Regularizing Class-wise Predictions via Self-knowledge Distillation

Sukmin Yun[1*] Jongjin Park[1*] Kimin Lee[2†] Jinwoo Shin[1]
[1]Korea Advanced Institute of Science and Technology, South Korea
[2]University of California, Berkeley, USA
{sukmin.yun, jongjin.park, jinwoos}@kaist.ac.kr kiminlee@berkeley.edu

Abstract

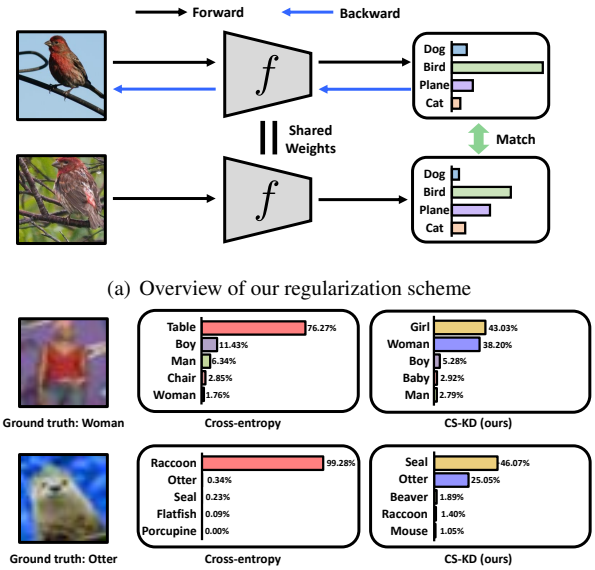
Deep neural networks with millions of parameters may suffer from poor generalization due to overfitting. To mitigate the issue, we propose a new regularization method that penalizes the predictive distribution between similar samples. In particular, we distill the predictive distribution between different samples of the same label during training. This results in regularizing the dark knowledge (i.e., the knowledge on wrong predictions) of a single network (i.e., a self-knowledge distillation) by forcing it to produce more meaningful and consistent predictions in a class-wise manner. Consequently, it mitigates overconfident predictions and reduces intra-class variations. Our experimental results on various image classification tasks demonstrate that the simple yet powerful method can significantly improve not only the generalization ability but also the calibration performance of modern convolutional neural networks.

1. Introduction

Deep neural networks (DNNs) have achieved state-of-the-art performance on many computer vision tasks, *e.g.*, image classification [19], generation [4], and segmentation [18]. 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 [36, 55]. To address the issue, many regularization strategies have been investigated in the literature: early stopping [3], L_1/L_2-regularization [35], dropout [42], batch normalization [40] and data augmentation [8].

Regularizing the predictive distribution of DNNs can be effective because it contains the most succinct knowledge of the model. On this line, several strategies such as label-smoothing [32, 43], entropy maximization [13, 36], and angular-margin based methods [5, 58] have been proposed in the literature. They were also influential in solving related problems such as network calibration [16], novelty detection [27], and exploration in reinforcement learning [17]. In this paper, we focus on developing a new output regularizer for deep models utilizing the concept of *dark knowledge* [22], *i.e.*, the knowledge on wrong predictions made by DNNs. Its importance has been first evidenced by the so-called knowledge distillation (KD) [22] and investigated in many following works [1, 39, 41, 54].

While the related works [15, 21] use the knowledge distillation 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 [53, 57]. 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 as shown in Figure 1(a). One can ex-

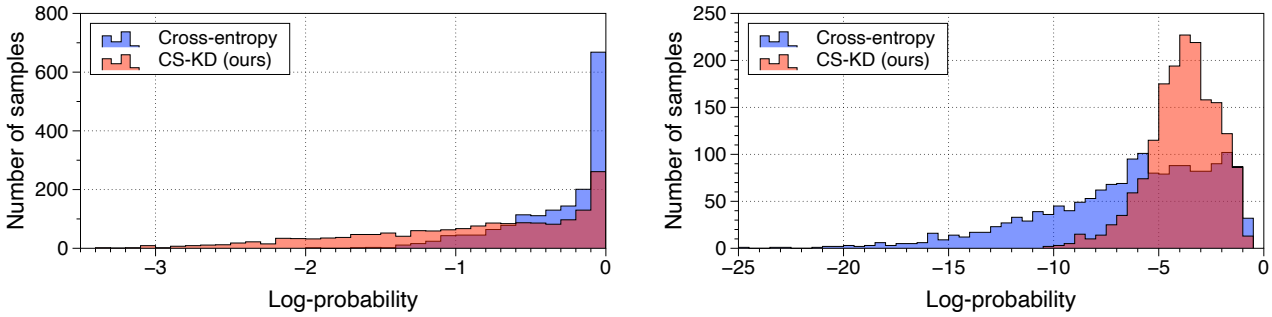


(a) Overview of our regularization scheme

(b) Top-5 softmax scores on misclassified samples

Figure 1. (a) Illustration of class-wise self-knowledge distillation (CS-KD). (b) Predictive distributions on misclassified samples. We use PreAct ResNet-18 trained on CIFAR-100 dataset. For misclassified samples, softmax scores of the ground-truth class are increased by training DNNs with class-wise regularization.

