Mask-Guided Divergence Loss Improves the Generalization and Robustness of Deep Neural Network

Xiangyuan Yang¹, Jie Lin¹*, Hanlin Zhang², Xinyu Yang¹, and Peng Zhao¹

¹ Xi’an Jiaotong University, Xi’an, China
ouyang_xy@stu.xjtu.edu.cn; {jielin, yxyphd, p.zhao}@mail.xjtu.edu.cn
² Qingdao University, Qingdao, China
hanlin@qdu.edu.cn

Abstract. Deep neural network (DNN) with dropout can be regarded as an ensemble model consisting of lots of sub-DNNs (i.e., an ensemble sub-DNN where the sub-DNN is the remaining part of the DNN after dropout), and through increasing the diversity of the ensemble sub-DNN, the generalization and robustness of the DNN can be effectively improved. In this paper, a mask-guided divergence loss function (MDL), which consists of a cross-entropy loss term and an orthogonal term, is proposed to increase the diversity of the ensemble sub-DNN by the added orthogonal term. Particularly, the mask technique is introduced to assist in generating the orthogonal term for avoiding overfitting of the diversity learning. The theoretical analysis and extensive experiments on 4 datasets (i.e., MNIST, FashionMNIST, CIFAR10, and CIFAR100) manifest that MDL can improve the generalization and robustness of standard training and adversarial training. For CIFAR10 and CIFAR100, in standard training, the maximum improvement of accuracy is 1.38% on natural data, 30.97% on FGSM (i.e., Fast Gradient Sign Method) attack, 38.18% on PGD (i.e., Projected Gradient Descent) attack. While in adversarial training, the maximum improvement is 1.68% on natural data, 4.03% on FGSM attack and 2.65% on PGD attack.

Keywords: Mask-guided divergence loss · Generalization · Robustness · Cosine similarity · Deep neural network

1 Introduction

Deep learning has penetrated various daily applications, and how to improve the performance of the deep neural network (DNN) has been received wide attention. Due to the existence of overfitting, the DNN does not realize the full potentialities. In addition, the imperceptible perturbations [9,29] cause the DNN to misclassify, resulting in significant reduction on classification accuracy. Therefore, to design a DNN with higher performance, the generalization and robustness of the DNN should be improved at the same time.

* Corresponding author
To improve the generalization of the DNN avoiding the overfitting, dropout has been proposed to prevent co-adaptation of feature detectors by omitting a part of neurons with a fixed probability. Subsequently, many variants of dropout gradually have been appeared, for example, dropconnect improves the generalization of neural networks by omitting links between neurons of adjacent layers and dropblock omits regions in the convolutional layer of the DNN. Multi-sample dropout was proposed to accelerate training and obtain better generalization, which dropouts the feature layer in the DNN multiple times and calculates the average loss in every feedforward and backward training step. Disout was proposed to replace with dropout, which introduces feature map distortion to reduce the Rademacher complexity of the DNN for improving generalization. In addition, Srivastava et al. and Konda et al. explained that dropout can improve the generalization of the DNN through ensemble model and data augmentation. However, most of these existing efforts are barely considered to improve the generalization of the DNN via increasing the diversity of the ensemble sub-DNN. Meanwhile, these regularization methods can not guarantee the robustness of the DNN.

To enhance the robustness of the DNN resisting adversarial attacks, adversarial training is the most effective defense method. Adversarial training methods can be divided into two categories: the only robustness of the DNN improved and the trade-off between the robustness and the accuracy of natural samples. Free adversarial training (Free-AT) and Fast adversarial training (Fast-AT) belong to the former category. Free-AT was proposed to eliminate the overhead cost of generating adversarial examples by recycling the gradient information computed. Fast-AT is a much weaker and cheaper adversary that is not more costly than standard training. The trade-off between robustness and accuracy (TRADES) and its variants, i.e., adversarial training with transferable adversarial examples (ATTA) and friendly adversarial training, belong to the latter category. TRADES is a new defense algorithm to identify a trade-off between robustness and accuracy. ATTA improves the training efficiency by accumulating adversarial perturbations through epochs. FAT was proposed to minimize the loss by using the least adversarial data that are confidently misclassified. However, all adversarial training methods will greatly reduce the accuracy of natural samples.

To solve the above issues, in this paper, a mask-guided divergence loss function, namely MDL, is proposed to increase the diversity of the ensemble sub-DNN, thereby improving the generalization and robustness of the DNN in standard training and adversarial training.

The contributions of our paper can be summarized as follows:

First, a mask-guided divergence loss function (MDL), which consists of the cross-entropy loss term and the orthogonal term, is proposed to increase the diversity of the ensemble sub-DNN. Specifically, the added orthogonal term are responsible for increasing the diversity of the ensemble sub-DNN. In addition, to avoid overfitting in diversity learning, the mask technique is also introduced in
our MDL to ensure that a part of training samples with high correct predicted probabilities participate in calculating the orthogonal term in MDL.

Second, the theoretical analysis is conducted to prove that the proposed MDL can improve the generalization and robustness of the DNN. Generally, when the correct predicted probability of a sample is less than 0.5, the sample may be classified incorrectly. However, for the DNN trained by MDL, when MDL converges to 0, the predicted probabilities of false categories are constrained to less than \( \frac{1}{M-1} \) where \( M \) is the number of the category. For MNIST, FashionMNIST, CIFAR10, CIFAR100, etc., \( \frac{1}{M-1} \) is less than 0.5. Therefore, more samples will be classified correctly. Meanwhile, a sample will be classified incorrectly only if it is attacked with greater strength.

