This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Online Knowledge Distillation via Collaborative Learning

Qiushan Guo[1], Xinjiang Wang[2], Yichao Wu[2], Zhipeng Yu[2], Ding Liang[2], Xiaolin Hu[3], Ping Luo[4]
[1]Beijing University of Posts and Telecommunications [2]SenseTime Group Limited
[3]Tsinghua University [4]The University of Hong Kong
qsguo@bupt.edu.cn xlhu@mail.tsinghua.edu.cn pluo@cs.hku.hk
{wangxinjiang,wuyichao,yuzhipeng,liangding}@sensetime.com

Abstract

*This work presents an efficient yet effective online Knowledge Distillation method via Collaborative Learning, termed KDCL, which is able to consistently improve the generalization ability of deep neural networks (DNNs) that have different learning capacities. Unlike existing two-stage knowledge distillation approaches that pre-train a DNN with large capacity as the "teacher" and then transfer the teacher's knowledge to another "student" DNN unidirectionally (*i.e. *one-way), KDCL treats all DNNs as "students" and collaboratively trains them in a single stage (knowledge is transferred among arbitrary students during collaborative training), enabling parallel computing, fast computations, and appealing generalization ability. Specifically, we carefully design multiple methods to generate soft target as supervisions by effectively ensembling predictions of students and distorting the input images. Extensive experiments show that KDCL consistently improves all the "students" on different datasets, including CIFAR-100 and ImageNet. For example, when trained together by using KDCL, ResNet-50 and MobileNetV2 achieve 78.2% and 74.0% top-1 accuracy on ImageNet, outperforming the original results by 1.4% and 2.0% respectively. We also verify that models pre-trained with KDCL transfer well to object detection and semantic segmentation on MS COCO dataset. For instance, the FPN detector is improved by 0.9% mAP.*

1. Introduction

Knowledge distillation [10] is typically formulated as "teacher-student" learning setting. It is able to improve performance of a compact 'student' deep neural network because the representation of a 'teacher' network can be used as structured knowledge to guide the training of student. The predictions (*e.g.* soft target) produced by the teacher can be easily learned by a student and encourage it to generalize better than that trained from scratch. However, in traditional offline knowledge distillation framework, the teacher is pre-trained first and then fixed, meaning that the knowledge can be only transferred from the teacher to the student (*i.e.* one-way) as shown in Fig. 1a.

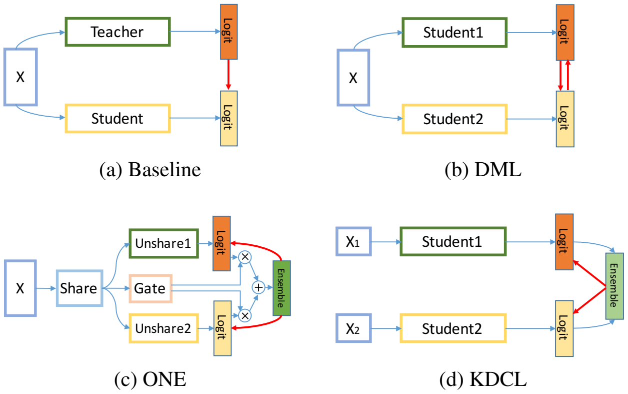

Figure 1: (a) [10] transfers knowledge from the static pre-trained teacher to student model. (b) Students can learn from each other in [32]. (c) [15] establishes teacher using multiple branch design. The gate is to ensemble all the branches. (d) KDCL consistently gains extra information from ensembling soft target produced by all students, outperforming existing approaches. The input of each model is randomly distorted separately to increase its generalization ability. When model pair is trained with KDCL on ImageNet, ResNet-18 is improved by 1.9% and ResNet-50 gets 1.0% improvement due to the knowledge from ResNet-18.

The online distillation methods [32, 15] are more attractive because the training process is simplified to a single stage, and all the networks are treated as students. These approaches merge the training processes of all student networks, enabling them to gain extra knowledge from each other. Students directly learn from the prediction of other students in Deep Mutual Learning (DML) [32], as illustrated in Fig. 1b. However, the output of students can be diverse, conflicting with each other and even the ground truth. When the performances are significantly different among models, this method does harm to the model of high performance.

An alternative method proposed by [15] (ONE) is to train a multi-branch network while establishing teacher on the fly, as shown in Fig. 1c. Nonetheless, this method is inflexible because the network is compelled to share lower layers and knowledge transfer occurs only at the upper layers within a single model rather than other models, limiting the extra knowledge and the performance. The gate module is not a guarantee of high quality soft target.

Self-distillation [6] shows that distilling a converged teacher model into a student model of identical network architecture can further improve the generalization ability compared to the teacher. The efficacy of self-distillation and online distillation leads us to the following question: *Could we use a small network to improve the model with larger capacity in a one-stage distillation framework?*

In this work, we propose a novel online knowledge distillation method via collaborative learning. In KDCL, student networks with different capacities learn collaboratively to generate high-quality soft target supervision, which distills the additional knowledge to each student as illustrated in Fig.1d. The high-quality soft target supervision aims at instructing students with significant performance gaps to consistently converge with higher generalization ability and less variance to the input perturbation in the data domain.

