomization. In *Int. Conf. Learn. Represent.*, 1–16.

Yan, Z.; Guo, Y.; and Zhang, C. 2018. Deep Defense: Training DNNs with Improved Adversarial Robustness. In *Adv. Neural Inf. Process. Syst.*, 419–428.

Zhang, H.; Cheng, M.; and Hsieh, C.-J. 2019. Enhancing Certifiable Robustness via a Deep Model Ensemble. In *Int. Conf. Learn. Represent. Safe Mach. Learn. Work.*, 1–12.

Zhang, H.; Wang, J.; Robotics, H.; and Research, B. 2019. Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training. In *Adv. Neural Inf. Process. Syst.*, 1831–1841.

Zhou, Z.-H. 2012. *Ensemble methods: foundations and algorithms*, 111–112. CRC press.