REGULARIZING PREDICTIONS VIA CLASS-WISE SELF-KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

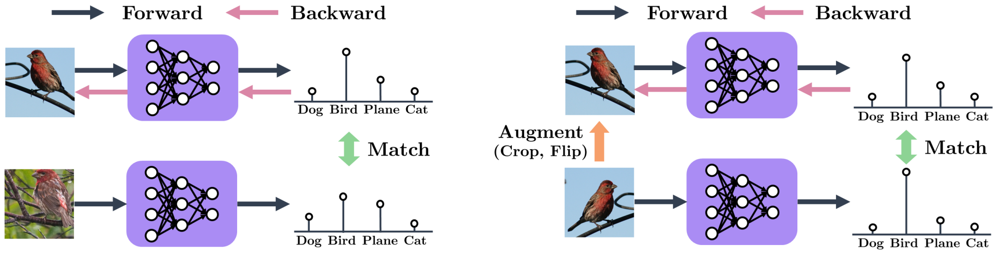
Deep neural networks with millions of parameters may suffer from poor generalizations due to overfitting. To mitigate the issue, we propose a new regularization method that penalizes the predictive distribution between similar samples. In particular, we distill the predictive distribution between different samples of the same label and augmented samples of the same source during training. In other words, we regularize the dark knowledge (i.e., the knowledge on wrong predictions) of a single network, i.e., a self-knowledge distillation technique, to force it output more meaningful predictions. We demonstrate the effectiveness of the proposed method via experiments on various image classification tasks: it improves not only the generalization ability, but also the calibration accuracy of modern neural networks.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved state-of-the-art performance on many machine learning applications, e.g., computer vision (He et al., 2016), natural language processing (Devlin et al., 2019), and reinforcement learning (Silver et al., 2016). As the scale of training dataset increases, the size of DNNs (i.e., the number of parameters) also scales up to handle such a large dataset efficiently. However, networks with millions of parameters may incur overfitting and suffer from poor generalizations (Pereyra et al., 2017; Zhang et al., 2017). To address the issue, many regularization strategies have been investigated in the literature: early stopping, L_1/L_2-regularization (Nowlan & Hinton, 1992), dropout (Srivastava et al., 2014), batch normalization (Sergey Ioffe, 2015) and data augmentation (Cubuk et al., 2019)

Regularizing the predictive or output distribution of DNNs can be effective because it contains the most succinct knowledge of the model. On this line, several strategies such as entropy maximization (Pereyra et al., 2017) and angular-margin based methods (Chen et al., 2018; Zhang et al., 2019) have been proposed in the literature. They can be also influential to solve related problems, e.g., network calibration (Guo et al., 2017), detection of out-of-distribution samples (Lee et al., 2018) and exploration of the agent in reinforcement learning (Haarnoja et al., 2018). In this paper, we focus on developing a new output regularizer for deep models utilizing the concept of *dark knowledge* (Hinton et al., 2015), i.e., the knowledge on wrong predictions made by DNN. Its importance has been first evidenced by the so-called knowledge distillation and investigated in many following works (Romero et al., 2015; Zagoruyko & Komodakis, 2017; Srinivas & Fleuret, 2018; Ahn et al., 2019).

While the related works (Furlanello et al., 2018; Hessam Bagherinezhad & Farhadi, 2018) use the knowledge distillation (KD; Hinton et al. 2015) to transfer the dark knowledge learned by a teacher network to a student network, we regularize the dark knowledge itself during training a single network, i.e., self-knowledge distillation. Specifically, we propose a new regularization technique, coined class-wise self-knowledge distillation (CS-KD) that matches or distills the predictive distribution of DNNs between different samples of the same label (class-wise regularization) and augmented samples of the same source (sample-wise regularization) as shown in Figure 1. One can expect that the proposed regularization method forces DNNs to produce similar wrong predictions if samples are of the same class, while the conventional cross-entropy loss does not consider such consistency on the wrong predictions.



(a) Class-wise regularization

(b) Sample-wise regularization

Figure 1: Illustration of class-wise self-knowledge distillation (CS-KD). We match or distill the output distribution of DNNs between (a) different samples with the same label and (b) augmented samples of the same source.

