ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230577

Deep Ensemble Robustness by Adaptive Sampling in Dropout-Based Simultaneous Training

Quanwei Wu[a], Bo Huang[b], Yi Wang[a;*], Zhiwei Ke[c] and Da Luo[a]

[a]Dongguan University of Technology
[b]The Hong Kong University of Science and Technology
[c]Shenzhen University

Abstract. Recent studies show that an ensemble of deep networks can have better adversarial robustness by increasing the deep feature learning diversity of base models to limit the adversarial transferability. However, existing schemes mostly rely on a second-order method for gradient regularization which usually involves a heavy computation overhead. In this paper, we propose a simple yet effective method which eliminates the use of a second-order optimization and significantly reduces the computation complexity of regularized simultaneous training of deep ensemble networks. For the first time, we show analytically that stochastic regularization by the proposed approach can promote both model smoothness and feature diversity of representation learning in the deep space. We also show that the proposed method is able to achieve a better gain of certified robustness. This is due to the effect of a prioritized feature selection enabled by an adaptive and continuous sampling of neuron activation among the base networks. Experimental results show that our method can improve adversarial robustness significantly comparing with the existing ensemble models on several image benchmark datasets. The ensemble performance can be further boosted by complementing the stochastic regularization approach with other defense paradigms such as adversarial training.

1 Introduction

Despite their prevalent applications, deep neural networks (DNNs) are found to be vulnerable to *adversarial examples* that are intentionally generated to induce predictive errors when input to the model [29]. To defend against the attack, it is critical to enhance adversarial robustness of the underlying networks. There are many defense strategies proposed in the literature [9, 1, 24, 17]. But most of them focus on strengthening individual models.

Deep learning architectures are widely applied in solving safety and security-critical problems such as self-driving cars and biometric identification. It is critical to enhance the adversarial robustness of DNN models under adversarial attacks. Moreover in many real-time applications, lightweight models are deployed in mobile devices where computation resources are limited, whereas several studies have shown that popular defense schemes such as adversarial training can be less effective on such small models [20, 22]. One possible solution is to build a robust ensemble of small deep networks as proposed in this paper.

* Corresponding Author. Email: wangyi@dgut.edu.cn.

Intuitively, it is harder to compromise an ensemble of models rather than a single one. However, conventional ensemble models train the base networks independently and often results in similar model predictions [18]. On the other hand, adversarial examples can be transfered across models especially similar ones [1], known as *transferred attacks*. This has become an obstacle for building robust countermeasures [30], especially in the context of ensemble models when the member networks demonstrate overlapping vulnerability [34].

Therefore, a recent line of research propose to increase diversity of base models at either outputs or internal representations for limiting adversarial transferability and improving ensemble robustness. For instance, [21] proposed the ADP scheme to diversify the predictive score distribution of base models over less-likely class labels at the model output. [17] proposed a GradDiv regularizer to disperse the gradient distribution by penalizing concentration in the ensemble loss function. [12] proposes a diversified learning strategy based on priority dropouts coupled with a dispersed ensemble gradients (DEG) regularizer to increase ensemble diversity on high-level feature representations. [35] showed that both the lower and upper bounds of adversarial transferability are tighter with smoother models, and accordingly proposed a TRS regularizer loss to enforce model smoothness and reducing loss gradient similarity between base models.

Although demonstrated effective in gaining robustness, the previous ensemble training methods are either limited by increasing class labels and more complex data scenes or involving a second-order method of gradient regularization which significantly impairs the simultaneous training efficiency. In this paper, we propose a simple yet effective method of ensemble training to promote both model smoothness and feature diversity of representation learning among base networks. Our strategy is in vein of the stochastic regularization scheme proposed in [12]. Yet, the adaptive dropout based scheme proposed in this paper does not require to work with second-order optimization, which significantly reduces the complexity of regularized simultaneous training. This is enabled by introducing a continuous utility function for sampling active neurons across members with diversified priority.

Our main contributions are:

- We propose a new keep rate generating function for adaptive sampling of neuron activations that can enforce model smoothness in the priority-based dropout training of deep ensembles, so called the LPD scheme.
- We show that the proposed adaptive dropout can promote both

model smoothness and feature diversified learning without using any second-order method previously required for gradient regularization. This significantly reduces the computation overhead of simultaneous training for deep ensembles.
- We study impacts of the proposed LPD scheme on intrinsic model properties including model smoothness, loss gradient diversity, and certified robustness, and show with extensive experiments that our method can effectively improve the deep ensemble model robustness under various attacks.

2 Related work

Adversarial attacks of DNN models can be roughly classified into *white-box* and *black-box* scenarios. In a white-box setting, an attacker can access target model details to build adversarial examples, typically via gradient descent [2, 9]. In black-box settings, however, an attacker attempts to attack by transferred adversarial examples built on a surrogate model [1, 5] or probing decision boundaries by queries [3, 4].

Many defense methods are proposed against adversarial examples. Most notably is *adversarial training* [9] which can be formulated as a min-max optimization problem [20] that builds a stronger model by attacking it on-the-fly. However, computing adversarial perturbations at each step requires many iterations of gradient-based optimization to be performed for each new mini-batch, which significantly increases the computation complexity of adversarial training especially as the model size and input dimensions increase [31].

Gradient regularization is another defense paradigm by altering the shape of decision boundaries for interpretable and qualitatively reasons, e.g., to have smooth input gradients with respect to its predictions [24] or increase gradient diversity [17, 12]. DNN trained with input gradient regularization often exhibit remarkable robustness against transferred examples that are generated to fool other models [5, 24]. However, such regularization methods in general have high computation cost because computing input gradients requires to take second derivatives in each mini-batch of parameter gradient descent [24].