The major challenge is to generate soft target supervision that can boost the performance of all students with high confidence, which have different learning capacities or significant performance gaps. Ensembling tends to yield better results when diversity presents among the outputs of models [14]. Therefore, we propose to generate high-quality soft target supervision by carefully ensembling the output of students with the information of ground truth in an online manner. Furthermore, we propose to estimate generalization error by measuring model on the validation set. The soft target is generated for stronger generalization ability on the validation set.

For improving invariance against perturbations in the input data domain, the soft target should encourage the students to output similarly with similar distorted input images. Therefore, students are fed with the images, which are individually perturbed from identical inputs, and the soft target is generated by combining the outputs and fusing the information of data augmentation. In this case, the benefits of model ensembling are further exploited.

To evaluate the effect of KDCL, we conduct extensive experiments on benchmarks for image classification, CIFAR-100 [13] and ImageNet-2012 [4]. We demonstrate that, with KDCL, ResNet-50 [8] and ResNet-18 trained in pair achieve 77.8% and 73.1% val accuracy. ResNet-18 outperforms the baseline by 1.9% and ResNet-50 gains 1.0% improvement as a benefit of the extra knowledge from ResNet-18. We also verify that models pre-trained with KDCL transfer well to object detection and semantic segmentation on the COCO dataset [17].

Our contributions are listed as follows.

- A new pipeline of knowledge distillation based on collaborative learning is designed. Models of various learning capacity can benefit from collaborative training.
- A series of model ensembling methods are designed to dynamically generate high-quality soft targets in a one-stage online knowledge distillation framework.
- Invariance against perturbations in the input domain is enhanced by transferring knowledge and fusing the output of images with different distortion.

2. Related work

Knowledge transfer for the neural network is advocated by [2, 10] to distill the knowledge from teacher to student. An obvious way is to let the student imitate the output of the teacher model. [2] proposes to improve shallow networks by penalizing the difference of logits between the student and the teacher. [10] realizes knowledge distillation by minimizing the Kullback-Leibler (KL) divergence loss of their output categorical probability.

Structure knowledge Based on the pioneering work, many methods have been proposed to excavate more information from the teacher. [20] introduces more supervision by further exploiting the feature of intermediate hidden layers. [31] defines additional attention information combined with distillation. [18] mines mutual relations of data examples by distance-wise and angle-wise losses. [23] establishes an equivalence between Jacobian matching and distillation. [9] transfers more accurate information via the route to the decision boundary. A few recent papers about self-distillation [29, 3, 6, 28] have shown that a converged teacher model supervising a student model of identical architecture could improve the generalization ability over the teacher. In contrast to mimicking complex models, KDCL involves all networks in learning and provides hint via fusing the information of the students. Without any additional loss for intermediate layers, KDCL reduces the difficulty of optimizing model.

Collaborative learning In online distillation framework, students imitate the teacher in the training process. DML [32] suggests peer students learn from each other through the cross-entropy loss between each pair of students. Co-distillation [1] is similar to DML, whereas it forces student networks to maintain their diversity longer by adding distillation loss after updating enough steps. Inspired by self-distillation, training a multiple branch variant of the target network is proposed to establish a strong teacher on-the-fly. ONE [15] constructs multiple branch classifiers and trains a gate controller to align the teacher's prediction. CLNN [22] promotes the diversity of each