We demonstrate the effectiveness of our regularization method using deep convolutional neural networks, such as ResNet (He et al., 2016) and DenseNet (Huang et al., 2017) trained for image classification tasks on various datasets including CIFAR-100 (Krizhevsky et al., 2009), TinyImageNet[1], CUB-200-2011 (Wah et al., 2011), Stanford Dogs (Khosla et al., 2011), and MIT67 (Quattoni & Torralba, 2009) datasets. We compare or combine our method with prior regularizers. In our experiments, the top-1 error rates of our method are consistently smaller than those of prior output regularization methods such as angular-margin based methods (Chen et al., 2018; Zhang et al., 2019) and entropy regularization (Dubey et al., 2018; Pereyra et al., 2017). In particular, the gain tends to be larger in overall for the top-5 error rates and the expected calibration errors (Guo et al., 2017), which confirms that our method indeed makes predictive distributions more meaningful. Moreover, we investigate a variant of our method by combining it with other types of regularization method for boosting performance, such as the mixup regularization (Zhang et al., 2018) and the original KD method. We improve the top-1 error rate of mixup from 37.09% to 31.95% and that of KD from 39.32% to 35.36% under ResNet (He et al., 2016) trained by the CUB-200-2011 dataset. Our method is very simple to use, and would enjoy a broader usage in the future.

2 Regularization via self-knowledge distillation

In this section, we introduce a new regularization technique, named class-wise self-knowledge distillation (CS-KD). Throughout this paper, we focus on fully-supervised or classification tasks, and denote $\mathbf{x} \in \mathcal{X}$ as an input and $y \in \mathcal{Y} = \{1, ..., C\}$ as its ground-truth label. Suppose that a softmax classifier is used to model a posterior distribution, i.e., given the input $\mathbf{x}$, the predictive distribution is as follows:

$$P\left(y|\mathbf{x};\theta,T\right) = \frac{\exp\left(f_y\left(\mathbf{x};\theta\right) / \, T\right)}{\sum_{i=1}^{C}\exp\left(f_i\left(\mathbf{x};\theta\right) / \, T\right)},$$

where $f = [f_i]$ denotes the logit-vector of DNN, parameterized by θ and $T > 0$ is the temperature scaling parameter.

2.1 Class-wise regularization

We first consider matching the predictive distributions on samples of the same class, which distills their dark knowledge into the model itself. To this end, we propose a class-wise regularization loss that enforces consistent predictive distributions in the same class. Formally, given input $\mathbf{x}$ and another randomly sampled input $\mathbf{x}'$ having the same label y, it is defined as follows:

$$\mathcal{L}_{\texttt{cls}}(\mathbf{x},\mathbf{x}') := \text{KL}\left(P(y|\mathbf{x}';\widetilde{\theta},T)\big\|P(y|\mathbf{x};\theta,T)\right),$$

where KL denotes the Kullback-Leibler (KL) divergence and $\widetilde{\theta}$ is a fixed copy of the parameters θ. As suggested by (Takeru Miyato & Ishii, 2018), the gradient is not propagated through $\widetilde{\theta}$ to avoid

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

Algorithm 1 Class-wise self-knowledge distillation (CS-KD)

```
Initialize parameters $\theta$.
while $\theta$ has not converged do
    for $(\mathbf{x}, y)$ in a sampled batch do
        $g_\theta \leftarrow 0$
        Get another sample $\mathbf{x}'$ randomly which has the same label $y$ from the training set.
        Generate $\mathbf{x}_{\texttt{aug}}$, ${\mathbf{x}'}_{\texttt{aug}}$ using data augmentation methods.
        Compute gradient: $g_\theta \leftarrow g_\theta + \nabla_\theta \mathcal{L}_{\texttt{tot}}(\mathbf{x}, \mathbf{x}_{\texttt{aug}}, {\mathbf{x}'}_{\texttt{aug}})$
    end for
    Update parameters $\theta$ using gradients $g_\theta$.
end while
```

the model collapsing issue. Similar to the knowledge distillation method (KD) by Hinton et al. (2015), $\mathcal{L}_{\texttt{cls}}$ matches two predictions. While the original KD matches predictions of a sample from two networks, we do predictions of different samples from a single network. Namely, our method performs self-knowledge distillation.