Randomization techniques are also explored as a promising defense against adversarial attacks [2]. Related work can be roughly categories into randomness added to the input [33] and randomness added to the model [6, 12]. The former intends to mitigate adversarial effects by destructing the intentionally crafted perturbations by input variations such as random resizing, padding, and discretization. The latter includes adding random noise layers [19] and randomly perturb the hidden layers to learn an anisotropic noise distribution [8], which is equivalent to training the original network with an extra regularization of Lipschitz constant to improve the learning-theoretic bound on adversarial robustness.

Recently, dropout-based mechanisms are added to the defense paradigm of randomization [11]. Novel use of dropout and variants has emerged to help resisting adversarial attacks. In [6], a stochastic activation pruning scheme is similar to random dropout in terms of sampling activations, and it is also applied at test time. [13] proposed a self-adaptive dropout that improves the adversarial robustness of a single DNN model. [12] proposed a Priority Diversified Dropouts (PDD) to promote ensemble diversity of high-level feature representation learning. The diversified feature learning is driven by a second-order gradient regularization scheme of DEG [24].

Intrinsic properties of adversarial examples are studied to better understand the attacks hence the design of defense methods. [5] systematically studies the underlying reasons of transferability at both training and testing time, and suggests three main impact factors: 1) size of the input gradients, 2) gradient alignment, and 3) variability of the loss landscape. Among the three factors, most existing work of defense are developed based on reducing gradient alignment between the target and surrogate models to reduce the attack transferability in ensemble models [17, 12, 35]. In particular, [35] demonstrates theoretically that increasing gradient diversity alone is not enough and it has to work with increased model smoothness to ensure lower transferability.

3 Proposed Method

Inspired from [5], we propose to facilitate simultaneous training by reducing the model complexity and increasing the variability of loss landscape for ensemble robustness without a second-order optimization of gradient alignment. This is achieved in a unified simple framework of designing an adaptive dropout with priority-based keep-rate generating functions across the base networks. The proposed stochastic regularization scheme is described as follows.

3.1 Problem Formulation

Dropout is a stochastic regularization technique that was conventionally used to combat overfitting in various DNN models. They are typically applied in the training phase by setting a random subset of neuron activations to zero, i.e., "dropping" units in the fully-connected (FC) layer, with a certain probability $(1-p)$ where p is the *keep rate*. In this way, the network is made more robust to input noise and has better generalization performance on information-degraded data points [11, 28].

Consider the ensemble model $\mathcal{F} = \{F(\theta_k)\}$ where $F(\theta_k)$ is a base network for $k = 1, 2, ..., K$. Typically, $\mathcal{F}$ is trained by aggregating the individual predictors on the mini-batch of data $\mathbf{x}$ such as

$$\hat{\mathbf{y}}_{\mathcal{F}} = \frac{1}{K}\sum_{k=1}^{K} F(\mathbf{x};\theta_k). \tag{1}$$

We refer the conventional independent training of $F(\mathbf{x};\theta_k)$ *without* interactions between the base networks as the *baseline* approach, and those *with* interactions as *simultaneous training*. There are different ways to impose interactions on members in the ensemble model. For instance, [21] defines an ADP regularizer on the output of predictive scores of $\{\hat{\mathbf{y}}_k\}$ for all k to encourage non-maximal predictions of each network to be mutually orthogonal. [34] generates training inputs on-the-fly with isolated vulnerabilities by utilizing a different set of features from other sub-models. [17, 12, 35] introduces additional gradient regularization to the ensemble loss function.

In this paper, we study the paradigm of stochastic regularization by adaptive dropout training over an ensemble of deep networks. Adaptive dropout serves to span the latent semantic feature space of the ensemble model by inducing sparsity in feature representation and reduce the base model complexity [28], which results in different activation patterns of deep features. On the other hand, previous study such as [14] showed that the ReLU activation strength contains critical feature information that can be exploited for dropout analysis and subsequently model improvement. Therefore, [12] designs an adaptive dropout as a feature selection scheme based on joint activation density estimation across the based networks to facilitate diversified learning of feature representations in simultaneous training. Here, we introduce necessary notations of the stochastic regularization by briefly reviewing the priority-based diversity dropout (PDD) proposed therein.

Algorithm 1 Laplace Priority Dropout (LPD)

Require: The last FC layer of $F(\theta_k)$ for $k = 1, 2, ..., K$ in each training period

for k=1 **to** K **do**
  Compute $\mu(v)$ by Eq. (3) for each neuron with activation strength $v > \epsilon$ in the last FC layer of $F(\theta_k)$;
  Count $n_k(\mu)$ for all $\mu(v)$ values in $F(\theta_k)$;
end for
Compute $N(\mu_l) = \sum_k n_k(\mu_l)$ for $l = 1, 2, ..., L$;
Sort $\mu_1, \mu_2, ..., \mu_L$ w.r.t. $N(\mu_l)$ in descending order;
for k=1 **to** K **do**
  Find $r_k \in \{\mu_l\}$ with the k-th largest count in $N(\mu_l)$;
  Assign r_k as the priority activation median to $F(\theta_k)$;
  Compute $z_k(\mu)$ by Eq. (4) for all μ values in $F(\theta_k)$;
  Compute the LPD keep rate $p_{\text{LPD}}(k, \mu)$ by Eq. (6) for each FC unit of $F(\theta_k)$ based on its μ value;
  Keep a FC unit in $F(\theta_k)$ with probability $p_{\text{LPD}}(k, \mu)$;
end for
return The last FC layer of $F(\theta_k)$ for $k = 1, 2, ..., K$.