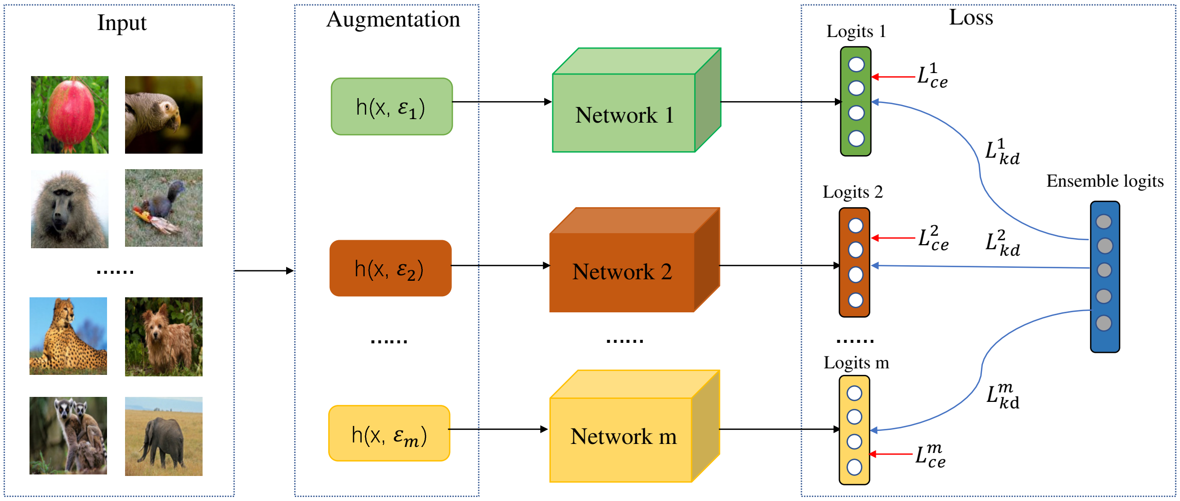

Figure 2: Overview of knowledge distillation via collaborative learning (KDCL). We input images distorted separately for each network to increase the invariance against perturbations in the data domain. KDCL dynamically ensembles soft target produced by all students to improve students consistently. $h(x, \epsilon)$ means random distortion and ϵ is the random seed.

branch by a hierarchical multiple branch design and proposes to scale the gradient accordingly. Different from generating a soft target by averaging the logits, which the above methods adopt, KDCL is proposed to generate a soft target dynamically to improve all students, even with significant performance gaps. The soft target is also designed to improve invariance in the input data domain.

Ensemble method Our work is also related to network ensemble approaches. Majority Voting, Stacked Generalization [27], GASEN [33], and Super Learner [12] explicitly select reliable prediction. Dropout [24], DropConnection [26] and Stochastic Depth [11] generally create exponent numbers of with shared weights during training and then average them at test time. Our method focuses on fusing information of students and guaranteeing the quality of soft target to improve the generalization ability of students.

3. Collaborative Learning for Knowledge Distillation

3.1. Background

Knowledge distillation is to optimize the student network under the supervision of the teacher network. More precisely, the loss is the KL divergence of the soften output of teacher network and student network as [10] defines

$$L_{KD} = \frac{1}{n} \sum_{i=1}^{n} T^2 KL(\mathbf{p}_i, \mathbf{q}_i), \tag{1}$$

where n is the batch size, T is the temperature parameter, $\mathbf{p}$ and $\mathbf{q}$ represent the soften probability distribution produced by teacher network and student network. We note the logit of student and teacher as $\mathbf{z}_s$ and $\mathbf{z}_t$. Then $\mathbf{q} = softmax(\mathbf{z}_s/T)$ and the soft target $\mathbf{p} = softmax(\mathbf{z}_t/T)$.

Teacher Model	Teacher Top-1	Student Top-1
ResNet-34	73.5	70.8
ResNet-50	76.5	71.2
ResNet-101	77.9	71.4

Table 1: Top-1 accuracy on ImageNet-2012 validation set. The second column is the pre-trained teacher model's performance, and the third one is student model accuracy trained with KD loss. The student gets 70.1% accuracy supervised by the hard target.

A high-quality teacher is important for optimizing a good student. If the teacher is not well optimized and provides noisy supervision, the risk that soft target and ground truth conflict with each other becomes high. We evaluate the impact of teacher quality using ResNet-18 [8] as the student model on ImageNet dataset. All the teacher models and student models are trained for 100 epochs. In Tab. 1, the performance of the same student network is compared under the supervision of different teacher models. When the teacher size is not too large for the student, teacher's performance increases, thus it provides better supervision for the student by being a better predictor.

3.2. Our method

Overview. We propose KDCL to automatically generate a soft target in an online manner, as illustrated in Fig. 2. The framework can be viewed as a super network composed of multiple individual sub-networks. The raw images are augmented separately with different random seeds, and the soft target is generated to supervise all networks. We propose a series of methods to generate a soft target, which ensures that students with different capacity benefit from collaborative learning and enhances the invariance of the network against input perturbations. Note that all the models can predict independently, so the improvement does not incur additional test computational cost.

Loss function To improve the generalization performance, we distill the knowledge of soft target to each sub-network via the KD loss. All the sub-networks are trained from scratch. With the standard cross-entropy loss, all the networks are trained end-to-end with a multi-task loss function:

$$L = \sum_{i=1}^{m} L_{CE}^{i} + \lambda L_{KD}^{i}, \tag{2}$$

where L_{KD} is the KL divergence between the output of students and the soft target and λ is the trade-off weight.

KDCL-Naive. In our framework, all the models are student models and the supervision is generated by combining the output of the models. Assuming that there are m sub-networks, the logit of the k-th sub-network is defined as $\mathbf{z}_k$. The teacher logit $\mathbf{z}_t$ is expressed as

$$\mathbf{z}_t = h(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_m), \tag{3}$$

where h is a function to produce higher quality logit compared with the logits of students. Assuming that training samples and test samples follow the same distribution, a model predicting with less loss on training set encourages students to converge faster.

A naive combination method is to choose the logit with the smallest cross-entropy loss among all students, which can be defined as

$$\mathbf{z}_t = \mathbf{z}_k, k = \arg\min_{i} L_{CE}(\mathbf{z}_i, \mathbf{y}), \tag{4}$$

where $\mathbf{y}$ is the one-hot label and L_{CE} is the standard cross-entropy loss.

