Peer Collaborative Learning for Online Knowledge Distillation

Guile Wu, Shaogang Gong
Queen Mary University of London
guile.wu@qmul.ac.uk, s.gong@qmul.ac.uk

Abstract

Traditional knowledge distillation uses a two-stage training strategy to transfer knowledge from a high-capacity teacher model to a compact student model, which relies heavily on the pre-trained teacher. Recent online knowledge distillation alleviates this limitation by collaborative learning, mutual learning and online ensembling, following a one-stage end-to-end training fashion. However, collaborative learning and mutual learning fail to construct an online high-capacity teacher, whilst online ensembling ignores the collaboration among branches and its logit summation impedes the further optimisation of the ensemble teacher. In this work, we propose a novel Peer Collaborative Learning method for online knowledge distillation, which integrates online ensembling and network collaboration into a unified framework. Specifically, given a target network, we construct a multi-branch network for training, in which each branch is called a peer. We perform random augmentation multiple times on the inputs to peers and assemble feature representations outputted from peers with an additional classifier as the peer ensemble teacher. This helps to transfer knowledge from a high-capacity teacher to peers, and in turn further optimises the ensemble teacher. Meanwhile, we employ the temporal mean model of each peer as the peer mean teacher to collaboratively transfer knowledge among peers, which helps each peer to learn richer knowledge and facilitates to optimise a more stable model with better generalisation. Extensive experiments on CIFAR-10, CIFAR-100 and ImageNet show that the proposed method significantly improves the generalisation of various backbone networks and outperforms the state-of-the-art methods.

Introduction

Deep learning has achieved incredible success in many computer vision tasks in recent years. Whilst many studies focus on developing deeper and/or wider networks for improving the performance (He et al. 2016; Zagoruyko and Komodakis 2016; Xie et al. 2017), these cumbersome networks require more computational resources, which hinders their deployments in resource-limited scenarios. To alleviate this problem, knowledge distillation is developed to transfer knowledge from a stronger teacher (Hinton, Vinyals, and Dean 2015) or an online ensemble (Lan, Zhu, and Gong 2018) to a student model, which is more suitable for deployment.

Traditionally, knowledge distillation (KD) requires to pre-train a high-capacity teacher model in the first stage, and then transfer the knowledge of the teacher to a smaller student model in the second stage (Hinton, Vinyals, and Dean 2015; Romero et al. 2015; Phuong and Lampert 2019). Via aligning soft predictions (Hinton, Vinyals, and Dean 2015) or feature representations (Romero et al. 2015) between the teacher and the student, a student model usually significantly reduces the model complexity for deployment but still achieves competitive accuracy as the teacher model. However, since the teacher and the student are trained in two separate stages, this traditional strategy usually requires more training time and computational cost.

Recent online knowledge distillation (Lan, Zhu, and Gong 2018; Zhang et al. 2018; Chen et al. 2020) alleviates this limitation by directly optimising a target network, following a one-stage end-to-end training fashion. Instead of pre-training a high-capacity teacher, online knowledge distillation typically integrates the teacher into the student using a hierarchical network with shared intermediate-level representations (Song and Chai 2018) (Fig. 1(a)), multiple parallel networks (Zhang et al. 2018)(Fig. 1(b)), or a multi-branch network with online ensembling (Lan, Zhu, and Gong 2018) (Fig. 1(c)). Although these methods have shown their superiority over the traditional counterparts, collaborative learning and mutual learning fail to construct an online high-capacity teacher to facilitate the optimisation of the student, whilst online ensembling ignores the collaboration among branches and its logit summation impedes the further optimisation of the ensemble teacher.

In this work, we propose a novel Peer Collaborative Learning (PCL) method for online knowledge distillation. As shown in Fig. 1(d), we integrate online ensembling and network collaboration into a unified framework to take full advantage of them for improving the quality of online knowledge distillation. Specifically, in training, we construct a multi-branch network by adding auxiliary branches (high-level layers) to a given target network. We call each branch "*a peer*" and design two types of online teachers for peer collaborative learning to improve the generalisation of a target network. The first teacher, *peer ensemble teacher*, is an online high-capacity model, which helps to distil knowledge

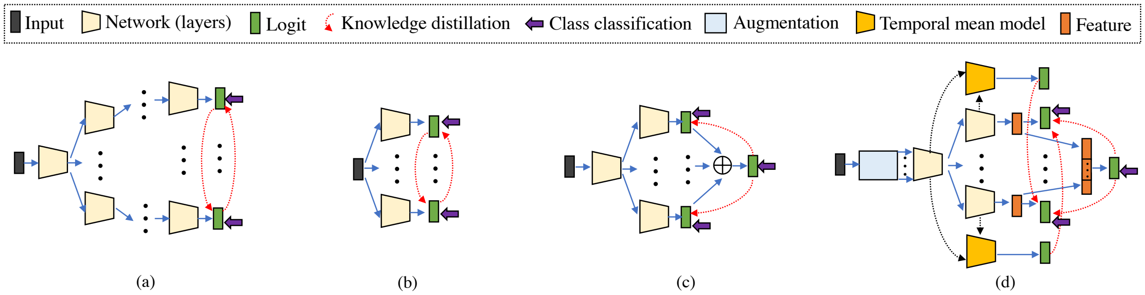


Figure 1: Comparing four online knowledge distillation mechanisms: (a) Collaborative learning. (b) Mutual learning. (c) Online ensembling. (d) Peer collaborative learning (Proposed). Our method integrates two types of peer collaborations (*i.e.* peer ensemble teacher and peer mean teacher) into a unified framework to improve the quality of online knowledge distillation.