2.2 Sample-wise regularization

In addition to enforcing the intra-class consistency of predictive distributions, we apply this idea to the single-sample scenario by augmenting the input data. For a given training sample $\mathbf{x}$, the proposed sample-wise regularization loss $\mathcal{L}_{\texttt{sam}}$ is defined as follows:

$$\mathcal{L}_{\texttt{sam}}(\mathbf{x}, \mathbf{x}_{\texttt{aug}}) := \mathrm{KL}\left(P(y|\mathbf{x}; \widetilde{\theta}, T) \middle\| P(y|\mathbf{x}_{\texttt{aug}}; \theta, T)\right),$$

where $\mathbf{x}_{\texttt{aug}}$ is an augmented input that is modified by some data augmentation methods, e.g., resizing, rotating, random cropping (Krizhevsky et al., 2009; Simonyan & Zisserman, 2015; Szegedy et al., 2015), cutout (DeVries & Taylor, 2017), and auto-augmentation (Cubuk et al., 2019). In our experiments, we use standard augmentation methods for ImageNet (i.e., flipping and random sized cropping) because they make training more stable.

In summary, the total training loss $\mathcal{L}_{\texttt{tot}}$ is defined as a weighted sum of the two regularization terms with cross-entropy loss as follows:

$$\mathcal{L}_{\texttt{tot}}(\mathbf{x}, \mathbf{x}_{\texttt{aug}}, {\mathbf{x}'}_{\texttt{aug}}) := -y \cdot \log P(y|\mathbf{x}_{\texttt{aug}}; \theta, 1) + \lambda_{\texttt{cls}} \mathcal{L}_{\texttt{cls}}(\mathbf{x}_{\texttt{aug}}, \mathbf{x}'_{\texttt{aug}}) + \lambda_{\texttt{sam}} \mathcal{L}_{\texttt{sam}}(\mathbf{x}, \mathbf{x}_{\texttt{aug}}),$$

where $\lambda_{\texttt{cls}}$ and $\lambda_{\texttt{sam}}$ are balancing weights for each regularization, respectively. Note that the first term is the cross-entropy loss of softmax outputs with temperature $T = 1$. In other words, we not only train the true label, but also regularize the wrong labels. The full training procedure with the proposed loss $\mathcal{L}_{\texttt{tot}}$ is summarized in Algorithm 1.

3 Experiments

3.1 Experimental setup

Datasets. To demonstrate our method under general situations of data diversity, we consider various image classification tasks including conventional classification and fine-grained classification tasks. We use CIFAR-100 (Krizhevsky et al., 2009) and TinyImageNet[2] datasets for conventional classification tasks, and CUB-200-2011 (Wah et al., 2011), Stanford Dogs (Khosla et al., 2011), and MIT67 (Quattoni & Torralba, 2009) datasets for fine-grained classification tasks. Note that fine-grained image classification tasks have visually similar classes and consist of fewer training samples per class compared to conventional classification tasks. We sample 10% of the training dataset randomly as a validation set for CIFAR-100 and TinyImageNet and report the test accuracy based on the validation accuracy. For the fine-grained datasets, we report the best validation accuracy.

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

Table 1: Top-1 error rates (%) on various image classification tasks and model architectures. We reported the mean and standard deviation over 3 runs with different random seed. Boldface values in parentheses indicate relative error rate reductions from cross-entropy.