KDCL-Linear. The naive combination is easy to implement, but the teacher logit is not quality enough. KDCL-Linear defines the teacher logit as the best linear combination of sub-network logits, which is a simple but useful information fusion. Finding out the best linear combination can be treated as an optimization problem. Let matrix $\mathbf{Z} = (\mathbf{z}_1^T; \mathbf{z}_2^T; ...; \mathbf{z}_m^T)$. Each column of the matrix $\mathbf{Z}$ represents the logit of a student. The problem can be illustrated as follow:

$$\min_{\alpha \in \mathbb{R}^m} L_{CE}(\alpha^T \mathbf{Z}, \mathbf{y}), \text{subject to} \sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0 \tag{5}$$

Eq. 5 is a convex optimization problem and is easy to solve.

KDCL-MinLogit. The KDCL-Linear incurs another optimization problem during training, whereas we hope network ensemble is efficient. As an alternative, we propose the KDCL-MinLogit method to generate the soft target. The difference between values in logit decides the probability distribution produced by the $softmax$ function. Therefore the output probability is expressed as

$$\mathbf{p} = softmax(\mathbf{z}) = softmax(\mathbf{z} - z^c), \tag{6}$$

where z^c is the element corresponding to target class c in logit. Define $\mathbf{z}^c = \mathbf{z} - z^c$, then the c-th element of $\mathbf{z}^c$ is 0 for all sub-networks. When the other elements in logit become smaller, the cross-entropy loss with the one-hot label will decrease. Then a neat way to generate teacher logit is to select the minimum element of each row of matrix $\mathbf{Z}^c$, which can be defined as $\mathbf{Z}^c = (\mathbf{z}_1^c, \mathbf{z}_2^c, ..., \mathbf{z}_m^c)$. More precisely, the teacher logit can be expressed as

$$z_{t,j} = \min\{Z_{j,i}^c | i = 1, 2, ..., m\}, \tag{7}$$

where $z_{t,j}$ is the j-th element of soft target $\mathbf{z}_t$ and $Z_{j,i}^c$ is the element of the j-th row and i-th column in $\mathbf{Z}^c$. This method is compatible with mainstream deep learning frameworks.

KDCL-General. The teacher with more generalization ability usually instructs the students to converge better. Performance on the validation dataset can be viewed as a measure of generalization ability. Therefore, we propose to find an optimal ensemble of the m component networks to approximate the general teacher. We randomly pick N examples from the training set to construct the validation set $\mathcal{D}_v$ and the predictions of the component networks are combined through weighted average. The weight should satisfy $w_i \in [0, 1]$ $(i = 1, 2, ..., m)$ and $\sum_{i=1}^{m} w_i = 1$. In this setup, we focus on measuring the generalization ability directly, so the probability is discussed rather than the logit. The generalization error on the input $\mathbf{x}$ is defined as

$$E(\mathbf{x}) = (f(\mathbf{x}) - t)^2, \tag{8}$$

where $f(\mathbf{x})$ is the prediction probability of target class and t is the ground truth. The generalization error of the ensemble network can be expressed as

$$\begin{aligned} E &= \int \left(\sum_{i=1}^{m} w_i f_i(\mathbf{x}) - t \right)^2 p(\mathbf{x}) d\mathbf{x} \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} w_i w_j C_{ij}, \end{aligned} \tag{9}$$

where $p(\mathbf{x})$ is the data distribution and C_{ij} is expressed as

$$\begin{aligned} C_{ij} &= \int (f_i(\mathbf{x}) - t)(f_j(\mathbf{x}) - t) p(\mathbf{x}) d\mathbf{x} \\ &\approx \frac{1}{N} \sum_{k=1}^{N} (f_i(\mathbf{x}_k) - t)(f_j(\mathbf{x}_k) - t) \end{aligned} \tag{10}$$

In general, the distribution of data is unknown or intractable. Hence the empirical distribution is adapted as an approximation. According to Eq. 9 and the constraint on weight, the optimum weight $\mathbf{w}$ can be solved by the Lagrange multiplier as follow:

$$w_k = \frac{\sum_{j=1}^{m} C_{kj}^{-1}}{\sum_{i=1}^{m} \sum_{j=1}^{m} C_{ij}^{-1}}, \tag{11}$$

where w_k is the k-th element of the optimum weight $\mathbf{w}$ and C_{ij}^{-1} is the value of the i-th row and the j-th column of the inverse matrix of $\mathbf{C}$.

Measuring the generalization error incurs the little computational cost and updating the parameters of the neural network a few steps does not change the output drastically due to the small learning rate. Consequently, we update the optimal weight vector $\mathbf{w}$ after each training epoch. Moreover, without any prior knowledge, we assume all the component networks have equal weight $\frac{1}{m}$ in initialization and the soft target is the weighted average probability distribution.