Finally, to evaluate the effectiveness of our proposed MDL, several experiments are conducted based on 4 datasets (i.e., MNIST, FashionMNIST, CIFAR10, and CIFAR100). The evaluation results demonstrate that our proposed MDL function can improve the generalization and robustness of the DNN in standard training and adversarial training. (i) In standard training, the accuracy of MDL method is greater than that of the other 9 compared regularization methods on natural samples. Under FGSM and PGD attacks, MDL can achieve much more robust than these regularization methods. Particularly, under FGSM (\( \varepsilon = 8/255 \)) attack, the accuracies of VGG16 and ResNet50 on CIFAR10 are 61.92%, 62.84%, respectively, which achieve a comparable level with that of adversarial training methods. In addition, MDL has good compatibility with these regularization methods. (ii) In adversarial training, MDL can enhance the robustness of TRADES \(^{[55]} \) under FGSM and PGD attacks.

2 Multi-Sample Dropout

In this section, multi-sample dropout \(^{[15]} \) is briefly mentioned, which can be considered as the foundation of our proposed MDL.

In multi-sample dropout \(^{[15]} \), the last feature layer in the DNN dropouts multiple times in each feedforward, and the average loss is calculated by all outputs. Then feedback is used to update the parameters of the DNN, which are equivalent to training multiple sub-DNNs in each training step of the DNN. Generally, in the DNN with dropout, only one sub-DNN is trained in each training step. Therefore, in the same training time, multi-sample dropout can make the DNN converge faster and achieve better and higher generalization performance than dropout.

3 Mask-Guided Divergence Loss (MDL)

In this section, the proposed mask-guided divergence loss function is described in detail. The basic idea is that due to the DNN with dropout can be regarded as an ensemble sub-DNN, the generalization and robustness of the DNN can be improved by increasing the diversity of the ensemble sub-DNN. However, to
increase the diversity of the ensemble sub-DNN, two problems should be solved: i) in each training step of the DNN with dropout, only one sub-DNN updates the parameters, which can not actively enhance the diversity of ensemble sub-DNN, and ii) how to define the diversity among sub-DNNs.

To actively enhance the diversity of the ensemble sub-DNN, multi-sample dropout [15] can be used to dropout the last feature layers multiple times in each training step, and then the outputs of multiple sub-DNNs are jointly calculated to increase the diversity among sub-DNNs. However, to define the diversity among sub-DNNs, the following statement can be devoted.

The output of the last convolutional layer is input to dropout two times. Then, after going through all fully connected layers and softmax, two classification results can be obtained, i.e., the outputs of the two sub-DNNs. The correct predicted probabilities of the two sub-DNNs are used to calculate the mask, and the false categories predicted vectors of the two sub-DNNs take part in calculating the diversity, where the mask and diversity constitute the orthogonal term in the proposed MDL. Assuming that the false categories predicted vectors for the sample \((s, y_s)\) output by the first and second sub-DNNs are denoted as \(\bar{P}_{1s} = P_{1s} \setminus y_s\) and \(\bar{P}_{2s} = P_{2s} \setminus y_s\), respectively, where \(P_{1s}\) and \(P_{2s}\) are the predicted probability vectors of the first sub-DNN and the second sub-DNN, respectively, \(\setminus y_s\) denotes removing the correct probability from the predicted probability vector. Here, cosine similarity can be used to define the diversity among sub-DNNs: cosine similarity based diversity \(D_1(P_{1s}, P_{2s})\) can be represented as

\[
D_1(P_{1s}, P_{2s}) = \frac{\bar{P}_{1s} \cdot \bar{P}_{2s}}{\|\bar{P}_{1s}\|_2 \|\bar{P}_{2s}\|_2} \tag{1}
\]

where \(\| \cdot \|_2\) is l2 norm. The smaller \(D_1(P_{1s}, P_{2s})\), the greater the diversity.

Generally, if there are \(K\) sub-DNNs in one training step, the calculation formula of the orthogonal term with cosine similarity based diversity in mask-guided divergence loss can be represented as

\[
O(P_{1s}, P_{2s}, \ldots, P_{Ks}) = \xi(P_s) \sum_{i=1}^{K} \sum_{j=i+1}^{K} D_1(P^i_s, P^j_s) \tag{2}
\]

where \(P^i_s\) denotes the predicted probability vector of \(i^{th}\) sub-DNN, the mask \(\xi(P_s)\) ensures that a certain percentage of samples (denoted as \(\eta\%)\) with high correct predicted probabilities in each batch size of training samples participate in the calculation of the orthogonal term, which can be represented as

\[
\xi(P_s) = \begin{cases} 
1, & -\log (p^{y_s}_s) \leq T \\
0, & -\log (p^{y_s}_s) > T 
\end{cases} \tag{3}
\]
The effect of the mask is to avoid overfitting because the existence of samples that are difficult to learn in the training set can make the learned diversity have a more complex expression.

Finally, for the sample \((s, y_s)\), the mask-guided divergence loss consists of the average cross-entropy loss of \(K\) sub-DNNs (i.e., the cross-entropy loss term) and the orthogonal term, which can be represented as

\[
L_{MDL} = \sum_{i \leq K} \frac{L^i_{CE}}{K} + \rho \cdot \frac{O \left( P^1_s, P^2_s, \ldots, P^K_s \right)}{K \cdot (K-1)}^\frac{1}{2}
\]

where the calculation formula of the cross-entropy of the \(i\)-th sub-DNN can be represented as

\[
L^i_{CE} = -\sum_{m=1}^{M} q (m \mid s) \log (p^i_s m)
\]

where \(q(m = y_s \mid s) = 1\) and \(q(m \neq y_s \mid s) = 0\), \(p^i_s m\) denotes the predicted probability of category \(m\) of \(i\)-th sub-DNN for sample \(s\), and \(\rho\) is a weight coefficient.

4 Theoretical Analysis

In this section, we will prove why MDL can improve the generalization and robustness of the DNN. For an input image \(s\), the correct category \(y_s\) is assumed as the \(M\)-th category where \(M\) is the number of categories, and the false categories predicted vector of the \(i\)-th sub-DNN is \([p^i_1 s, p^i_2 s, \ldots, p^i_{(M-1)} s]\).