*Equal contribution
†Work was done while the author was at KAIST



(a) Log-probabilities of predicted labels on misclassified samples

(b) Log-probabilities of ground-truth labels on misclassified samples

Figure 2. Histogram of log-probabilities of (a) the predicted label, *i.e.*, top-1 softmax score, and (b) the ground-truth label on misclassified samples by networks trained by the cross-entropy (baseline) and CS-KD. The networks are trained on PreAct ResNet-18 for CIFAR-100.

pect 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 predictive distributions. Furthermore, it could achieve two desirable goals simultaneously: preventing overconfident predictions and reducing the intra-class variations. We remark that they have been investigated in the literature via different methods, *i.e.*, entropy regularization [13, 32, 36, 43] and margin-based methods [5, 58], respectively, while we achieve both using a single principle.

We demonstrate the effectiveness of our simple yet powerful regularization method using deep convolutional neural networks, such as ResNet [19] and DenseNet [23] trained for image classification tasks on various datasets including CIFAR-100 [26], TinyImageNet[1], CUB-200-2011 [46], Stanford Dogs [25], MIT67 [38], and ImageNet [10]. In our experiments, the top-1 error rates of our method are consistently lower than those of prior output regularization methods such as angular-margin based methods [5, 58] and entropy regularization [13, 32, 36, 43]. In particular, the gain tends to be larger in overall for the top-5 error rates and the expected calibration errors [16], which confirms that our method indeed makes predictive distributions more meaningful. We also found the top-1 error rates of our method are lower than those of the recent self-distillation methods [53, 57] in overall. Moreover, we investigate variants of our method by combining it with other types of regularization methods for boosting performance, such as the Mixup regularization [56] and the original KD method [22]. For example, we improve the top-1 error rate of Mixup from 37.09% to 30.71%, and that of KD from 39.32% to 34.47% using the CUB-200-2011 dataset under ResNet-18 and ResNet-10, respectively.

We remark that the idea of using a consistency regularizer like ours has been investigated in the literature [2, 7, 24, 31, 37, 44, 53]. While most prior methods proposed to regularize the output distributions of original and perturbed inputs to be similar, our method forces the consistency between different samples having the same class. To the best of our knowledge, no work is known to study such a class-wise regularization. We believe that the proposed method may be influential to enjoy a broader usage in other applications, *e.g.*, face recognition [11, 58], and image retrieval [45].

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

Algorithm 1 Class-wise self-knowledge distillation

Initialize parameters θ.
while θ has not converged **do**
  Sample a batch $(\mathbf{x}, y)$ from the training dataset.
  Sample another batch $\mathbf{x}'$ randomly, which has the same label y from the training dataset.
  Update parameters θ by computing the gradients of the proposed loss function $\mathcal{L}_{\texttt{CS-KD}}(\mathbf{x}, \mathbf{x}', y; \theta, T)$ in (1).
end while

2. Class-wise 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 classification tasks and denote $\mathbf{x} \in \mathcal{X}$ as input and $y \in \mathcal{Y} = \{1, ..., C\}$ as its ground-truth label. Suppose that a softmax classifier is used to model a posterior predictive distribution, *i.e.*, given the input $\mathbf{x}$, the predictive distribution is:

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

where f_i denotes the logit of DNNs for class i which are parameterized by θ, and $T > 0$ is the temperature scaling parameter.

2.1. Class-wise regularization

We consider matching the predictive distributions on samples of the same class, which distills their dark knowledge from the model itself. To this end, we propose a class-wise regularization loss that enforces consistent predictive

distributions in the same class. Formally, given an input $\mathbf{x}$ and another randomly sampled input $\mathbf{x}'$ having the same label y, it is defined as follows:

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

where KL denotes the Kullback-Leibler (KL) divergence, and $\widetilde{\theta}$ is a fixed copy of the parameters θ. As suggested by Miyato *et al*. [31], the gradient is not propagated through $\widetilde{\theta}$ to avoid the model collapse issue. Similar to the original knowledge distillation method (KD; [22]), the proposed loss $\mathcal{L}_{\texttt{cls}}$ matches two predictions. While the original KD matches predictions of a single sample from two networks, we make predictions of different samples from a single network, *i.e.*, self-knowledge distillation. Namely, the total training loss $\mathcal{L}_{\texttt{CS-KD}}$ is defined as follows:

$$\mathcal{L}_{\texttt{CS-KD}}(\mathbf{x}, \mathbf{x}', y; \theta, T) := \mathcal{L}_{\texttt{CE}}(\mathbf{x}, y; \theta) + \lambda_{\texttt{cls}} \cdot T^2 \cdot \mathcal{L}_{\texttt{cls}}(\mathbf{x}, \mathbf{x}'; \theta, T), \quad (1)$$

where $\mathcal{L}_{\texttt{CE}}$ is the standard cross-entropy loss, and $\lambda_{\texttt{cls}} > 0$ is a loss weight for the class-wise regularization. Note that we multiply the square of the temperature T^2 by following the original KD [22]. The full training procedure with the proposed loss $\mathcal{L}_{\texttt{CS-KD}}$ is summarized in Algorithm 1.

2.2. Effects of class-wise regularization

The proposed CS-KD is arguably the simplest way to achieve two goals, preventing overconfident predictions and reducing the intra-class variations, via a single mechanism. To avoid overconfident predictions, it utilizes the model-prediction of other samples as the soft-label. It is more 'realistic' than the label-smoothing method [32, 43], which generates 'artificial' soft-labels. Besides, ours directly minimizes the distance between two logits within the same class, and it would reduce intra-class variations.