The adaptive dropout is applied to the last fully connected (FC) layer across all member networks $F(\theta_k) \in \mathcal{F}$ for $k = 1, 2, ..., K$. It first finds the spanning range of activation strength for all hidden units in the last FC layer, and then collect the ensemble statistics of neuron activation density through $M > K$ quantized activation intervals over the K base networks. The ensemble statistic indicates the "importance" of underlying features as more neurons with particular strength are activated in the networks. Thus, intervals with a higher neuron density are considered having a higher *priority* for feature selection. The top-K such intervals are then assigned to the K base networks to diversify high-level feature representation learning in the deep ensemble. This is enabled by a different generating function of keep rates for each network [12]:

$$p_{\text{PDD}}(k, m) = \begin{cases} \alpha\,, & m = t_k \\ \beta \cdot (1 - N_k(m)/C_k)\,, & m \neq t_k \end{cases} \tag{2}$$

where m and t_k can be viewed as the quantized activation strength of a hidden unit and activation priority interval of the k-th network. The parameters α and β are coefficients in $[0, 1]$. In particular, a relatively large α is selected to activate priority neurons, i.e., $m = t_k$, with a higher probability and a small β is used to cap the total number of neuron activations in all other cases, i.e., $m \neq t_k$. The latter is based on the neuron activation density $N_k(m)/C_k$ of a hidden unit with particular activation strength, where $C_k = \sum_{m=1}^{M} N_k(m)$ is the total number of activated units with a valid activation strength for $m = 1, 2, ..., m$ in the last FC layer of $F(\theta_k)$ for $k = 1, 2, ...K$.

3.2 Laplace Priority Dropout

The priority-based adaptive dropout can be viewed as a *feature selection* by the base network [12]. However, it is not clear why the stochastic regularization of dropout training can improve the model adversarial robustness. Moreover, the PDD scheme has to work with a *second-order* gradient regularization (DEG) to maximize its performance [12]. In this paper, we aim to explain and evaluate the impact of adaptive sampling from the perspective of several intrinsic properties that are highly related to improving the adversarial robustness as introduced in Section 2.

In [35], it was shown that *model smoothness* plays a critical role in constraining loss gradients similarity and hence tightening the bounds of adversarial transferability. Thus, we are inspired to study and hence improve the model smoothness of dropout-based stochastic regularization. We notice that the sampling function of active hidden units is the key to the dropout training, and that the generating function of PDD in (2) is composed of a two-case discriminative function. In this vein, we propose a new keep rate generating function for adaptive sampling of neuron activations that can enforce model smoothness in the priority-based dropout training of deep ensembles.

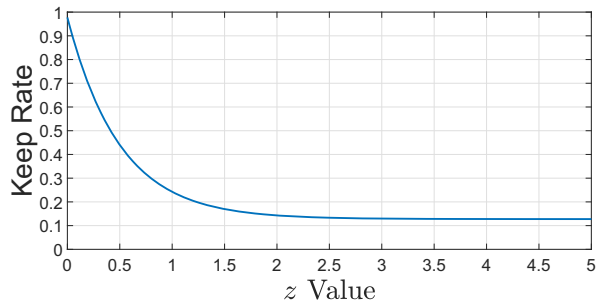

Figure 1. Adaptive keep rates generated by the proposed LPD with respect to the value of Laplace kernel z.

For diversified learning, the design goal is to have priority features retained with higher probability and non-priority ones suppressed much faster for different base networks. Thus, a viable solution is to use a Laplace kernel that has the shape of desired property as shown in Figure 1. The use of Laplacian kernel not only satisfies the design goal but also enables analytical properties of dropout analysis for priority-based regularized training of the ensemble models. We call the proposed method *Laplace Priority Dropout* (LPD) which is outlined in Algorithm 1. The following describes the main steps of how LPD generates the keep rates for adaptive sampling of hidden units across the networks.

For the ease of expression, a hyper-parameter s is set up to indicate the size of activation interval. Without specification, we use $s = 0.1$ empirically. We then compute the median value μ for a hidden unit with activation strength v in the last FC layer as

$$\mu(v) = \left\lceil \frac{v}{s} \right\rceil \cdot s - 0.5s \tag{3}$$

Suppose the k-th network $F(\theta_k)$ has C_k activated neurons in the last hidden layer and each is computed with a median value using its activation strength by Eq. (3). We gather statistical counts of μ_m across all base networks, i.e., $N(\mu_m) = \sum_k n_k(\mu_m)$ for $k = 1, 2, ..., K$ and $m = 1, 2, ..., M$ where M is the total number of activation median values in the ensemble networks.

We then sort $\mu_1, \mu_2, ..., \mu_M$ with respect to $N(\mu_m)$ in descending order to get $\mu_1^*, \mu_2^*, ..., \mu_M^*$. The top-K values of $\{\mu_m^*\}$ are considered as *priority* median values for neuron activation and each is assigned to one of the K base networks in $\mathcal{F}$. The priority median values are denoted by r_k for the k-th network.

To facilitate the diversified learning, we want neurons whose activation median value close to the designated r_k to have a higher probability and retain in the last FC layer of $F(\mathbf{x}; \theta_k)$. For the k network, we map the activation strength v with a median value μ into

$$z_k(\mu) = \gamma \cdot \frac{n_k(\mu)}{C_k} \cdot (r_k - \mu)^2 \tag{4}$$

where the first term γ is a scaling parameter, the second term $n_k(\mu)/C_k$ indicates the neuron density in the corresponding activation interval, and the third term is a distance between μ and the

priority median value r_k of the k-th network. We then build a utility function with the kernel

$$f_k(\mu) = \frac{1}{2\lambda} \cdot \exp(-\frac{z_k(\mu)}{\lambda}), \tag{5}$$

which generates the adaptive keep rates as

$$p_{\texttt{LPD}}\,(k, \mu) = \alpha \cdot f_k(\mu) + \beta. \tag{6}$$

Note that the kernel $z_k(\mu)$ is not only based on the Euclidean distance from the priority median but also the neuron density which is also a variable of the activation median μ. In this way, the keep rate is higher for hidden units whose activation strength is closer to the priority and activated more frequently in the local network.