**Proposition 1.** When MDL loss function converges \(L_{MDL} \rightarrow 0\), the diversity of \(N\) sub-DNNs is maximized. Therefore, for the input image \(s\), each axis is paralleled by \(\frac{N}{M-1}\) sub-DNNs’ false categories predicted vectors where \(M\) is the number of the category.

**Proof.** Proposition 1 can be simplified as: the scheme of minimizing the sum of cosine values between \(N\) number of \((M-1)\)-dimensional vectors is that each axis is paralleled by \(\frac{N}{M-1}\) vectors. First, there are only 0 and \(\frac{\pi}{2}\) angles between the vectors, because the existence of acute angle will increase the sum of cosine values. For example, when \(a_1\) number of 0 and \(a_2\) number of \(\frac{\pi}{2}\) angles are adjusted to \(a_1 + a_2\) acute angles \(\theta_1^{a_1}, \theta_2^{a_2}, \ldots, \theta_{a_1+a_2}^{a_1}\),

\[
\min_{\theta_1^{a_1}, \theta_2^{a_2}, \ldots} \cos \theta_1^{a_1} + \cos \theta_2^{a_1} + \cdots + \cos \theta_{a_1+a_2}^{a_1} > a_1
\]

s.t. \(\theta_1^{a_1} + \theta_2^{a_1} + \cdots + \theta_{a_1+a_2}^{a_1} = \frac{a_2 \pi}{2}\)

\(\forall 1 \leq i \leq a_1 + a_2, 0 < \theta_i^{a_1} < \frac{\pi}{2}\)

\(\theta_1^{a_1} \geq \theta_2^{a_1} \geq \cdots \geq \theta_{a_1+a_2}^{a_1}\)
which is proved by mathematical induction as follow: when \( a_1 = 1 \), if \( a_2 \) is an odd number,

\[
\cos \theta_1 + \cos \theta_2 + \cdots + \cos \theta_{1+a_2} > \pm \cos \left( \theta_1 + \theta_2 + \cdots + \theta_{a_2} \right) + \cos \theta_{1+a_2} \tag{7}
\]

\[
= \pm \cos \left( \frac{a_2}{2} \pi - \theta_{1+a_2} \right) + \cos \theta_{1+a_2} = \sin \theta_{1+a_2} + \cos \theta_{1+a_2} > 1 \tag{8}
\]

if \( a_2 \) is an even number,

\[
\cos \theta_1 + \cos \theta_2 + \cdots + \cos \theta_{1+a_2} > \pm \sin \left( \theta_1 + \theta_2 + \cdots + \theta_{a_2} \right) + \cos \theta_{1+a_2} \tag{9}
\]

\[
= \pm \sin \left( \frac{a_2}{2} \pi - \theta_{1+a_2} \right) + \cos \theta_{1+a_2} = \sin \theta_{1+a_2} + \cos \theta_{1+a_2} > 1 \tag{10}
\]

where Eq. 7 and Eq. 9 can be obtained by the undetermined coefficient method. Therefore, when \( a_1 = 1 \), Eq. 6 is satisfied. Assuming \( a_1 = k \), Eq. 6 is also satisfied, so

\[
\cos \theta_1 + \cos \theta_2 + \cdots + \cos \theta_{k+a_2} > k \tag{11}
\]

When \( a_1 = k + 1 \),

\[
\cos \theta_1^{k+1} + \cos \theta_2^{k+1} + \cdots + \cos \theta_{k+a_2}^{k+1} = \cos \theta_1^k + \cos \theta_2^k + \cdots + \cos \theta_{k+a_2}^k + \left( \cos \theta_{k+1}^k - \cos \theta_1^k \right) + \left( \cos \theta_{k+2}^k - \cos \theta_{k+1}^k \right) + \cdots + \left( \cos \theta_{k+a_2}^k - \cos \theta_{k+a_2}^k \right) + \left( \cos \theta_{k+a_2+1}^k - \cos \theta_{k+a_2}^k \right) > k + 1 + \theta_{k+1}^{k+1} \sin \left( \min_{1 \leq i \leq k+a_2} \theta_{i}^{k+1} \right) - \sin \left( \theta_{k+1}^{k+1} \right) \tag{12}
\]

Therefore, Eq. 6 is successful. Second, according to the mean value theorem, when the number of vectors paralleled to each axis is equal, i.e. \( \frac{N}{M-1} \), the sum of cosine values is the smallest. In summary, the Proposition 1 is proved.

According to Proposition 1 for the input image \( s \), the probability of \( j \)-th false category predicted by the DNN trained with MDL is

\[
p_s^j = \frac{p_s^{1j} + p_s^{2j} + \cdots + p_s^{Nj}}{N} \leq \frac{N}{M-1} = \frac{1}{M-1} \tag{13}
\]

where \( 1 \leq j \leq M-1 \) when the correct category of \( s \) is \( M \). For a 10-category task (e.g., MNIST and CIFAR10), \( p_s^j \leq \frac{1}{5} \). Ideally, if \( p_s^M \) is greater than \( \frac{1}{9} \), the input image \( s \) can confidently be classified correctly. However, for the DNN trained without MDL, the input image \( s \) can confidently be classified correctly if and only if \( p_s^M \) is greater than \( \frac{1}{2} \). Therefore, MDL can increase the generalization of the DNN. In the case of being attacked, greater attack strength is required to make \( p_s^M \) less than \( \frac{1}{9} \). Hence, the robustness of the DNN is also enhanced.

5 Evaluations and Limitations

5.1 Evaluation Setup

The effectiveness evaluation of the proposed mask-guided divergence loss function is conducted based on MNIST [22], FashionMNIST [50], CIFAR10 [21] and
CIFAR100 [21] datasets. MNISTNet [33], which is a 7-layer network with 4 convolutional layers and 3 fully connected layers, is used to evaluate the generalization and robustness of MDL on MNIST and FashionMNIST with standard training and adversarial training, VGG16 [41] and ResNet50 [11] on CIFAR10 and CIFAR100 with standard training, and ResNet18 [11] on CIFAR10 and CIFAR100 with adversarial training.