from a stronger ensemble teacher to each peer, and in turn further improves the ensemble teacher. Instead of using peer logit summation to construct the ensemble teacher (Lan, Zhu, and Gong 2018), we perform random augmentation multiple times on the inputs to peers and then assemble feature representations outputted from peers with an additional classifier as the peer ensemble teacher. This design helps to learn discriminative information among feature representations of peers and facilitates to assemble a stronger teacher for online knowledge distillation. Furthermore, to generate a more stable model with better generalisation, we use the second teacher, *peer mean teacher*, to collaboratively distil knowledge among peers. Instead of directly distilling knowledge among peers using mutual learning (Zhang et al. 2018), we utilise the temporal mean model of each peer to construct the peer mean teacher which can produce more stable predictions. As a result, this design helps each peer to learn richer knowledge and facilitates to optimise a more stable model with better generalisation for deployment. In testing, we use a temporal mean model of a peer for deployment, which has the same number of parameters as the given target network, so there is no extra inference cost for deployment. Besides, the outputted feature representations from peer mean teachers plus the additional classifier can form a high-capacity ensemble for deployment to get better performance in the scenarios where computational cost is less constrained.

Our **contributions** are: **(I)** We propose a novel Peer Collaborative Learning method for online knowledge distillation, which integrates online ensembling and network collaboration into a unified framework; **(II)** We construct a peer ensemble teacher via performing random augmentation multiple times on the inputs to peers and assembling feature representations outputted from peers with an additional classifier. This helps to simultaneously optimise peers and the ensemble teacher for online knowledge distillation. **(III)** We utilise the temporal mean model of each peer to construct the peer mean teacher for peer collaborative distillation, resulting in a more stable model with better generalisation; **(IV)** Extensive experiments on CIFAR-10 (Krizhevsky and Hinton 2009), CIFAR-100 (Krizhevsky and Hinton 2009) and ImageNet (Russakovsky et al. 2015) using a variety of backbone networks show that the proposed method significantly improves the generalisation of the backbone networks and outperforms the state-of-the-art alternative methods.

Related Work

Traditional Knowledge Distillation (Hinton, Vinyals, and Dean 2015) is one of the most effective solutions to compress a cumbersome model or an ensemble of models into a smaller model for deployment. In (Hinton, Vinyals, and Dean 2015), Hinton, Vinyals, and Dean propose to distil the knowledge from a high-capacity teacher model to a compact student model, which is accomplished by aligning soft output predictions between the teacher and the student. In recent years, many promising methods have been designed to transfer various "knowledge", such as intermediate representations (Romero et al. 2015), flow between layers (Yim et al. 2017), attention maps (Zagoruyko and Komodakis 2017), structural relations (Park et al. 2019) and activation similarity (Tung and Mori 2019), to facilitate the optimisation process of distillation. Although these methods have shown competitive performance for compressing a model, they typically follow a two-stage training solution by pre-training a high-capacity teacher model for transferring knowledge to a compact student model, which requires more training time and computational cost.

Online Knowledge Distillation (Lan, Zhu, and Gong 2018; Chen et al. 2020; Zhang et al. 2018) proposes to directly optimise a target network via distilling knowledge among multiple networks or branches without pre-training a high-capacity teacher, which follows a one-stage end-to-end training strategy. Since online KD directly optimises a target network, there is no need to store or download a teacher model, which saves time and computational cost. In (Song and Chai 2018), Song and Chai propose to distil knowledge among multiple classifier heads of a hierarchical network for improving the generalisation of a target network. In (Zhang et al. 2018), Zhang et al. introduce a mutual learning solution to distil knowledge among multiple parallel networks with the same input. Although these methods help to improve the generalisation of the target network, they only distil limited knowledge among parallel networks or heads and fail to construct a stronger online teacher to further improve the student. In (Guo et al. 2020), Guo et al. employ multiple

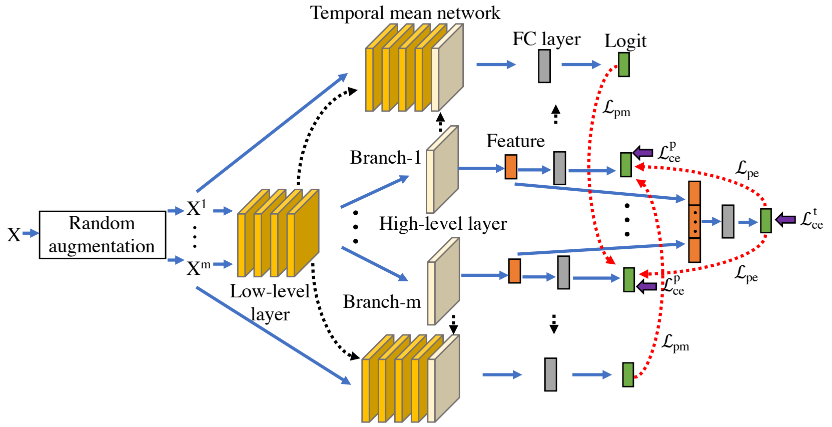

Temporal mean network
FC layer
Logit
$\mathcal{L}_{\text{pm}}$
Feature
$\mathcal{L}_{\text{ce}}^{\text{p}}$
$\mathcal{L}_{\text{pe}}$
Branch-1
High-level layer
$\mathcal{L}_{\text{ce}}^{\text{t}}$
X
Random augmentation
X^1
X$^\text{m}$
Low-level layer
Branch-m
$\mathcal{L}_{\text{ce}}^{\text{p}}$
$\mathcal{L}_{\text{pe}}$
$\mathcal{L}_{\text{pm}}$