Dataset	Method	ResNet-18	DenseNet-121
CIFAR-100	Cross-entropy	25.82±0.26	23.54±0.27
	AdaCos	25.72±0.49	24.22±0.34
	Virtual-softmax	24.13±0.12	23.51±0.04
	Maximum-entropy	23.53±0.24	23.02±0.31
	CS-KD (ours)	**22.74±0.14 (-11.9%)**	**22.66±0.24 (- 3.7%)**
TinyImageNet	Cross-entropy	45.16±0.22	40.85±0.24
	AdaCos	44.14±0.41	40.71±0.22
	Virtual-softmax	43.88±0.31	42.92±1.56
	Maximum-entropy	43.56±0.04	40.10±0.58
	CS-KD (ours)	**42.95±0.43 (- 4.9%)**	**39.65±0.58 (- 2.9%)**
CUB-200-2011	Cross-entropy	46.00±1.43	42.30±0.44
	AdaCos	35.47±0.07	30.84±0.38
	Virtual-softmax	35.03±0.51	33.85±0.75
	Maximum-entropy	39.86±1.11	37.51±0.71
	CS-KD (ours)	**33.50±0.31 (-27.2%)**	**30.79±0.36 (-27.2%)**
Stanford Dogs	Cross-entropy	36.29±0.32	33.39±0.17
	AdaCos	32.66±0.34	27.87±0.65
	Virtual-softmax	31.48±0.16	30.55±0.72
	Maximum-entropy	32.41±0.20	29.52±0.74
	CS-KD (ours)	**31.06±0.51 (-14.4%)**	**27.65±0.59 (-17.2%)**
MIT67	Cross-entropy	44.75±0.80	41.79±0.19
	AdaCos	42.66±0.43	40.25±0.68
	Virtual-softmax	42.86±0.71	43.66±0.30
	Maximum-entropy	43.36±1.62	43.48±1.30
	CS-KD (ours)	**40.77±1.05 (- 8.9%)**	**39.75±0.38 (- 4.9%)**

Network architecture. We consider two state-of-the-art convolutional neural network architectures: ResNet (He et al., 2016) and DenseNet (Huang et al., 2017). We use standard ResNet-18 with 64 filters and DenseNet-121 with growth rate of 32 for image size 224 × 224. For CIFAR-100 and TinyImageNet, we modify the first convolutional layer[3] with kernel size 3×3, strides 1 and padding 1, instead of the kernel size 7×7, strides 2 and padding 3, for image size 32×32.

Evaluation metric. For evaluation, we measure the following metrics:

- **Top-1 / 5 error rate.** Top-k error rate is the fraction of test samples for which the correct label is amongst the top-k confidences. We measured top-1 and top-5 error rates to evaluate the generalization performance of the models.
- **Expected Calibration Error (ECE).** ECE (Naeini et al., 2015; Guo et al., 2017) approximates the difference in expectation between confidence and accuracy, by partitioning predictions into M equally-spaced bins and taking a weighted average of bins' difference of confidence and accuracy, i.e., ECE $= \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|$, where n is the number of samples, B_m is the set of samples whose confidence falls into the m-th interval, $\mathrm{acc}(B_m)$ is the accuracy of B_m and $\mathrm{conf}(B_m)$ is the average confidence within B_m. We compare ECE values to evaluate whether the model represents the true likelihood.
- **Recall at *k* (R@*k*).** Recall at k is the percentage of test samples that have at least one example from the same class in k nearest neighbors on the feature space. To measure the distance between two samples, we use L_2-distance between their average-pooled features in the penultimate layer. We compare the recall at 1 scores to evaluate intra-class variations of learned features.

[3] We used a reference implementation: https://github.com/kuangliu/pytorch-cifar.

Table 2: Top-1 error rates (%) of compatibility experiments with mixup regularization on various image classification tasks. We reported the mean and standard deviation over 3 runs with different random seed, and the best results are indicated in bold.

Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Cross-entropy	25.82±0.26	45.16±0.22	46.00±1.43	36.29±0.32	44.75±0.80
CS-KD (ours)	22.74±0.14	42.95±0.43	33.50±0.31	31.06±0.51	40.77±1.05
Mixup	23.28±0.17	43.03±0.37	37.09±0.27	32.54±0.04	41.67±1.05
Mixup + CS-KD (ours)	**21.51±0.44**	**42.73±0.58**	**31.95±0.65**	**29.64±0.28**	**40.17±1.12**

Table 3: Top-1 error rates (%) of compatibility experiments with knowledge distillation (KD) on various image classification tasks. Teacher network is pre-trained on DenseNet-121 (large) by CS-KD, and student network trained on ResNet-10 (small). We reported the mean and standard deviation over 3 runs with different random seed, and the best results are indicated in bold.

Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Cross-entropy	27.93±0.04	48.09±0.54	48.36±0.61	38.96±0.40	44.75±0.62
CS-KD (ours)	26.79±0.22	45.71±0.32	39.12±0.09	34.07±0.46	41.54±0.67
KD	26.77±0.22	44.63±0.10	39.32±0.65	34.23±0.42	38.63±0.11
KD + CS-KD (ours)	**26.38±0.25**	**43.85±0.04**	**35.36±0.26**	**32.08±0.16**	**37.91±0.27**

Hyper-parameters. All networks are trained from scratch and optimized by stochastic gradient descent (SGD) with momentum 0.9, weight decay 0.0001 and an initial learning rate of 0.1. The learning rate is divided by 10 after epochs 100 and 150 for all datasets and total epochs are 200. We set batch size as 128 for conventional, and 32 for fine-grained classification tasks. We use standard flips, random resized crops, 32 for conventional and 224 for fine-grained classification tasks, overall experiments. Furthermore, we set $T = 4$, $\lambda_{\texttt{cls}} = 1$ for all experiments and $\lambda_{\texttt{sam}} = 1$ for experiments on fine-grained classification tasks, and $\lambda_{\texttt{sam}} = 0$ on conventional classification tasks. To compute expected calibration error (ECE), we set the number of bins M as 20.

Baselines. We compare our method with prior regularization methods such as the state-of-the-art angular-margin based methods (Zhang et al., 2019; Chen et al., 2018) and entropy regularization (Dubey et al., 2018; Pereyra et al., 2017). They also regularize predictive distributions as like ours.

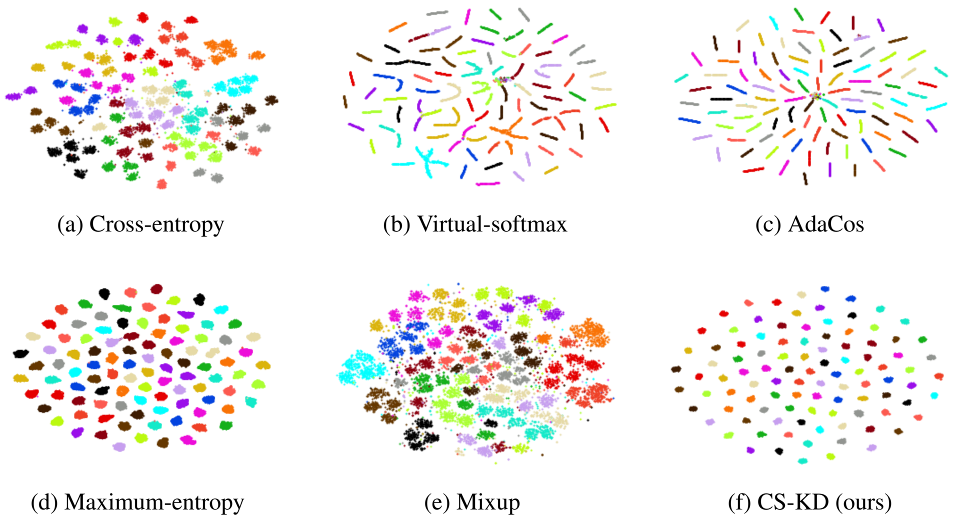
- **AdaCos** (Zhang et al., 2019).[4] AdaCos dynamically scales the cosine similarities between training samples and corresponding class center vectors to maximize angular-margin.
- **Virtual-softmax** (Chen et al., 2018). Virtual-softmax injects an additional virtual class to maximize angular-margin.
- **Maximum-entropy** (Dubey et al., 2018; Pereyra et al., 2017). Maximum-entropy is a typical entropy regularization, which maximizes the entropy of the predictive distribution.

Note that AdaCos and Virtual-softmax regularize the predictive or output distribution of DNN to learn feature representation by reducing intra-class variations and enlarging inter-class margins.

3.2 Experimental results

Comparison with output regularization methods. We measure the top-1 error rates of the proposed method (denoted by CS-KD) by comparing with Virtual-softmax, AdaCos, and Maximum-entropy on various image classification tasks. Table 1 shows that CS-KD outperforms other baselines consistently. In particular, CS-KD improves the top-1 error rate of cross-entropy loss from 46.00% to 33.50% in the CUB-200-2011 dataset, while the top-1 error rates of other baselines are even worse than the cross-entropy loss (e.g., AdaCos in the CIFAR-100, Virtual-softmax in the MIT67, and Maximum-entropy in the TinyImageNet and the MIT67 under DenseNet). The results imply that our method is more effective and stable than other baselines.