Invariant collaborative learning. In order to improve invariance against the perturbations in the data domain, we generate identical soft target for all students fed with similar distorted images. Therefore, we randomly sample images with the same data augmentation policy for each subnetwork and fuse the knowledge with the above ensemble methods. This method increases the amount and diversity of training data. With the knowledge of the additional training data, the soft target encourages the sub-networks to have a lower generalization error.

4. Experiments

In this section, we perform a series of experiments to evaluate our training mechanism on image classification benchmarks and conduct transfer experiments on COCO dataset [17], a widely used benchmark in the field of object detection and segmentation.

4.1. Results on ImageNet

In experiments on ImageNet, we analyze the effectiveness of our method of generating soft target based on the student pair, ResNet-50 and ResNet-18, and evaluate a series of network architectures on ImageNet.

Dataset and training details. ImageNet dataset contains 1000 object classes with about 1.28 million images for training and 50,000 images for validation. We separate 20,000 images from the train set, 20 samples on each class, as the validation set to measure the generalization ability of sub-network for KDCL-General. So the original validation set is treated as the test set.

We follow the ResNet training procedure. The learning rate starts at 0.1 and warms up to 0.8 linearly after 5 epochs. We set the weight decay to 0.0001, the batch size to 2048, and the momentum is 0.9. All the ResNet models are trained for 200 epochs and the learning rate drop by 0.1 at 60, 120, and 180 epoch. As a default, the MobileNetV2 [21] models are optimized for 300 epochs by stochastic gradient descent (SGD) with a warm-up learning rate to 0.8 and decay it by 0.1 at 90, 180 and 270 epoch. We apply the scale and aspect ration augmentation and the photometric distortions [25]. During test time, the images are scaled to 256 × 256 followed by a 224 × 224 center crop.

Method	ResNet-50	ResNet-18	Gain
Vanilla	76.8	71.2	0
KD[10]	76.8	72.1	0.9
DML[32]	75.8	71.7	-0.5
ONE[15]	-	72.2	-
CLNN[22]	-	72.4	-
KDCL-Naive	77.5	72.9	2.4
KDCL-Linear	**77.8**	**73.1**	**2.9**
KDCL-MinLogit	**77.8**	**73.1**	**2.9**
KDCL-General	77.1	72.0	1.1

Table 2: Top-1 accuracy rate (%) on ImageNet. All the models are reimplemented with our training procedure for a fair comparison. Gain indicates the sum of the component student network improvement. ONE and CLNN are incompatible with different network structures. Therefore, only the accuracy of ResNet-18 is compared.

Method	Top-1	Top-5	Params
Vanilla	71.2	90.0	11.7M
ONE[15]	72.2	90.6	29.5M
KDCL MobileNetV2x1.2	**72.9**	**90.8**	**16.4M**
CLNN[22]	72.4	90.7	40.5M
KDCL ResNet-50	**73.1**	**91.2**	**37.2M**

Table 3: Top-1 and Top-5 accuracy rate (%) on ImageNet. The backbone is ResNet-18. ONE is trained with 3 branches (Res4 block) and CLNN has a hierarchical design with 4 heads. For KDCL, ResNet-18 is trained with a peer network.

Quantitative comparison. Tab. 2 shows top-1 accuracy rate on ImageNet. KD indicates Knowledge Distillation [10]. DML represents the Deep Mutual Learning [32]. ONE [15] and CLNN [22] are self-distillation methods, which are incompatible with different network structures. Therefore, only the accuracy of ResNet-18 is compared.

From the results in Tab. 2, we can make the following key observations. DML can generate an appropriate soft target for the compact model but harm the complex model when there is a significant performance gap, as the prediction of compact model conflicts with the complex model and ground truth. KDCL-Linear outperforms KDCL-Naive as a result of the higher quality soft target.

Model 1	Top-1	Model 2	Top-1	Method
MBV2	72.0	MBV2x0.5	64.8	Vanilla
MBV2	**73.1**	MBV2x0.5	66.2	Linear
MBV2	**73.1**	MBV2x0.5	**66.3**	MinLogit
ResNet-18	71.2	MBV2x0.5	64.8	Vanilla
ResNet-18	71.8	MBV2x0.5*	**65.6**	Linear
ResNet-18	**71.9**	MBV2x0.5*	**65.6**	MinLogit
ResNet-18	71.2	MBV2	72.0	Vanilla
ResNet-18	72.1	MBV2*	**72.8**	Linear
ResNet-18	**72.2**	MBV2*	**72.8**	MinLogit
ResNet-50	76.8	MBV2x0.5	64.8	Vanilla
ResNet-50	77.5	MBV2x0.5*	**67.1**	Linear
ResNet-50	**77.7**	MBV2x0.5*	66.8	MinLogit
ResNet-50*	76.5	ResNet-18*	71.2	Vanilla
ResNet-50*	76.8	ResNet-18*	72.0	Linear
ResNet-50*	**77.0**	ResNet-18*	**72.1**	MinLogit

Table 4: The comparative result of different sub-network on ImageNet validation set. MBV2 is the abbreviation of MobileNetV2. MBV2x0.5 represents the width multiplier is 0.5. ResNet-50* and ResNet-18* are trained for 100 epochs. MBV2* and MBV2x0.5* are trained for 200 epochs.