We also examined whether the proposed method forces DNNs to produce meaningful predictions. To this end, we investigate prediction values in softmax scores, *i.e.*, $P(y|\mathbf{x})$, from PreAct ResNet-18 [20] trained on the CIFAR-100 dataset [26] using the standard cross-entropy loss and the proposed CS-KD loss. Specifically, we analyze the predictions of two concrete misclassified samples in the CIFAR-100 dataset. As shown in Figure 1(b), CS-KD not only relaxes the overconfident predictions but also enhances the prediction values of classes correlated to the ground-truth class. This implies that CS-KD induces meaningful predictions by forcing DNNs to produce similar predictions on similar inputs. To evaluate the prediction quality, we also report log-probabilities of the softmax scores on the predicted class and ground-truth class on samples that are commonly misclassified by both the cross-entropy and our method. As shown in Figure 2(a), our method produces less confident predictions on misclassified samples compared to the cross-entropy method. Interestingly, our method increases ground-truth scores for misclassified samples, as reported in Figure 2(b). In our experiments, we found that the classification accuracy and calibration effects can be improved by forcing DNNs to produce such meaningful predictions (see Section 3.2 and 3.4).

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.[2] Specifically, we use CIFAR-100 [26] and TinyImageNet[3] datasets for conventional classification tasks, and CUB-200-2011 [46], Stanford Dogs [25], and MIT67 [38] datasets for fine-grained classification tasks. The fine-grained image classification tasks have visually similar classes and consist of fewer training samples per class compared to conventional classification tasks. ImageNet [10] is used for a large-scale classification task.

Network architecture. We consider two state-of-the-art convolutional neural network architectures: ResNet [19] and DenseNet [23]. We use standard ResNet-18 with 64 filters and DenseNet-121 with a growth rate of 32 for image size 224×224. For CIFAR-100 and TinyImageNet, we use PreAct ResNet-18 [20], which modifies the first convolutional layer[4] 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 by following [56]. We use DenseNet-BC structure [23], and the first convolution layer of the network is also modified in the same way as in PreAct ResNet-18 for image size 32×32.

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 the standard data augmentation technique for ImageNet [10], *i.e.*, flipping and random cropping. For our method, the temperature T is chosen from $\{1, 4\}$, and the loss weight $\lambda_{\texttt{cls}}$ is chosen from $\{1, 2, 3, 4\}$. The optimal parameters are chosen to minimize the top-1 error rates on the validation set. More detailed ablation studies on the hyper-parameters T and $\lambda_{\texttt{cls}}$ are provided in the supplementary material.

[2] Code is available at https://github.com/alinlab/cs-kd.
[3] https://tiny-imagenet.herokuapp.com/
[4] We used a reference implementation: https://github.com/kuangliu/pytorch-cifar.

Model	Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
ResNet-18	Cross-entropy	24.71$_{\pm 0.24}$	43.53$_{\pm 0.19}$	46.00$_{\pm 1.43}$	36.29$_{\pm 0.32}$	44.75$_{\pm 0.80}$
	AdaCos	23.71$_{\pm 0.36}$	42.61$_{\pm 0.20}$	35.47$_{\pm 0.07}$	32.66$_{\pm 0.34}$	42.66$_{\pm 0.43}$
	Virtual-softmax	23.01$_{\pm 0.42}$	42.41$_{\pm 0.20}$	35.03$_{\pm 0.51}$	31.48$_{\pm 0.16}$	42.86$_{\pm 0.71}$
	Maximum-entropy	22.72$_{\pm 0.29}$	41.77$_{\pm 0.13}$	39.86$_{\pm 1.11}$	32.41$_{\pm 0.20}$	43.36$_{\pm 1.62}$
	Label-smoothing	22.69$_{\pm 0.28}$	43.09$_{\pm 0.34}$	42.99$_{\pm 0.99}$	35.30$_{\pm 0.66}$	44.40$_{\pm 0.71}$
	CS-KD (ours)	**21.99**$_{\pm 0.13}$ **(-11.0%)**	**41.62**$_{\pm 0.38}$ **(- 4.4%)**	**33.28**$_{\pm 0.99}$ **(-27.7%)**	**30.85**$_{\pm 0.28}$ **(-15.0%)**	**40.45**$_{\pm 0.45}$ **(- 9.6%)**
DeseNet-121	Cross-entropy	22.23$_{\pm 0.04}$	39.22$_{\pm 0.27}$	42.30$_{\pm 0.44}$	33.39$_{\pm 0.17}$	41.79$_{\pm 0.19}$
	AdaCos	22.17$_{\pm 0.24}$	38.76$_{\pm 0.23}$	30.84$_{\pm 0.38}$	27.87$_{\pm 0.65}$	40.25$_{\pm 0.68}$
	Virtual-softmax	23.66$_{\pm 0.10}$	41.58$_{\pm 1.58}$	33.85$_{\pm 0.75}$	30.55$_{\pm 0.72}$	43.66$_{\pm 0.30}$
	Maximum-entropy	22.87$_{\pm 0.45}$	38.39$_{\pm 0.33}$	37.51$_{\pm 0.71}$	29.52$_{\pm 0.74}$	43.48$_{\pm 1.30}$
	Label-smoothing	21.88$_{\pm 0.45}$	38.75$_{\pm 0.18}$	40.63$_{\pm 0.24}$	31.39$_{\pm 0.46}$	42.24$_{\pm 1.23}$
	CS-KD (ours)	**21.69**$_{\pm 0.49}$ **(- 2.4%)**	**37.96**$_{\pm 0.09}$ **(- 3.2%)**	**30.83**$_{\pm 0.39}$ **(-27.1%)**	**27.81**$_{\pm 0.13}$ **(-16.7%)**	**40.02**$_{\pm 0.91}$ **(- 4.2%)**

Table 1. Top-1 error rates (%) on various image classification tasks and model architectures. We report the mean and standard deviation over three runs with different random seeds. Values in parentheses indicate relative error rate reductions from the cross-entropy, and the best results are indicated in bold.

Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Cross-entropy	24.71$_{\pm 0.24}$	43.53$_{\pm 0.19}$	46.00$_{\pm 1.43}$	36.29$_{\pm 0.32}$	44.75$_{\pm 0.80}$
DDGSD	23.85$_{\pm 1.57}$	**41.48**$_{\pm 0.12}$	41.17$_{\pm 1.28}$	31.53$_{\pm 0.54}$	41.17$_{\pm 2.46}$
BYOT	23.81$_{\pm 0.11}$	44.02$_{\pm 0.57}$	40.76$_{\pm 0.39}$	34.02$_{\pm 0.14}$	44.88$_{\pm 0.46}$
CS-KD (ours)	**21.99**$_{\pm 0.13}$ **(-11.0%)**	41.62$_{\pm 0.38}$ (- 4.4%)	**33.28**$_{\pm 0.99}$ **(-27.7%)**	**30.85**$_{\pm 0.28}$ **(-15.0%)**	**40.45**$_{\pm 0.45}$ **(- 9.6%)**

Table 2. Top-1 error rates (%) of ResNet-18 with self-distillation methods on various image classification tasks. We report the mean and standard deviation over three runs with different random seeds. Values in parentheses indicate relative error rate reductions from the cross-entropy, and the best results are indicated in bold. The self-distillation methods are re-implemented under our code-base.

Baselines. We compare our method with prior regularization methods such as the state-of-the-art angular-margin based methods [5, 58], entropy regularization [13, 32, 36, 43] and self-distillation methods [53, 57]. They also regularize predictive distributions like ours.

- **AdaCos** [58].[5] AdaCos dynamically scales the cosine similarities between training samples and corresponding class center vectors to maximize angular-margin.
- **Virtual-softmax** [5]. Virtual-softmax injects an additional virtual class to maximize angular-margin.
- **Maximum-entropy** [13, 36]. Maximum-entropy is a typical entropy regularization, which maximizes the entropy of the predictive distribution.
- **Label-smoothing** [32, 43]. Label-smoothing uses soft labels that are a weighted average of the one-hot labels and the uniform distribution.
- **DDGSD** [53]. Data-distortion guided self-distillation (DDGSD) is one of the consistency regularization techniques, which forces the consistent outputs across different augmented versions of the data.
- **BYOT** [57]. Be Your Own Teacher (BYOT) transfers the knowledge in the deeper portion of the networks into the shallow ones.

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

- **Top-1 / 5 error rate.** The top-k error rate is the fraction of test samples for which the correct label is not in the top-k confidences. We measure top-1 and top-5 error rates to evaluate the generalization performances.
- **Expected Calibration Error (ECE).** ECE [16, 33] approximates the difference in expectation between confidence and accuracy. It is calculated 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, and $\mathrm{acc}(B_m)$, $\mathrm{conf}(B_m)$ are the accuracy and the average confidence of B_m, respectively. We measure ECE with 20 bins to evaluate whether the model represents the true correctness likelihood.
- **Recall at k (R@k).** Recall at k is the percentage of test samples that have at least one 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 pooled features of the penultimate layer. We compare the recall at $k = 1$ scores to evaluate intra-class variations of learned features.

3.2. Classification accuracy

Comparison with output regularization methods. We measure the top-1 error rates of the proposed method (de-

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

Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Cross-entropy	24.71$_{\pm 0.24}$	43.53$_{\pm 0.19}$	46.00$_{\pm 1.43}$	36.29$_{\pm 0.32}$	44.75$_{\pm 0.80}$
CS-KD (ours)	21.99$_{\pm 0.13}$	41.62$_{\pm 0.38}$	33.28$_{\pm 0.99}$	30.85$_{\pm 0.28}$	40.45$_{\pm 0.45}$
Mixup	21.67$_{\pm 0.34}$	41.57$_{\pm 0.38}$	37.09$_{\pm 0.27}$	32.54$_{\pm 0.04}$	41.67$_{\pm 1.05}$
Mixup + CS-KD (ours)	**20.40**$_{\pm \mathbf{0.31}}$	**40.71**$_{\pm \mathbf{0.32}}$	**30.71**$_{\pm \mathbf{0.64}}$	**29.93**$_{\pm \mathbf{0.14}}$	**39.65**$_{\pm \mathbf{0.85}}$

Table 3. Top-1 error rates (%) of ResNet-18 with Mixup regularization on various image classification tasks. We report the mean and standard deviation over three runs with different random seeds, and the best results are indicated in bold.

Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Cross-entropy	26.72$_{\pm 0.33}$	46.61$_{\pm 0.22}$	48.36$_{\pm 0.61}$	38.96$_{\pm 0.40}$	44.75$_{\pm 0.62}$
CS-KD (ours)	25.80$_{\pm 0.10}$	44.67$_{\pm 0.12}$	39.12$_{\pm 0.09}$	34.07$_{\pm 0.46}$	41.54$_{\pm 0.67}$
KD	25.84$_{\pm 0.07}$	43.31$_{\pm 0.11}$	39.32$_{\pm 0.65}$	34.23$_{\pm 0.42}$	41.47$_{\pm 0.79}$
KD + CS-KD (ours)	**25.58**$_{\pm \mathbf{0.16}}$	**42.82**$_{\pm \mathbf{0.33}}$	**34.47**$_{\pm \mathbf{0.17}}$	**32.59**$_{\pm \mathbf{0.50}}$	**40.27**$_{\pm \mathbf{0.78}}$