[4] We used a reference implementation: https://github.com/4uiiurz1/pytorch-adacos


(a) Cross-entropy

(b) Virtual-softmax

(c) AdaCos

(d) Maximum-entropy

(e) Mixup

(f) CS-KD (ours)

Figure 2: Visualization of features on the penultimate layer using t-SNE, from 10,000 number of randomly chosen training samples of CIFAR-100. Note that 20 superclasses in CIFAR-100 are drawn by 20 different colors. (a) Cross-entropy, (b) Virtual-softmax, (c) AdaCos, (d) Maximum-entropy, (e) Mixup and (f) CS-KD (ours).

Compatibility with other types of regularization methods. We investigate orthogonal usage with other types of regularization methods such as mixup (Zhang et al., 2018) and knowledge distillation (KD). Mixup utilizes convex combinations of input pairs and corresponding label pairs for training. We combine our method with mixup regularization by applying the class-wise regularization loss $\mathcal{L}_{\texttt{cls}}$ to mixed inputs and mixed labels, instead of standard inputs and labels. Table 2 shows the effectiveness of our method combined with mixup regularization. Interestingly, this simple idea significantly improves the performances of fine-grained classification tasks. In particular, our method improves the top-1 error rate of mixup regularization from 37.09% to 31.95%, where the top-1 error rate of cross-entropy loss is 46.00% in the CUB-200-2011.

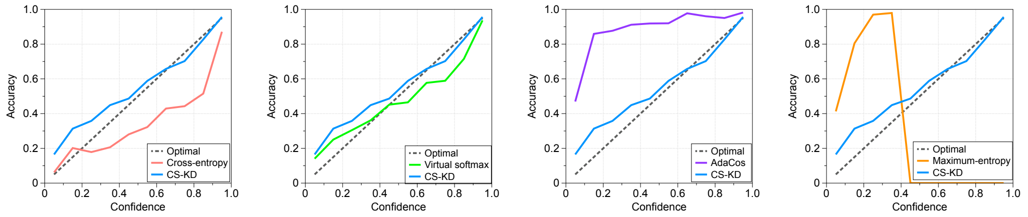
KD regularizes predictive distributions of student network to learn the dark knowledge of a teacher network. We combine our method with KD to learn dark knowledge from the teacher and itself simultaneously. Table 3 shows that the top-1 error rate under using our method solely is close to that of KD, although ours do not use additional teacher networks. Besides, learning knowledge from a teacher network improves the top-1 error rate of our method from 39.32% to 35.36% in the CUB-200-2011 dataset. The results show a wide applicability of our method, compatible to use with other regularization methods.

3.3 Analysis of feature embedding and calibration

One can expect that our method forces DNNs to produce meaningful predictions by reducing the intra-class variations. To verify this, we analyze feature embedding and various evaluation metrics, including the top-1, top-5 error, expected calibration error (Guo et al., 2017) and R@1. In Figure 2, we visualize feature embedding of the penultimate layer from ResNet-18 trained with various regularization techniques by t-SNE (Maaten & Hinton, 2008) in the CIFAR-100 dataset. One can note that intra-class variations are significantly decreased by our method (Figure 2f), while Virtual-softmax (Figure 2b) and AdaCos (Figure 2c) only reduce the angular-margin. We also provide quantitative analysis on the feature embedding by measuring the R@1 values, which are related to intra-class variations. Note that the larger value of R@1 means the more reduced intra-class variations on the feature embedding (Wengang Zhou, 2017). As shown in Table 4, R@1 values can be significantly improved when ResNet-18 is trained with our methods. In particular, R@1 of our method is 59.22% in the CUB-200-2011 dataset, while R@1 of Virtual-softmax and Adacos are 55.56% and 54.86%, respectively. Moreover, Table 4 shows the top-5 error rates of our method significantly outperform other regularization methods. Figure 3 and Table 4 show that our method enhances model calibration significantly, which also confirm that ours forces DNNs to produce more meaningful predictions.

Table 4: Top-1 / 5 error, ECE, and Recall at 1 rates (%) of ResNet-18. The arrow on the right side of the evaluation metric indicates ascending or descending order of the value. We reported the mean and standard deviation over 3 runs with different random seed, and the best results are indicated in bold.