Figure 2: An overview of Peer Collaborative Learning (PCL) for online knowledge distillation.

parallel networks and aggregate logits from all student networks based on the cross-entropy loss to generate soft targets for online distillation. More similar to our work, Lan, Zhu, and Gong (Lan, Zhu, and Gong 2018) use a multi-branch network and assemble logits from multiple branches (students) as the teacher to improve the generalisation of each student. However, logit aggregation impedes the further optimisation of the ensemble teacher and online ensembling ignores the collaboration among branches, resulting in suboptimal performance. In (Kim et al. 2020), Kim et al. integrate feature representations of multiple branches into the online ensembling, but their method requires more convolutional operations for the feature fusion and also fails to exploit the collaboration among branches. To address these limitations, in our work: (1) we assemble feature representations from peers with an additional classifier as the *peer ensemble teacher*, which helps to distil knowledge from an online high-capacity teacher to each peer (student) and in turn further optimises the teacher; (2) we exploit the temporal mean model of each peer as the *peer mean teacher* to distil knowledge among peers, which helps each peer to learn richer knowledge and facilitates to optimise a more stable model. Integrating these two teachers into a unified framework significantly improves the generalisation of each peer and the ensemble, resulting in better performance.

Neural Network Ensembling is a simple and effective solution for improving the generalisation performance of a model (Hansen and Salamon 1990; Zhou, Wu, and Tang 2002; Moghimi et al. 2016). Although this can usually bring better performance, training multiple neural networks to create an ensemble requires significantly more training time and computational cost. The recent trend in neural network ensembling focuses on training a single model and exploiting different training phases of a model as an ensemble. In (Huang et al. 2017a), Huang et al. force the model to visit multiple local minima and use the corresponding models as the snapshots for neural network ensembling. In (Laine and Aila 2017), Laine and Aila propose to use temporal ensembling of network predictions over multiple training epochs as the teacher to facilitate the optimisation of the current model for semi-supervised learning. Our work differs from these works in that we use feature representations of peers from a multi-branch network with an additional classifier as the ensemble teacher for online knowledge distillation, rather than using the network predictions from different phases or generating multiple networks for ensembling. In our method, the peer mean teacher shares the merit of the traditional Mean Teacher (Tarvainen and Valpola 2017). In (Tarvainen and Valpola 2017), network weights over previous training epochs are aggregated as a teacher to minimise the distance of predictions between the student and the teacher as the consistency regularisation for semi-supervised learning. In contrast, our method uses the shared low-level layers and multiple separated high-level layers to construct multiple peer mean teachers for aligning soft prediction distributions between the peer and its counterparts' mean teacher, resulting in a more stable model for improving the quality of online knowledge distillation.

Peer Collaborative Learning

Approach Overview

The overview of the proposed Peer Collaborative Learning (PCL) is depicted in Fig. 2. We employ an m-branch network for model training and call each branch "*a peer*". Since the low-level layers across different branches usually contain similar low-level features regarding minor details of images, sharing them enables to reduce the training cost and improve the collaboration among peers (Lan, Zhu, and Gong 2018). We therefore share the low-level layers and separate the high-level layers in the m-branch network.

As shown in Fig. 2, to facilitate online knowledge distillation, we assemble the feature representation of peers with an additional classifier as the *peer ensemble teacher* and use the temporal mean model of each peer as the *peer mean teacher*. The training optimisation objective of PCL contains three components: The first component is the standard cross-entropy loss for classification of the peers ($\mathcal{L}_{ce}^{p}$) and the peer ensemble teacher ($\mathcal{L}_{ce}^{t}$); The second component is the peer ensemble teacher distillation loss $\mathcal{L}_{pe}$ for transferring knowledge from a stronger teacher to a student, which in turn further improves the ensemble teacher; The third component is the peer mean teacher distillation loss $\mathcal{L}_{pm}$ for collaboratively distilling knowledge among peers. Thus, the overall objective $\mathcal{L}$ is formulated as:

$$\mathcal{L} = \mathcal{L}^p_{ce} + \mathcal{L}^t_{ce} + \mathcal{L}_{pe} + \mathcal{L}_{pm}. \tag{1}$$

In testing, we use a temporal mean model of a peer for deployment, which has the same number of parameters as the backbone network, so there is no extra inference cost for deployment. In the scenarios where computational cost is less constrained, feature representations from peer mean teachers plus the additional classifier can form an ensemble model for deployment to get better performance.