3.3 Analytical Properties of LPD

The kernel mapping of Eq. (6) enables analytical properties of the keep rates hence improving explainability of the adaptive dropout scheme for ensemble robustness. As shown in Eq. (6), $p_{\texttt{LPD}}$ is linear function of the utility function $f(z_k)$ which follows $La(0, \lambda)$. Thus, the probability density function of $p_{\texttt{LPD}}$ also follows the Laplace distribution, i.e., $p_{\texttt{LPD}} \sim La(\beta, \alpha\lambda)$. Without specification, we set $\lambda = 0.5$, $\alpha = 0.85$ and $\beta = 0.127$ empirically in our experiments. Accordingly, $p_{\texttt{LPD}} \sim La(0.127, 0.425)$. Figure 1 plots the adaptive keep rates $p_{\texttt{LPD}}$ with respect to the value of our Laplace kernel z. It can be seen that when z is close to 0, i.e., the FC unit is nearby the priority median for activation, the keep rate is high with a probability close to 1. After the inflection point, i.e., at about $\beta+\sqrt{\alpha\lambda} = 0.78$ in Figure 1, the keep rate is gradually reaching the plateau of $\beta = 0.127$ which ensures that even the activated neurons with low priority have a chance to retain for better generalizability. According to properties of the Laplace distribution, the adaptive keep rates have an expected value $E(p_{\texttt{LPD}}) = \beta$ and a variance $V(p_{\texttt{LPD}}) = 2(\alpha\lambda)^2$, which are 0.127 and 0.361 in our case, respectively.

4 Impact of LPD on Model Properties

4.1 Ensemble Diversity of Generalizability

We study ensemble diversity of the proposed method in terms of three perspectives, namely the ambiguity of member outputs Specifically, we exploit entropy to summarize the *ambiguity* level of the member outputs as in [15]

$$E = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{K - \lceil K/2 \rceil} \min\{l(\mathbf{x}_i), K - l(\mathbf{x}_i)\} \tag{7}$$

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+DEG or LPD. The output ambiguity becomes larger for LPD when the number of ensemble size K increases, which indicates a better generalizability of the ensemble networks.

Table 1. The entropy measure E computed by (7) on the ensemble learning methods.

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	0.2277	0.2462	0.4117	0.4126
LPD	**0.2631**	**0.2784**	**0.4624**	**0.4667**

4.2 Diversity of Input Loss Gradients

As introduced in Related Work, several studies have shown that increasing the loss gradient diversity or reducing the gradient alignment with respect to input helps to boost the performance of adversarial robustness and lower the success rate of transferred attacks [5, 12, 17, 35]. To measure the gradient diversity, we also use the ensemble cosine similarity, denoted by $\mathcal{L}_d = \sum_{1\le i<j\le K} \cos\langle g_i, g_j\rangle$ for $K = 3$, between pairwise loss gradients of input by any two base networks. Note that $\mathcal{L}_d = 0$ indicates *gradient orthogonality*. While $\mathcal{L}_d > 0$ indicates higher gradient similarity, $\mathcal{L}_d < 0$ indicates negative correlation in opposite gradient directions.

Table 2 lists the 25%, 50%, 75% quantiles of $\mathcal{L}_d$ by the comparing methods on the first 2000 CIFAR-10 test samples. It can be seen that the three diversity-driven simultaneous training methods have all improved loss gradient diversity amongst the base networks. In particular, PDD and LPD have $\mathcal{L}_d$ very close to 0. Given the small magnitudes, the difference between the two is almost negligible. For instance, LPD is about 10^7 times smaller than ADP in terms of L_d at the 75% quantile. The results suggest that LPD-based diversified learning is effective on constraining gradient similarity even without an explicit gradient regularization scheme. This helps to significantly reduce the computation complexity by eliminating the use of a second-order method as in [12].

Table 2. Loss gradient diversity by $\mathcal{L}_d$ on CIFAR-10. The best performance is marked in **bold**.

Q	Baseline	ADP	PDD	LPD
25%	3.76e-01	-1.48e-01	-4.11e-08	**1.25e-14**
50%	6.06e-01	2.73e-02	**-9.93e-12**	1.04e-11
75%	9.24e-01	2.77e-01	**6.23e-09**	2.49e-08

4.3 Model Smoothness

LPD is designed to improve model smoothness in the priority-based diversified learning by using a continuous utility for generating the adaptive keep rates. To measure model smoothness, [32] derives a universal lower bound of *Lipschitz Continuity* which is closely associated to the maximum norm of local input gradients. In [27], model smoothness is also upper bounded by the l_2 norm of input gradients. Accordingly, we also evaluate model smoothness using the ensemble l_2 norm of gradients, denoted by $\mathcal{L}_s = \sum_{1\le k\le K} \|g_k\|_2$ for $K = 3$. In general, the ensemble model is smoother with a smaller value of $\mathcal{L}_s$.

Table 3 shows the 25%, 50%, 75% quantiles of $\mathcal{L}_s$ also on the CIFAR-10 dataset. Among the four comparing methods, LPD has the smallest $\mathcal{L}_s$ which is, in general, two magnitudes smaller than that by the second best performing PDD. This indicates that LPD

improves model smoothness significantly while constraining the ensemble gradient similarity.

Previous study [35] shows theoretically that increasing both gradient diversity and model smoothness are critical to ensure lower transferability of adversarial examples for better model robustness. Our experimental results support this view by showing that LPD further improves the results of PDD by enforcing model smoothness in addition to diversified feature learning by dropout-based stochastic regularization. In this way, LPD is also able to boost the performance of adversarial robustness as will be shown in the next section.

Table 3. Model smoothness by $\mathcal{L}_s$ on CIFAR-10. The best performance is marked in **bold**.

Q	Baseline	ADP	PDD	LPD
25%	3.86e-02	4.45e-01	8.02e-07	**3.57e-09**
50%	7.03e-02	6.49e-01	7.40e-06	**3.50e-08**
75%	3.98e-01	2.21e+00	5.08e-05	**6.78e-07**

4.4 Certified Robustness

Certified robustness provides a *provable* guarantee that no adversarial example is capable of fooling the target model in the neighborhood of a given input [26]. In general, the robustness verification can be formulated as an optimization problem with maximization objective under a set of constrains placed by the underlying model $F(x;\theta)$ which can be viewed as a composite function involving L hidden layers with parameter sets θ_l for $l = 1, ..., L$. Given an input data point $x \in \mathbb{R}^d$ and a bounded l_p-norm perturbation $\epsilon \in \mathbb{R}_+$, the set of adversarial examples of x is denoted by $\mathbb{S}(x) = \left\{x' \mid \|x' - x\|_p < \epsilon\right\}$. The robustness verification is to find the output margin of $F(x)$ on $\mathbb{S}(x)$, which can be formulated as the layer-wise optimization objective function [38]. Due to the non-linear constraint introduced by the activation function, e.g. ReLU function, solving exactly the optimization objective is an NP-complete problem [25]. To address the issues, a line of research work is focus on studying the bound algorithms via mixed-integer programming solvers or layer-wise convex relaxation framework [7]. Among these techniques, CROWN [38] typically can give the tightest bound. It proposes to find a lower bound on the output margin between the ground-truth class and other classes to verify if the norm-bounded perturbation can change the model prediction. We follow the work of certified robustness for ensemble models in [37] to calculate the lower bound. The CROWN method defines $\hat{F}(x)$ with the same parameters as those of $F(x)$ except the last layer which is reformed by

$$\begin{gathered} \widehat{w}^L_{(j,:)} = w^L_{(i,:)} - w^L_{(s(j,i),:)} \\ \widehat{b}^L_j = b^L_i - b^L_{s(j,i)} \\ \text{where } j \in [C-1], s(j,i) = \begin{cases} j & j < i \\ j+1 & j \geq i \end{cases} \end{gathered} \tag{8}$$

such that $\hat{F}(x)$ produces $C-1$ margins $[F(x)]_i - [F(x)]_j$ for the ground-truth class i to another class j where $j \neq i$. The optimization objective can then be solved alternatively by running the CROWN algorithm on $\hat{F}(x)$. In this way, we obtain a lower bound for each margin, denoted by

$$\hat{\mathbb{M}}(x;\epsilon) = \{\hat{M}_1, \hat{M}_2, ..., \hat{M}_{C-1}\} \tag{9}$$

It should be noted that the CROWN method is a verification algorithm with very high computational complexity and huge GPU memory requirements. Even if we utilize Tesla-V100 GPUs with 32G memory, running the CROWN algorithm by using standard ResNet18 on CIFAR10 dataset will produce an insufficient memory error based on the library $\mathrm{auto\_LiRPA}$ [1] developed in part by the authors of CROWN. To show the performance of our proposed LPD algorithm and some the-state-of-art methods in terms of certified robustness, we choose the model integrated by 3 scale-down ResNet18 models, in which we reduce the parameter '$\mathrm{in\_channels}$' in the standard ResNet18 from 64 to 4. We compare LPD with Baseline and ADP. In particular, the first 200 samples that could be correctly classified by all three models are selected from the test dataset for evaluation.

Figure 2 compares the certified robustness of different models by their minimal lower bound in $\hat{\mathbb{M}}(x;\epsilon)$ from (9) over the 200 CIFAR-10 samples. The l_∞-norm bounded perturbation has a maximum magnitude of $\epsilon = 0.0005$. The minimal lower bound measures how difficult it is to find an adversarial example x' by imposing $\|x' - x\|_p < \epsilon$ on x. A positive point indicates that no such an x' can be found for x. Note that the more points located above the line of minimal lower bound = 0, the better adversarial robustness the model achieves. It can be seen that the enhanced model trained with both LPD and PDD have far more robust points in Figure 2, which indicates a better gain of certified robustness by the CROWN method [37].

It should also be noted that the provable verification measure only provides a lower bound of *guaranteed* robustness on the output margin under norm-bounded input perturbations. APD promotes the model robustness by diversifying the predictive score distribution over less-likely class labels at the model outputs. It is effective by empirical evaluations under specific attacks as shown in Figure 3, but not as much by certified robustness tests in Figure 2. Certified robustness tests give the worst-case scenario under the perturbation constraint even for unseen attacks. It is usually a conservative estimate in the vicinity of a particular input example and tends to be loose on expressive networks [26]. Thus, there can be a discrepancy of certified robustness from the empirical evaluation under specific attacks. This is demonstrated by LPD whose certified robustness is slightly better than PDD in Figure 2 but can achieve a significant gain of model performance by up to over 20% under actual attacks as shown in Table 4 of the next section.

5 Performance Evaluations

Datasets. Three benchmarks datasets are used, namely CIFAR-10, CIFAR-100 and Tiny-ImageNet. In all cases, the pixel values of images are scaled to be in the interval [0, 1].

Target Models. The deep ensemble model contains $K = 3$ ResNet-18 networks [10]. Dropout-based methods are applied to the last FC layer of 512 neurons before the softmax layer. Both APD and PDD (with DEG enhancement) are implemented with best performing parameters [21, 12].

Attack Methods. Five representative attack methods, namely FGSM [9], BIM [16], PGD [20], JSMA [23], and the C&W attack [2], are implemented with the ART v1.1 library [2] with a pytorch on Tesla V100 GPU. Unless otherwise specified, the attack parameter (Para.) is the attack strength ϵ for FGSM, BIM and PGD, the γ control of

[1] https://github.com/KaidiXu/auto_LiRPA

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

Table 4. Model recognition accuracy (%) under various white-box attacks.

Attacks	CIFAR-10					Tiny-ImageNet				
	Para.	Base.	ADP	PDD	LPD	Para.	Base.	ADP	PDD	LPD
No Attack	-	95.48	95.49	95.58	**95.83**	-	**67.36**	66.94	64.68	64.17
FGSM	0.02	47.07	62.93	58.42	**73.26**	0.02	18.87	12.46	45.63	**50.85**
	0.04	34.32	43.27	41.75	**65.55**	0.04	7.44	3.79	24.92	**30.61**
BIM	0.02	6.01	38.17	30.12	**42.69**	0.01	16.87	12.41	34.22	**41.24**
	0.04	1.00	16.93	12.43	**21.95**	0.02	3.30	2.32	16.35	**21.62**
PGD	0.02	7.49	38.97	33.92	**45.90**	0.01	22.85	19.85	39.16	**44.22**
	0.04	0.37	15.27	13.38	**27.50**	0.02	5.30	3.81	17.49	**25.34**
JSMA	0.05	27.20	50.42	48.78	**68.69**	0.05	43.64	47.74	65.78	**69.68**
	0.10	7.73	21.29	18.67	**49.50**	0.10	26.18	28.95	52.09	**57.47**
C&W	0.10	0.14	0.75	61.60	**68.60**	0.10	3.57	1.02	**27.76**	25.34

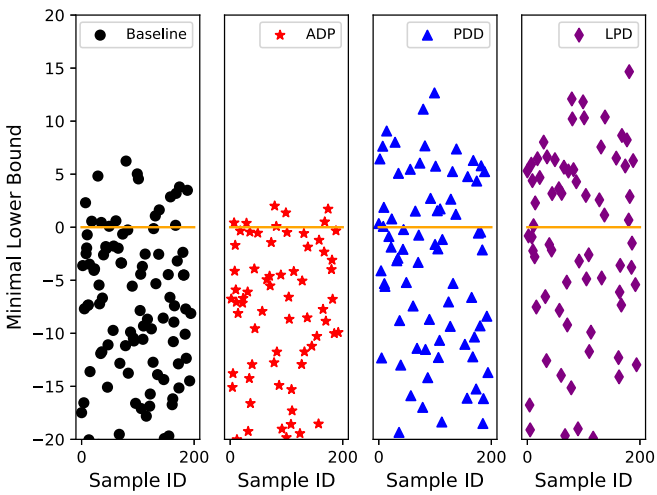

Figure 2. Certified robustness over 200 CIFAR10 samples.

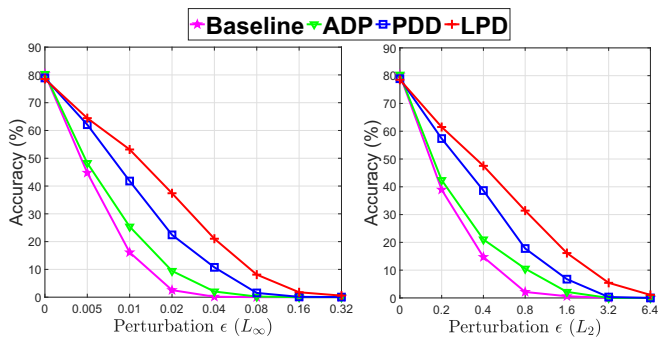

Figure 3. Model robustness by accuracy (%) under the PGD attack with an increasing perturbation value (in $\log_2$ scale) on CIFAR-100.

L_0 perturbation intensity $\theta = 0.1$ for JSMA, and the c control of the classification loss for C&W.

5.1 *Model Adversarial Robustness*

Table 4 shows model recognition accuracy of the comparing ensemble methods under various white-box attacks. It can be seen that the baseline (Base.) results have very low recognition accuracy in general, indicating that the attacks are fairly strong with high success rates. In most cases, the proposed LPD demonstrates better adversarial robustness with a significant gain up to 20% even comparing with the PDD (with DEG enhancement). Comparing with other methods, LPD has improved the model robustness against the BIM and PGD attacks by about 5-15% on CIFAR-10 and Tiny-ImageNet. We also note that the amount of gain varies across different attacks and their settings. This may be due to the stochastic nature of dropout-based regularization for diversified feature learning. The performance can be further boosted by combining our approach with other robustness training methods such as adversarial training as shown in Table 6.

We also evaluate the model accuracy under black-box attacks by leveraging the transferability of adversarial examples built on surrogate models assuming different attacker knowledge [12]. In particular, Type II attacks are adaptive methods by assuming the surrogates are trained with the same defence method (if there is any). The results are reported in Table 5. Again, LPD performs the best in most cases by reducing transferability of adversarial samples.This indicates that the proposed scheme is able to improve adversarial defense by improving model smoothness of the ensemble model under constrained gradient similarity.

Table 5. Model adversarial robustness by accuracy (%) under different black-box attacks on CIFAR-100.

