Progressive Network Grafting for Few-Shot Knowledge Distillation

Chengchao Shen,[1] **Xinchao Wang,** [2] **Youtan Yin,** [1] **Jie Song,** [1] **Sihui Luo,** [1]
Mingli Song [1, 3*]
[1] Zhejiang University
[2] Stevens Institute of Technology
[3] Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies
{chengchaoshen, youtanyin, sjie, sihuiluo829, brooksong}@zju.edu.cn, xinchao.wang@stevens.edu

Abstract

Knowledge distillation has demonstrated encouraging performances in deep model compression. Most existing approaches, however, require massive labeled data to accomplish the knowledge transfer, making the model compression a cumbersome and costly process. In this paper, we investigate the practical *few-shot* knowledge distillation scenario, where we assume only a few samples without human annotations are available for each category. To this end, we introduce a principled dual-stage distillation scheme tailored for few-shot data. In the first step, we graft the student blocks one by one onto the teacher, and learn the parameters of the grafted block intertwined with those of the other teacher blocks. In the second step, the trained student blocks are progressively connected and then together grafted onto the teacher network, allowing the learned student blocks to adapt themselves to each other and eventually replace the teacher network. Experiments demonstrate that our approach, with only a few unlabeled samples, achieves gratifying results on CIFAR10, CIFAR100, and ILSVRC-2012. On CIFAR10 and CIFAR100, our performances are even on par with those of knowledge distillation schemes that utilize the full datasets. The source code is available at https://github.com/zju-vipa/NetGraft.

Introduction

Deep neural networks have been widely applied to various computer vision tasks, such as image classification (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016), semantic segmentation (Long, Shelhamer, and Darrell 2015; Chen et al. 2018; Badrinarayanan, Kendall, and Cipolla 2017), and object detection (Ren et al. 2015; Liu et al. 2016; Zhang et al. 2018). The state-of-the-art deep models, however, are often cumbersome in size and time-consuming to train and run, which precludes them from being deployed in the resource-critical scenarios such as Internet of Things (IoT).

To this end, many model compression methods, such as network pruning (Li et al. 2017; Han et al. 2015; Wen et al. 2016) and knowledge distillation (Hinton, Vinyals, and Dean 2015; Romero et al. 2014), are proposed to trade off between model performance and size. Network pruning methods remove the weights of trained network based on priors and then retrain the pruned network to recover the performance of the original network. They require massive labeled data and an iterative retraining procedure, which is often time-consuming. Knowledge distillation methods (Hinton, Vinyals, and Dean 2015), on the other hand, train student networks by making them imitate the output of a given teacher. However, as student network is in many cases initialized randomly and trained from scratch, knowledge distillation approaches also rely on numerous training data to explore the large parameter space of student so as to train a well-behaved model.

To alleviate data hunger in knowledge distillation, several few-shot distillation methods have been proposed to transfer knowledge from teacher to student with less dependency on the amount of data. The work of (Li et al. 2018) proposes a few-shot approach by combining network pruning and block-wise distillation to compress the teacher model. The one of (Wang et al. 2020) introduces an active mixup augmentation strategy that selects hard samples from a pool of the augmented images. The work of (Bai et al. 2019) designs a cross distillation method, combined with network pruning to reduce the layer-wise accumulated errors in the few-shot setting. In spite of the encouraging results achieved, existing methods still heavily rely on pre- or post-processing techniques, such as network pruning, which *per se* are unstable and error-prone.

In this paper, we propose a principled dual-stage progressive network grafting strategy for few-shot knowledge distillation, which allows us to eliminate the dependency on other techniques that are potentially fragile and hence strengthen the robustness of knowledge distillation. At the heart of our proposed approach is a block-wise "grafting" scheme, which learns the parameters of the student network by injecting them into the teacher network and optimizing them intertwined with the parameters of the teacher in a progressive fashion. Such a grafting strategy takes much better advantage of the well-trained parameters of the teacher network and therefore significantly shrinks parameter space of the student network, allowing us to training the student with much fewer samples.

Specifically, our grafting-based distillation scheme follows a two-step procedure. In the first step, the student net-

*Corresponding author

work is decomposed into several blocks, each of which contains fewer parameters to be optimized. We then take the block of the student to replace the corresponding one of the teacher, and learn the parameters of the student intertwined with the well-trained parameters of the teacher, enabling the knowledge transfer from the teacher to the student. In the second step, the trained student blocks are progressively connected and then grafted onto teacher network, in which the student blocks learn to adapt each other and together replace more teacher blocks. Once all the student blocks are grafted onto the teacher network, the parameter learning is accomplished. The proposed dual-stage distillation, by explicitly exploiting the pre-trained parameters and refined knowledge of the teacher, largely eases the student training process and reduces the the risk of overfitting.

In sum, our contribution is a novel grafting strategy for few-shot knowledge distillation, which removes the dependency on other brittle techniques and therefore reinforces robustness. By following a principled two-step procedure, the proposed grafting strategy dives into the off-the-shelf teacher network and utilizes the well-trained parameters of the teacher to reduce the parameter search space of the student, thus enabling the efficient training the student under the few-shot setup. With only a few unlabeled samples for each class, the proposed approach achieves truly encouraging performances on CIFAR10, CIFAR100, and ILSVRC-2012. On CIFAR10 and CIFAR100, the proposed method even yields results on par with those obtained by knowledge distillation using the full datasets.

Related Work

Few-Shot Learning. The mainstream of few-shot learning research focuses on image classification, which learns to classify using few samples per category. Two kinds of approaches are widely adopted: metric learning based method (Koch, Zemel, and Salakhutdinov 2015; Vinyals et al. 2016; Snell, Swersky, and Zemel 2017) and meta learning based one (Ravi and Larochelle 2017; Wang and Hebert 2016; Santoro et al. 2016). The difference of problem setting between few-shot distillation and few-shot classification can be summarized as two fold. Firstly, a trained teacher model is available in few-shot distillation, but few-shot classification has none. Secondly, the model is trained on various related tasks in few-shot classification, but few-shot distillation is trained on the same task.

Knowledge Distillation. Resuing pre-trained networks has recently attracted attentions from researchers in the field (Chen et al. 2020; Yu et al. 2017). Bucilua et al (Bucilua, Caruana, and Niculescu-Mizil 2006) propose prototype knowledge distillation method, which trains a neural network using predictions from an ensemble of heterogeneous models. Hinton et al (Hinton, Vinyals, and Dean 2015) propose the knowledge distillation concept, where temperature is introduced to soften the predictions of teacher network. Following (Hinton, Vinyals, and Dean 2015), researchers pay more attention to the supervision from intermediate representations for better optimization performance (Romero et al. 2014; Wang et al. 2018; Shen et al. 2019a,b; Ye et al. 2020; Luo et al. 2020). To reduce total training time, online distillation (Yang et al. 2019; Zhang et al. 2019) is proposed to unify the training of student and teacher into one step. The above methods consume tremendous labeled data to transfer knowledge from teacher network, which significantly affects the convenience of deployment in practice. The works of (Song et al. 2019, 2020), on the other hand, focus on estimating knowledge transferability across different tasks, while those of (Yang et al. 2020b,a) explore distillation on the graph domain.

To reduce the data dependency, data-efficient and data-free knowledge distillation are investigated. FSKD (Li et al. 2018) adopts block-wise distillation to align the pruned student and teacher, where the blocks are not optimized to imitate the final teacher predictions. Wang et al (Wang et al. 2020) combine image mixup and active learning to augment the dataset, which still requires considerable data. ZSKD (Nayak and Chakraborty 2019) adapts synthetic data impressions from teacher model to replace original training data to achieve knowledge transfer. ZSKT (Micaelli and Storkey 2019) and DFAD (Fang et al. 2019) introduce an adversarial strategy to synthesize training samples for knowledge distillation. DAFL (Chen et al. 2019) and KEGNET (Yoo et al. 2019) are proposed to synthesize images by random given labels and then student learns to imitate teacher.

However, data-free methods need to learn knowledge from massive low-quality synthetic samples, which is a time-consuming procedure. In the meanwhile, data-free methods need to redesign and train a dedicated generator to synthesize massive data, where image generation is largely limited by the capacity of the generator, especially for high-resolution images. Few-shot methods have higher training efficiency, which only involve few real samples during optimization, even only one sample.

Grafting. Grafting is adopted as an opposite operation against pruning in decision tree (Webb 1997; Penttinen and Virtamo 2003), which adds new branches to the existing tree to increase the predictive performance. Li et al (Li, Tao, and Lu 2012) graft additional nodes onto the hidden layer of trained neural network for domain adaption. NGA (Hu, Delbruck, and Liu 2020) is proposed to graft front end network onto a trained network to replace its counterpart, which adapts the model to another domain. Meng et al (Meng et al. 2020) propose an adaptive weighting strategy, where filters from two networks are weighted and summed to reactivate invalid filters. Our proposed network grafting strategy replaces cumbersome teacher blocks with the corresponding lightweight student ones in a progressive manner, which aims to smoothly transfer knowledge from teacher to student.

The Proposed Method

Overview

The goal of few-shot knowledge distillation is to transfer knowledge from teacher network $\mathcal{T}$ to student network $\mathcal{S}$ using only few samples per category. For K-shot distillation, the optimization algorithm needs to search a large parameter space of student $\mathcal{S}$ with only K samples per category. Hence,

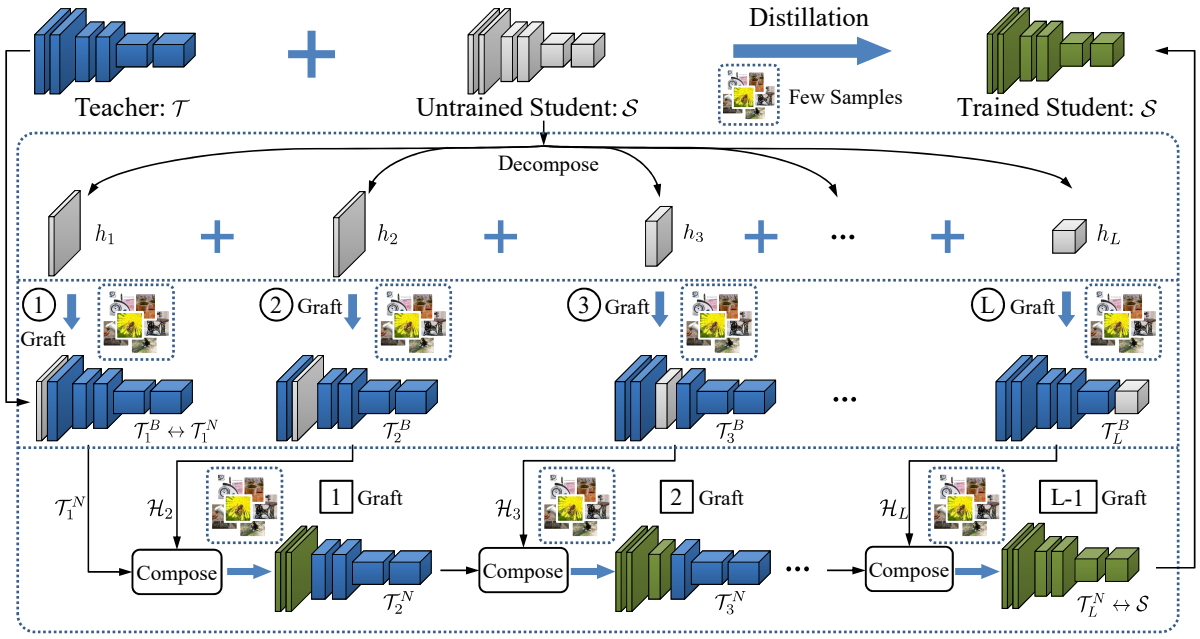


Figure 1: The dual-stage knowledge distillation strategy for few-shot knowledge distillation. Firstly, student network $\mathcal{S}$ is decomposed into several blocks: $\{h_l\}_{l=1}^L$, each of which is grafted onto teacher and then optimized by few-shot distillation between $\mathcal{T}_l^B$ and $\mathcal{T}$. Secondly, trained blocks in the first stage: $\{\mathcal{H}_l\}_{l=1}^L$ are sequentially composed into the trained student network: $\mathcal{S}$ using few-shot distillation between $\mathcal{T}_l^N$ and $\mathcal{T}$.

it is hard to directly optimize the student network with plain knowledge distillation scheme. To accomplish this goal, we adopt network grafting strategy to decompose the student network into several blocks, each of which only contains fewer parameters to be optimized.

Let $\mathcal{D}_c = \{x_{c,k}\}_{k=1}^K$ denotes the K samples from class c. The whole training dataset for K-shot N-way distillation can be presented as $\mathcal{D} = \cup_{c=1}^N \mathcal{D}_c$. The teacher network can be regarded as a composite function as $\mathcal{T}(x) = f_L \circ \cdots \circ f_l \circ \cdots \circ f_1(x)$, where f_l denotes the l-th block of teacher. In the same way, the student network can be presented as $\mathcal{S}(x) = h_L \circ \cdots \circ h_l \circ \cdots \circ h_1(x)$, where h_l denotes the l-th block of student. The knowledge distillation can be decomposed into a series of block distillation problems, where student block h_l learns to master the functions of the corresponding block in the teacher: f_l.

To this end, we adopt a dual-stage knowledge distillation strategy, which is depicted in Figure 1. In the first stage, each block of student is grafted onto teacher network, which replaces the corresponding block of teacher. Note that only single block of the teacher is replaced with the corresponding student block each time. Then the grafted student block is trained with the distillation procedure between the grafted teacher network and the original teacher one. In the second stage, all the trained student blocks are progressively grafted on teacher network, until the whole teacher network is fully replaced by a series of student block.

Block Grafting

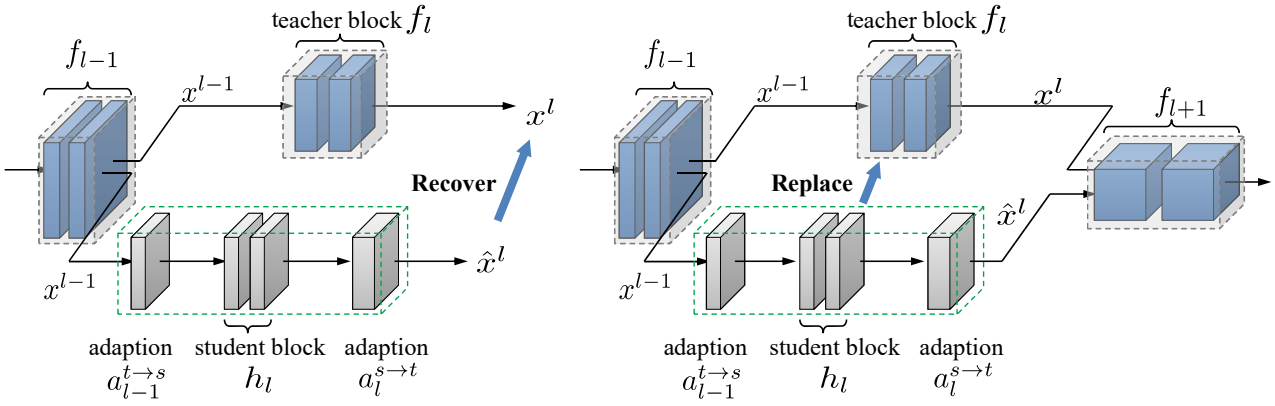
In the setting of few samples available, it's believed that the optimization of neural network is difficult, especially the network with massive parameters. To reduce the complexity of network optimization, the student network is decomposed into a series of blocks, each of which contains fewer parameters. For the training of block module, two alternative solutions are available, as shown in Figure 2.

For the first solution, the input and output of the corresponding block from teacher network are used to train the student block, as depicted in Figure 2 (a). This solution aims to imitate and recover the output of teacher block originally. However, due to model compression, the student block is generally much smaller than the teacher one. In other word, the capacity of student block is smaller than teacher block. There is some noise in teacher block output, which is trivial for final classification decision. Learning without information selection may hurt the performance of the student.

For the second solution, the student block is grafted on teacher network, as shown in Figure 2 (b). The grafted teacher network can be denoted as:

$$\mathcal{T}_l^B(x) = f_L \circ \cdots f_{l+1} \circ h_l \circ f_{l-1} \circ \cdots f_1(x), \quad (1)$$

where the l-th block of student: h_l replaces the teacher block f_l. Note that the block number of student needs to be equal to the teacher one, but the inner structure of the h_l and the f_l can be different. The grafted teacher $\mathcal{T}_l^B$ is trained to recover the predictions of the original teacher $\mathcal{T}$ and only the parameters of h_l are optimized. This solution encourages the block h_l to focus on the knowledge, which is vital for final



(a) Block Distillation (b) Block Grafting

Figure 2: Two alternative block-wise distillation solutions. (a) Block distillation tries to recover the output of teacher block using a lightweight student block. (b) Block grafting is trained to replace teacher block with student one to recover the final prediction of teacher. Note that only the modules in green dashed box are updated during optimization.

classification predictions. It effectively avoids the problem in the first solution, where the block only learns the original outputs of teacher block but not the function of the block in the whole network.

Due to the channel dimension difference between student block and teacher one, we introduce adaption module to align the channel dimension difference. The adaption module can be categorized into two kinds: adaption module for the dimension transform from teacher block to student one $a_{l-1}^{t\to s}(x^{l-1})$ and the one from student to teacher $a_l^{s\to t}(x^l)$. Combined with adaption modules, the wrapped scion $\mathcal{H}_l(x^{l-1})$ can be written as:

$$\mathcal{H}_l(x^{l-1}) = a_l^{s\to t} \circ h_l \circ a_{l-1}^{t\to s}(x^{l-1}), \tag{2}$$

where x^l denotes the output of l-th block from network. The adaption module is implemented with 1×1 convolution operation. It achieves a linear recombination of input features across channel dimension and doesn't change the size of receptive field, which is designed to align features between student and teacher. There are two special cases to be clarified:

$$\mathcal{H}_1(x) = a_1^{s\to t} \circ h_1(x), \mathcal{H}_L(x) = h_L(x) \circ a_{L-1}^{t\to s}(x^{L-1}), \tag{3}$$

where $\mathcal{H}_1(x)$ and $\mathcal{H}_L(x)$ denote the first wrapped scion and the last one, respectively. Combining Eq.1, Eq.2 and Eq.3, the final grafted teacher can be written as:

$$\mathcal{T}_l^B(x) = f_L \circ \cdots f_{l+1} \circ \mathcal{H}_l \circ f_{l-1} \circ \cdots f_1(x). \tag{4}$$

Progressive Network Grafting

A series of trained student blocks can be obtained from the above section, each of which can make decision with the blocks of teacher network. However, these trained blocks are not trained to work with each other. In this section, we adopt a network grafting strategy to progressively increase the student blocks in the grafted teacher network and reduce the dependency of the original teacher.

Algorithm 1 Network Grafting for Few-Shot Knowledge Distillation

Input: Teacher model $\mathcal{T}$; Few unlabeled training data $\mathcal{D} = \{x_i\}_{i=1}^{N\cdot K}$.
Output: The parameters of student block $\mathcal{S}$.

```
for l = 1 to L do
    Wrap h_l in a_l^{s→t} and a_{l-1}^{t→s} to obtain H_l by Eq. 2 and Eq. 3;
    Graft H_l onto T to obtain T_l^B by Eq. 4;
    for number of training epochs do
        Feed x into T and T_l^B to compute L_l^B by Eq. 7;
        Update the parameters of H_l using Adam;
    end for
end for
Initialize grafted teacher by T_1^N ← T_1^B;
for l = 2 to L do
    Combine T_{l-1}^N and H_l to obtain T_l^N by Eq. 5;
    for number of training epochs do
        Feed x into T and T_l^N to compute L_l^N by Eq. 8;
        Update the parameters of {H_j}_{j=1}^l using Adam;
    end for
end for
Merge {a_l^{t→s}}_l and {a_l^{s→t}}_l into {h_l}_l and obtain T_L^N, namely S, by Eq. 6.
```

On the base of $\mathcal{T}_1^B(x) = f_L \circ \cdots \circ f_2 \circ \mathcal{H}_1(x)$, the trained blocks: $\mathcal{H}_2, \mathcal{H}_3, \cdots, \mathcal{H}_L$ are sequentially grafted onto $\mathcal{T}$, as shown in Figure 1. The grafted teacher for network grafting can be denoted as:

$$\mathcal{T}_l^N(x) = f_L \circ \cdots f_{l+1} \circ \mathcal{H}_l \circ \mathcal{H}_{l-1} \circ \cdots \mathcal{H}_1(x), \tag{5}$$

where $\mathcal{T}_1^N(x) = \mathcal{T}_1^B(x)$. During network grafting, a sequence of models: $\{\mathcal{T}_l^N(x)\}_{l=1}^L$ are optimized. Finally, $\mathcal{T}_L^N(x)$ is obtained, which connects all student blocks and forms a complete network. However, $\mathcal{T}_L^N(x)$ is still different from original student network $\mathcal{S}(x)$. $\mathcal{T}_L^N(x)$ is composed

of a series of $\mathcal{H}_l$, but $\mathcal{S}(x)$ is composed of h_l. Compared to h_l, $\mathcal{H}_l$ contains additional adaption module: $a_l^{s\to t}$ or $a_{l-1}^{t\to s}$, which means the obtained $\mathcal{T}_L^N(x)$ has more parameters than $\mathcal{S}(x)$.

Thanks to the linearity of adaption module, the parameters of adaption module can be merged into the convolution layer in the next block h_{l+1} without increasing any parameters. For $\mathcal{H}_{l+1} \circ \mathcal{H}_l = a_{l+1}^{s\to t} \circ h_{l+1} \circ a_l^{t\to s} \circ a_l^{s\to t} \circ h_l \circ a_{l-1}^{t\to s}$, $(a_l^{t\to s} \circ a_l^{s\to t})$ can be merged into h_{l+1}. We denote $\hat{h}_{l+1} = h_{l+1} \circ a_l^{t\to s} \circ a_l^{s\to t}$. Then, $\mathcal{T}_L^N(x)$ can be transformed into the following form:

$$\mathcal{T}_L^N(x) = \hat{h}_L \circ \cdots \hat{h}_{l+1} \circ \hat{h}_l \circ \hat{h}_{l-1} \circ \cdots \hat{h}_1(x), \quad (6)$$

which has the same network architecture as $\mathcal{S}$. In other word, we achieve the knowledge transfer from teacher $\mathcal{T}$ to student $\mathcal{S}$.

Optimization

The scale of logits from different network architectures may have a large gap, which may result in optimization difficulty. To this end, we propose an L_2 loss function on normalized logits for knowledge transfer between teacher block f_l and student block h_l as follow:

$$\mathcal{L}_l^B(x) = \frac{1}{N}\|\widetilde{\mathcal{T}}_l^B(x) - \widetilde{\mathcal{T}}(x)\|_2^2, \quad (7)$$

where $\widetilde{\mathcal{T}}(x) = \mathcal{T}(x)/\|\mathcal{T}(x)\|_2$. In the optimization of block grafting, only the wrapped student block $\mathcal{H}_l$ is learnable and updated with the gradient $\nabla_{\Theta_l}\mathcal{L}_l^B$, where Θ_l denotes the parameters of $\mathcal{H}_l$. For network grafting, a similar distillation is adopted,

$$\mathcal{L}_l^N(x) = \frac{1}{N}\|\widetilde{\mathcal{T}}_l^N(x) - \widetilde{\mathcal{T}}(x)\|_2^2. \quad (8)$$

The difference to block grafting is that a sequence of wrapped student blocks: $\{\mathcal{H}_i\}_{i=1}^l$ need to be optimized, for the l-th step of network grafting. The complete algorithm for our proposed approach can be summarized as Algorithm 1.

Experiments

Experimental Settings

Datasets and Models. Both CIFAR10 and CIFAR100 are composed of $60,000$ colour images with 32×32 size, where $50,000$ images are used as training set and the rest $10,000$ images are used as test set. The CIFAR 10 dataset contains 10 classes, and the CIFAR100 contains 100 classes. In few-shot setting, we randomly sample K samples per class from the original CIFAR datasets as training set, where $K \in \{1, 5, 10\}$. Random crop and random horizontal flip are applied to training images to augment the dataset. The test set is the same as the original one. Due to low resolution of images in CIFAR, a modified VGG16 (Li et al. 2017) are used as the teacher model. Following (Li et al. 2018) and (Li et al. 2017), we adopt VGG16-half as student model.

The ILSVRC-2012 dataset (Russakovsky et al. 2015) contains 1.2 million images as training set, $50,000$ images as validation set, from $1,000$ categories. We randomly sample 10 images per class from the training set for few-shot training. The training images are augmented with random crop and random horizontal flip. During test, the images are cropped into 224×224 size in the center. We adopt PyTorch official trained ResNet34 (He et al. 2016) as the teacher model, ResNet18 as the student one.

Implementation Details. The proposed method is implemented using PyTorch on a Quadro P6000 24G GPU. The batch size is set to 64 for 10-shot training. For K-shot training, we set batch size to $\lfloor 64 \times K/10 \rfloor$. For all experiments, we adopt Adam algorithm for network optimization. Without extra clarification, the following learning rates work when batch size is 64. For other batch size: B, the learning rate is scaled by the factor: $B/64$ as (He et al. 2016). The learning rates for block grafting and network grafting on CIFAR10 are set to 2.5×10^{-4}, 1×10^{-4}, respectively. For CIFAR100, the learning rates are 1×10^{-3}, 5×10^{-5}, respectively. Following (Kingma and Ba 2014), we set the weight decay to zero and the running averages of gradient and its square to 0.9 and 0.999, respectively. We adopt the weight initialization proposed by (He et al. 2015). For ResNet18 on ILSVRC-2012, we adopt the same optimizer and weight initialization method as VGG16-half. During block grafting, we set the learning rate of *block1* and *block2* to 10^{-4}, the one of *block3* and *block4* to 10^{-3}. During network grafting, the learning rates for *block1∼2*, *block1∼3* and *block1∼4* are set to 10^{-4}, 2×10^{-3} and 10^{-3}, respectively.

Experimental Results

Homogeneous Architecture Knowledge Distillation. Intuitively, the knowledge between homogeneous network architectures tends to have a more clear correlation than heterogeneous networks, especially for block-wise distillation situation. In this section, we first investigate knowledge distillation between homogeneous network to validate the effectiveness of the proposed method. In Table 1, VGG16-half has a similar network architecture as VGG16 but has fewer channels than VGG16 in the corresponding layer. With full dataset, both KD (Hinton, Vinyals, and Dean 2015) and FitNet (Romero et al. 2014) for VGG16-half achieve comparable performance as their larger teacher. However, when only few samples are available, both KD and FitNet encounter a significant performance drop. FSKD (Li et al. 2018) adopts a dual-step strategy: first network pruning and then block-wise distillation, which achieves an efficient improvement against KD and FitNet in few-shot setting. Our proposed method achieves comparable performance as KD trained with full dataset, even better than teacher in 10-shot setting on CIFAR10. And our method is also superior to other few-shot knowledge distillation methods in all settings. FSKD focuses on the imitation of the intermediate teacher outputs, whose student is not optimized to imitate the predictions of teacher. We owe the improvement of our method to the end-to-end imitation of the teacher predictions, which is more robust to some trivial intermediate noise.

Heterogeneous Architecture Knowledge Distillation. To further verify our method, we conduct few-shot distillation between heterogeneous networks. We adopt ResNet18 as the student, whose network architecture is significantly

Method	# Samples	CIFAR10 # Params	CIFAR10 # FLOPs	CIFAR10 Accuracy (%)	CIFAR100 # Params	CIFAR100 # FLOPs	CIFAR100 Accuracy (%)
Teacher (VGG16)	5,000	15.0M	0.31G	92.83	15.0M	0.31G	69.82
KD (full)	5,000	5.4M	0.22G	92.06±0.17	5.4M	0.22G	68.31±0.15
FitNet (full)	5,000	5.4M	0.22G	92.75±0.11	5.4M	0.22G	70.12±0.12
KD (few)	1	5.4M	0.22G	71.80±1.84	5.4M	0.22G	37.73±1.53
	5			87.49±1.01			64.03±0.06
	10			88.48±0.16			65.27±0.17
FitNet (few)	1	5.4M	0.22G	74.43±1.43	5.4M	0.22G	41.67±1.23
	5			88.53±0.92			64.13±0.34
	10			88.76±0.14			65.64±0.15
FSKD	1	5.4M	0.22G	87.42±0.84	5.4M	0.22G	48.57±1.14
	5			88.83±0.63			65.47±0.18
	10			92.37±0.00†			67.34±0.12
Cross Distillation	1	5.4M	0.22G	69.57±1.39	5.4M	0.22G	41.32±0.18
	5			84.91±0.98			63.81±0.15
	10			86.61±0.71			64.95±0.24
Ours (VGG16-half)	1	5.4M	0.22G	**90.74±0.49**	5.4M	0.22G	**64.22±0.17**
	5			**92.88±0.07**			**68.16±0.20**
	10			**92.89±0.06**			**68.86±0.03**
Ours (ResNet18)	1	11.2M	0.22G	73.69±0.91	11.2M	0.22G	55.51±0.92
	5			91.79±0.34			66.64±0.12
	10			92.48±0.13			67.77±0.04

† denotes the experimental result reported in (Li et al. 2018).

Table 1: The performance of few-shot distillation on CIFAR dataset. "full" denotes training with full dataset. "few" denotes few-shot training. The second column: "# Samples" denotes the number of samples per class. All student networks adopt VGG16-half, except "Ours (ResNet18)". All results are averaged over five runs and error bars correspond to the standard deviation.

different from VGG16. Despite more parameters than VGG16-half, the ResNet18 does not achieve better performance in few-shot distillation setting than VGG16-half. Especially when only one sample is available per category, the performance drop of ResNet18 is dramatic on both CIFAR10 and CIFAR100. We guess that the knowledge distribution of ResNet18 is significantly different from the VGG16 one across blocks, which harms the performance of knowledge distillation. We will explore knowledge transfer between very different network architectures in future.

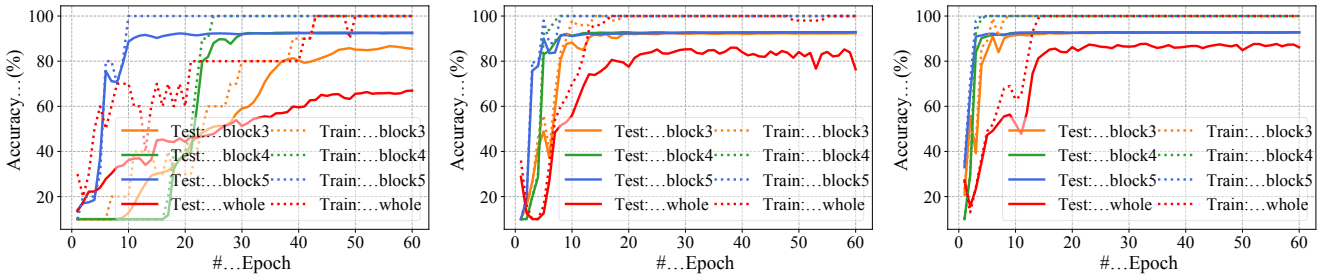
Knowledge Distillation for Large-Scale Dataset. To validate the generality of the proposed method, we also conduct experiments on a more challenging large-scale dateset: ILSVRC-2012. As shown in Table 2, the experimental results demonstrate that our proposed method significantly outperforms other few-shot distillation baselines. The performance of our method is slightly below ResNet18 trained on full ILSVRC-2012 with classification loss. We deduce that the dataset for few-shot distillation struggles to cover the high diversity of full ILSVRC-2012.

Learning with Different Numbers of Samples. To investigate the effect of different numbers of training samples, we conduct a series of few-shot distillation experiments. The experiments are implemented on CIFAR10 and CIFAR100, as shown in Figure 4. For KD and FitNet, the distillation performances are significantly improved with the increment of the training sample number. When the number of training samples is extremely small, the performances of KD and FitNet are both far from the teacher one. Our proposed method achieves comparable performance as teacher, even when only one training sample is available. Due to smaller parameter search space, our method is more robust to the size of training dataset.

Method	#Samples	#Params	#FLOPs	Acc@1	Acc@5
Teacher (ResNet34)	1,000	21.8M	3.6G	73.31%	91.42%
ResNet18 (xenc)	1,000	11.7M	1.8G	69.76%	89.08%
KD (ResNet18)	10	11.7M	1.8G	46.40%	73.71%
FitNet (ResNet18)	10	11.7M	1.8G	45.82%	73.46%
Ours (ResNet18)	10	11.7M	1.8G	68.15%	88.83%

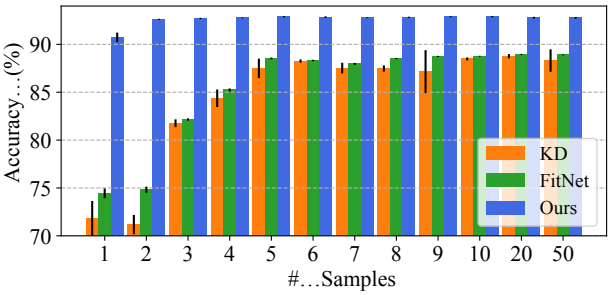
Table 2: The performance of few-shot distillation on ILSVRC-2012 dataset. "Acc@1" and "Acc@5" denote top 1 accuracy and top 5 accuracy, respectively. "xenc" denotes training with cross entropy classification loss.

Optimization Efficiency of Block Grafting. To evaluate the efficiency of our proposed block grafting strategy, we compare the training curves of single grafted block optimization and the whole student optimization, denoted as "whole", in different few-shot settings. Specifically, the last three blocks of VGG16-half, which contain most of parameters in network, are selected as the target to validate block grafting algorithm. As shown in Figure 3, our proposed method not only converges more quickly, but also outperforms the "whole" by a large margin. We observe that our

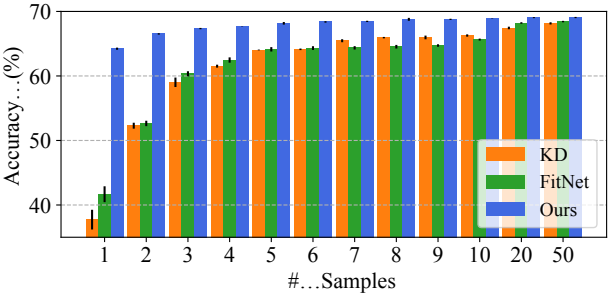


(a) 1-Shot Distillation (b) 5-Shot Distillation (c) 10-Shot Distillation

Figure 3: The comparison of optimization efficiency on block grafting and the whole student on CIFAR10.



(a) CIFAR10



(b) CIFAR100

Figure 4: Learning with different numbers of samples. All experiments are averaged over five runs and error bars correspond to the standard deviation.

method has a smaller gap between training accuracy and test one, which demonstrates our method can significantly reduce the risk of overfitting.

Partial Network Grafting. Some trained networks contain several cumbersome blocks, which can be simplified. Partial network grafting provides an alternative solution to improve the compactness of the existing trained network. We graft some lightweight blocks onto the target trained network to reduce the original model size. In this section, we evaluate the effect of different blocks' graft from ResNet18 to ResNet34, as shown in Table 3. We split the backbone

	block1	**block2**	**block3**	**block4**
Params (M)	0.22→0.14	1.12→0.52	6.82→2.10	13.11→8.40
	36.4%↓	53.6%↓	69.2%↓	35.9%↓
Accuracy (%)	72.55	70.33	68.11	68.40

Table 3: The performance of block grafting from ResNet18 to ResNet34 on ILSVRC-2012. The accuracy of teacher network is 73.31%.

of ResNet34 and ResNet18 into four blocks: block1∼4, according to down-sampling layer. As depicted in Table 3, the *block3* and *block4* contain most of parameters in ResNet34. Replacing these blocks with the corresponding blocks from ResNet18 can significantly reduce the original model size. For example, replacing *block3* with the one from ResNet18 can reduce the parameter number from 6.82M to 2.10M, a 69.2% reduction. Compared to the ResNet18 trained on full ILSVRC-2012, the performance only drops 1.65%, where only 10 samples per category are used to train network.

Conclusion and Future Work

In this paper, we propose an progressive network grafting method to distill knowledge from teacher with few unlabeled samples per class. This is achieved by a dual-stage approach. In the first stage, the student is split into blocks and grafted onto the corresponding position of teacher. The student blocks are optimized with fixed teacher blocks using distillation loss. In the second stage, the trained student blocks are incrementally grafted onto the teacher and trained to connect to each other, until the whole student network replaces the teacher one. Experimental results on several benchmarks demonstrate that the proposed method successfully transfers knowledge from teacher in few-shot setting and achieves comparable performance as knowledge distillation using full dataset.

Few-shot distillation can significantly reduce the training cost of neural network. For future work, we plan to explore network architecture search on distillation using block grafting, which aims to find a series of more efficient block modules. We believe that this work is one step toward efficient network architecture search.

Acknowledgements

This work is supported by National Natural Science Foundation of China (U20B2066, 61976186), the Major Scientifc Research Project of Zhejiang Lab (No. 2019KD0AC01) and Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies.

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39(12): 2481–2495.

Bai, H.; Wu, J.; King, I.; and Lyu, M. 2019. Few shot network compression via cross distillation. *arXiv preprint arXiv:1911.09450* .

Bucilua, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 535–541.

Chen, H.; Wang, Y.; Xu, C.; Shi, B.; Xu, C.; Tian, Q.; and Xu, C. 2020. AdderNet: Do We Really Need Multiplications in Deep Learning? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3514–3522.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40(4): 834–848.

Fang, G.; Song, J.; Shen, C.; Wang, X.; Chen, D.; and Song, M. 2019. Data-Free Adversarial Distillation. *arXiv preprint arXiv:1912.11006* .

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems (NeurIPS)*, 1135–1143.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1026–1034.

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

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hu, Y.; Delbruck, T.; and Liu, S.-C. 2020. Exploiting Event Cameras by Using a Network Grafting Algorithm. *arXiv preprint arXiv:2003.10959* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Koch, G.; Zemel, R.; and Salakhutdinov, R. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1097–1105.

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. P. 2017. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*.

Li, T.; Li, J.; Liu, Z.; and Zhang, C. 2018. Few sample knowledge distillation for efficient network compression. *arXiv preprint arXiv:1812.01839* .

Li, Y.; Tao, X.; and Lu, J. 2012. Network grafting: Transferring learned features from trained neural networks. In *IEEE International Conference on Computational Intelligence and Cybernetics (CyberneticsCom)*, 40–44.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 21–37. Springer.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440.

Luo, S.; Pan, W.; Wang, X.; Wang, D.; Tang, H.; and Song, M. 2020. Collaboration by Competition: Self-coordinated Knowledge Amalgamation for Multi-talent Student Learning. In *European Conference on Computer Vision (ECCV)*, 631–646. Springer.

Meng, F.; Cheng, H.; Li, K.; Xu, Z.; Ji, R.; Sun, X.; and Lu, G. 2020. Filter Grafting for Deep Neural Networks. *arXiv preprint arXiv:2001.05868* .

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, 9547–9557.

Nayak, G. K., M. K. R. S. V. B. R. V.; and Chakraborty, A. 2019. Zero-Shot Knowledge Distillation in Deep Networks. In *International Conference on Machine Learning (ICML)*, 4743–4751.

Penttinen, A.; and Virtamo, J. 2003. Improving multicast tree construction in static ad hoc networks. In *IEEE International Conference on Local Computer Networks (ICLCN)*, 762–765.

Ravi, S.; and Larochelle, H. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 91–99.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3): 211–252.

Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; and Lillicrap, T. 2016. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065* .

Shen, C.; Wang, X.; Song, J.; Sun, L.; and Song, M. 2019a. Amalgamating Knowledge towards Comprehensive Classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 3068–3075.

Shen, C.; Xue, M.; Wang, X.; Song, J.; Sun, L.; and Song, M. 2019b. Customizing student networks from heterogeneous teachers via adaptive knowledge amalgamation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 4077–4087.

Song, J.; Chen, Y.; Wang, X.; Shen, C.; and Song, M. 2019. Deep Model Transferability from Attribution Maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 6182–6192.

Song, J.; Chen, Y.; Ye, J.; Wang, X.; Shen, C.; Mao, F.; and Song, M. 2020. DEPARA: Deep Attribution Graph for Deep Knowledge Transferability. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 3630–3638.

Wang, D.; Li, Y.; Wang, L.; and Gong, B. 2020. Neural Networks Are More Productive Teachers Than Human Raters: Active Mixup for Data-Efficient Knowledge Distillation from a Blackbox Model. *arXiv preprint arXiv:2003.13960* .

Wang, H.; Zhao, H.; Li, X.; and Tan, X. 2018. Progressive Blockwise Knowledge Distillation for Neural Network Acceleration. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Wang, Y.-X.; and Hebert, M. 2016. Learning to learn: Model regression networks for easy small sample learning. In *European Conference on Computer Vision (ECCV)*, 616–634. Springer.

Webb, G. I. 1997. Decision tree grafting. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 846–851.

Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H. 2016. Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2074–2082.

Yang, C.; Xie, L.; Su, C.; and Yuille, A. L. 2019. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2859–2868.

Yang, Y.; Feng, Z.; Song, M.; and Wang, X. 2020a. Factorizable Graph Convolutional Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020b. Distilling Knowledge From Graph Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ye, J.; Ji, Y.; Wang, X.; Gao, X.; and Song, M. 2020. Data-Free Knowledge Amalgamation via Group-Stack Dual-GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12516–12525.

Yoo, J.; Cho, M.; Kim, T.; and Kang, U. 2019. Knowledge Extraction with No Observable Data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2701–2710.

Yu, X.; Liu, T.; Wang, X.; and Tao, D. 2017. On Compressing Deep Models by Low Rank and Sparse Decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3713–3722.

Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; and Li, S. Z. 2018. Single-shot refinement neural network for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4203–4212.