Peer Ensemble Teacher

Input Augmentation for Peers. Suppose there are n samples in a training dataset $\{(x_i, y_i)\}_{i=1}^n$, where x_i is the i-th input sample, $y_i \in \{1, 2, ..., C\}$ is the corresponding label, and C is the number of classes in the dataset ($C \leq n$). Existing multi-branch online distillation methods (Lan, Zhu, and Gong 2018; Chen et al. 2020) directly use x_i (applying random augmentation once) as the input to all the branches, which causes the homogenisation among peers and decreases the generalisation of the network. To alleviate this problem, we perform random augmentation m times to x_i to generate m counterparts of x_i (*i.e.* $\{x_i^1, x_i^2, ..., x_i^m\}$, and use each counterpart as the input to each peer. This stochastic augmentation fashion is similar to (Laine and Aila 2017), but we perform it multiple times to assemble discriminative features as the ensemble teacher in a multi-branch network, rather than to generate two predictions for consistency regularisation or distillation.

Online Ensembling. To construct a stronger online teacher for online knowledge distillation, logits from multiple networks/branches are usually aggregated (w/ or w/o attention gates) (Lan, Zhu, and Gong 2018; Chen et al. 2020). However, this impedes the ensemble teacher from further optimising the ensemble model and ignores the discriminative information among feature representations of peers, which might lead to a suboptimal solution since the logit summation is not further learned. In our work, we concatenate the features outputted from peers and use an additional fully connected layer for classification to construct a learnable stronger online teacher. Thus, the multi-class classification is performed for both the peers and the ensemble teacher as:

$$\mathcal{L}^p_{ce} = -\sum_{j=1}^{m}\sum_{c=1}^{C} y_c log \frac{exp(z^p_{j,c})}{\sum_{k=1}^{C} exp(z^p_{j,k})}, \tag{2}$$

$$\mathcal{L}^t_{ce} = -\sum_{c=1}^{C} y_c log \frac{exp(z^t_c)}{\sum_{k=1}^{C} exp(z^t_k)}, \tag{3}$$

where $z^p_{j,c}$ is the output logit from the last fully connected layer of the j-th peer over a class c, y_c is the ground-truth label indicator, z^t_c is the output logit from the fully connected layer of the peer ensemble teacher over a class c.

Then, to transfer knowledge from the ensemble teacher to each peer, we compute the soft prediction of the j-th peer and the ensemble teacher as:

$$p^p_{j,c} = \frac{exp(z^p_{j,c}/T)}{\sum_{k=1}^{C} exp(z^p_{j,k}/T)},\ p^t_c = \frac{exp(z^t_c/T)}{\sum_{k=1}^{C} exp(z^t_k/T)}, \tag{4}$$

where T is a temperature parameter (Hinton, Vinyals, and Dean 2015), $p^p_{j,c}$ is the soft prediction of the j-th peer over a class c, and p^t_c is the soft prediction of the ensemble teacher over a class c. Using Kullback Leibler (KL) divergence, the peer ensemble distillation loss $\mathcal{L}_{pe}$ is formulated as:

$$\mathcal{L}_{pe} = \omega(e) \cdot T^2 \sum_{j=1}^{m}\sum_{c=1}^{C} p^t_c \cdot log \frac{p^t_c}{p^p_{j,c}}, \tag{5}$$

where T^2 ensures contributions of ground-truth and teacher probability distributions keep roughly unchanged (Hinton, Vinyals, and Dean 2015), e is the current training epoch, $\omega(\cdot)$ is a weight ramp-up function (Laine and Aila 2017) stabilises model training, which is defined as:

$$\omega(e) = \begin{cases} \lambda \cdot exp(-5 * (1 - \frac{e}{\alpha})^2) & , e \leq \alpha; \\ \lambda & , e > \alpha; \end{cases} \tag{6}$$

where α is the epoch threshold for the ramp-up function and λ is a parameter weighting the gradient magnitude.

Remarks. The proposed peer ensemble teacher differs from existing feature fusion (Hou, Liu, and Wang 2017; Kim et al. 2020; Chen, Zhu, and Gong 2017) in that we construct an online high-capacity teacher model by performing random augmentation multiple times on the input to peers and assembling feature representations from peers of a multi-branch network with an additional classifier, without using additional convolutional operations or multiple networks. This helps to effectively distil knowledge from a stronger ensemble teacher to each peer, and in turn further improves the ensemble teacher.

Peer Mean Teacher

Online ensembling helps to construct a stronger teacher for online knowledge distillation, but it ignores the collaboration among peers. On the other hand, mutual learning (Zhang et al. 2018) and collaborative learning (Song and Chai 2018) benefit from mutual distillation among networks or heads, but they fail to construct a high-capacity teacher for online distillation. In our work, we further use peer mutual distillation for improving collaboration among peers. Instead of directly distilling knowledge among peers, we use the temporal mean model (Tarvainen and Valpola 2017) of each peer as the peer mean teacher for peer collaborative distillation. We denote the weights of the shared low-level layers as θ_l and the weights of the separated high-level layers of the j-th peer as $\theta_{h,j}$. At the g-th global training step [1], the j-th peer mean teacher $\{\theta^t_{l,g}, \theta^t_{h,j,g}\}$ is formulated as:

$$\begin{cases} \theta^t_{l,g} = \phi(g) \cdot \theta^t_{l,g-1} + (1 - \phi(g)) \cdot \theta_{l,g}, \\ \theta^t_{h,j,g} = \phi(g) \cdot \theta^t_{h,j,g-1} + (1 - \phi(g)) \cdot \theta_{h,j,g}, \end{cases} \tag{7}$$