KDCL-MinLogit can be more efficient and the performance is equal to KDCL-Linear. The result of KDCL-General is not good enough because of the imprecise estimates. The weight for ensembling the prediction is updated each epoch rather than each iteration to save computational cost, so the generated soft target is not as good as KDCL-Linear and KDCL-MinLogit.

Following the setting of ONE [15] and CLNN [22], the low-level layers are shared to save parameters. Multiple branches are ensembled, which are equal to several identical networks. For a fair comparison, we choose a single model as the peer network with fewer parameters than the multiple branch architecture. Our method surpasses the complex ONE with gate controller predicting learnable ensemble logits and carefully designed CLNN with hierarchical architecture as shown in Tab. 3. The result proves that ONE and CLNN are limited with extra knowledge due to the multi-branch design.

Our proposed method applies to various architectures. Therefore, we conduct experiments on the complex-compact pair and compact-compact pair. Tab. 4 shows that the more compact model MobileNetV2×0.5 can provide hint for MobileNetV2, ResNet-18 and even ResNet-50 because the compact model can beat the complex model on some samples as illustrated in Fig. 3. MobileNetV2×0.5 with 1.9M parameters helps to improve ResNet-50 with 25.6M. The fact proves our approach is suitable for the situation that there is a significant performance gap between models. Long training runs can improve accuracy by providing more various soft targets by randomly distorting the training images. Therefore, the top-1 accuracy of ResNet-50 and ResNet-18 is further increased by 0.8% and 1.0% for another 100 training epochs.

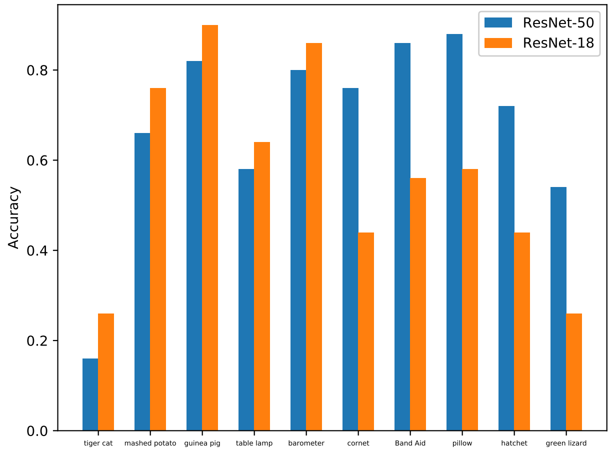

Figure 3: The comparison of ResNet-50 and ResNet-18 on the part categories of ImageNet validation set.

Network	1	2	3	4	5
Top-1 (%)	70.1	71.3	71.61	71.75	71.87

Table 5: KDCL benefits from ensembling more sub-networks. All the networks are ResNet-18 to prevent the impact of network performance differences.

Model	Res-50	Res-18	MBV2	MBV2x0.5	Gain
Vanilla	76.8	71.2	72.0	64.8	0
KDCL	**78.2**	**73.5**	**74.0**	**66.9**	**7.8**

Table 6: Top-1 accuracy rate (%) on ImageNet. ResNet-50 is significantly improved with the knowledge from three compact models.

Ensemble more models. The main experiments show good results using two sub-networks. We also prove that ensembling more models generally give better accuracy. Tab. 5 shows that ensembling two sub-networks can outperform the baseline model with a remarkable margin. KDCL scales well with more sub-networks but the gain decreases as the networks increase. We conjecture that the mutual information between the strong ensemble network and the additional network increases as the ensemble size raises. Experiments are further conducted when utilizing neural networks with different capacities on ImageNet. Tab. 6 shows that ResNet-50 achieves 78.2% top-1 accuracy with knowledge from three compact models.

4.2. Results on CIFAR

Dataset and training details. CIFAR-100 consists of 32×32 color images containing 100 classes. The dataset

Method	ICL	ResNet-32 Acc %	WRN-16-2 Acc %	Gain
Vanilla		69.9	72.2	0
2 distill 1 [10]		73.3	72.2	3.4
1 distill 2 [10]		69.9	74.5	2.3
DML [32]		73.3	74.8	6.0
ONE [15]		73.6	-	-
CLNN [22]		73.4	-	-
KDCL-Naive		73.7	74.8	6.4
KDCL-Naive	√	73.8	74.9	6.6
KDCL-Linear		73.4	74.6	5.9
KDCL-Linear	√	73.6	74.9	6.4
KDCL-MinLogit		73.0	74.1	5.0
KDCL-MinLogit	√	73.5	74.6	6.0
KDCL-General		74.0	75.2	7.1
KDCL-General	√	**74.3**	**75.5**	**7.7**

Table 7: The comparative and ablative result of our generate distillation method on CIFAR-100 dataset. ICL is invariant collaborative learning. We only report the accuracy of ResNet-32 as ONE and CLNN are incompatible with WRN-16-2.

is split into a training set with 50,000 images and a test set with 10,000 images. For KDCL-General, we separate 5,000 images from the training set, 50 samples on each class, as the validation set to measure the generalization ability of students. All the models are trained for 200 epochs with learning rate starting at 0.1 and the learning rate drops by 0.1 at 100 and 150 epoch. We set the weight decay to 0.0005, batch size to 128, and the momentum is 0.9. All the training images are padded with 4 pixels and a 32 × 32 crop is randomly sampled from the padded images or its horizontal flip. The temperature T and λ are 2 and 1 separately. Accuracy is computed as the median of 5 runs.