Table 4. Top-1 error rates (%) of ResNet-10 (student) with knowledge distillation (KD) on various image classification tasks. Teacher networks are pre-trained on DenseNet-121 by CS-KD. We report the mean and standard deviation over three runs with different random seeds, and the best results are indicated in bold.

noted by CS-KD) by comparing with Virtual-softmax, AdaCos, Maximum-entropy, and Label-smoothing 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 the cross-entropy loss from 46.00% to 33.28% under the CUB-200-2011 dataset. We also observe that the top-1 error rates of other baselines are often worse than the cross-entropy loss, *e.g.*, Virtual-softmax, Maximum-entropy, and Label-smoothing under MIT67 and DenseNet). As shown in Table 6, top-5 error rates of CS-KD outperform other regularization methods, as it encourages meaningful predictions. In particular, CS-KD improves the top-5 error rate of the cross-entropy loss from 6.91% to 5.69% under the CIFAR-100 dataset, while the top-5 error rates of AdaCos is even worse than the cross-entropy loss. These results imply that our method induces better predictive distributions than other baseline methods.

Comparison with self-distillation methods. We also compare our method with recent proposed self-distillation techniques such as DDGSD [53] and BYOT [57]. As shown in Table 2, CS-KD shows better top-1 error rates on ResNet-18 in overall. For example, CS-KD shows the top-1 error rate of 33.28% on the CUB-200-2011 dataset, while DDGSD and BYOT have 41.17% and 40.76%, respectively. All tested self-distillation methods utilize regularization effects of knowledge distillation. The superiority of CS-KD could be explained by its unique effect of reducing intra-class variations.

Model	Method	Top-1 (1-crop)
ResNet-50	Cross-entropy	24.0
	CS-KD (ours)	**23.6**
ResNet-101	Cross-entropy	22.4
	CS-KD (ours)	**22.0**
ResNeXt-101-32x4d	Cross-entropy	21.6
	CS-KD (ours)	**21.2**

Table 5. Top-1 error rates (%) on ImageNet dataset with various model architectures trained for 90 epochs with batch size 256. The best results are indicated in bold.

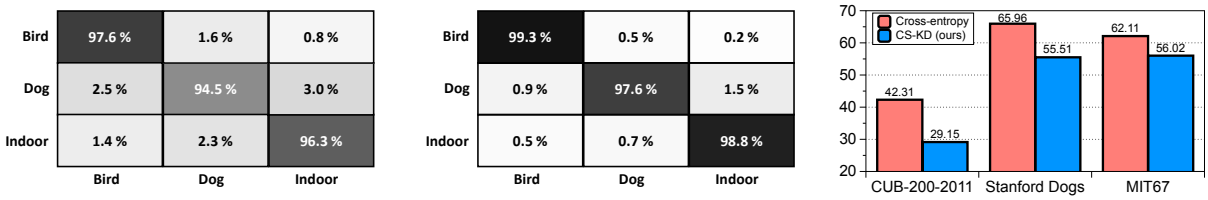
Evaluation on large-scale datasets. To verify the scalability of our method, we have evaluated our method on the ImageNet dataset with various model architecture such as ResNet-50, ResNet-101, and ResNeXt-101-32x4d [52]. As reported in Table 5, our method improves 0.4% of the top-1 error rates across all the tested architectures consistently. The 0.4% improvement is comparable to, *e.g.*, adding 51 more layers on ResNet-101 (*i.e.*, ResNet-152) [19].

Compatibility with other regularization methods. We investigate orthogonal usage with other types of regularization methods such as Mixup [56] and knowledge distillation (KD) [22]. 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 3 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 30.71%, where the top-1 error rate of the cross-entropy loss is 46.00% under ResNet-18 on the CUB-200-2011 dataset.

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 4 shows that our method achieves a similar performance of KD, although

(a) Cross-entropy (b) Virtual-softmax (c) AdaCos (d) CS-KD (ours)

Figure 3. Visualization of various feature embeddings on the penultimate layer using t-SNE on PreAct ResNet-18 for CIFAR-100. The proposed method (d) shows the smallest intra-class variation that leads to the best top-1 error rate.



(a) Cross-entropy (b) CS-KD (ours) (c) Top-1 error rates (%)

Figure 4. Experimental results of ResNet-18 on the mixed dataset. The hierarchical classification accuracy (%) of each model trained by (a) the cross-entropy and (b) our method. One can observe that the model trained by CS-KD is less confusing classes across different domains. (c) Top-1 error rates (%) of fine-grained label classification.

ours does not use additional teacher networks. Besides, combining our method with KD further improves the top-1 error rate of our method from 39.32% to 34.47%, where the top-1 error rate of the cross-entropy loss is 48.36% under ResNet-10 trained on the CUB-200-2011 dataset. These results show the wide applicability of our method, compatible to use with other regularization methods.

3.3. Ablation study

Feature embedding analysis. One can expect that the intra-class variations can be reduced by forcing DNNs to produce meaningful predictions. To verify this, we analyze the feature embedding of the penultimate layer of ResNet-18 trained on CIFAR-100 dataset by t-SNE [30] visualization method. As shown in Figure 3, the intra-class variations are significantly decreased by our method (Figure 3(d)) compared to other baselines, including Virtual-softmax (Figure 3(b)) and AdaCos (Figure 3(c)), which are designed to reduce intra-class variations. We also provide quantitative results on the metric Recall at 1 (R@1), which has appeared in Section 3.1. We remark that the larger value of R@1 implies small intra-class variations on the feature embedding [50]. As shown in Table 6, R@1 values can be significantly improved when ResNet-18 is trained by our method. In particular, R@1 of CS-KD is 47.15% under the TinyImageNet dataset, while R@1 of Adacos, Virtual-softamx, and the cross-entropy loss are 44.66%, 44.69%, and 30.59%, respectively.