[1] Here, g = $e \cdot$Batch$_{num}$ + Batch$_{ind}$, where Batch$_{num}$ is the total number of training mini-batches in each epoch and Batch$_{ind}$ is the index of the current mini-batch.

where $\theta^t_{l,g}$ are the weights of the shared low-level layers of the peer mean teachers, $\theta^t_{h,j,g}$ are the weights of the separated high-level layers of the j-th peer mean teacher, $\phi(g)$ is a smoothing coefficient function defined as:

$$\phi(g) = min(1 - \frac{1}{g}, \beta), \tag{8}$$

where β is the smoothing coefficient hyper-parameter. Note that, the additional classifier of the peer ensemble teacher is also aggregated for the ensemble deployment. We compute the soft prediction $p^{mt}_{j,c}$ of the j-th mean teacher over a class c as Eq. (4) with the output logit $z^{mt}_{l,c}$ of this mean teacher. Thus, the peer mean teacher distillation loss $\mathcal{L}_{pm}$ is formulated as:

$$\mathcal{L}_{pm} = \omega(e)\cdot\frac{T^2}{m-1}\sum_{j=1}^{m}\sum_{l=1,l\neq j}^{m}\sum_{c=1}^{C} p^{mt}_{l,c}\cdot log\frac{p^{mt}_{l,c}}{p^{p}_{j,c}}. \tag{9}$$

Remarks. The traditional mean teacher is used for semi-supervised/unsupervised learning (Tarvainen and Valpola 2017; Mittal, Tatarchenko, and Brox 2019; Ge, Chen, and Li 2020), which mainly enforces the distance between the model predictions to be close. In contrast, we employ the temporal mean model of each peer in a multi-branch network as the peer mean teacher, and use it for peer collaborative distillation by aligning the soft distributions between the peer and its counterparts' mean teacher. Compared with mutual distillation among peers (Zhang et al. 2018), averaging model weights temporally over training epochs enables the peer mean teacher to stabilise soft predictions for improving peer collaboration and generating a more stable network.

Summary. As shown in Algorithm 1, PCL follows a one-stage training fashion without pre-training a high-capacity teacher. In test, we use a single peer mean model as the target model (PCL) without adding extra inference cost. Besides, the ensemble of peer mean teachers (with the additional classifier) can be used as a high-capacity ensemble (PCL-E).

Experiment

Datasets. We used three image classification benchmarks for evaluation: (1) *CIFAR-10* (Krizhevsky and Hinton 2009) contains 60000 images in 10 classes, with 5000 training images and 1000 test images per class. (2) *CIFAR-100* (Krizhevsky and Hinton 2009) consists of 60000 images in 100 classes, with 500 training images and 100 test images per class. (3) *ImageNet ILSVRC 2012* (Russakovsky et al. 2015) contains 1.2 million training images and 50000 validation images in 1000 classes.

Implementation Details. To verify the effectiveness of our method, we used a variety of backbone networks, including ResNet (He et al. 2016), VGG (Simonyan and Zisserman 2015), DenseNet (Huang et al. 2017b), WRN (Zagoruyko and Komodakis 2016), and ResNeXt (Xie et al. 2017). Following (Lan, Zhu, and Gong 2018), the last block of each backbone network was separated from the parameter sharing (on ImageNet, the last two blocks were separated), while the other low-level layers were shared. We set m=3 peers in the multi-branch architecture. We used standard random crop and horizontal flip for the random augmentation in training, and did not use random augmentation in testing. We used SGD as the optimiser with Nesterov momentum 0.9 and weight decay $5e$-4. We trained the network for $Epoch_{max}$=300 epochs on CIFAR-10/100 and 90 epochs on ImageNet. We set the initial learning rate to 0.1, which decayed to {0.01, 0.001} at {150, 225} epochs on CIFAR-10/100 and at {30, 60} epochs on ImageNet. We set the batch size to 128, the temperature T=3, α=80 for ramp-up weighting, β=0.999 to learn temporal mean models, λ=1.0 for CIFAR-10/100 and λ=0.1 for ImageNet. By default, in PCL, we used the first branch as the target network. Our models were implemented with Python 3.6 and PyTorch 0.4, and trained on TESLA V100 GPU (32GB).

Algorithm 1 Peer Collaborative Learning for Online KD.

```
Input: Training data $\{(x_i, y_i)\}_{i=1}^{n}$.
Output: A trained target model $\{\theta^t_l, \theta^t_{h,1}\}$,
        and a trained ensemble model $\{\theta^t_l, \theta^t_{h,j}\}_{j=1}^{m}$.
/* Training */
Initialisation: Randomly initialise model parameters
for $e = 0 \rightarrow Epoch_{max}$ do    /* Mini-Batch SGD */
    Randomly transform $x_i$ to get counterparts $\{x_i\}_{j=1}^{m}$
    Compute features and logits of peers
    Assembling features as the peer ensemble teacher
    Compute the logits of the teachers
    Compute peer classification loss $\mathcal{L}^p_{ce}$ (Eq.(2))
    Compute ensemble classification loss $\mathcal{L}^t_{ce}$ (Eq.(3))
    Compute peer ensemble distillation loss $\mathcal{L}_{pe}$(Eq.(5))
    Compute mean teacher distillation loss $\mathcal{L}_{pm}$(Eq.(9))
    Update peer models with Eq.(1)
    Update peer mean teachers with Eq.(7)
end for
/* Testing */
Deploy with a single target model $\{\theta^t_l, \theta^t_{h,1}\}$
Deploy with an ensemble model $\{\theta^t_l, \theta^t_{h,j}\}_{j=1}^{m}$
```

Evaluation Metrics. We used the top-1 classification error rate (%) and reported the average results with the standard deviation over 3 runs.