Knowledge distillation relieves over-fitting. The second and third row of Tab. 7 show that the student model becomes more general for knowledge distillation, even surpasses the teacher model. For distillation (2nd row in Tab. 7) from Wide-ResNet-16 [30] with a widening factor 2 (WRN-16-2) to ResNet-32, we observe that the accuracy of the student network ResNet-32 is 93.37% on the train set, behind the teacher network WRN-16-2 99.39%, while the test error is lower than WRN-16-2. This phenomenon demonstrates the knowledge distillation can relieve over-fitting.

Quantitative comparison. Most of our proposed methods outperform DML as a result of the effective learning mechanism and the end-to-end training manner compared with multi-stage parameters updating. An interesting phenomenon is observed that KDCL-MinLogit and KDCL-Linear do worse than KDCL-Naive, which conflicts with the results on ImageNet. We conjecture that the soft target with less cross-entropy loss on CIFAR-100 train set leads to over-fitting like the one-hot label. KDCL-General significantly improves the performance by the more general teacher model according to the optimal weighted average on the validation set. This result proves that our approaches can further improve the ability of knowledge distillation to alleviate over-fitting. It also shows that there is a trade-off between the fitting ability and generalization ability when the amount of data is limited.

The ablation study in Tab. 7 shows that invariant collaborative learning is promising. The improvement comes from fusing information of different distorting images and the shared soft target also encourages the sub-network to output similarly with similar input.

Backbone	Type	box mAP	mask mAP
ResNet-18 (BaseLine)	Faster	32.2	-
ResNet-18 (Our)	Faster	**33.1**	-
ResNet-18 (BaseLine)	Mask	33.4	30.7
ResNet-18 (Our)	Mask	**34.0**	**31.3**

Table 8: Average precision (AP) on COCO 2017 validation set with pre-trained ResNet-18. All models are used as backbones for Faster-RCNN [19], Mask-RCNN [7] based on FPN [16].

4.3. Transfer Learning

Dataset and training details. We follow the commonly used practice [19, 7] to divide the 40k validation set into a 35k and 5k subset. The training set containing 80k images and the 35k subset are used for training. The 5k subset that denoted as *minival* set is used to validate our result. All the models are trained for 14 epochs on 8 GPUs with 4 images on each GPU. The learning rate starts at 0.04 and is decayed by 0.1 at 9 and 12 epoch. The weight decay is 0.0001 and momentum is 0.9. To fully utilize the capacity of the model, all the batch normalization layers are in sync mode and no weight is frozen. We replace ROI-Pooling with ROI-Align [7] for better results by default.

Results. Tab. 8 reports that the validation set performance of object detection and instance segmentation on standard AP metric (corresponds to the average AP for IOU from 0.5 to 0.95 with a step size of 0.05.). Based on ResNet-18 trained with KDCL, the detection head outperforms baseline by 0.9%. Our proposed learning mechanism also brings improvements on instance segmentation by 0.6%. The improvements come from the more powerful generalization. In summary, this set of experiments demonstrates that the improvements induced by our learning mechanism can be realized across a broad range of tasks and datasets.

5. Analysis

Definition of generalisation error. The L2-norm for generalisation error (Eq.8) is for simplicity and results in a neat analytic solution, which is easy and efficient to implement. Actually Eq. 11 holds as long as the function $E(\mathbf{x}) = g(f_i(\mathbf{x}), t)$ is convex and nonnegative(e.g. cross entropy loss). We can derive this using Jensen's inequality as follow:

$$\begin{aligned}
&E(\mathbf{x}) = g(\sum_i^m w_i f_i(\mathbf{x}), t) \leq (\sum_i^m w_i g(f_i(\mathbf{x}), t)) = E^{'}(\mathbf{x}) \\
&E^{'2} = \int E^{'}(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \sum_i^m \sum_j^m w_i w_j C_{ij} \\
&C_{ij} = \int g(f_i(\mathbf{x}), t) g(f_j(\mathbf{x}), t) p(\mathbf{x}) d\mathbf{x},
\end{aligned} \tag{12}$$

Minimizing the upper boundary $E^{'}$ can reduce the generalization error.

Sensitivity to loss hyper-parameter. In the main experiments, we simply set the hyper-parameter λ to 1, aimed to highlight the effectiveness of KDCL. The results in Tab. 9 show that performance is insensitive to the choice of λ over a large range from $\frac{1}{5}$ to 5, yet a carefully adjusted hyper-parameter does bring some extra performance gain.

λ	0.1	0.2	0.5	1	2	5	10
R	71.5	73.3	73.8	74.3	**74.7**	74.0	72.9
W	73.9	74.5	75.0	75.5	**76.1**	74.9	74.5

Table 9: The model pair ResNet-32 and WRN-16-2 is trained with KDCL-General method on CIFAR-100.

Sensitivity to data augmentation policies. We adopted Cutout[5], which yields state-of-the-art results, to evaluate our method on CIFAR-100. The results in Tab. 10 the following table proves that our method is insensitive to the choice of data augmentation.

Method	ICL	ResNet-32	WRN-16-2
Vanilla		72.3	74.3
KDCL		74.3	75.5
KDCL	√	**74.6**	**75.8**

Table 10: The model pair is trained with Cutout policy.

How does ICL contribute? Invariant collaborative learning (ICL) leads to similar class probability predictions, meaning small intra-class distance. We measure the similarity of predictions by calculating the standard deviation of probability predictions of each class. The comparison is based on ResNet-32 and WRN-16-2 trained on CIFAR-100. The averages of the standard deviation are 0.347 and 0.343. ICL reduces the averages to 0.337 and 0.336.

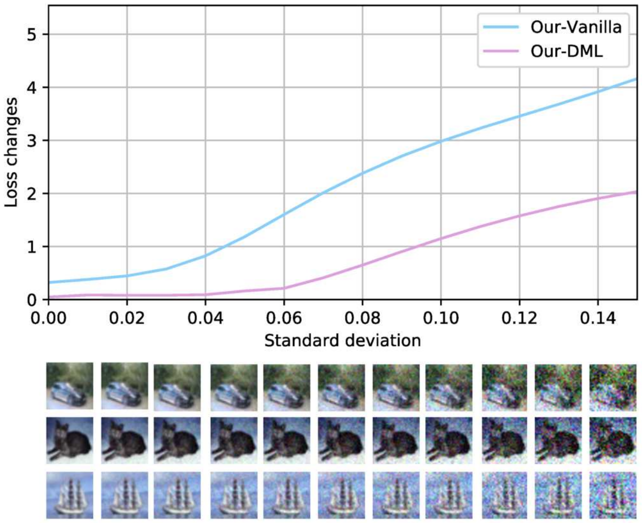

Figure 4: The curve of loss gap from KDCL-General to baseline and DML on CIFAR-100. The robustness test is based on ResNet-32 solutions found by KDCL, DML and Vanilla training.

Robustness to Noise. We bring the noise to the testing images and observe that our method is robust to random noise. The noise is sampled from independent Gaussian distribution $\mathcal{N}(0, \delta^2)$ and added to each pixel. The std δ is set ranging from 0.0 to 0.15 by step 0.01. At each magnitude scale, the loss is the average of 10 independent tests. Fig. 4 shows that the loss gap from KDCL to baseline and DML becomes more significant with the increasing of δ.

6. Conclusion

In this work, we introduce a novel online knowledge distillation framework based on collaborative learning strategy. With our ensemble method, models of various capacity can benefit from collaborative learning. Further, we improve the invariance by transferring knowledge via fusing information of image distorting. Experiments on ImageNet, CIFAR, and COCO show that KDCL significantly reduces the generalization error while keeping that students predict independently. The performance gain persists when the trained model is fine-tuned on other vision tasks. KDCL could be further extended to other recognition tasks and may help to optimize the collaborative learning of multiple relative tasks, like joint human parsing and pose estimation.

Acknowledgments This work was partially supported by HKU Seed Fund for Basic Research, Start-up Fund, Research Donation from SenseTime, the National Natural Science Foundation of China under Grant Nos. U19B2034 and 61836014.

References

[1] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018. 2

[2] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014. 2

[3] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. Label refinery: Improving imagenet classification through label progression. *arXiv preprint arXiv:1805.02641*, 2018. 2

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2

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

[6] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018. 2

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 7

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2, 3

[9] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge distillation with adversarial samples supporting decision boundary. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3771–3778, 2019. 2

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2, 3, 5, 7

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016. 3

[12] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, 2018. 3

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 2

[14] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003. 2

[15] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7528–7538. Curran Associates Inc., 2018. 1, 2, 5, 6, 7

[16] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 7

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[18] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7

[20] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2

[21] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 5

[22] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1832–1841, 2018. 2, 5, 6, 7

[23] Suraj Srinivas and Francois Fleuret. Knowledge transfer with Jacobian matching. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4723–4731, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 2

[24] 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. 3

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 5

[26] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013. 3

[27] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. 3

[28] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L. Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[29] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network

minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4133–4141, 2017. 2

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 7

[31] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 2

[32] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4328, 2018. 1, 2, 5, 7

[33] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002. 3