Hierarchical image classification. By producing more semantic predictions, *i.e.*, increasing the correlation between similar classes in predictions, we expect the trained classifier can capture a hierarchical (or clustering) structure of label space. To verify this, we evaluate the proposed method on a mixed dataset with 387 fine-grained labels and three hierarchy labels, *i.e.*, bird (CUB-200-2011; 200 labels), dog (Stanford Dogs; 120 labels), and indoor (MIT67; 67 labels). Specifically, we randomly choose 30 samples per each fine-grained label for training, and original test datasets are used for the test. For evaluation, we train ResNet-18 to classify the fine-grained labels and measure a hierarchical classification accuracy by predicting a hierarchy label (bird, dog, or indoor) as that of predicted fine-grained label.

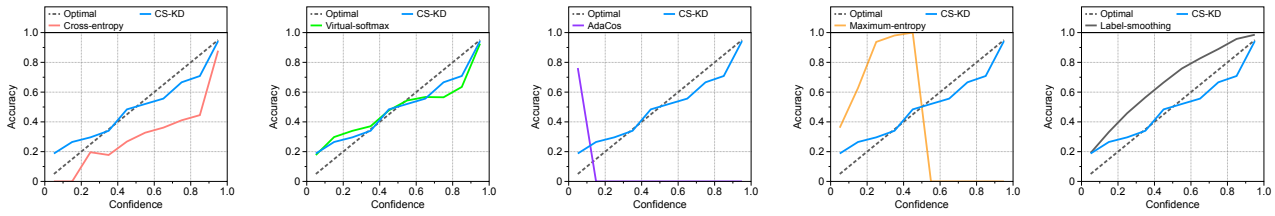
First, we extract the hierarchical structure as confusion matrices, where each element indicates the hierarchical image classification accuracy. As shown in Figure 4(a) and 4(b), our method captures the hierarchical structure of the mixed dataset almost perfectly, *i.e.*, showing the identity confusion matrix. In particular, our method enhances the hierarchical image classification accuracy significantly up to 99.3% in the bird hierarchy (CUB-200-2011). Moreover, as shown in Figure 4(c), our method also improves the top-1 error rates of fine-grained label classification significantly. Interestingly, the error rate of CUB-200-2011 is even lower than the errors reported in Table 1. This is because the model learns additional information by utilizing the dark knowledge of more labels.

3.4. Calibration effects

In this section, we also evaluate the calibration effects of the proposed regularization method. Specifically, we provide reliability diagrams [9, 34], which plot the expected sample accuracy as a function of confidence of PreAct ResNet-18 for the CIFAR-100 dataset in Figure 5. We remark that the plotted identity function (dashed diagonal)