Comparison with the State-of-the-Arts

Competitors. We compared PCL with backbone networks (Baseline) and six online KD state-of-the-arts (DML (Zhang et al. 2018), CL (Song and Chai 2018), ONE (Lan, Zhu, and Gong 2018), FFL-S (Kim et al. 2020), OKDDip (Chen et al. 2020), KDCL(-Naive) (Guo et al. 2020)).

Setting. For fair comparisons, following (Lan, Zhu, and Gong 2018), we used three-branch architectures in compared methods (ONE, CL, FFL-S, OKDDip and PCL) unless they have to be used with network-based architectures (three parallel networks in DML and KDCL). Here, although the network-based OKDDip usually yields better performance than the branch-based one, the former one uses more parameters for training, so we used the branch-based OKDDip.

Results. As shown in Table 1, the proposed PCL improves the performance of various backbone networks (baseline) by

Dataset	Network	DML	CL	ONE	FFL-S	OKDDip	KDCL	Baseline	PCL(ours)
C10	ResNet32	6.06±0.07	5.98±0.28	5.80±0.12	5.99±0.11	5.83±0.15	5.99±0.08	6.74±0.15	**5.67±0.12**
	ResNet110	5.47±0.25	4.81±0.11	4.84±0.30	5.28±0.06	4.86±0.10	4.89±0.16	5.31±0.10	**4.47±0.16**
	VGG16	5.87±0.07	5.86±0.15	5.86±0.23	6.78±0.08	6.02±0.06	5.91±0.12	6.04±0.13	**5.26±0.02**
	DenseNet40-12	6.41±0.26	6.95±0.25	6.92±0.21	6.72±0.16	7.36±0.22	6.13±0.08	6.81±0.02	**5.87±0.13**
	WRN20-8	4.80±0.13	5.41±0.08	5.30±0.14	5.28±0.13	5.17±0.15	4.73±0.16	5.32±0.01	**4.58±0.04**
	ResNeXt29	4.46±0.16	4.45±0.18	4.27±0.10	4.67±0.04	4.34±0.02	4.02±0.27	4.72±0.03	**3.93±0.09**
C100	ResNet32	26.32±0.14	27.67±0.46	26.21±0.41	27.82±0.11	26.75±0.38	26.24±0.34	28.72±0.19	**25.86±0.16**
	ResNet110	22.14±0.50	21.17±0.58	21.60±0.36	22.78±0.41	21.46±0.26	21.72±0.32	23.79±0.57	**20.02±0.55**
	VGG16	24.48±0.10	25.67±0.08	25.63±0.39	29.13±0.99	25.32±0.05	24.33±0.22	25.68±0.19	**23.11±0.25**
	DenseNet40-12	26.94±0.31	28.55±0.34	28.40±0.38	28.75±0.35	28.77±0.14	27.48±0.42	28.97±0.15	**26.91±0.16**
	WRN20-8	20.23±0.07	20.60±0.12	20.90±0.39	21.78±0.14	21.17±0.06	20.63±0.30	21.97±0.40	**19.49±0.49**
	ResNeXt29	18.94±0.01	18.41±0.07	18.60±0.25	20.18±0.33	18.50±0.11	18.64±0.18	20.57±0.43	**17.38±0.23**
ImgNet	ResNet18	30.18±0.08	29.96±0.05	29.82±0.13†	31.15±0.07	30.07±0.06	30.40±0.05	30.49±0.14	**29.58±0.13**

Table 1: Comparisons with the state-of-the-arts on CIFAR-10, CIFAR-100 and ImageNet. Top-1 error rates (%) are reported. ResNeXt29: ResNeXt29-2×64d. Implementations of all methods are mainly based on [1] and [2]. †: Reported result 29.45±0.23.

	Component	CIFAR-100
	Backbone	23.79±0.57
Proposed	Backbone+L_{ce}^{p}	23.56±0.50
	Backbone+L_{ce}^{p}+L_{ce}^{t}	23.48±0.99
	Backbone+L_{ce}^{p}+L_{ce}^{t}+L_{pe}	21.19±0.62
	Backbone+L_{ce}^{p}+L_{ce}^{t}+L_{pe}+L_{pm} (full)	**20.02±0.55**
Variant	P.E.+Mutual Distillation	21.09±0.18
	P.E.+Traditional Mean Model(weighted)	21.36±0.73
	Logit Sum + P.M.	20.43±0.71
	P.E. + P.M. (full model)	**20.02±0.55**

Table 2: Component effectiveness evaluation with ResNet-110 on CIFAR-100. Top-1 error rates (%). P.E.: Peer Ensemble teacher. P.M.: Peer Mean teacher.