Type	Attacks	Para.	Base.	ADP	PDD	LPD
I	PGD	0.02	40.31	41.10	60.49	**65.14**
		0.04	15.47	17.94	30.38	**34.80**
	JSMA	0.05	66.81	68.59	72.35	**77.45**
		0.1	55.93	59.18	62.24	**68.93**
II	PGD	0.01	77.15	66.35	**87.37**	85.27
		0.02	56.63	47.98	**77.60**	73.08
	JSMA	0.05	56.33	58.46	69.32	**74.24**
		0.1	35.91	39.64	58.71	**66.21**

5.2 Incorporating with Adversarial Training

Our method complements exiting defense paradigms acting on individual models such as adversarial training [9] and other data augmentation schemes [36]. Table 6 shows the results by incorporating the dropout-based regularization schemes with adversarial training with adversarial examples generated by an attack method. In all cases, LPD further boosts the defence performance, especially under attacks of large perturbation intensities and under unseen attacks, e.g., $\mathrm{AdvT}_{\mathrm{FGSM}}$ against PGD and $\mathrm{AdvT}_{\mathrm{PGD}}$ against FGSM.

Table 6. Model accuracy (%) by incorporating adversarial training to the ensemble model on CIFAR-100.

Attacks	CIFAR-100			
	FGSM		PGD	
Para. (ϵ)	0.04	0.08	0.02	0.04
$\mathrm{AdvT}_{\mathrm{FGSM}}$	41.06	20.80	20.15	3.62
$\mathrm{AdvT}_{\mathrm{FGSM}}$ + PDD	59.64	27.09	38.17	19.50
$\mathrm{AdvT}_{\mathrm{FGSM}}$ + LPD	**61.23**	**29.45**	**40.42**	**21.30**
$\mathrm{AdvT}_{\mathrm{PGD}}$	44.14	21.67	44.11	15.69
$\mathrm{AdvT}_{\mathrm{PGD}}$ + PDD	55.56	34.68	51.02	26.13
$\mathrm{AdvT}_{\mathrm{PGD}}$ + LPD	**56.74**	**36.35**	**53.11**	**27.52**

5.3 Increasing Attack Strength

Figure 3 plots the model recognition accuracy by increasing the normalized perturbation intensity ϵ under L_∞ and L_2 norm attacks using PGD, respectively. Note that the plots are on a semi-log scale with the ϵ value doubly increased over the x-axis. It can be seen that LPD is consistently more robust than other methods even under large perturbation attacks. For example, LPD is able to withstand L_∞ perturbations with a normalized intensity of 0.32 (i.e., 82/255 pixels), which is about 8 times stronger than the Baseline limit on CIFAR-100.

5.4 Complexity Analysis

Most of the existing approach aiming at promoting gradient diversity or improving model smoothness inevitably introduce second-order optimization that largely increase the cost of training, which hinders their application in practice. Specifically, the calculation of DEG penalty in [12] requires $\mathcal{O}(K^2)$ for pair-wise operations and $\mathcal{O}(n^2)$ for computing cosine similarity, while the min-max framework used in TRS to compute the regularization term of model smoothness performs back propagation twice and requires $\mathcal{O}(n^2)$, where K is the number of networks and n is the gradient dimension. Different form that, the proposed LPD method involves counting in $\mathcal{O}(C_k)$ and sorting in $\mathcal{O}(I \log I)$, where C_k is the number of FC units in the k-th network and I is the number of intervals. In practice, the training cost is trivial so that the LPD method can be efficiently applied to large-scale models toward complex datasets. We test the training time per epoch with a mini-batch size of 64 on CIFAR-10. When $K = 3$, for instance, it takes 80s/epoch for LPD, 703s/epoch for DEG, 4106s/epoch for TRS, 64s/epoch for ADP, and 54s/epoch for baseline on Tesla V100. It is clear that the proposed LPD method are comparable to the baseline and ADP while significantly superior than two methods based on second-order optimization in terms of training efficiency.

6 Conclusion

In this paper, we propose a new dropout-based simultaneous training strategy that promotes both model smoothness and gradient diversity in the deep learning space of ensemble models. The proposed approach does not involve any second-order optimization and thus have a low computation complexity of simultaneous training for enhancing the deep ensembles. Our evaluations show that our method is simple yet effective under different attacks, especially against adaptive attacks and transferred examples in black-box settings.

Acknowledgements

The work of Yi Wang was supported in part by Natural Science Foundation of China (grant no. 61876038). The work of Da Luo was supported in part by Guangdong Natural Science Key Field Project (2019KZDZX1008).

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner, 'Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples', in *Int. Conf. Mach. Learn.*, volume 80, pp. 274–283, (2018).

[2] Nicholas Carlini and David Wagner, 'Towards evaluating the robustness of neural networks', in *IEEE Symp. Secur. Priv.*, pp. 39–57. IEEE, (2017).

[3] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright, 'HopSkipJumpAttack: A Query-Efficient Decision-Based Attack', in *IEEE Symp. Secur. Priv.*, pp. 1277–1294, San Francisco, CA, US, (apr 2020).

[4] Mingyang Chen, Junda Lu, Yi Wang, Jianbin Qin, and Wei Wang, 'DAIR: A Query-Efficient Decision-based Attack on Image Retrieval Systems', *ACM SIGIR Conf. Res. Dev. Inf. Retr.*, 1064–1073, (2021).

[5] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli, 'Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks', in *USENIX Secur. Symp.*, pp. 321–338, (2019).

[6] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar, 'Stochastic Activation Pruning for Robust Adversarial Defense', in *Int. Conf. Mach. Learn.*, (mar 2018).

[7] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann, and Pushmeet Kohli, 'A dual approach to scalable verification of deep networks', in *Conf. Uncertain. Artif. Intell.*, volume 2, pp. 550–559, (2018).

[8] Panagiotis Eustratiadis, Henry Gouk, Da Li, and Timothy Hospedales, 'Weight-Covariance Alignment for Adversarially Robust Neural Networks', in *Int. Conf. Mach. Learn.*, pp. 1–12, (2021).

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, 'Explaining and Harnessing Adversarial Examples', in *Int. Conf. Learn. Represent.*, (2015).

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, (2016).

[11] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, 'Improving neural networks by preventing co-adaptation of feature detectors', *Comput. Sci.*, **3**(4), 212–223, (2012).

[12] Bo Huang, Zhiwei Ke, Yi Wang, Wei Wang, Linlin Shen, and Feng Liu, 'Adversarial Defence by Diversified Simultaneous Training of Deep Ensembles', in *Proc. AAAI Conf. Artif. Intell.*, pp. 7823–7831, (2021).

[13] Zhiwei Ke, Zhiwei Wen, Weicheng Xie, Yi Wang, and Linlin Shen, 'Group-wise dynamic dropout based on latent semantic variations', in *Proc. AAAI Conf. Artif. Intell.*, pp. 11229–11236, (2020).

[14] Rohit Keshari, Richa Singh, and Mayank Vatsa, 'Guided dropout', in *Proc. AAAI Conf. Artif. Intell.*, pp. 4065–4072, (2019).

[15] L. I. Kuncheva and C. J. Whitaker, 'Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy', *Mach. Learn.*, **51**, 73–82, (2003).

[16] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, 'Adversarial Machine Learning at Scale', in *Int. Conf. Learn. Represent.*, (2017).
[17] Sungyoon Lee, Hoki Kim, and Jaewook Lee, 'GradDiv: Adversarial Robustness of Randomized Neural Networks via Gradient Diversity Regularization', *IEEE Trans. Pattern Anal. Mach. Intell.*, **99**(1), 1–15, (2022).
[18] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft, 'Convergent Learning: Do different neural networks learn the same representations?', in *Int. Conf. Learn. Represent.*, volume 44 of *Proceedings of Machine Learning Research*, pp. 196–212. PMLR, (2016).
[19] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh, 'Towards Robust Neural Networks via Random Self-ensemble', in *Eur. Conf. Comput. Vis.*, pp. 369–385, (dec 2018).
[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, 'Towards Deep Learning Models Resistant to Adversarial Attacks', in *Int. Conf. Learn. Represent.*, (2018).
[21] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu, 'Improving Adversarial Robustness via Promoting Ensemble Diversity', in *Int. Conf. Mach. Learn.*, pp. 4970–4979, (may 2019).
[22] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu, 'Bag of tricks for adversarial training', in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, (2021).
[23] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami, 'The limitations of deep learning in adversarial settings', in *IEEE Eur. Symp. Secur. Priv.*, pp. 372–387. IEEE, (2016).
[24] Andrew Slavin Ros and Finale Doshi-Velez, 'Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients', *Assoc. Adv. Artif. Intell.*, 1660–1669, (2018).
[25] Hadi Salman, Greg Yang, Huan Zhang, Cho Jui Hsieh, and Pengchuan Zhang, 'A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks', in *Adv. Neural Inf. Process. Syst.*, number 32, pp. 9835–9846, (2019).
[26] Samuel Henrique Silva and Peyman Najafirad, 'Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey', *arXiv Prepr.*, **99**(1), (2020).
[27] Aman Sinha, Hongseok Namkoong, and John Duchi, 'Certifying some distributional robustness with principled adversarial training', in *Int. Conf. Learn. Represent.*, pp. 1–49, (2018).
[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting', *J. Mach. Learn. Res.*, **15**(1), 1929–1958, (2014).
[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, 'Intriguing properties of neural networks', in *Int. Conf. Learn. Represent.*, pp. 1–10, Banff, Canada, (2014).
[30] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel, 'Ensemble Adversarial Training: Attacks and Defenses', in *Int. Conf. Learn. Represent.*, (2018).
[31] Theodoros Tsiligkaridis and Jay Roberts, 'Second Order Optimization for Adversarial Robustness and Interpretability', in *Assoc. Adv. Artif. Intell.*, (2020).
[32] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Aurelie Lozano, Cho-Jui Hsieh, and Luca Daniel, 'On Extensions of CLEVER: A Neural Network Robustness Evaluation Algorithm', in *IEEE Glob. Conf. Signal Inf. Process.*, (oct 2018).
[33] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille, 'Mitigating Adversarial Effects Through Randomization', in *Int. Conf. Learn. Represent.*, Palais des Congrès Neptune, Toulon, France, (nov 2018).
[34] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li, 'DVERGE: Diversifying Vulnerabilities for Enhanced Robust Generation of Ensembles', in *Adv. Neural Inf. Process. Syst.*, pp. 1–20, (2020).
[35] Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Benjamin Rubinstein, Pan Zhou, Ce Zhang, and Bo Li, 'TRS: Transferability Reduced Ensemble via Encouraging Gradient Diversity and Model Smoothness', in *Adv. Neural Inf. Process. Syst.*, pp. 1–34, (2021).
[36] Haichao Zhang, Jianyu Wang, Horizon Robotics, and Baidu Research, 'Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training', in *Adv. Neural Inf. Process. Syst.*, pp. 1831–1841, (2019).
[37] Huan Zhang, Minhao Cheng, and Cho-Jui Hsieh, 'Enhancing Certifiable Robustness via a Deep Model Ensemble', in *Safe Mach. Learn. Work. ICLR*, (2019).
[38] Huan Zhang, Tsui Wei Weng, Pin Yu Chen, Cho Jui Hsieh, and Luca Daniel, 'Efficient neural network robustness certification with general activation functions', in *Adv. Neural Inf. Process. Syst.*, volume 2018-Decem, pp. 4939–4948, (2018).