All DNNs are trained with a batch size of 128 using stochastic gradient descent (SGD) with 0.9 momentum and $5 \cdot 10^{-4}$ decay rate. In standard training, MNISTNet trains 20 epochs on MNIST and 50 epochs on FashionMNIST, both VGG16 and ResNet50 train 200 epochs on CIFAR10 and CIFAR100, respectively. In adversarial training, MNISTNet trains 50 epochs on MNIST and FashionMNIST, respectively, ResNet18 trains 100 epochs on CIFAR10 and CIFAR100, respectively.

In addition, two popular learning rate schedulers are used: Multistep and Cyclic [42]. For Multistep, the learning rate is initially set to 0.01 (or 0.1) and decays at the $\frac{1}{2}$ and $\frac{3}{4}$ of the total epochs by a factor of 0.1, respectively. For Cyclic, the max and min learning rate are set to 0.01 and 0, respectively. The metric of all evaluations is the accuracy (%). In the evaluation of our MDL, the number of sub-DNN $K$ in each training step is set to 4, the dropout rate is 0.5, $\eta$ choose from $\{100, 90, 80, 70\}$ and the weight coefficient $\rho$ is 1.

**Generalization Evaluation Setup.** The 9 compared methods in the evaluation are consist of 5 model-level methods (i.e., dropout (DO) [12], dropconnect (DC) [46], dropblock (DB) [7], multi-sample dropout (MSD) [15], disout [45]) and 4 loss-level methods (i.e., label smoothing (LS) [30], focal loss (FL) [26], label relaxation (LR) [25], online label smoothing (OLS) [52]). The training parameters of these compared methods are selected from the original experiment. In addition, cosine similarity based diversity $D_1 (P_1^s, P_2^s)$ in Eq. 1 replaces with Pearson correlation coefficient (PCC) based diversity $D_2 (P_1^s, P_2^s)$, called MDL-PCC:

$$Pe(P_1^s, P_2^s) = \frac{\sum_{i=1}^{M-1} (\bar{p}_{si} - u(\bar{P}_1^s)) (\bar{p}_{si} - u(\bar{P}_2^s))}{\| \bar{P}_1^s - u(\bar{P}_1^s) \|_2 \| \bar{P}_2^s - u(\bar{P}_2^s) \|_2}$$ (14)

$$D_2(P_1^s, P_2^s) = (Pe(P_1^s, P_2^s) + 1) / 2$$ (15)

where $\bar{p}_{si}^{1i}$ and $\bar{p}_{si}^{2i}$ are the $i$th element in $\bar{P}_1^s$ and $\bar{P}_2^s$, $\| \cdot \|_2$ is l2 norm, $u(\cdot)$ is the average of vector. The smaller $D_2 (P_1^s, P_2^s)$, the greater the diversity.

**Robustness Evaluation Setup.** TRADES [55] is selected as the compared adversarial training method with $\beta = 1$ on MNIST and $\beta = 6$ on FashionMNIST, CIFAR10, and CIFAR100. For the combination of TRADES and MDL, MDL only applies to natural samples. FGSM [9] and PGD [29] with different attack strength, e.g., $\varepsilon = [1/255, 2/255, \cdots, 8/255, 12/255, 16/255]$ on CIFAR10 and CIFAR100, $\varepsilon = [0.05, 0.1, \cdots, 0.3, 0.35, 0.4]$ on MNIST and $\varepsilon =$
Table 1. Comparison between our method and the state-of-the-art approaches on accuracy (%) and average rank. Note that M, FM, C10 and C100 represent MNIST, Fashion-MNIST, CIFAR10 and CIFAR100 respectively.

| Method      | MNISTNet | VGG16 | ResNet50 |
|-------------|----------|-------|----------|
|             | Cyclic   | Multistep | Cyclic | Multistep | Cyclic | Multistep | Avg. rank |
|             | M        | FM     | M        | C10      | C100    | C10      | C10       | C100       | C10       | C100       | C10       | C100       | C10       | C100       | C10       |
| DO          | 99.56    | 93.41  | 99.61    | 93.73    | 93.05   | 70.67    | 93.45    | 72.16   | 92.69   | 70.67    | 94.53    | 76.04    | 4.33 |
| FL          | 99.49    | 93.11  | 99.55    | 93.18    | 93.14   | 71.37    | 92.92    | 71.36   | 91.68   | 69.62   | 92.12    | 74.06    | 6.08 |
| DB          | 99.45    | 92.58  | 99.53    | 93.46    | 93.46   | 71.12    | 93.52    | 71.58   | 92.99   | 66.05   | 94.57    | 75.96    | 5.25 |
| MSD         | 99.55    | 92.89  | 99.58    | 93.36    | 93.31   | 71.39    | 93.59    | 72.17   | 92.74   | 70.04   | 94.66    | 75.42    | 4.0  |
| Disout      | 99.54    | 92.15  | 99.51    | 92.45    | 93.07   | 71.48    | 93.57    | 71.42   | 93.00   | 67.47   | 94.62    | 76.00    | 5.25 |
| LR          | 99.38    | 92.77  | 99.45    | 93.00    | 92.39   | 70.30    | 88.42    | 71.10   | 92.31   | 69.32   | 92.29    | 73.80    | 7.42 |
| MDL         | 99.61    | 93.53  | 99.65    | 93.95    | 93.51   | 71.99    | 93.67    | 72.59   | 93.99   | 72.05   | 94.69    | 76.75    | 1.17 |
| MDL-PCC     | 99.62    | 93.49  | 99.64    | 93.81    | 93.35   | 71.45    | 92.96    | 72.03   | 93.54   | 71.86   | 94.89    | 76.56    | 2.5  |

Table 2. Compatibility verification of our method with dropout variant, label smoothing and its variant on accuracy(%). Note that M, FM, C10 and C100 represent MNIST, Fashion-MNIST, CIFAR10 and CIFAR100 respectively.