Measurement	Method	CIFAR-100	TinyImageNet	CUB-200-2011	Stanford Dogs	MIT67
Top-5 ↓	Cross-entropy	6.91$_{\pm 0.09}$	22.21$_{\pm 0.29}$	22.30$_{\pm 0.68}$	11.80$_{\pm 0.27}$	19.25$_{\pm 0.53}$
	AdaCos	9.99$_{\pm 0.20}$	22.24$_{\pm 0.11}$	15.24$_{\pm 0.66}$	11.02$_{\pm 0.22}$	19.05$_{\pm 2.33}$
	Virtual-softmax	8.54$_{\pm 0.11}$	24.15$_{\pm 0.17}$	13.16$_{\pm 0.20}$	8.64$_{\pm 0.21}$	19.10$_{\pm 0.20}$
	Maximum-entropy	7.29$_{\pm 0.12}$	21.53$_{\pm 0.50}$	19.80$_{\pm 1.21}$	10.90$_{\pm 0.31}$	20.47$_{\pm 0.90}$
	Label-smoothing	7.18$_{\pm 0.08}$	20.74$_{\pm 0.31}$	22.40$_{\pm 0.85}$	13.41$_{\pm 0.40}$	19.53$_{\pm 0.75}$
	CS-KD (ours)	**5.69**$_{\pm \mathbf{0.03}}$	19.21$_{\pm 0.04}$	**13.07**$_{\pm \mathbf{0.26}}$	**8.55**$_{\pm \mathbf{0.07}}$	**17.46**$_{\pm \mathbf{0.38}}$
	CS-KD-E (ours)	5.93$_{\pm 0.06}$	**19.12**$_{\pm \mathbf{0.34}}$	13.74$_{\pm 0.91}$	8.57$_{\pm 0.13}$	18.21$_{\pm 0.45}$
ECE ↓	Cross-entropy	15.45$_{\pm 0.33}$	14.08$_{\pm 0.76}$	18.39$_{\pm 0.76}$	15.05$_{\pm 0.35}$	17.99$_{\pm 0.72}$
	AdaCos	73.76$_{\pm 0.35}$	55.09$_{\pm 0.41}$	63.39$_{\pm 0.06}$	65.38$_{\pm 0.33}$	54.00$_{\pm 0.52}$
	Virtual-softmax	8.02$_{\pm 0.55}$	4.60$_{\pm 0.67}$	11.68$_{\pm 0.66}$	7.91$_{\pm 0.38}$	11.21$_{\pm 1.00}$
	Maximum-entropy	56.41$_{\pm 0.36}$	42.68$_{\pm 0.31}$	50.52$_{\pm 1.20}$	51.53$_{\pm 0.28}$	42.41$_{\pm 1.74}$
	Label-smoothing	13.20$_{\pm 0.60}$	**2.67**$_{\pm \mathbf{0.48}}$	15.70$_{\pm 0.81}$	11.60$_{\pm 0.40}$	8.79$_{\pm 2.47}$
	CS-KD (ours)	5.17$_{\pm 0.40}$	7.26$_{\pm 0.93}$	15.44$_{\pm 0.92}$	10.46$_{\pm 1.08}$	15.56$_{\pm 0.29}$
	CS-KD-E (ours)	**4.69**$_{\pm \mathbf{0.56}}$	3.79$_{\pm 0.35}$	**8.75**$_{\pm \mathbf{0.49}}$	**4.70**$_{\pm \mathbf{0.18}}$	**8.06**$_{\pm \mathbf{1.90}}$
R@1 ↑	Cross-entropy	61.38$_{\pm 0.64}$	30.59$_{\pm 0.42}$	33.92$_{\pm 1.70}$	47.51$_{\pm 1.02}$	31.42$_{\pm 1.00}$
	AdaCos	67.95$_{\pm 0.42}$	44.66$_{\pm 0.52}$	54.86$_{\pm 0.24}$	58.37$_{\pm 0.43}$	42.39$_{\pm 1.91}$
	Virtual-softmax	68.35$_{\pm 0.48}$	44.69$_{\pm 0.58}$	55.56$_{\pm 0.74}$	59.71$_{\pm 0.56}$	44.20$_{\pm 0.90}$
	Maximum-entropy	**71.51**$_{\pm \mathbf{0.29}}$	39.18$_{\pm 0.79}$	48.66$_{\pm 2.10}$	60.05$_{\pm 0.45}$	38.06$_{\pm 3.32}$
	Label-smoothing	71.44$_{\pm 0.03}$	34.79$_{\pm 0.67}$	41.59$_{\pm 0.94}$	54.48$_{\pm 0.68}$	35.15$_{\pm 1.54}$
	CS-KD (ours)	71.15$_{\pm 0.15}$	**47.15**$_{\pm \mathbf{0.40}}$	**59.06**$_{\pm \mathbf{0.38}}$	**62.67**$_{\pm \mathbf{0.07}}$	**46.74**$_{\pm \mathbf{1.48}}$
	CS-KD-E (ours)	70.57$_{\pm 0.57}$	45.52$_{\pm 0.35}$	58.44$_{\pm 1.09}$	62.03$_{\pm 0.30}$	44.82$_{\pm 1.22}$

Table 6. Top-5 error, ECE, and Recall at 1 (R@1) rates (%) of ResNet-18 on various image classification tasks. We denote our method combined with the sample-wise regularization by CS-KD-E. 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 three runs with different random seeds, and the best results are indicated in bold.



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

Figure 5. Reliability diagrams [9, 34] show accuracy as a function of confidence, for PreAct ResNet-18 trained on CIFAR-100 using (a) Cross-entropy, (b) Virtual-softmax, (c) AdaCos, (d) Maximum-entropy, and (e) Label-smoothing. All methods are compared with our proposed method, CS-KD. Perfect calibration [16] is plotted by dashed diagonals (Optimal) for all.

implies perfect calibration [16], and our method is the closest one among the baselines, as shown in Figure 5. Moreover, we evaluate our method by ECE [16, 33], which is a quantitative metric of calibration, in Table 6. The results demonstrate that our method outperforms the cross-entropy loss consistently. In particular, CS-KD enhances ECE of the cross-entropy from 15.45% to 5.17% under the CIFAR-100 dataset, while AdaCos and Maximum-entropy are significantly worse than the cross-entropy with 73.76% and 56.41%, respectively.

As a natural extension of CS-KD, we also consider combining our method with an existing consistency loss [2, 7, 31, 37, 44], which regularizes the output distributions of a given sample and its augmented one. Specifically, for a given training sample $\mathbf{x}$ and another sample $\mathbf{x}'$ having the same label, the combined regularization loss $\mathcal{L}_{\texttt{CS-KD-E}}$ is defined as follows:

$$\mathcal{L}_{\texttt{CS-KD-E}}(\mathbf{x}, \mathbf{x}', y; \theta, T) := \mathcal{L}_{\texttt{CS-KD}}(\mathbf{x}_{\texttt{aug}}, \mathbf{x}'_{\texttt{aug}}, y; \theta, T) + \lambda_{\texttt{E}} \cdot T^2 \cdot \mathrm{KL}\left(P(y|\mathbf{x}; \widetilde{\theta}, T) \middle\| P(y|\mathbf{x}_{\texttt{aug}}; \theta, T)\right),$$

where $\mathbf{x}_{\texttt{aug}}$ is an augmented sample that is generated by the data augmentation technique[6], and $\lambda_{\texttt{E}} > 0$ is the loss weight for balancing. The corresponding results are reported in

[6]We use standard data augmentation techniques (*i.e.*, flipping and random sized cropping) for all tested methods in this paper.

Table 6, denoted by CS-KD-E. We found that CS-KD-E significantly enhances the calibration performance of CS-KD, and also outperforms the baseline methods over top-1 and top-5 error rates consistently. In particular, CS-KD-E enhances ECE of CS-KD from 5.17% to 4.69% under the CIFAR-100 dataset. We think that investigating the effect of such combined regularization could be an interesting direction to explore in the future, *e.g.*, utilizing other augmentation methods such as cutout [12] and auto-augmentation [8].

4. Related work

Regularization techniques. Numerous techniques have been introduced to prevent overfitting of neural networks, including early stopping [3], L_1/L_2-regularization [35], dropout [42], and batch normalization [40]. Alternatively, regularization methods for the predictive distribution also have been explored: Szegedy *et al*. [43] proposed label-smoothing, which is a mixture of the ground-truth and the uniform distribution, and Zhang *et al*. [56] proposed a data augmentation method called Mixup, which linearly interpolates a random pair of training samples and corresponding labels. Müller *et al*. [32] investigated a method called Label-smoothing and empirically showed that it improves not only generalization but also model calibration in various tasks, such as image classification and machine translation. Similarly, Pereyra *et al*. [36] proposed penalizing low entropy predictive distribution, which improved exploration in reinforcement learning and supervised learning. Moreover, several works [2, 7, 37, 44] investigated consistency regularizers between the predictive distributions of corrupted samples and original samples for semi-supervised learning. We remark that our method enjoys orthogonal usages with the prior methods, *i.e.*, our method can be combined with the prior methods to further improve the generalization performance.

Knowledge distillation. Knowledge distillation [22] 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. Recently, some of the self-distillation approaches [53, 57], which distill knowledge itself, are proposed. Data-distortion guided self-distillation method [53] transfers knowledge between different augmented versions of the same training data. Be Your Own Teacher [57], on the other hand, utilizes ensembling predictions from multiple branches to improve its performance. We remark that our method and these knowledge distillation methods have a similar component, *i.e.*, using a soft target distribution, but ours only reduces intra-class variations. We also remark that the joint usage of our method and the prior knowledge distillation methods is also possible.

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 [48] and contrastive loss [6]. To make the model extract discriminative features, center loss [49] and range loss [51] were proposed to minimize distances between samples belong to the same class. Recently, angular-margin based losses were proposed for further improvement. L-softmax [29] and A-softmax [28] combined angular-margin constraints with softmax loss to encourage the model to generate more discriminative features. CosFace [47], AM-softmax [14], and ArcFace [11] introduced angular-margins for a similar purpose, by reformulating softmax loss. Different from L-Softmax and A-Softmax, Virtual-softmax [5] 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 the generalization performance of deep neural networks. We propose the regularization term, which penalizes the predictive distribution between different samples of the same label by minimizing the Kullback-Leibler divergence. We remark that our idea regularizes the dark knowledge (*i.e.*, the knowledge on wrong predictions) itself and encourages 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 technique would enjoy a broader range of applications, such as exploration in deep reinforcement learning [17], transfer learning [1], face verification [11], and detection of out-of-distribution samples [27].

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00563, Research on Adaptive Machine Learning Technology Development for Intelligent Autonomous Digital Companion) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)). We also thank Sungsoo Ahn and Hankook Lee for helpful discussions.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, 2019. 1, 8

[2] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NeurIPS*, 2014. 2, 7, 8

[3] Christopher Bishop. Regularization and complexity control in feed-forward networks. In *ICANN*, 1995. 1, 8

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 1

[5] Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. In *NeurIPS*, 2018. 1, 2, 4, 8

[6] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 8

[7] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. In *EMNLP*, 2018. 2, 7, 8

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019. 1, 8

[9] 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. 6, 7

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 3

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2, 8

[12] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 8

[13] Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Maximum-entropy fine grained classification. In *NeurIPS*, 2018. 1, 2, 4

[14] Haijun Liu Feng Wang, Weiyang Liu and Jian Cheng. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018. 8

[15] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018. 1

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 1, 2, 4, 7

[17] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018. 1, 8

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3, 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 3

[21] Mohammad Rastegari Hessam Bagherinezhad, Maxwell Horton and Ali Farhadi. Label refinery: Improving imagenet classification through label progression. In *ECCV*, 2018. 1

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2, 3, 5, 8

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2, 3

[24] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018. 2

[25] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR*, 2011. 2, 3

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2, 3

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018. 1, 8

[28] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 8

[29] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 8

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 6

[31] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, 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. 2, 3, 7

[32] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *NeurIPS*, 2019. 1, 2, 3, 4, 8

[33] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015. 4, 7

[34] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005. 6, 7

[35] Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight-sharing. *Neural computation*, 4(4):473–493, 1992. 1, 8

[36] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR workshops*, 2017. 1, 2, 4, 8

[37] Eduard Hovy Minh-Thang Luong Qizhe Xie, Zihang Dai and Quoc V. Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 2, 7, 8

[38] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 2, 3

[39] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 1

[40] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1, 8

[41] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *ICML*, 2018. 1

[42] 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. 1, 8

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1, 2, 3, 4, 8

[44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 2, 7, 8

[45] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016. 2

[46] 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. 2, 3

[47] 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. 8

[48] 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. 8

[49] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 8

[50] Qi Tian Wengang Zhou, Houqiang Li. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017. 6

[51] Yandong Wen Zhifeng Li Xiao Zhang, Zhiyuan Fang and Yu Qiao. Range loss for deep face recognition with long-tail. In *ICCV*, 2017. 8

[52] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 5

[53] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *AAAI*, 2019. 1, 2, 4, 5, 8

[54] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 1

[55] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 5, 8

[57] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *ICCV*, 2019. 1, 2, 4, 5, 8

[58] 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. 1, 2, 4