DGD: Densifying the Knowledge of Neural Networks with Filter Grafting and Knowledge Distillation

Hao Cheng 1 Fanxu Meng 2 Ke Li 1 Huixiang Luo 3 Guangming Lu 2 Xiaowei Guo 1 Feiyue Huang 1 Xing Sun 1

Abstract
With a fixed model structure, knowledge distillation and filter grafting are two effective ways to boost single model accuracy. However, the working mechanism and the differences between distillation and grafting have not been fully unveiled. In this paper, we evaluate the effect of distillation and grafting in the filter level, and find that the impacts of the two techniques are surprisingly complementary: distillation mostly enhances the knowledge of valid filters while grafting mostly reactsivate invalid filters. This observation guides us to design a unified training framework called DGD, where distillation and grafting are naturally combined to increase the knowledge density inside the filters given a fixed model structure. Through extensive experiments, we show that the knowledge densified network in DGD shares both advantages of distillation and grafting, lifting the model accuracy to a higher level.

1. Introduction
Convolutional Neural Networks (CNNs) have achieved state of art results in many application fields, including computer vision (Krizhevsky et al., 2012; Lotter et al., 2016), natural language processing (Zhang & LeCun, 2015), speech recognition (Graves et al., 2013), etc. Generally speaking, a CNN model is usually evaluated in terms of accuracy and efficiency. Given a pre-trained model, the network is sometimes pruned for efficiency consideration when its accuracy is already satisfying (Li et al., 2016; Suau et al., 2018; Yang et al., 2018). By contrast, when the model efficiency has met the deployment requirement, one of the common choices for performance improvement is to train the model with the help of other models, e.g., knowledge distillation (Hinton et al., 2015; Parisotto et al., 2015; Bucilu et al., 2006; Phuong & Lampert, 2019; Nayak et al., 2019) and filter grafting (Fanxu et al., 2020). However, the uses of these techniques are isolated and their working mechanisms have not been fully unveiled.

Since deep networks are usually over-parameterized, a pre-trained network always contains some invalid (unimportant) filters. These filters have relatively small $l_1$ norm and contribute little to the output. Pruning part of these filters and re-training the network can decrease the computation complexity while keeping a comparable accuracy. Conversely, instead of pruning the model structure for efficiency, some methods learn from other models for accuracy. Knowledge distillation first trains a bigger model as a teacher, then trains a smaller student to mimic the teacher. Mimicking is performed by enforcing the outputs of the student and teacher to be close to each other. The outputs can be class probabilities (Hinton et al., 2015) or internal feature representations (Ba & Caruana, 2014; Romero et al., 2014). More recently, filter grafting proposes to re-activate the invalid filters of the network with the help of other networks. Specifically, given a network, this method grafts (adds) the knowledge (weights) of valid filters from other parallelly trained networks onto invalid filters of the network. The grafted network has better representation ability since more valid filters are involved in the data flow. Unlike filter pruning, whose effects can be easily understood by invalid computation decreasing, the reason for why dis-
tillation and grafting can improve the network accuracy has not been fully studied.

In this paper, we dissect how these training techniques influence the network in the filter level, and propose to use knowledge density as a unified metric to measure the accuracy and efficiency of a network. Knowledge density is defined as a proportion of the layer-wise filter \( l_1 \) norm to layer-wise filter number. By comparing the effects of pruning, grafting and distillation, we argue that all these methods work because they improve the network knowledge density in essence. Specifically, pruning compresses the filter number while grafting improves the \( l_1 \) norm of invalid filter and distillation improves the \( l_1 \) norm of valid filter, i.e., making the network knowledge become denser. Since filter pruning densifies the network knowledge at a cost of accuracy, we focus on improving the network density by improving the \( l_1 \) norm of both valid and invalid filters in this paper. To achieve this, we design a training framework called Densifying with Grafting and Distillation (DGD). As shown in Figure 1, DGD densifies the knowledge of both kinds of filters. The main contributions of this paper are listed as follows:

- We analyze the difference between distillation and grafting in terms of network filters, and find that grafting mostly improves the knowledge of invalid filters while distillation mostly improves that of valid filters. This observation inspires us to design a unified metric to measure the accuracy and efficiency of a network.

- Based on the definition of knowledge density, we design a new learning framework called DGD, which improves the knowledge of both valid and invalid filters. In DGD, a student network learns knowledge from the teacher network and other student networks.

- We evaluate DGD on multiple learning tasks. Results show that given a structure fixed network, DGD could boost the accuracy of the network to a higher level.

2. Related Works

Knowledge Distillation: The original idea of knowledge distillation can be traced back to model compression (Buciu et al., 2006), where authors demonstrate that the knowledge acquired by a large ensemble of models can be transferred to a single small model. Hinton et al. (2015) generalize this idea to deep neural networks and show a small, shallow network can be improved through a teacher-student framework. Meanwhile, past works have also been focusing on improving the quality and applicability of knowledge distillation and understanding how knowledge distillation works. The work (Tang et al., 2016) employs the knowledge transfer learning approach to train RNN using a deep neural network model as the teacher. This is different from the original knowledge distillation, since the teacher used is weaker than the student. Lopes et al. (2017) present a method for data-free knowledge distillation, which is able to compress deep neural networks trained on large-scale datasets to a fraction of their size leveraging only some extra metadata with a pre-trained model. The work (Mirzadeh et al., 2019; Cho & Hariharan, 2019) shows that the student network performance degrades when the gap between student and teacher is large. They introduce multi-step knowledge distillation which employs an intermediate-size network to solve this. However, no efforts has been paid on interpreting knowledge distillation in the filter level. In our work, we evaluate knowledge distillation in terms of the filters and find interesting observations.

Filter Grafting: The motivation of filter grafting (Fanxu et al., 2020) comes from the observation that DNNs have unimportant (invalid) filters. These filters limit the potential of DNNs since they are identified as having little effect on the network output. While filter pruning removes these invalid filters for efficiency consideration, filter grafting re-activates them from an accuracy boosting perspective. Specifically, for filters whose \( l_1 \) norm is small, we can graft the weights from other networks’ filters onto them. This work also finds that the positions of invalid filters in each network are statistically different, thus grafting can be efficiently processed in the layer level (Formulation of grafting can be seen in Section 3.3).

Collaborative Learning: DGD framework involves learning from multiple models. Similar ideas have been shown in Dual Learning (He et al., 2016) and Cooperative Learning (Batra & Parikh, 2017). Dual learning deals with the special translation problem where an unconditional within-language model is available to evaluate the quality of the prediction, and ultimately provides the supervision that drives the learning process. In dual learning, different models have different learning tasks. Cooperative Learning aims to learn multiple models jointly for the same task but in different domains, e.g., recognizing the same set of object categories with one model inputting RGB images and the other inputting depth images. The models communicate via object attributes which are domain invariant. Dual learning and cooperate learning are different from DGD where all models address the same task and domain.

3. Knowledge density in Neural Networks

3.1. Definition of knowledge density

Given a network \( M \), suppose there are \( Q \) filters inside \( M \) and \( M_q \) is the \( q \)-th filter. Let \( E(M) \) denotes the knowledge density of \( M \). \( E(M) \) can be expressed as:

\[
E(M) = \frac{\sum_{q=1}^{Q} \text{knowledge}_q}{\sum_{q=1}^{Q} \text{filter}_q}
\]
where \( f(W_q) \) evaluates the quality of the single filter \( W_q \). We adopt \( f \) to be \( l_1 \) norm, which is commonly used in filter pruning and filter grafting. Specifically, \( f \) can be expressed by:

\[
f(W_q) = \|W_q\|_1,
\]

where \( \|W_q\|_1 \) is the \( l_1 \) norm of \( W_q \). A network with larger \( E(M) \) is a denser network in this sense. We argue the reason that knowledge distillation and filter grafting work is that they could densify the knowledge of neural networks given a fixed model structure (See Section 3.4 for details).

### 3.2. Formulation of Knowledge Distillation

Concretely, given \( N \) samples \( X = \{x_i\}_{i=1}^N \) from \( K \) classes, we denote the corresponding label set as \( Y = \{y_i\}_{i=1}^N \) with \( y_i \in \{1, 2, \cdot \cdot \cdot, K\} \). For any input image \( x_i \), the teacher network produces a vector of scores \( s^t(x_i) = [s^t_1(x_i), s^t_2(x_i), \cdot \cdot \cdot, s^t_K(x_i)] \) that are converted into probabilities:

\[
p^t_k(x_i) = \frac{e^{s^t_k(x_i)}}{\sum_{j} e^{s^t_j(x_i)}}.
\]

Trained neural networks produce peaky probability distributions, which may be less informative. Distillation (Hinton et al., 2015) therefore propose to soften these probabilities using temperature scaling:

\[
p^s_k(x_i) = \frac{e^{s^s_k(x_i)/\tau}}{\sum_{j} e^{s^s_j(x_i)/\tau}},
\]

where \( \tau > 1 \) is a hyper-parameter.

Similarly, a student network also produces a softened class probability distribution \( p^s(x_i) \). The loss for the student is then a linear combination of the typical cross entropy loss \( L_C \) and a knowledge distillation loss \( L_{KD} \):

\[
L = \alpha_s L_C + \alpha_t L_{KD},
\]

where \( L_{KD} = -\sum_{i=1}^N \tau^2 \sum_k p^t_k(x_i) \log p^s_k(x_i) \). \( \alpha_s, \alpha_t \) and \( \tau \) are hyper-parameters and usually \( \tau \in \{2, 3, 4, 5\} \).

### 3.3. Formulation of Filter Grafting

Suppose there are \( C \) networks in filter grafting algorithm. \( M_c \) denotes the \( c \)-th network. Denote \( W^M_c \) as the weight of the \( l \)-th layer of \( M_c \). Then grafting procedure can be expressed as:

\[
W^{M'}_l = \alpha W^M_c + (1 - \alpha) W^{M_c-1}_l,
\]

where \( \alpha \) is a weighting coefficient that determines which network is more valuable (i.e., balancing the self-network information and external information from other networks). In grafting (Fanxu et al., 2020), \( \alpha \) is calculated from:

\[
\alpha = A * (arctan(d * (H(W^M_c) - H(W^{M_c-1}))) + 0.5,
\]

where \( A \) and \( d \) are fixed hyper-parameters. \( H(W^M_c) \) denotes the entropy (information) of \( W^M_c \). Suppose \( H(W^M_c) > H(W^{M_c-1}) \), then \( \alpha \) is bigger than 0.5.

Filter grafting (Fanxu et al., 2020) also proves that the grafted network has more valid filters compared to a non-rafted network. Also, the performance gets better as the number of network increases in grafting algorithm. We depict a simple picture to illustrate the grafting process in Figure 2.

### 3.4. Evaluating knowledge distillation and filter grafting in the filter level

In this section, knowledge distillation and filter grafting are evaluated in terms of the network’s filters, i.e., how distillation and grafting influence the knowledge density of the valid and invalid filter. Different thresholds are set to determine which filters are valid or invalid. Specifically, all the filters are ranked according to their \( l_1 \) norm values. For filters whose \( l_1 \) norms are larger than the threshold, we consider these filters as valid. Inversely, filters are considered to be invalid if their \( l_1 \) norms are smaller than the threshold. Then for each network trained by different methods, we calculate its knowledge density of both the valid and the invalid filter. Datasets and training setting are listed below:

**Training datasets:** The CIFAR-10 and CIFAR-100 (Krizhevsky et al.) datasets are selected for this experiment. The datasets consist of \( 32 \times 32 \) color images with objects from 10 and 100 classes respectively. Both are split into a
DGD: Densifying the Knowledge of Neural Networks with Filter Grafting and Knowledge Distillation

| Dataset | Threshold | The number of valid filters | The number of invalid filters | Method | Density of valid filters | Density of invalid filters | Accuracy on test set |
|---------|-----------|-----------------------------|-------------------------------|--------|-------------------------|--------------------------|----------------------|
| CIFAR-10 | 0.1       | 1936                        | 96                            | baseline | 6.982                  | 0.003                   | 93.64                |
|         |           |                              |                               | distillation | **13.648** | 0.000                   | **93.70**             |
|         |           |                              |                               | grafting | 6.427                  | **1.633**               | **94.14**            |
|         | 1         | 1920                        | 112                           | baseline | 7.035                  | 0.095                   | 93.64                |
|         |           |                              |                               | distillation | **13.762** | 0.000                   | **93.70**             |
|         |           |                              |                               | grafting | 6.460                  | **1.757**               | **94.14**            |
| CIFAR-100 | 0.1      | 1968                        | 64                            | baseline | 12.086                 | 0.001                   | 71.26                |
|         |           |                              |                               | distillation | **26.914** | 0.000                   | **71.92**             |
|         |           |                              |                               | grafting | 10.802                 | **1.765**               | **72.60**            |
|         | 1         | 1961                        | 71                            | baseline | 12.126                 | 0.075                   | 71.26                |
|         |           |                              |                               | distillation | **27.011** | 0.000                   | **71.92**             |
|         |           |                              |                               | grafting | 10.830                 | **1.879**               | **72.60**            |

Table 1. This table records the density of valid filters and invalid filters for each method.

50000 images train set and a 10000 images test set. The Top-1 accuracy is used as the evaluation metric.

Training setting: The baseline training settings are listed as follows: mini-batch size (256), optimizer (SGD), initial learning rate (0.1), momentum (0.9), weight decay (0.0005), number of epochs (200) and learning rate decay (0.1 at every 60 epochs). Standard data augmentation is applied to the dataset. For knowledge distillation hyper-parameters, we set $\alpha_s = \alpha_t = 1$ and $\tau = 2$ in (3), which are consistent with the work (Yang et al., 2019). For the hyper-parameters regarding filter grafting, we use the same setting from (Fanxu et al., 2020), where grafting is performed at the end of each epoch with $A = 0.4$ and $d = 500$ in (6). We use ResNet-56 as the baseline model. Filter grafting involves two ResNet-56 networks that learn knowledge from each other and one network is selected for testing. Knowledge distillation uses ResNet-110 as the teacher and ResNet-56 as the student. After training, the student network is used for testing.

It is worth noting that we identify a filter as valid or invalid by the baseline model with a threshold. To see how grafting and distillation influence filters, the number of valid and invalid filters is fixed for baseline, grafting and distillation (i.e., the denominator of (1) is the same for each method when calculating density). The results are listed in Table 1. We can see that both knowledge distillation and filter grafting improve the accuracy of the baseline model. However, they behave differently in the filter level. For knowledge distillation, it greatly densifies the knowledge of valid filters and sparsifies invalid filters. For example, the $l_1$ norms of invalid filters trained by distillation are very close to 0. On the other hand, filter grafting mostly densifies invalid filters since the $l_1$ norms of invalid filters are prominently improved by the grafting algorithm. We further visualize the filters of the networks trained by each method in Figure 3.

Figure 3. This figure shows the filter weight of each layer. Each filter is represented by a colored cube which is relational to the value of its $l_1$ norm. The black cubes represent the filters whose $l_1$ norm are very close to 0. (best viewed in color)

3. We can see that distillation could globally densify the knowledge of the network. However, the network trained by distillation has more invalid filters ($l_1$ norm close to 0) than baseline. In contrast, grafting could greatly densify the invalid filters and keep more filters functional in networks. This observation means that the two methods boost the neural network in an opposite way which is naturally complementary. So it inspires us to design a unified framework to further enhance DNNs. In the next section, we introduce our DGD framework that unites filter grafting and knowledge distillation in a single framework.

4. DGD: A Unified Framework Improving Knowledge Density of Networks

In Section 3, it is observed that knowledge distillation and filter grafting are complementary in terms of the filter. Thus it motivates us to design a unified framework that improves
the knowledge density of networks. We propose our DGD in Figure 4. In DGD, each student learns knowledge from teachers and other students. Suppose there are $C$ students and $M$ teachers in DGD, $s_c$ is the $c$-th student and $t_m$ is the $m$-th teacher. Except for the base cross-entropy loss, the training of the student has two improvements:

- **Learning knowledge from teachers:** We add KD-loss in loss function which is consistent with the vanilla knowledge distillation. This step helps students densify valid filters inside the network. Following the formulation in 3.2, we denote $p^k_t(x_i)$ and $p^m_t(x_i)$ as the probability of class $k$ for sample $x_i$ given student network $s_c$ and teacher network $t_m$ respectively. For the training of $s_c$, the total loss can be expressed as:

$$L_{s_c} = L_{C_{s_c}} + L_{KD_{t_1 \rightarrow s_c}},$$

where $L_{C_{s_c}}$ is the typical cross entropy loss:

$$L_{C_{s_c}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} I(y_i, k) \log(p^k_{s_c}(x)).$$

The indicator function $I$ is defined as

$$I(y_i, k) = \begin{cases} 1 & y_i = k \\ 0 & y_i \neq k \end{cases}$$

$L_{KD_{t_1 \rightarrow s_c}}$ is the knowledge distillation loss defined as:

$$L_{KD_{t_1 \rightarrow s_c}} = -\sum_{i=1}^{N} \sum_{k=1}^{K} p^k_t(x_i) \log(p^c_{s_c}(x_i)),$$

where $\bar{p}^k_t(x)$ is the average probability of all the teachers given $x_i$:

$$\bar{p}^k_t(x) = \frac{1}{M} \sum_{m=1}^{M} p^m_t(x).$$

- **Learning knowledge from other students:** This step is different with the work (Fanxu et al., 2020) when calculate the weighting coefficient $\alpha$ in (5). In the traditional grafting algorithm, the weighting coefficient is determined by the information of each student network (students with more information have larger weights than the other). However, since we have teacher networks besides student networks guiding the learning process, the weighting coefficient can be determined more efficiently. In DGD, students that are more similar to the teacher are given higher importance. It means that a bad student should learn more from a good student. We find this weighting strategy is better than the original one and could improve the invalid filters more effectively.

Specifically, let $W_{s_c}$ denotes the weights of $s_c$ and define $S_{t \leftrightarrow s_c}$ as the similarity between $t$ and $s_c$, then

$$S_{t \leftrightarrow s_c} = -D_{KL}(\bar{p}^t||p^c),$$

where $D_{KL}(\bar{p}^t||p^c)$ is the KL-divergence between the outputs of $t$ and $s_c$:

$$D_{KL}(\bar{p}^t||p^c) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \bar{p}^k_t(x_i) \log \left( \frac{\bar{p}^k_t(x_i)}{p^c_k(x_i)} \right).$$

When performing grafting, the weights of the $s_c$ are refined by the previous student:

$$\begin{cases} W_{s_c} = \alpha W_{s_c} + (1 - \alpha) W_{s_{c-1}} & 1 \leq c \leq C \\ W_{s_c} = \alpha W_{s_c} + (1 - \alpha) W_{s_C} & c = 1 \end{cases}$$

where $\alpha$ is determined by:

$$\alpha = A \times (\arctan(c \times (S_{t \leftrightarrow s_c} - S_{t \leftrightarrow s_{c+1}}))) + 0.5.$$
DGD: Densifying the Knowledge of Neural Networks with Filter Grafting and Knowledge Distillation

| Teacher   | Student   | Method                  | CIFAR-10     | CIFAR-100    |
|-----------|-----------|-------------------------|--------------|--------------|
| —         | ResNet-32 | baseline                | 93.16 ± 0.21 | 69.72 ± 0.38 |
| ResNet-56 | filter    | 93.29 ± 0.19            | 71.00 ± 0.42 |
|           | grafting  | 93.18 ± 0.31            | 70.70 ± 0.35 |
|           | knowledge | DGD 93.77 ± 0.28        | 72.49 ± 0.33 |
| ResNet-56 | baseline  | 93.64 ± 0.20            | 71.26 ± 0.39 |
|           | filter    | 94.14 ± 0.28            | 72.60 ± 0.32 |
|           | grafting  | 93.70 ± 0.19            | 71.92 ± 0.21 |
|           | knowledge | DGD 94.3 ± 0.19         | 73.37 ± 0.31 |
| ResNet-110| baseline  | 92.25 ± 0.31            | 71.95 ± 0.48 |
|           | filter    | 93.53 ± 0.32            | 72.48 ± 0.43 |
|           | grafting  | 92.95 ± 0.38            | 73.63 ± 0.46 |
|           | knowledge | DGD 94.15 ± 0.32        | 74.85 ± 0.42 |

Table 2. This table records the accuracy of CIFAR-10 and CIFAR-100 for each method. Grafting trains two student networks simultaneously while distillation maintains one teacher network and one student network. DGD involves one teacher network and two students networks. For each method, we use the single student network for testing.

It is easier to determine which student is better with the teacher, since the teacher network is usually exceeding over the students. The measure of similarity between each student and the teacher provides a convenient way for comparing the quality of students. 2) In original grafting algorithm (Fanxu et al., 2020), $\alpha$ varies for each layer (see (5)). However, for DGD, we regard the weights of all the layers as a whole to perform grafting which is more fast and efficient. It is worth noting that grafting is performed at the end of each epoch and distillation is performed at each optimization step. The frequencies of applying these two methods in DGD are different.

Testing for DGD: DGD involves training multiple student networks. But each student has the same network structure and learn knowledge from other students and teachers. Thus at the end of the training, the performance for each student network is quite similar to each other. To avoid ambiguity, we always select the first student network for testing in the remaining experiments.

5. Experiments

This section is arranged as follows: in Section 5.1, we perform DGD on the image classification task. In Section 5.2, we perform DGD on the person re-identification task. Section 5.3 extend DGD to multiple teachers and multiple students. In Section 5.4, we evaluate DGD in the filter level to prove that DGD does densify the knowledge of both valid and invalid filters. All the experiments are reproducible and the code is available at supplementary material.

5.1. Classification Task on CIFAR-10 and CIFAR-100

We compare DGD with the baseline, knowledge distillation and filter grafting on CIFAR-10 and CIFAR-100 datasets. The training setting of the baseline, knowledge distillation and filter grafting is introduced in Section 3.4. For DGD, we use one teacher network and two student networks in this experiment. Knowledge distillation is performed at each optimization step while grafting is performed at the end of each epoch. At the end of the training, we select one student network for testing. For each method on each dataset, we do five runs and report the mean and std of the accuracy. The results are shown in Table 2. We can see that DGD gives the best results on both CIFAR-10 and CIFAR-100 datasets. Especially for MobileNetV2, the student network trained by DGD outperforms the baseline by about 3 percent on CIFAR-100 and 2 percent on CIFAR-10.
DGD: Densifying the Knowledge of Neural Networks with Filter Grafting and Knowledge Distillation

| Teacher | Student | Method                | Market-1501 | Duke   |
|---------|---------|-----------------------|-------------|--------|
| —       | ResNet-18 | baseline              | 50.5 ± 0.1  | 61.9 ± 0.3 |
| ResNet-34 | —      | filter grafting       | 52.6 ± 0.2  | 64.9 ± 0.2 |
|         |         | knowledge distillation| 54.3 ± 0.1  | 66.5 ± 0.1 |
|         |         | DGD                   | **56.5 ± 0.4** | **68.8 ± 0.2** |
| —       | ResNet-34 | baseline              | 55.0 ± 0.1  | 66.7 ± 0.3 |
|         |         | filter grafting       | 57.2 ± 0.1  | 68.3 ± 0.1 |
|         |         | knowledge distillation| 58.3 ± 0.2  | 70.3 ± 0.2 |
|         |         | DGD                   | **59.4 ± 0.2** | **71.8 ± 0.1** |

Table 3. This table records the mAP of Market-1501 and Duke for each method. Grafting trains two student networks simultaneously while distillation maintains one teacher network and one student network. DGD involves one teacher network and two student networks. For each method, we use the single student network for testing.

The reason may be that MobileNetV2 is based on depth separable convolutions, thus the filters may learn insufficient knowledge. Since DGD could densify the knowledge of the filters, MobileNetV2 trained by DGD could learn better than vanilla baseline training.

We further depict the accuracy with the training epochs of each method in Figure 5. It’s interesting that grafting mostly influences the model at the early training epochs while distillation mostly makes an impact at later stages. Also, the network trained by DGD achieves the best accuracy among all the methods.

### 5.2. Recognition Task on Market-1501 and Duke

In this section, we further evaluate DGD on the person re-identification tasks (ReID). ReID is an open set retrieval problem under distributed multi-camera surveillance, aiming to match people appeared in different non-overlapping camera views (Ye et al., 2020; Leng et al., 2019; Zheng et al., 2016). We conduct experiments on two popular person ReID datasets: Market-1501 (Zheng et al., 2015) and Duke (Ristani et al., 2016). Market-1501 contains 32668 images of 1501 identities captured from six camera views, with 751 identities for training and 750 identities for testing. Duke contains 36441 images of 1404 identities captured from eight camera views, with 702 identities for training and 702 identities for testing The hyper-parameter setting for the experiment is consistent with (Zhou et al., 2019): mini-batch size (32), pretrained (True), optimizer (amsgrad), initial learning rate (0.1), learning rate decay (0.1 at every 20 epochs) and number of epochs (60). The mAP (mean average precision) is adopted as the evaluation metric. Results are shown in Table 3. We can see that DGD could consistently improve the performance for recognition tasks.

### 5.3. Extending DGD to multiple teachers and students

In Section 5.1 and Section 5.2, DGD only considers one teacher network and two student networks. However, this is the simplest form of this framework. We find that DGD could further improve the network by bringing more teachers and students to the framework. In this experiment, we use ResNet-110 as the teacher and MobileNetV2 as the student. The model accuracy with the number of teachers and students in DGD is listed in Table 4. It can be found that as we raise the number of teachers and students, the model accuracy increases. The results have shown that given more networks, there exists great potential for the DGD framework.

### 5.4. Evaluating DGD in the filter level

In order to show that the network trained by DGD does densify both valid and invalid filters, we calculate the density of the valid and invalid filters in Table 5 and plot all the filters’ $l_1$ norm in Figure 6. The results show that DGD could greatly densify both valid and invalid filters given a fixed model structure. As we state that distillation and grafting are complementary in the filter level, their combination could boost a filter-efficient network. The experiments in classification and recognition tasks also confirm this statement.
| Dataset    | Threshold | Method | Density of valid filters | Density of invalid filters | Accuracy on test set |
|------------|-----------|--------|--------------------------|---------------------------|----------------------|
| CIFAR-10   | 0.1       | baseline | 6.982                    | 0.003                     | 93.64                |
|            |           | DGD    | 12.117                   | 1.385                     | 94.30                |
|            | 1         | baseline | 7.035                    | 0.095                     | 93.64                |
|            |           | DGD    | 12.191                   | 1.635                     | 94.30                |
| CIFAR-100  | 0.1       | baseline | 12.086                   | 0.001                     | 71.26                |
|            |           | DGD    | 18.064                   | 0.654                     | 73.37                |
|            | 1         | baseline | 12.126                   | 0.075                     | 71.26                |
|            |           | DGD    | 18.119                   | 0.842                     | 73.37                |

Table 5. This table records the density of valid filters and invalid filters for each method. The backbone is ResNet-56.

Figure 6. This figure shows all the filters’ \( l_1 \) norm for baseline and DGD. (best viewed in color)

6. Ablation study

**Strategy on calculating weighting coefficient**: Instead of determining the weighting coefficient \( \alpha \) by the entropy of each student network (Equation (6)), we use the similarity of the student and the teacher to evaluate which student is better (Equation (15)). Table 6 records the model accuracy trained by DGD framework with different strategies. Results show that ‘similarity’ is a better strategy for the DGD framework.

| Teacher    | Student    | similarity | entropy   |
|------------|------------|------------|-----------|
| ResNet-56  | ResNet-32  | 93.77      | 93.64     |
| ResNet-110 | ResNet-56  | 94.30      | 94.04     |
| ResNet-110 | MobileNetV2| 94.15      | 93.17     |

Table 6. This table records the model accuracy trained by DGD framework with different weighting strategy. We conduct experiments with variant teacher and student on CIFAR-10 dataset.

**Sensibility on Hyper-parameters of Distillation**: We further evaluate the model sensibility on the hyper-parameter of knowledge distillation. Temperature scaling parameter \( T \) from (3) is chosen to conduct experiments. Table 7 shows the model accuracy with variant temperature scaling in DGD framework. We find the model trained by DGD is stable to variant temperature scaling parameters.

| Temperature Scaling | CIFAR-10 | CIFAR-100 |
|---------------------|----------|-----------|
| 2                   | 93.77    | 72.49     |
| 3                   | 93.80    | 72.54     |
| 4                   | 93.49    | 72.94     |
| 5                   | 93.77    | 72.56     |

Table 7. This table records the model accuracy trained by DGD framework with different temperature scaling. The teacher network is ResNet-56 and the student network is ResNet-32.

**Sensibility on Hyper-parameters of Grafting**: We evaluate the model sensibility on the hyper-parameter of grafting. \( A \) and \( d \) from (6) are chosen to conduct experiments. Table 8 show the model accuracy with variant \( A \) and \( d \) in DGD framework. We find the model trained by DGD is also relatively stable to the \( A \) and \( d \). This suggests that the model trained by DGD is robust to the hyper-parameters.

| \( A \) | \( d \) | CIFAR-10 | CIFAR-100 |
|--------|--------|----------|-----------|
| 0.2    | 500    | 93.66    | 72.67     |
| 0.4    | 500    | 93.51    | 72.78     |
| 0.8    | 500    | 93.65    | 72.46     |
| 0.4    | 10     | 93.75    | 73.11     |
| 0.4    | 10     | 93.46    | 73.01     |
| 0.4    | 100    | 93.71    | 72.3      |

Table 8. This table records the model accuracy trained by DGD framework with different \( A \) and \( d \). The teacher network is ResNet-56 and the student network is ResNet-32.

7. Conclusion and Discussion

In this paper, we evaluate knowledge distillation and filter grafting in the filter level and find these two techniques are surprisingly complementary: distillation mostly enhances the knowledge of valid filters while grafting mostly reacti-
vates invalid filters. This observation leads to a better understanding of neural networks and guides us to design a unified DGD training framework. The network trained by DGD could both densify the knowledge of valid and invalid filters, boosting the accuracy of neural networks to a higher level. There are some future directions to be considered: 1) In current DGD framework, students have the same network structures with each other. How can we extend DGD to heterogeneous structures for student network? 2) Filter pruning also leads to a filter-efficient DNN. Can we further help pruning process with DGD framework?

References

Ba, J. and Caruana, R. Do deep nets really need to be deep? In Advances in neural information processing systems, pp. 2654–2662, 2014.

Batra, T. and Parikh, D. Cooperative learning with visual attributes. arXiv preprint arXiv:1705.05512, 2017.

Bucilu, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 535–541. ACM, 2006.

Cho, J. H. and Hariharan, B. On the efficacy of knowledge distillation. In Proceedings of the IEEE International Conference on Computer Vision, pp. 4794–4802, 2019.

Fanxu, M., Hao, C., Ke, L., