Collaborative Group Learning

Shaoxiong Feng,[1] **Hongshen Chen,**[2] **Xuancheng Ren,**[3] **Zhuoye Ding,**[2] **Kan Li,**[1] **Xu Sun**[3,4]

[1]School of Computer Science & Technology, Beijing Institute of Technology
[2]JD.com
[3]MOE Key Laboratory of Computational Linguistics, School of EECS, Peking University
[4]Center for Data Science, Peking University
{shaoxiongfeng, likan}@bit.edu.cn, {renxc, xusun}@pku.edu.cn
ac@chenhongshen.com, dingzhuoye@jd.com

Abstract

Collaborative learning has successfully applied knowledge transfer to guide a pool of small student networks towards robust local minima. However, previous approaches typically struggle with drastically aggravated student homogenization when the number of students rises. In this paper, we propose Collaborative Group Learning, an efficient framework that aims to diversify the feature representation and conduct an effective regularization. Intuitively, similar to the human group study mechanism, we induce students to learn and exchange different parts of course knowledge as collaborative groups. First, each student is established by randomly routing on a modular neural network, which facilitates flexible knowledge communication between students due to random levels of representation sharing and branching. Second, to resist the student homogenization, students first compose diverse feature sets by exploiting the inductive bias from subsets of training data, and then aggregate and distill different complementary knowledge by imitating a random subgroup of students at each time step. Overall, the above mechanisms are beneficial for maximizing the student population to further improve the model generalization without sacrificing computational efficiency. Empirical evaluations on both image and text tasks indicate that our method significantly outperforms various state-of-the-art collaborative approaches whilst enhancing computational efficiency.

Introduction

Deep neural network has achieved impressive performance in various fields. Combining multiple individual networks, an ensemble model gains better predictive performance than a single network. One important reason is that an ensemble model usually aggregates a robust local minimum rather than a sharp local minimum that a single model may be stuck in. To alleviate the prohibitive computational cost of those high-capacity ensemble networks, Knowledge Distillation (KD) method is therefore proposed to achieve more compact yet accurate models by transferring knowledge (Ba and Caruana 2014; Romero et al. 2015; Hinton, Vinyals, and Dean 2015; Han, Mao, and Dally 2016). KD comprises two pipelined learning stages, a pre-training stage and a knowledge transfer stage. Recently, attempts on group-based online knowledge distillation, also known as collaborative

learning, explore less costly and unified models to eliminate the necessity of pre-training a large teacher model (Zhang et al. 2018; Anil et al. 2018), where a group of students simultaneously discovers knowledge from the ground-truth labels and distills group-level knowledge (multi-view feature representation) from each other.

Collaborative learning shares the benefits of finding a more robust local minimum than a single model learning while accelerating the model learning efficiency compared with conventional KD. In terms of the implementation of student networks in collaborative learning, DML (Zhang et al. 2018) uses a pool of network-based students, where each student is an individual network and they asynchronously collaborate, whereas CL-ILR (Song and Chai 2018) proposed branch-based collaborative learning that all the student networks share the bottom layers while dividing into branches in the upper layers. Benefiting from representation sharing (an extreme form of hint training (Romero et al. 2015)), CL-ILR not only is more compact and efficient but also shows better generalization performance. As observed in (Zhang et al. 2018; Song and Chai 2018; Lan, Zhu, and Gong 2018; Chen et al. 2020), the model performance continually improves along with the increasing number of students.

However, the students in collaborative learning tend to homogenize, damage the generalization ability, and degrade to the original individual network. Although the students are randomly initialized, they learn from the same entire training set and are prone to converging to similar feature representations (Li et al. 2016; Morcos, Raghu, and Bengio 2018). Moreover, each student distills knowledge from all other students, which further aggravates the homogenization problem due to ignoring the diversity of students' group-level knowledge (Schwenker 2013; Lan, Zhu, and Gong 2018; Chen et al. 2020). Another insurmountable obstacle for collaborative learning is that the computational cost boosts greatly as well when more students join in collaborative learning.

To overcome these challenges, in this paper, we propose a collaborative group learning framework that improves and maintains the diversity of feature representation to conduct an effective regularization. Intuitively, under the spirit of knowledge distillation by learning as collaborative groups, we divide the whole course into multiple segments and assign students into several non-isolated sub-groups. Each

student learns one piece of course and grasps the whole course through efficient knowledge distillation. Specifically, we *first* introduce a conceptually novel method, called **random routing**, to build student networks, where each student is regarded as a group of network modules, and the connections between network modules are established as randomly routing a modular neural network. After randomly routing, we break the limitation of sharing representation only at the bottom layers and extend its range to any layer. Modules are shared by different involved students, which facilitates knowledge sharing and distillation between students. *Second*, to tackle the student homogenization problem aggravating with the increasing number of students, **sub-set data learning** is proposed for each student to learn different parts of the training set. It increases the model diversity by introducing the inductive bias of the data subset into the student training. Moreover, to compensate for the knowledge (training data) loss of individual students while maintaining the student diversity, we further propose **sub-group imitation**, where a sub-group of students is randomly selected and assigned to aggregate group-level knowledge in each iteration, rather than aggregating knowledge from all other students as in previous approaches. It allows a student to internalize dynamic and evolving group-level knowledge while adjusting the sub-group size for adapting to various computational environments. In addition to be collaboratively devoted to the student homogenization problem so as to regularize the feature learning effectively, the above three mechanisms also enhance computational efficiency (e.g., reduce the number of parameters and the number of forward and backward propagation), which means our framework can maximize the student population to further improve the model generalization under restricted computational resources.

In summary, our contributions are as follows: 1) Collaborative group learning strikes a better balance between diversifying feature representation and enhancing knowledge transfer to induce students toward robust local minima. 2) Random routing builds students by randomly connecting the module path, which enables random levels of representation sharing and branching. 3) To overcome the student homogenization problem, sub-set data learning first draws on the inductive bias from the data subset to improve the model diversity. Then, sub-group imitation further transfers supplementary knowledge from a random and dynamic sub-group of students, maintaining the diversity of students. 4) Besides, the proposed three mechanisms are computationally efficient, allowing more students to join in collaboration and further boost the model generalization. We conducted detailed analyses to verify the advantages of our framework on generalization, computational cost, and scalability.

Method

Compared with previous approaches, collaborative group learning has the superiority in generalization performance and computational efficiency. In this section, we first elaborate how to build students using **random routing**, then introduce **sub-set data learning** and **sub-group imitation** to discover and transfer diverse knowledge effectively, and finally present the training objective.
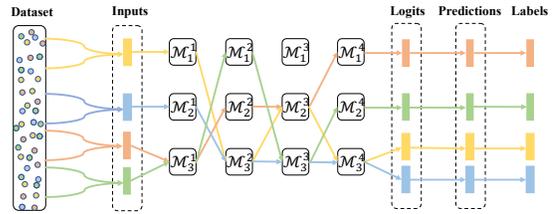

Figure 1: *An overview of collaborative group learning with random routing and sub-set data learning.*

Random Routing for Student Network

Previous work constructs a pool of students by continually introducing new networks or branches, which rapidly expands the model capacity with the increasing number of students. In our proposed collaborative learning, students are built from a modular neural network, using random routing, where students can be built as many as possible given restricted-capacity networks. More importantly, random module sharing and branching also benefit knowledge interaction between students. The modular neural network consists of L distinct layers with each layer $\ell \in [1, L]$ containing M modules, arranged in parallel, i.e., $\mathcal{M}^\ell = \left\{ \mathcal{M}_m^\ell \right\}_{m=1}^M$ (see Figure 1). Each module $\mathcal{M}_m^\ell$ is a learnable sub-network embedded in the modular neural network, consisting of different combinations of layers. It extracts different types of features in accordance with various tasks, such as residual block for vision features and transformer block for semantic features. For the ℓth layer, the index of the selected module is uniformly sampled using $\mathbb{U}(\cdot)$ over the set of integers $[1, M]$. After L times of selection, we construct a pathway $\mathbf{P}_k \in \mathbb{R}^{L \times M}$ to form a random routing network for the kth student:

$$\mathbf{P}_k(\ell, m) = \begin{cases} 1, & \text{if the module } \mathcal{M}_m^\ell \text{ is present in the path,} \\ 0, & \text{otherwise.} \end{cases}$$
(1)

When training the kth student, any m^{th} module in ℓ^{th} layer with $\mathbf{P}_k(\ell, m) = 1$ is activated. All established students are simultaneously trained by two supervised losses that we will elaborate later. With the help of random routing, one student can share the selected module of any other student in the same layer, which means that compared with previous alternative work, our method can build more students with the same number of parameters. Also, more students participating in collaboration imply more diverse feature sets in the student pool. Meanwhile, the fine-grained representation sharing and branching across multi-layers between students, as an extreme form of hint training (Romero et al. 2015), implicitly and flexibly boost knowledge sharing and transfer. Consequently, it naturally imposes efficient regularization on the feature learning for each student (Song and Chai 2018; Lan, Zhu, and Gong 2018).

Sub-set Data Learning

In prior collaborative approaches, all students learn from the same set of training data. The inductive bias contained in the training data significantly affects the features learning (Zhang, Wang, and Zhu 2018), facilitates or hinders the

model training (Such et al. 2020). We consider utilizing the inductive bias of training data to enrich the diversity of students and propose a sub-set data learning. Concretely, given N samples $\mathcal{X} = \{x_i\}_{i=1}^{N}$ from C classes with the corresponding labels $\mathcal{Y} = \{y_i\}_{i=1}^{N}$, where $y_i \in \{1, 2, \ldots, C\}$, the entire training set is randomly divided into K subsets $\mathcal{X}^k = \{x_i^k\}_{i=1}^{N/K}$ to train the corresponding K students (see Figure 1). The kth student produces the probability of class c for sample x_i^k by normalizing the logit,

$$p_k^c\left(x_i^k\right) = \frac{\exp\left(z_k^c\right)}{\sum_{j=1}^{C} \exp\left(z_k^j\right)} \tag{2}$$

where the logit z_k is the output of the kth student.

As a multi-class classifier, the general training criterion of the kth student is to minimize the cross-entropy between the ground-truth labels and the predicted distributions,

$$L_{ce}^k = -\sum_{i=1}^{N/K}\sum_{c=1}^{C} \mathbb{I}\left(y_i^k = c\right) \log\left(p_k^c\left(x_i^k\right)\right) \tag{3}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

Sub-group Imitation

Conventional collaborative learning usually introduces extra supplementary information in the form of group-level knowledge. However, aggregating group-level knowledge with the naive or the weighted average faces two main drawbacks: First, the homogenization phenomenon is more likely to occur due to the similar and redundant group-level knowledge of students; Besides, the computational cost increase linearly as the number of students continually grows. In order to improve the generalization of each student, we propose a sub-group imitation (see Figure 2), which randomly selects a sub-group instead of the whole group of students for imitating in each iteration. Intuitively, in our collaborative framework, each student follows a dynamic and evolving "teacher" to gain experiences while learns to denoise the random perturbation of soft knowledge (prediction alignment) and hard knowledge (parameter sharing) that alleviates student homogenization but hinders the stability of student learning. In practice, we can adjust the sub-group size flexibly to balance the performance and training computational cost. The group-level knowledge for the kth student is computed as:

$$L_{kl}^k = \sum_{i=1}^{N/K}\sum_{c=1}^{C} p_t^c\left(x_i^k; T\right) \log \frac{p_t^c\left(x_i^k; T\right)}{p_k^c\left(x_i^k; T\right)};$$
$$p_t^c\left(x_i^k; T\right) = \frac{\exp\left(z_t^c/T\right)}{\sum_{j=1}^{C} \exp\left(z_t^j/T\right)} \tag{4}$$

where T is the temperature, used to soften the predictions, and z_t is defined as:

$$z_t = \frac{1}{H}\sum_{k=1}^{K} Select\left(z_k\right) \tag{5}$$

where H is the expected number of imitated students, and $Select(\cdot)$ is the selection function with an imitating probability p. Note that we first select which students to imitate and then calculate the corresponding z_k in practice.
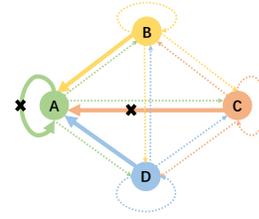

Figure 2: *Sub-group imitation*. For example, with the imitating probability $p = 0.5$, student A may only choose student B and D to aggregate the group-level knowledge for one iteration.

Optimization

We obtain the overall loss function as:

$$L = \sum_{k=1}^{K}\left(L_{ce}^k + \phi(t) * L_{kl}^k\right), \tag{6}$$

where $\phi(t)$ is a ramp-up coefficient function (Laine and Aila 2017) that maintains an equilibrium of the contribution of labels and group-level knowledge. The imbalance of contribution will result in either exacerbating the homogenization of students or weaken knowledge transfer between students. The ramp-up coefficient function can prevent students from getting prematurely stuck in the homogenization problem, which causes that students can not learn enough diverse knowledge to regularize each other effectively.

$$\phi(t) = \begin{cases} 1, & \text{if } t \text{ not in } [J^s, J^e], \\ \exp\left(-5 * (1-\lambda)^2\right), & \text{otherwise.} \end{cases} \tag{7}$$

where t is the index of training epoch, and λ is a scalar that increases linearly from zero to one during the ramp-up range $[J^s, J^e]$. Once a pool of students are collected from the proposed modular neural network, and the training set is randomly divided, we conduct the sub-group imitation throughout the whole training process. All students are trained simultaneously at each iteration until convergence. In inference, we can randomly select one student or choose the best student by a hold-out set to predict the class of input data.

Experiments

Datasets and Architectures

We present our results on six public available datasets of three classification tasks covering image classification, topic classification, and sentiment analysis. To validate the effectiveness of the proposed collaborative group learning framework in depth, we conduct the evaluation tasks in various tasks ranging from image field to more challenging text classifications, especially the fine-grained sentiment analysis tasks. Table 1 summarizes the statistics of all datasets. For the image-related tasks, we adopt augmentation and normalization procedure following (He et al. 2016). For the text-related tasks, following (Conneau et al. 2017), we do not conduct any preprocessing except lower-casing. Four network architectures are used in our experiments for different tasks, ResNet-18 and ResNet-34 (He et al. 2016) for CIFAR-10 and CIFAR-100, Transformer (Vaswani et al. 2017) for

Dataset	# Train	# Holdout	# Test	# Classes	Classification Task
CIFAR-10 (Krizhevsky 2009)	45k	5k	10k	10	Image classification
CIFAR-100 (Krizhevsky 2009)	45k	5k	10k	100	Image classification
IMDB Review (Maas et al. 2011)	23k	2k	25k	2	Sentiment analysis
Yelp Review Full (Zhang, Zhao, and LeCun 2015)	630k	20k	50k	5	Sentiment analysis
Yahoo! Answers (Zhang, Zhao, and LeCun 2015)	1 350k	50k	60k	10	Topic classification
Amazon Review Full (Zhang, Zhao, and LeCun 2015)	2 900k	100k	650k	5	Sentiment analysis

Table 1: Statistics of six classification datasets used in our experiments.

Datasets	Baseline	DML	CL-ILR	ONE	OKDDip	CGL
ResNet-18						
CIFAR-10	93.97 ± 0.09	94.18 ± 0.09	94.11 ± 0.12	94.19 ± 0.06	94.29 ± 0.04	$\mathbf{94.61 \pm 0.06}$
CIFAR-100	74.68 ± 0.13	76.13 ± 0.10	76.61 ± 0.03	76.17 ± 0.12	76.69 ± 0.04	$\mathbf{78.01 \pm 0.07}$
ResNet-34						
CIFAR-100	76.06 ± 0.11	76.73 ± 0.12	77.09 ± 0.12	76.96 ± 0.10	77.39 ± 0.09	$\mathbf{78.31 \pm 0.10}$

Table 2: Top-1 accuracy (%) on the image datasets.

IMDB Review, and VDCNN-9 (Conneau et al. 2017) for the rest of datasets.

Comparison Approaches

We compare Collaborative Group Learning (CGL) to several recently proposed collaborative approaches, including network-based DML (Zhang et al. 2018), branch-based CL-ILR (Song and Chai 2018), ONE (Lan, Zhu, and Gong 2018), and OKDDip (Chen et al. 2020). We also report a "Baseline" model that trains only one student on ground-truth labels. For branch-based approaches, all students share the first several blocks of layers and separate from the last block to form a multi-branch structure as (Lan, Zhu, and Gong 2018). The students in all the comparison models are set to the same amount and architecture for different tasks, i.e., 3 students for image datasets and 5 students for text datasets. The number of parameters increases as more students join in, whereas in our method, given 9 layers of modules and 2 modules in each layer, with random routing, theoretically we can build 2 to 512 different students without extra computational cost. In our experiment, we set 8 students for collaborative group learning. The imitating probability is set to 0.25 for image tasks and 0.5 for text tasks. The student that obtains the best score on the holdout set is used for evaluation. In OKDDip (Chen et al. 2020), the group leader student is chosen for prediction.

Experiment Settings

For ResNet-18 and ResNet-34, we use Adam (Kingma and Ba 2015) for optimization with a mini-batch of size 64. The initial learning rate is 0.001, divided by 2 at 60, 120, and 160 of the total 200 training epochs. For VDCNN-9, we adopted the same experimental settings as (Conneau et al. 2017; Zhang, Zhao, and LeCun 2015). Training is performed with Adam, using a mini-batch of size 64, a learning rate of 0.001 for the total 20 training epochs. We use Sentence-Piece[1] (BPE) to tokenize IMDB Review and set vocabulary size, embedding dimension, and maximum sequence length

to 16000, 512, and 512. For Transformer, the size of blocks and heads is 3 and 4 separately. We set the size of the hidden state and feed-forward layer to 128 and 512. Training is performed with Adam, using a mini-batch of size 64, a learning rate of 0.0001 for the total 30 training epochs. We run each method 3 times and report "mean (std)".

Comparison on Image Classification

Table 2 summarises the Top-1 accuracy (%) of CIFAR-10 and CIFAR-100 obtained by ResNet-18 and ResNet-34 with the existing state-of-the-art and our methods. We observe that our method significantly outperforms all other methods with substantial accuracy gains, which shows that with the same computational cost, our collaborative framework is more effective than previous methods on improving model generalization. The branch-based methods, especially OKDDip, yield more generalizable models compared to the network-based method (DML). This suggests parameter sharing benefits the transfer of diverse and complementary knowledge between students as observations in (Song and Chai 2018; Lan, Zhu, and Gong 2018). We also found that all collaborative frameworks achieve more performance improvement in the smaller architecture according to the results of ResNet-18 and ResNet-34 on CIFAR-100.

Comparison on Text Classification

Table 3 reports the Top-1 accuracy (%) of all text datasets based on VDCNN-9 and Transformer. It can be seen that our method also achieves better performance than all prior methods as above, indicating that our method can be generically applied to more challenging text classification tasks. The prior methods obtain slightly better performance than "Baseline" on all datasets except for Yahoo! Answers (Topic classification), which means that the difficulty of clearly discriminate fine-grained sentiment labels hinders students from discovering diverse feature sets and transferring supplementary knowledge from the others. The superiority of our method on both image and text datasets demonstrates the generalization and robustness of the proposed collaborative framework.

---

[1] https://github.com/google/sentencepiece

Datasets	Baseline	DML	CL-ILR	ONE	OKDDip	CGL
			VDCNN-9			
Yelp Review Full	62.15 ± 0.15	62.53 ± 0.10	62.66 ± 0.08	62.74 ± 0.05	62.75 ± 0.18	$\mathbf{63.32 \pm 0.04}$
Yahoo! Answers	69.02 ± 0.07	69.79 ± 0.11	70.09 ± 0.07	70.08 ± 0.09	70.10 ± 0.09	$\mathbf{70.35 \pm 0.05}$
Amazon Review Full	60.25 ± 0.11	60.54 ± 0.10	60.59 ± 0.07	60.49 ± 0.03	60.63 ± 0.04	$\mathbf{61.03 \pm 0.05}$
			Transformer			
IMDB Review	82.30 ± 0.10	82.45 ± 0.05	83.10 ± 0.07	82.66 ± 0.08	82.74 ± 0.12	$\mathbf{83.81 \pm 0.10}$

Table 3: Top-1 accuracy (%) on the text datasets.

Condition	w/o RR	w/o SDL	w/o SGI
Accuracy (%)	75.75	77.48	77.23

Table 4: Results of the ablation study.

Ramp-up Range (%)	0	20	40	80
Accuracy (%)	77.79	78.01	77.22	77.10

Table 5: Impact of the Ramp-up coefficient.

Dataset (Architecture)	Condition	Diversity
CIFAR-100 (ResNet-18)	(w/.)	0.535
	(w/o.)	0.178
Yelp Review Full (VDCNN-9)	(w/.)	0.185
	(w/o.)	0.131
IMDB Review (Transformer)	(w/.)	0.056
	(w/o.)	0.039

Table 6: Effect of sub-set data learning.

Ablation Study and Analysis

In this section, we further investigate the effectiveness and robustness of our method, including random routing, sub-set data learning, and sub-group imitation. We also provide detailed analyses to demonstrate how and why our method works. We conduct ablation comparisons with the branch-based approaches, as they have the advantages of better performance and lower computational cost. The score reported below is all obtained by running each model 3 times and providing "mean". Unless otherwise stated, the following results are based on CIFAR-100 with ResNet-18.

Ablation Study　In Table 4, we report the Top-1 accuracy (%) of models "w/o random routing (RR)" (i.e., built by individual networks), "w/o sub-set data learning (SDL)" (i.e., using the same entire training data), and "w/o sub-group imitation (SGI)" (i.e., imitating all other students). According to the results, we can see that 1) without the parameter sharing generated by random routing, students, only based on the assigned sub-set data and the logits-based imitation, do not obtain sufficient information for the feature learning, which demonstrates that random routing indeed brings a highly effective knowledge transfer. 2) without sub-set data learning or sub-group imitation, the student homogenization will aggravate and the model may converge to a worse sub-optimal. These phenomena verify that the reason that CGL works well is the ***effective balance*** of conducting knowledge transfer and maintaining the model diversity.

To analyze the impact of the ramp-up coefficient, we set the ramp-up interval to 0%, 20%, 40% and 80% of the training epochs. The results of Table 5 indicate the ramp-up coefficient can alleviate the homogenization problem but too long ramp-up ranges weaken knowledge transfer.

Impact of Student Population　It is well known that the increasing number of students benefits the model performance (Lan, Zhu, and Gong 2018; Chen et al. 2020). Fig-
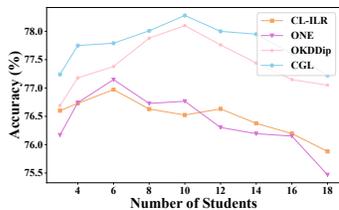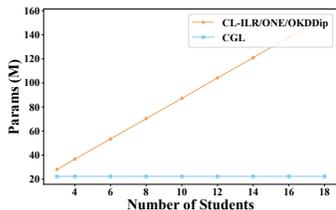
ure 3(a) shows the Top-1 accuracy (%) of all comparative methods with respect to the number of students. Our method consistently achieves the best accuracy in varying numbers of students, which demonstrates its superiority. We also observe that the curves of all methods rise first and then decline, which implies that excessive students also damage the model performance. We conjecture that excessive students may not be sufficiently trained due to weakening knowledge discovery of each student (for our methods) or exacerbating the similarity and redundancy of group-level knowledge (for comparative methods). In such a scenario, our method is undergoing under-fitting, whereas comparative methods are usually struggling with over-fitting (too many students homogenizing). Such an under-fitting problem can be simply adjusted by allowing sub-set data overlapping or increasing imitation probability. The performance of CI-ILR, ONE declines much earlier than OKDDip and CGL, which indicates that the latter effectively handles more diverse students.

However, more students significantly increase the number of model parameters and the computational cost of training (i.e., **computational efficiency**), which limits the deployment of collaborative learning. Our method can alleviate these problems by random routing and sub-group imitation. In terms of model parameters, as shown in Figure 3(b), our method remains a constant number of model parameters as more students join in, whereas the parameter number in comparison models explodes. Moreover, our method also maintains constant computational cost (the number of forward and backward propagation) by variable imitation probability. As for the training time, similar to codistillation (a variant of DML) (Anil et al. 2018), our method can be easily implemented in parallel. Therefore it still consumes the constant training time as the number of students increases. Please refer to the appendix for a more detailed discussion.

Model Diversity Analysis　We improve the model diversity in collaborative learning from two aspects: sub-set data learning diversifies the training sets of students to learn di-

(a) Comparative performances

(b) Number of parameters

Figure 3: Impact of student population. The parameter number of comparative methods explodes as more students are involved, while our method maintains a constant computational cost.
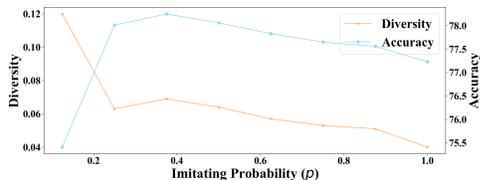

Figure 4: Effect of imitating probability.

CIFAR-10								
Architecure	1	2	3	4	5	6	7	8
Score	**94.55**	94.50	94.46	94.48	94.45	**94.53**	**94.63**	94.45
Rank	**2**	4	6	5	7	**3**	**1**	7
CIFAR-100								
Architecure	1 (**1**)	2	3	4	5 (**6**)	6	7	8 (**7**)
Score	**77.94**	77.82	77.93	77.87	**77.97**	77.81	77.88	**78.15**
Rank	**3**	7	4	6	**2**	8	5	**1**

Table 7: Transfer of parameter sharing structure. Architecture: ResNet-18. "(#)" is corresponding to the index of architectures on CIFAR-10.

Impact of Parameter Sharing Besides considering the logits of the output layer, parameter sharing is also an implicit and efficient way to boost knowledge transfer by aligning the intermediate features between selected students (Song and Chai 2018; Lan, Zhu, and Gong 2018). Network-based collaborative learning does not support parameter sharing, while branch-based one shares parameters only at the bottom layers. Benefited from random routing, our method naturally allows flexible knowledge communication based on fine-grained and random levels of parameter sharing structure. We first investigate the effect of parameter sharing ratio on model performance. We fix the number of students and the imitating probability, and then vary the parameter sharing ratio by adjusting the number of modules per layer or manually setting shared layers. From Figure 5, we discover that for collaborative learning, sharing too many layers will cause students to homogenize, and sharing too few layers weaken knowledge transfer, which implies previous parameter sharing structure is not flexible enough to maintain a trade-off between diversity and generalization of students due to dense and consecutive multi-layer parameter sharing.

Similar to neural architecture search (Zoph and Le 2017), our collaborative framework is capable of finding an efficient and generalizable parameter sharing structure. Concretely, one can collect a set of parameter sharing structures by random routing, and then select the best structure that can be applied directly to a new dataset. We validate this assumption by randomly generating eight parameter sharing structures on CIFAR-10 based on ResNet-18, and then choosing the best three structure with another five randomly formed structure to train and test on CIFAR-100. The results in Table 7 show that the top 3 structures in CIFAR-10 also obtain the top 3 performances in CIFAR-100, which verifies the generalization of naturally formed parameter sharing structure. Compared to manually designed parameter sharing structure, our method is obviously more efficient.

verse feature sets; sub-group imitation randomly selects various sub-groups of imitated students from which students aggregate and distill different complementary knowledge. Table 6 reports the diversity of students $w.$ and $w/o.$ subset data learning based on a comprehensive set of architectures and datasets. The diversity is calculated by averaging L_2 distance between the probability distribution of each pair of students. To isolate the effect of sub-set data learning, we revoke sub-group imitation in this analysis. The results in Table 6 verify sub-set data learning indeed boosts the diversity of students on various architectures and datasets.

We further vary the imitating probability p to analyze its effect on diversity and accuracy. The imitating probability p is selected from $[0.125, 1.0]$ and applied to ResNet-18 on CIFAR-100. From Figure 4, we discover that the diversity shows a downward trend, and the accuracy first ascends and then slowly declines. This phenomenon demonstrates that when each student aggregates knowledge from too many students, the model performance declines as they may homogenize to each other; however, when it mimics very few students, it is unable to distill a sufficient amount of knowledge. Our method achieves a better balance between diversity and performance with limited computational resources by choosing a proper imitating probability.

Model Generalization Analysis We demonstrate why our collaborative group learning obtains better generalization than comparison methods. Recently, a collection of work (Chaudhari et al. 2017; Keskar et al. 2017) has proved that comparing a wider local minimum with a narrow one, the former is more beneficial for the model resisting small perturbations that dramatically damage the model accuracy. In-
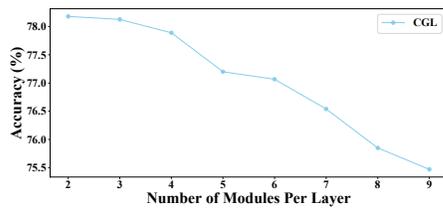
(a) Autonomous parameter sharing
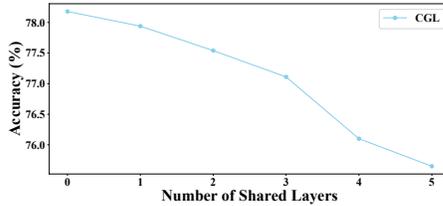
(b) Enforced parameter sharing

Figure 5: Impact of parameter sharing ratio. We can sparse parameter sharing by increasing the number of modules per layer or densify parameter sharing by manually setting more shared layers in which all students choose the same module.
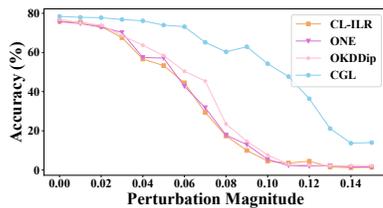

Figure 6: Model generalization analysis.

spired by this insight, we manually inject perturbations into the models to measure the width of local minima reached by all methods. Specially, we first generate different magnitudes of perturbations drawn from independent Gaussian distribution with variable standard deviation σ, and then add them to the model parameters. In Figure 6, we plot the accuracy drop under different perturbation magnitudes. We can see that the accuracy of comparison methods declines much faster when the perturbation magnitude becomes larger. In contrast, our method is more stable, reflecting that collaborative group learning imposes effective regularization on the feature learning and guides the model towards a wider local minimum.

Related Work

Knowledge Distillation To deploy high-performance neural networks on mobile devices and embedded systems, Knowledge Distillation (KD) (Bucila, Caruana, and Niculescu-Mizil 2006; Ba and Caruana 2014; Hinton, Vinyals, and Dean 2015; Romero et al. 2015) has been proposed to transfer fine-grained and hierarchical knowledge from a pre-trained large model (teacher) to a small model (student) by aligning the predictions or intermediate features of teacher and student. The student not only obtains similar performance as the teacher but also is easily deployed

to the limited computation environment. Recently, several works (Zagoruyko and Komodakis 2017; Yim et al. 2017; Srinivas and Fleuret 2018; Ahn et al. 2019) try to design new forms of teacher-learned knowledge or feature matching loss to facilitate knowledge transfer. KD suffers from pre-training a large teacher, which consumes more computational resources and training time; whereas we resort to collaborative learning and distill knowledge from a random sub-group of peer students.

Collaborative Learning Collaborative learning (Zhang et al. 2018; Song and Chai 2018; Lan, Zhu, and Gong 2018; Chen et al. 2020) is more lightweight than KD in terms of learning stages. It facilitates each student to find a robust local minimum to achieve better generalization performance (Chaudhari et al. 2017; Keskar et al. 2017) in comparison to KD. Currently, there are two mainstream implementations of student networks. One is network-based (Zhang et al. 2018), where students are independent networks, and the parameter capacity increases linearly with the number of students; the other is branch-based (CL-ILR (Song and Chai 2018) and ONE (Lan, Zhu, and Gong 2018)), where the bottom layers of students are shared. In our framework, we enable more flexible representation sharing with random routing mechanism (Fernando et al. 2017; Rajasegaran et al. 2019), where layers at any level can be shared by different involved students. More importantly, students can be constructed as many as possible under restricted computational resources, whereas previous collaborative learning approach is more resource-intensive.

In terms of knowledge distillation in collaborative learning, OKDDip (Chen et al. 2020) aggregates knowledge of all the students through weighted average. In contrast, we alleviate the student homogenization and enhance the model generalization ability by distilling knowledge from a random and dynamic sub-group of students, and each student learns different parts of the training data.

Conclusion

In this work, we present a novel knowledge distillation-based learning paradigm, collaborative group learning, which obtains better generalization performance and consumes lower computational cost than prior collaborative approaches. Specifically, adopting random routing to build students not only is more parameter-efficient but also enables flexible knowledge communication between students. Besides, building more students promises more diverse knowledge at the beginning of training. To alleviate the student homogenization problem during training, sub-set data learning is introduced to diversify the feature sets of students, and sub-group imitation further boosts the diversity of group-level knowledge as well as enhances computational efficiency. Overall, our framework generates dynamic and diverse multi-view representations for the same input that effectively regularize the feature learning. Extensive experiments validate the effectiveness and robustness of our framework, and detailed analysis further proves that maintaining a balance between diversifying feature sets and internalizing group knowledge is essential for collaborative learning.

Acknowledgements

References

Ahn, S.; Hu, S. X.; Damianou, A. C.; Lawrence, N. D.; and Dai, Z. 2019. Variational Information Distillation for Knowledge Transfer. In *CVPR*, 9163–9171.

Anil, R.; Pereyra, G.; Passos, A.; Ormándi, R.; Dahl, G. E.; and Hinton, G. E. 2018. Large scale distributed neural network training through online distillation. In *ICLR (Poster)*.

Ba, J.; and Caruana, R. 2014. Do Deep Nets Really Need to be Deep? In *NIPS*, 2654–2662.

Bucila, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *KDD*, 535–541.

Chaudhari, P.; Choromanska, A.; Soatto, S.; LeCun, Y.; Baldassi, C.; Borgs, C.; Chayes, J. T.; Sagun, L.; and Zecchina, R. 2017. Entropy-SGD: Biasing Gradient Descent Into Wide Valleys. In *ICLR (Poster)*.

Chen, D.; Mei, J.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. In *AAAI*, 3430–3437.

Conneau, A.; Schwenk, H.; Barrault, L.; and LeCun, Y. 2017. Very Deep Convolutional Networks for Text Classification. In *EACL (1)*, 1107–1116.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. PathNet: Evolution Channels Gradient Descent in Super Neural Networks. *CoRR* abs/1701.08734.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531.

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. In *ICLR*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Krizhevsky, A. 2009. *Learning multiple layers of features from tiny images*. Master's thesis, University of Toronto.

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *ICLR (Poster)*.

Lan, X.; Zhu, X.; and Gong, S. 2018. Knowledge Distillation by On-the-Fly Native Ensemble. In *NeurIPS*, 7528–7538.

Li, Y.; Yosinski, J.; Clune, J.; Lipson, H.; and Hopcroft, J. E. 2016. Convergent Learning: Do different neural networks learn the same representations? In *ICLR*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *ACL*, 142–150.

Morcos, A. S.; Raghu, M.; and Bengio, S. 2018. Insights on representational similarity in neural networks with canonical correlation. In *NeurIPS*, 5732–5741.

Rajasegaran, J.; Hayat, M.; Khan, S. H.; Khan, F. S.; and Shao, L. 2019. Random Path Selection for Continual Learning. In *NeurIPS*, 12648–12658.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. FitNets: Hints for Thin Deep Nets. In *ICLR (Poster)*.

Schwenker, F. 2013. Ensemble Methods: Foundations and Algorithms [Book Review]. *IEEE Comput. Intell. Mag.* 8(1): 77–79.

Song, G.; and Chai, W. 2018. Collaborative Learning for Deep Neural Networks. In *NeurIPS*, 1837–1846.

Srinivas, S.; and Fleuret, F. 2018. Knowledge Transfer with Jacobian Matching. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, 4730–4738.

Such, F. P.; Rawal, A.; Lehman, J.; Stanley, K. O.; and Clune, J. 2020. Generative Teaching Networks: Accelerating Neural Architecture Search by Learning to Generate Synthetic Training Data. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, 9206–9216.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In *CVPR*, 7130–7138.

Zagoruyko, S.; and Komodakis, N. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *ICLR (Poster)*.

Zhang, Q.; Wang, W.; and Zhu, S. 2018. Examining CNN Representations With Respect to Dataset Bias. In *AAAI*, 4464–4473.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *NIPS*, 649–657.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep Mutual Learning. In *CVPR*, 4320–4328.

Zoph, B.; and Le, Q. V. 2017. Neural Architecture Search with Reinforcement Learning. In *ICLR*.