| Method      | MNISTNet | VGG16 | ResNet50 |
|-------------|----------|-------|----------|
|             | Cyclic   | Multistep | Cyclic | Multistep | Cyclic | Multistep | Avg. rank |
|             | M        | FM     | M        | C10      | C100    | C10      | C10       | C100       | C10       | C100       | C10       | C100       | C10       |
| DC          | 99.50    | 92.18  | 99.50    | 92.60    | 93.08   | 71.39    | 93.52    | 71.55   | 92.89   | 67.27   | 94.50    | 76.01 |
| DC+MDL      | 99.55    | 92.80  | 99.56    | 93.21    | 93.55   | 71.86    | 93.33    | 72.11   | 94.47   | 72.98   | 94.55    | 77.11 |
| LS          | 99.47    | 93.43  | 99.55    | 93.53    | 93.17   | 72.38    | 93.49    | 72.39   | 92.51   | 69.19   | 93.99    | 76.15 |
| LS+MDL      | 99.48    | 93.55  | 99.55    | 94.05    | 93.37   | 72.41    | 93.42    | 72.46   | 93.36   | 68.34   | 94.58    | 76.05 |
| OLS         | 99.55    | 93.37  | 99.57    | 93.75    | 93.26   | 72.51    | 93.69    | 72.35   | 92.55   | 70.23   | 94.47    | 76.00 |
| OLS+MDL     | 99.52    | 93.53  | 99.58    | 93.88    | 93.34   | 72.91    | 93.44    | 72.72   | 93.42   | 69.29   | 94.63    | 76.42 |

[2/255, 4.255, ..., 16/255, 24/255] on FashionMNIST, are used to evaluate the robustness of our MDL in standard training and adversarial training. All robustness test DNNs are trained with the Multistep scheduler. Note that PGD with 40 steps (i.e., PGD-40) on MNIST and PGD with 20 steps (i.e., PGD-20) on FashionMNIST, CIFAR10, and CIFAR100 are used.

All experiments are executed on four NVIDIA V100 GPUs, and the deep learning framework used is pytorch 3.7. Note that the percentage sign (%) is omitted in the accuracies of all evaluation results.

5.2 Generalization Evaluation of MDL

Comparison among MDL, Dropout, Label Smoothing and Their Variants. Tables 1 and 2 show the accuracies of MDL, MDL-PCC and the other 9 compared regularization methods. In Table 1, the average rank of MDL is the best, and MDL-PCC is the second, which show that the improved diversity of the DNN can improve the generalization performance, and cosine similarity based diversity is better than Pearson correlation coefficient based. As shown in Tables 1 and 2, the improvements of MDL compared with the maximum accuracy of the 9 regularization methods are 0.05%, 0.1%, 0.05%, 0.99%, 1.38% on MNIST, FashionMNIST, both VGG16 and ResNet50 of CIFAR10, and ResNet50 of CIFAR100 by using the Cyclic scheduler, respectively, and 0.04%, 0.2%, 0.03%,
Fig. 1. Relationship between the parameter $\eta$ and accuracy (%). Note that MC, MM, VC, VM, RC and RM represent MNIST-Net with Cyclic and Multistep, VGG16 with Cyclic and Multistep, ResNet50 with Cyclic and Multistep, respectively.

Fig. 2. The t-SNE visualization of the model logits on CIFAR10. The subgraph (1), (2), (3) and (4) are VGG16 and ResNet50 without and with MDL, respectively.

0.2%, 0.6% on MNIST, FashionMNIST, both VGG16 and ResNet50 of CIFAR10, and ResNet50 of CIFAR100 by using the Multistep scheduler, respectively. Although OLS has the maximum accuracy on VGG16 of CIFAR100 by using the Cyclic scheduler, the combination of OLS and MDL can improve 0.4% to OLS. In summary, MDL has the best comprehensive generalization.

Compatibility between MDL and Other Methods. Table 2 shows the generalization improvements of MDL combined with dropconnect, LS, and OLS on 4 datasets, respectively. For the DNN with dropconnect, MDL can effectively improve the accuracies of the DNNs on 4 datasets and 2 learning schedulers, i.e., 0.05%, 0.62%, 0.47%, 1.58%, 0.47%, and 5.71% on MNIST, FashionMNIST, both VGG16 and ResNet50 of CIFAR10 and CIFAR100 respectively by using the Cyclic scheduler, and 0.06%, 0.61%, 0.05%, 0.56% and 1.1% on MNIST, FashionMNIST, ResNet50 of CIFAR10, both VGG16 and ResNet50 of CIFAR100 respectively by using the Multistep scheduler, except for the accuracy reduction of 0.19% on VGG16 of CIFAR10 by using the Multistep scheduler. For label smoothing and online label smoothing, MDL can improve the accuracies of the DNNs on MNIST and Fashion-MNIST datasets, and the accuracies of the most DNNs on CIFAR10 and CIFAR100. Therefore, MDL has good compatibility with dropconnect (a variant of dropout), and certain compatibility with label smoothing and online label smoothing (a variant of label smoothing).
Fig. 3. The accuracy (%) lines of dropout, MSD and our MDL on 4 datasets under FGSM and PGD attacks with different strength, respectively. The first row of subgraphs are under FGSM attack, the second row are under PGD attack.

Impacts of $\eta$ on Generalization. Fig. 1 shows the impacts of the size of $\eta$ on 4 datasets. As Fig. 1 shows, the parameter $\eta$ has a certain impact on the accuracies of all DNNs on 4 datasets. Generally, when the size of $\eta$% is equal or near to the percentage of the accuracy, the DNN using MDL can obtain the optimal or sub-optimal generalization, which saves the time consumption on adjusting the parameter $\eta$. For example, for MNIST, FashionMNIST, CIFAR10, and CIFAR100, when $\eta$% is set to 100%, 90%, 100%, and 70% near the percentage of the accuracies, respectively (i.e., 99%, 93%, 94%, and 72%), all DNNs can obtain the optimal generalization. Therefore, the optimal generalization of the DNN can be achieved by setting the parameter $\eta$% as the percentage of the accuracy approximately. Note that the initial value of the parameter $\eta$ is set to 100.

The t-SNE Visualization of Features Extracted from the DNNs Trained by Different Methods. To study the influence of MDL on feature extraction, we visualize the model logits of the CIFAR10 test dataset by using t-SNE [28]. Fig. 2(1) and Fig. 2(2) are the t-SNE visualization of VGG16 with Dropout and MDL, respectively. Fig. 2(3) and Fig. 2(4) are the t-SNE visualization ResNet50 with Dropout and MDL, respectively. In Fig. 2(1) and Fig. 2(3), there are many separate small clusters, which verifies that the DNN with dropout has learned more complex classification boundaries. However, in Fig. 2(2) and Fig. 2(4), due to few separate small clusters, the classification boundaries of the DNN with MDL are simple. Therefore, dropout with MDL can achieve better generalization. Note that because MDL can ensure that the probability of each false category is less than $\frac{1}{M-1}$, the probabilities of all categories are close around $\frac{1}{M-1}$ for wrong classification samples and low confidence samples, thereby Fig. 2(2) and Fig. 2(4) have a radial distribution.
Fig. 4. The accuracy (%) lines of DC and the combination of MDL and DC on 4 datasets under FGSM and PGD attacks with different strength, respectively. The first row of subgraphs are under FGSM attack, the second row are under PGD attack.

Fig. 5. The accuracy (%) lines of TRADES and the combination of MDL and TRADES on 4 datasets under FGSM and PGD attacks with different strength, respectively. The first row of subgraphs are under FGSM attack, the second row are under PGD attack.

5.3 Robustness Evaluation of MDL

Comparison among MDL, Dropout and MSD in Standard Training. Fig. 3 evaluated the robustness of the DNN with different regularization methods on 4 datasets. Under FGSM and PGD attacks with any strength, the accuracy of MDL is greater than that of dropout and MSD. Under FGSM attack, the maximum accuracy improvement is 29.73%, 34.88%, 23.11%, 30.97%, 26.67% and 9.04% on MNIST, FashionMNIST, both VGG16 and ResNet50 of CIFAR10 and CIFAR100, respectively. Under PGD attack, the maximum accuracy improvement is 12.05%, 12.15%, 38.18%, 31.35%, 34.48% and 7.56%, respectively. Therefore, MDL significantly improves the robustness of the DNN in standard training. In addition, under FGSM ($\varepsilon=8/255$) attack, the accuracies of VGG16 and ResNet50 with MDL on CIFAR10 are 61.92%, 62.84%, respectively, which have reached a comparable level with that of adversarial training methods.
Compatibility between MDL and Dropconnect in Standard Training.
To verify that MDL is compatible with dropconnect, the dropout operator in MDL replaces with dropconnect. Fig. 4 evaluated the robustness of the DNN trained by the combination of MDL and dropconnect on 4 datasets. Under FGSM and PGD attacks with any strength, the accuracy of the combination of MDL and dropconnect is greater than that of dropconnect. Under FGSM attack, the maximum accuracy improvement is 34.02%, 21.34%, 18.22%, 21.28%, 14.14% and 9.63% on MNIST, FashionMNIST, both VGG16 and ResNet50 of CIFAR10 and CIFAR100, respectively. Under PGD attack, the maximum accuracy improvement is 33.8%, 14.43%, 27.85%, 12.13%, 11.91%, 4.17%, respectively. Therefore, the combination of MDL and dropconnect does not affect the significant improvement in the robustness of the DNN.

Compatibility between MDL and TRADES in Adversarial Training.
To verify that MDL is compatible with adversarial training methods, TRADES is used to train the DNN with MDL. Fig. 5 evaluated the robustness of the DNN with or without MDL trained by TRADES on 4 datasets. Under FGSM and PGD attacks with any strength, the accuracy of the DNN with MDL is greater than that of without MDL. Under FGSM attack, the maximum accuracy improvement is 21.01%, 0.82%, 4.03% and 2.13% on MNIST, FashionMNIST, CIFAR10 and CIFAR100, respectively. Under PGD attack, the maximum accuracy improvement is 20.61%, 0.95%, 2.63% and 2.65%, respectively.

For MNIST, when the attack strength of FGSM and PGD is small, the accuracies of the DNN with and without MDL are the same and approximated to 100%, but the accuracies of the DNN with MDL can be improved in the large attack strength of FGSM and PGD. For FashionMNIST, when the attack strength of FGSM and PGD is large, the accuracies of the DNN with and without MDL are the same, but the accuracies of the DNN with MDL can be steadily improved in the range of \((0,16/255]\) attack strength of FGSM and PGD. For CIFAR10 and CIFAR100, the accuracies of ResNet18 are steadily improved for any attack strength of FGSM and PGD. In summary, MDL has good compatibility with the adversarial training method, TRADES, and can further improve the robustness of the DNN.

5.4 Limitations
MDL is suitable for the adversarial training methods of the trade-off between robustness and accuracy, e.g., TRADES \[55\] and its variants, but it is not suitable for the adversarial training methods with only adversarial samples participating in the calculation of the loss function, e.g., Fast-AT \[49\], and Free-AT \[38\].

6 Related Work
In this section, we introduce regularization methods that can improve the generalization of the DNN and adversarial training methods that can enhance the robustness of the DNN.
6.1 Regularization Methods

Zhang et al. [53] concluded two types of regularization methods: explicit regularization and implicit regularization, but the improvement of the DNNs is limited. Novak et al. [31] explored the sensitivity of trained neural networks in perturbation, and the experimental results show that poor generalization corresponds to lower robustness and good generalization gives rise to more robust functions. Presently, the regulariztion methods can be categorized into three types: dropout, label smoothing and, batch normalization.

**Dropout and Its Variants.** StochasticDepth [14] was proposed to avoid over-fitting and improve the inference speed of ResNet by adding a probability on residual connect. Different from dropout which randomly selects the neurons to set to zero, weighted channel dropout [13] operates on the channels in the stack of convolutional layers. Some Dropout variants are suitable for different DNN structures and different applications, such as auto-encoder [23], Transformer [5], recurrent neural network [19], image restoration [35], adversarial training [37] and reinforcement learning [19].