Dataset	Method	Top-1 ↓	Top-5 ↓	ECE ↓	R@1 ↑
CIFAR-100	Cross-entropy	25.82±0.26	7.42±0.29	16.31±0.25	59.42±1.03
	AdaCos	25.72±0.49	10.53±1.10	71.79±0.51	66.26±0.83
	Virtual-softmax	24.13±0.12	8.89±0.26	7.11±0.72	67.40±0.25
	Maximum-entropy	23.53±0.24	7.53±0.14	56.21±0.46	**70.66±0.21**
	CS-KD (ours)	**22.74±0.14**	**5.79±0.13**	**5.05±0.41**	70.04±0.17
TinyImageNet	Cross-entropy	45.16±0.22	22.21±0.29	14.08±0.76	30.59±0.42
	AdaCos	44.14±0.41	22.24±0.11	55.09±0.41	44.66±0.52
	Virtual-softmax	43.88±0.31	24.15±0.17	4.60±0.67	44.69±0.58
	Maximum-entropy	43.56±0.04	21.53±0.50	42.68±0.31	39.18±0.79
	CS-KD (ours)	**42.95±0.43**	**20.22±0.13**	**3.96±0.67**	**44.79±0.26**
CUB-200-2011	Cross-entropy	46.00±1.43	22.30±0.68	18.39±0.76	33.92±1.70
	AdaCos	35.47±0.07	15.24±0.66	63.39±0.06	54.86±0.24
	Virtual-softmax	35.03±0.51	13.16±0.20	11.68±0.66	55.56±0.74
	Maximum-entropy	39.86±1.11	19.80±1.21	50.52±1.20	48.66±2.10
	CS-KD (ours)	**33.50±0.31**	**13.06±0.35**	**5.17±0.33**	**59.22±0.97**
Stanford Dogs	Cross-entropy	36.29±0.32	11.80±0.27	15.05±0.35	47.51±1.02
	AdaCos	32.66±0.34	11.02±0.22	65.38±0.33	58.37±0.43
	Virtual-softmax	31.48±0.16	8.64±0.21	7.91±0.38	59.71±0.56
	Maximum-entropy	32.41±0.20	10.9±0.31	51.53±0.28	60.05±0.45
	CS-KD (ours)	**31.06±0.51**	**8.40±0.10**	**4.36±0.46**	**62.37±0.28**
MIT67	Cross-entropy	44.75±0.80	19.25±0.53	17.99±0.72	31.42±1.00
	AdaCos	42.66±0.43	19.05±2.33	54.00±0.52	42.39±1.91
	Virtual-softmax	42.86±0.71	19.10±0.20	11.21±1.00	44.20±0.90
	Maximum-entropy	43.36±1.62	20.47±0.90	42.41±1.74	38.06±3.32
	CS-KD (ours)	**40.77±1.05**	**17.64±0.16**	**8.12±1.13**	**44.73±2.09**



(a) Cross-entropy (b) Virtual-softmax (c) AdaCos (d) Maximum-entropy

Figure 3: Reliability diagrams (DeGroot & Fienberg, 1983; Niculescu-Mizil & Caruana, 2005) which show accuracy as a function of confidence, for ResNet-18 trianed on CIFAR-100 using (a) Cross-entropy, (b) Virtual-softmax, (c) AdaCos, and (d) Maximum-entropy. All methods are compared with our proposed method, CS-KD.

4 Related work

Regularization techniques. Numerous techniques have been introduced to prevent overfitting of neural networks, including early stopping, weight decay, dropout (Srivastava et al., 2014), and batch normalization (Sergey Ioffe, 2015). Alternatively, regularization methods for the output distribution also have been explored: Szegedy et al. (2016) showed that label-smoothing, which is a mixture of the ground-truth and the uniform distribution, improves generalization of neural networks. Similarly, Pereyra et al. (2017) proposed penalizing low entropy output distributions, which improves exploration in reinforcement learning and supervised learning. Zhang et al. (2018) proposed a powerful data augmentation method called mixup, which works as a regularizer that can be utilized with smaller weight decay. We remark that our method enjoys orthogonal usage with the prior methods, i.e., our methods can be combined with prior methods to further improve the generalization performance.

Knowledge distillation. Knowledge distillation (Hinton et al., 2015) is an effective learning method to transfer the knowledge from a powerful teacher model to a student. This pioneering work showed that one can use softmax with temperature scaling to match soft targets for transferring *dark knowledge*, which contains the information of non-target labels. There are numerous follow-up studies to distill knowledge in the aforementioned teacher-student framework. FitNets (Romero et al., 2015) tried to learn features of a thin deep network using a shallow one with linear transform. Similarly, Zagoruyko & Komodakis (2017) introduced a transfer method that matches attention maps of the intermediate features, and Ahn et al. (2019) tried to maximize the mutual information between intermediate layers of teacher and student for enhanced performance. Srinivas & Fleuret (2018) proposed a loss function for matching Jacobian of the networks output instead of the feature itself. We remark that our method and knowledge distillation (Hinton et al., 2015) have a similar component, i.e., using a soft target distribution, but our method utilizes the soft target distribution from itself. We also remark that joint usage of our method and the prior knowledge distillation methods is effective.

Margin-based softmax losses. There have been recent efforts toward boosting the recognition performances via enlarging inter-class margins and reducing intra-class variation. Several approaches utilized metric-based methods that measure similarities between features using Euclidean distances, such as triplet (Weinberger & Saul, 2009) and contrastive loss (Chopra et al., 2005). To make the model extract discriminative features, center loss (Wen et al., 2016) and range loss (Xiao Zhang & Qiao, 2017) were proposed to minimize distances between samples belong to the same class. COCO loss (Liu et al., 2017b) and NormFace (Feng Wang & Yuille, 2017) optimized cosine similarities, by utilizing reformulations of softmax loss and metric learning with feature normalization. Similarly, Yutong Zheng & Savvides (2018) applied ring loss for soft normalization which uses a convex norm constraint. More recently, angular-margin based losses were proposed for further improvement. L-softmax (Liu et al., 2016) and A-softmax (Liu et al., 2017a) combined angular margin constraints with softmax loss to encourage the model to generate more discriminative features. CosFace (Wang et al., 2018), AM-softmax (Feng Wang & Cheng, 2018) and ArcFace (Deng et al., 2019) introduced angular margins for a similar purpose, by reformulating softmax loss. Different from L-Softmax and A-Softmax, Virtual-softmax (Chen et al., 2018) encourages a large margin among classes via injecting additional virtual negative class.

5 Conclusion

In this paper, we discover a simple regularization method to enhance generalization performance of deep neural networks. We propose two regularization terms which penalizes the predictive distribution between different samples of the same label and augmented samples of the same source by minimizing the Kullback-Leibler divergence. We remark that our ideas regularize the dark knowledge (i.e., the knowledge on wrong predictions) itself and encourage the model to produce more meaningful predictions. Moreover, we demonstrate that our proposed method can be useful for the generalization and calibration of neural networks. We think that the proposed regularization techniques would enjoy a broader range of applications, e.g., deep reinforcement learning (Haarnoja et al., 2018) and detection of out-of-distribution samples (Lee et al., 2018).

References

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, 2019.

Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *NeurIPS*, 2018.

Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019.

Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine grained classification. In *NeurIPS*, 2018.

Haijun Liu Feng Wang, Weiyang Liu and Jian Cheng. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

Jian Cheng Feng Wang, Xiang Xiang and Alan L. Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 2017.

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Mohammad Rastegari Hessam Bagherinezhad, Maxwell Horton and Ali Farhadi. Label refinery: Improving imagenet classification through label progression. In *ECCV*, 2018.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR*, 2011.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017a.

Yu Liu, Hongyang Li, and Xiaogang Wang. Learning deep features via congenerous cosine loss for person recognition. *arXiv preprint arXiv:1702.06890*, 2017b.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *ICML*, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Masanori Koyama Takeru Miyato, Shin-ichi Maeda and Shin Ishii. Virtual adversarial training : A regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

Qi Tian Wengang Zhou, Houqiang Li. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017.

Yandong Wen Zhifeng Li Xiao Zhang, Zhiyuan Fang and Yu Qiao. Range loss for deep face recognition with long-tail. In *ICCV*, 2017.

Dipan K. Pal Yutong Zheng and Marios Savvides. Ring loss: Convex feature normalization for face recognition. In *CVPR*, 2018.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *CVPR*, 2019.