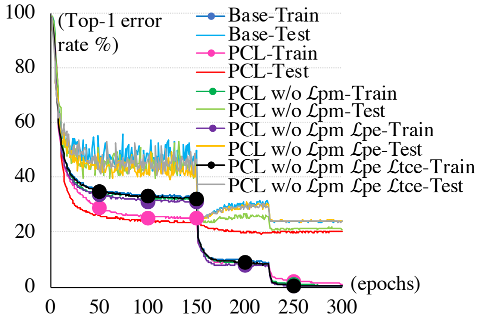


Figure 3: Component effectiveness comparison during training and testing with ResNet-110 on CIFAR-100.

approximately 1% and 2% on CIFAR-10 and CIFAR-100, respectively. This shows the effectiveness of PCL for improving the generalisation performance of various backbone networks. On CIFAR-10 and CIFAR-100, PCL achieves the best top-1 error rates compared with the state-of-the-art online distillation methods. For example, on CIFAR-10, PCL improves the state-of-the-arts by approximately 0.1% and 0.3% with ResNet-32 and ResNet-110, respectively; Whilst on CIFAR-100, PCL improves the state-of-the-arts by about 0.3% and 1.1% with ResNet-32 and ResNet-110, respectively. These improvements attribute to the integration of the peer mean teacher and the peer ensemble teacher into a unified framework. When extended to the large-scale ImageNet benchmark, as shown in Table 1, PCL improves the baseline by approximately 0.9% with ResNet-18. Compared with the state-of-the-art alternative methods, PCL still achieves competitive top-1 error rate (about 29.6% with ResNet-18).

Discussion. These results show the performance advantages of PCL for online KD. Note that with the peer ensemble teacher and the peer mean teachers, PCL requires extra computational cost but: (1) the inference cost of PCL is the same as the target backbone because we only use a single peer mean model as the target model for test; (2) the peer mean teachers are updated with Eq.(7) without the need to perform backpropagation (Tarvainen and Valpola 2017); (3) the peer ensemble teacher is a multi-branch model and the extra computational cost is mainly to train the additional classifier.

Component Effectiveness Evaluation

From Table 2, we can see that: (1) With all components, PCL (full model) achieves the best result, demonstrating the effectiveness of integrating of peer ensemble teacher and peer mean teacher into a unified framework for online KD. (2) Backbone+L_{ce}^{p}+L_{ce}^{t}+L_{pe} significantly improves the performance of Backbone by about 2.6%, showing the effectiveness of the peer ensemble teacher. (3) PCL (full model) improves Backbone+L_{ce}^{p}+L_{ce}^{t}+L_{pe} by about 1.1%, which indicates the effectiveness of the peer mean teacher. (4) Replacing P.E. or P.M. with some contemporary variants causes performance degradation, which demonstrates the superiority of the proposed PCL. Besides, from Fig. 3, we can see that PCL with all components (red curve) gets better generalisation. Interestingly, the test top-1 error rate (red curve) of PCL (full model) drops rapidly from 0 to 50 epochs, and then gradually reaches to the optimal value; In contrast, other methods (w/o L_{pm}) fluctuate dramatically, especially from 0 to 225 epochs. This shows the importance of the peer

[1] https://github.com/DefangChen/OKDDip-AAAI2020

[2] https://github.com/Lan1991Xu/ONE_NeurIPS2018

Method	CIFAR-10		CIFAR-100	
	Top-1	Param.	Top-1	Param.
ONE-E	4.75±0.27	**2.89M**	20.10±0.24	**2.96M**
FFL (fused)	4.99±0.07	3.10M	21.78±0.28	3.19M
OKDDip-E	4.79±0.12	2.91M	20.93±0.57	2.98M
PCL-E(ours)	**4.42±0.12**	2.90M	**19.49±0.49**	3.04M

Table 3: Ensemble effectiveness evaluation with ResNet-110 on CIFAR-10/100. Top-1 error rates (%) and the number of model parameters are reported.

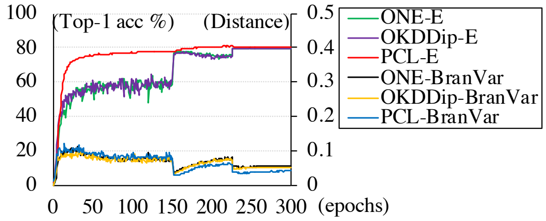


Figure 4: Peer variance for online ensembling analysis with ResNet-110 on CIFAR-100. '-BranVar': the branch variance. Here, we use top-1 accuracy for better visualisation.

mean teacher for learning richer knowledge among peers and optimising a more stable model.

Ensemble Effectiveness Evaluation

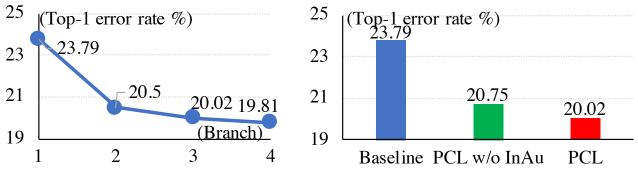
We compare our PCL-E with three online KD ensembles: ONE-E (ensemble of all branches) (Lan, Zhu, and Gong 2018), FFL (FFL-S with fused ensembles) (Kim et al. 2020), and OKDDip-E (ensemble of peers) (Chen et al. 2020). As shown in Table 3, PCL-E improves the state-of-the-arts by about 0.3% and 0.6% on CIFAR-10 and CIFAR-100, respectively. Besides, compared with ONE-E (the alternative method with the fewest model parameters), PCL-E achieves significantly better performance but only increases the number of model parameters by 0.01M and 0.08M with ResNet-110 on CIFAR-10 and CIFAR-100, respectively.