For dynamic dropout variants, their parameters change dynamically based on prior information. A reinforcement learning approach [34] was proposed to find better dropout patterns for various architectures. Contextual dropout [6] was proposed to determine the dropout rate by input covariates. In group-wise dynamic dropout [17], the dropout rate of different groups is dynamically changed based on local feature densities. Guided dropout [18] chooses guided neurons for intelligent dropout, which leads to better generalization as compared to the traditional dropout.

Arora et al. [1] investigated the capacity control of dropout in various machine learning problems, and calculated the generalization error bound by Rademacher complexity. Zhang et al. [54] verified that dropout can prevent the co-adaptation of features in the DNN with theories and experiments. Wei et al. [48] analyzed the effectiveness of dropout from two aspects of explicit regularization effect and implicit regularization effect, where explicit regularization effect is explained as that dropout changes the objective loss function of neural networks and implicit regularization effect is regarded as that dropout introduces mean-zero noise in gradient. Pal et al. [32] theoretically verified that dropblock induces spectral k-support norm regularization and dropconnect is equivalent to dropout for single hidden-layer linear networks.

**Label Smoothing and Its Variants.** Bahri et al. [2] presented several baselines to reduce churn and a K-Nearest Neighbor based label smoothing outperforming the baselines on churn. Ghoshal et al. [8] presented a low-rank adaptive label smoothing, which can summarize label smoothing and be adaptive to the latent structure of label space in structured representations. Rosenfeld et al. [36] presented a unifying view of randomized smoothing over arbitrary functions.
and a new strategy for building classifiers that are robust to general adversarial attacks. Some label smoothing variants appear in co-saliency detection [57], recommender systems [47], and arbitrary-oriented object detection [51].

Label smoothing is not the root cause of the poor performance of knowledge distillation. Lukasik et al. [27] studied how label smoothing relates to loss-correction techniques, and show that when distilling models from noisy data, label smoothing of the teacher is beneficial, which is in contrast to recent studies for noise-free problems. Shen et al. [40] explained that teachers with label smoothing provide less subtle semantic information than without label soothing, which is not the main reason for label smoothing failure but the training dataset appears a long-tailed distribution, and the number of the class is increased.

**Batch Normalization and Its Variants.** Batch normalization [16] solved the problem of internal covariate shift, which leads to an unstable training process and slow convergence speed. Summers et al. [44] presented four tricks to improve batch normalization, especially proposing a method for reasoning about the current example in inference normalization statistics, fixing a training vs. inference discrepancy. Transformer and meta learning also have their batch normalization versions, PowerNorm [39] and TaskNorm [3] respectively. Batch normalization has been demonstrated as an effective strategy to avoid networks collapsing quickly with depth [4]. Li et al. [24] proposed two solutions to alleviate the variance shift between dropout and batch normalization.

### 6.2 Adversarial Training Methods

Due to the linear nature [10] of the DNN, the DNN is vulnerable to the imperceptible perturbation [9,29]. The adversarial training is the most effective defense methods, which are divided into the only adversarial samples trained [49,38] and the trade-off between robustness and accuracy [55,58,56], which have been introduced in Sect. [1]

### 7 Conclusion

In this paper, the mask-guided divergence loss function (MDL) is proposed to enhance the generalization and robustness of the DNN by improving the diversity of the ensemble sub-DNN. The orthogonal term in MDL is responsible for improving diversity. The masking technique in MDL is introduced to prevent the overfitting of diversity learning. Theoretical analysis is conducted to prove that MDL enhances the generalization and robustness of the DNN by constraining the probabilities of false categories to be less than $\frac{1}{M-1}$. Through several experimental evaluations compared with the 9 regularization methods on 4 datasets, the evaluation results show that MDL can improve the generalization of the DNN and enhance the robustness of the DNN significantly under FGSM and PGD attacks. In addition, our MDL has good compatibility with regularization methods and adversarial training methods with the trade-off between robustness and accuracy.
References

1. Arora, R., Bartlett, P., Mianjy, P., Srebro, N.: Dropout: Explicit forms and capacity control. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021. Proceedings of Machine Learning Research, vol. 139, pp. 351–361. PMLR (2021)
2. Bahri, D., Jiang, H.: Locally adaptive label smoothing improves predictive churn. In: Proceedings of the 38th International Conference on Machine Learning, ICML 2021. Proceedings of Machine Learning Research, vol. 139, pp. 532–542. PMLR (2021)
3. Bronskill, J., Gordon, J., Requeima, J., Nowozin, S., Turner, R.E.: Tasknorm: Rethinking batch normalization for meta-learning. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 1153–1164. PMLR (2020)
4. Daneshmand, H., Kohler, J., Bach, F., Hofmann, T., Lucchi, A.: Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In: Advances in Neural Information Processing Systems. vol. 33, pp. 18387–18398. Curran Associates, Inc. (2020)
5. Fan, A., Grave, E., Joulin, A.: Reducing transformer depth on demand with structured dropout. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)
6. Fan, X., Zhang, S., Tanwisuth, K., Qian, X., Zhou, M.: Contextual dropout: An efficient sample-dependent dropout module. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
7. Ghiasi, G., Lin, T., Le, Q.V.: Dropblock: A regularization method for convolutional networks. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018. pp. 10750–10760 (2018)
8. Ghoshal, A., Chen, X., Gupta, S., Zettlemoyer, L., Mehdad, Y.: Learning better structured representations using low-rank adaptive label smoothing. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016. pp. 770–778. IEEE Computer Society (2016)
12. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors (2012)
13. Hou, S., Wang, Z.: Weighted channel dropout for regularization of deep convolutional neural network. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. pp. 8425–8432. AAAI Press (2019)
14. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: Computer Vision - ECCV 2016 - 14th European Conference. Lecture Notes in Computer Science, vol. 9908, pp. 646–661. Springer (2016)
15. Inoue, H.: Multi-sample dropout for accelerated training and better generalization (2019)

16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015. JMLR Workshop and Conference Proceedings, vol. 37, pp. 448–456. JMLR.org (2015)

17. Ke, Z., Wen, Z., Xie, W., Wang, Y., Shen, L.: Group-wise dynamic dropout based on latent semantic variations. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020. pp. 11229–11236. AAAI Press (2020)

18. Keshari, R., Singh, R., Vatsa, M.: Guided dropout. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. pp. 4065–4072. AAAI Press (2019)

19. Kim, W., Cho, M., Sung, Y.: Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019. pp. 6079–6086. AAAI Press (2019)

20. Konda, K.R., Bouthillier, X., Memisevic, R., Vincent, P.: Dropout as data augmentation (2015)

21. Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)

22. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

23. Lee, H., Nam, T., Yang, E., Hwang, S.J.: Meta dropout: Learning to perturb latent features for generalization. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)

24. Li, X., Chen, S., Hu, X., Yang, J.: Understanding the disharmony between dropout and batch normalization by variance shift. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019. pp. 2682–2690. Computer Vision Foundation / IEEE (2019)

25. Lienen, J., Hüllermeier, E.: From label smoothing to label relaxation. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021. pp. 8583–8591. AAAI Press (2021)

26. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell. 42(2), 318–327 (2020)

27. Lukasik, M., Bhojanapalli, S., Menon, A.K., Kumar, S.: Does label smoothing mitigate label noise? In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 6448–6458. PMLR (2020)

28. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research 9(86), 2579–2605 (2008), http://jmlr.org/papers/v9/vandermaaten08a.html

29. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations, ICLR 2018. OpenReview.net (2018)

30. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019. pp. 4696–4705 (2019)

31. Novak, R., Bahri, Y., Abolafia, D.A., Pennington, J., Sohl-Dickstein, J.: Sensitivity and generalization in neural networks: an empirical study. In: 6th International Conference on Learning Representations, ICLR 2018. OpenReview.net (2018)

32. Pal, A., Lane, C., Vidal, R., Haefele, B.D.: On the regularization properties of structured dropout. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020. pp. 7668–7676. IEEE (2020)
33. Papernot, N., McDaniel, P.D., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy, SP 2016. pp. 582–597. IEEE Computer Society (2016)
34. Pham, H., Le, Q.V.: Autodropout: Learning dropout patterns to regularize deep networks. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021. pp. 9351–9359. AAAI Press (2021)
35. Quan, Y., Chen, M., Pang, T., Ji, H.: Self2self with dropout: Learning self-supervised denoising from single image. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020. pp. 1887–1895. IEEE (2020)
36. Rosenfeld, E., Winston, E., Ravikumar, P., Kolter, J.Z.: Certified robustness to label-flipping attacks via randomized smoothing. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 8230–8241. PMLR (2020)
37. S., V.B., Babu, R.V.: Single-step adversarial training with dropout scheduling. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020. pp. 947–956. IEEE (2020)
38. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J.P., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019. pp. 3353–3364 (2019)
39. Shen, S., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Powernorm: Rethinking batch normalization in transformers. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 8741–8751. PMLR (2020)
40. Shen, Z., Liu, Z., Xu, D., Chen, Z., Cheng, K., Savvides, M.: Is label smoothing truly incompatible with knowledge distillation: An empirical study. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
42. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017. pp. 464–472. IEEE Computer Society (2017)
43. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15(1), 1929–1958 (2014)
44. Summers, C., Dinneen, M.J.: Four things everyone should know to improve batch normalization. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)
45. Tang, Y., Wang, Y., Xu, Y., Shi, B., Xu, C., Xu, C., Xu, C.: Beyond dropout: Feature map distortion to regularize deep neural networks. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020. pp. 5964–5971. AAAI Press (2020)
46. Wan, L., Zeiler, M.D., Zhang, S., LeCun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013. JMLR Workshop and Conference Proceedings, vol. 28, pp. 1058–1066. JMLR.org (2013)
47. Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., Wang, Z.: Knowledge-aware graph neural networks with label smoothness regularization for
recommender systems. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019. pp. 968–977. ACM (2019)

48. Wei, C., Kakade, S.M., Ma, T.: The implicit and explicit regularization effects of dropout. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 10181–10192. PMLR (2020)

49. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. In: 8th International Conference on Learning Representations, ICLR 2020. OpenReview.net (2020)

50. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)

51. Yang, X., Yan, J.: Arbitrary-oriented object detection with circular smooth label. In: Computer Vision - ECCV 2020 - 16th European Conference. Lecture Notes in Computer Science, vol. 12353, pp. 677–694. Springer (2020)

52. Zhang, C., Jiang, P., Hou, Q., Wei, Y., Han, Q., Li, Z., Cheng, M.: Delving deep into label smoothing. IEEE Trans. Image Process. 30, 5984–5996 (2021)

53. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization (2016)

54. Zhang, H., Li, S., Ma, Y., Li, M., Xie, Y., Zhang, Q.: Interpreting and boosting dropout from a game-theoretic view. In: 9th International Conference on Learning Representations, ICLR 2021. OpenReview.net (2021)

55. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019. Proceedings of Machine Learning Research, vol. 97, pp. 7472–7482. PMLR (2019)

56. Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., Kankanhalli, M.S.: Attacks which do not kill training make adversarial learning stronger. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020. Proceedings of Machine Learning Research, vol. 119, pp. 11278–11287. PMLR (2020)

57. Zhang, K., Li, T., Liu, B., Liu, Q.: Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019. pp. 3095–3104. Computer Vision Foundation / IEEE (2019)

58. Zheng, H., Zhang, Z., Gu, J., Lee, H., Prakash, A.: Efficient adversarial training with transferable adversarial examples. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020. pp. 1178–1187. Computer Vision Foundation / IEEE (2020)