Peer Variance for Online Ensembling Analysis

In Fig. 4, we analyse the peer (branch) variance for online ensembling over the training epochs. Here, we computed the average Euclidean distance between the predictions of every two branches as the branch diversity and used the average diversity of m branches as the branch variance. From Fig. 4, we can see that: (1) From 0 to 150 epochs, the top-1 accuracy of PCL-E soars to a high level outperforming other methods, and meanwhile, the branch variance of PCL (PCL-BranVar) is larger than other methods. This indicates that at the early stage, although the generalisation capability of the model is poor, each branch in PCL collaborates better to facilitate online knowledge distillation. (2) From 150 to 300 epochs, the top-1 accuracy of PCL-E is still better than the alternatives, whilst the branch variance of PCL becomes smaller than the alternatives. The main reason is that at this stage, the generalisation of each peer is significantly improved and the temporally aggregated model of each peer becomes stable (with accurate and similar predictions). This also results in a stronger ensemble model (see Table 3) and a more generalised target model (see Table 1).

Dataset	Baseline	KD†	PCL
CIFAR-10	6.74±0.15	5.82±0.12	**5.67±0.12**
CIFAR-100	28.72±0.19	26.23±0.21	**25.86±0.16**

Table 4: Comparison with two-stage distillation with ResNet-32 on CIFAR-10/100. Top-1 error rates (%). †: Use ResNet-110 as the teacher model.



(a) Branch Number Impact

(b) Input Augmentation Impact

Figure 5: Evaluating the impact of (a) different number of branches and (b) input augmentation for PCL with ResNet-110 on CIFAR-100.

Further Analysis and Discussion

Comparison with Two-Stage Distillation. In Table 4, we compare PCL with the traditional two-stage KD (Hinton, Vinyals, and Dean 2015). We can see that although PCL does not pre-train a high-capacity teacher model (*e.g.* ResNet-110), it still achieves better performance than the two-stage KD. This attributes to the integration of the peer ensemble teacher and the peer mean teacher into a unified framework for online knowledge distillation.

Branch Number. As shown in Fig. 5(a), the performance of PCL improves when more branches are used. In the four-branch setting, PCL (19.8%) still performs competitively against OKDDip (21.1% as reported in (Chen et al. 2020)).

Input Augmentation. As shown in Fig. 5(b), without using multiple input augmentation (PCL w/o InAu), the performance of PCL decreases (by approximately 0.7%), but it still achieves compelling performance. This further verifies the effectiveness of the model design in PCL.

Conclusion

In this work, we presented a novel Peer Collaborative Learning (PCL) method for online knowledge distillation, which integrates online ensembling and network collaboration into a unified framework. We assembled feature representations from peers as the online high-capacity peer ensemble teacher and used the temporal mean model of each peer as the peer mean teacher. Doing so allows improving the quality of online knowledge distillation in a one-stage end-to-end trainable fashion. Extensive experiments with a variety of backbone networks show the superiority of the proposed method over the state-of-the-art methods on CIFAR-10, CIFAR-100 and ImageNet.

Acknowledgements

This work is supported by Vision Semantics Limited, Alan Turing Institute Turing Fellowship, and Innovate UK Industrial Challenge Project on Developing and Commercialising Intelligent Video Analytics Solutions for Public Safety (98111-571149), Queen Mary University of London Principal's Scholarship.

References

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Chen, Y.; Zhu, X.; and Gong, S. 2017. Person re-identification by deep learning multi-scale representations. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*, 2590–2600.

Ge, Y.; Chen, D.; and Li, H. 2020. Mutual Mean-Teaching: Pseudo Label Refinery for Unsupervised Domain Adaptation on Person Re-identification. In *International Conference on Learning Representations*.

Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; and Luo, P. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 11020–11029.

Hansen, L. K.; and Salamon, P. 1990. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(10): 993–1001.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Hou, S.; Liu, X.; and Wang, Z. 2017. Dualnet: Learn complementary features for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 502–510.

Huang, G.; Li, Y.; Pleiss, G.; Liu, Z.; Hopcroft, J. E.; and Weinberger, K. Q. 2017a. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017b. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708.

Kim, J.; Hyun, M.; Chung, I.; and Kwak, N. 2020. Feature Fusion for Online Mutual Knowledge Distillation. In *International Conference on Pattern Recognition*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report.

Laine, S.; and Aila, T. 2017. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*.

Lan, X.; Zhu, X.; and Gong, S. 2018. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, 7517–7527.

Mittal, S.; Tatarchenko, M.; and Brox, T. 2019. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Moghimi, M.; Belongie, S. J.; Saberian, M. J.; Yang, J.; Vasconcelos, N.; and Li, L.-J. 2016. Boosted convolutional neural networks. In *British Machine Vision Conference*.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3967–3976.

Phuong, M.; and Lampert, C. 2019. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, 5142–5151.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3): 211–252.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Song, G.; and Chai, W. 2018. Collaborative learning for deep neural networks. In *Advances in Neural Information Processing Systems*, 1832–1841.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 1195–1204.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1365–1374.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1492–1500.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4133–4141.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. In *British Machine Vision Conference*.

Zagoruyko, S.; and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4320–4328.

Zhou, Z.-H.; Wu, J.; and Tang, W. 2002. Ensembling neural networks: many could be better than all. *Artificial Intelligence* 137(1-2): 239–263.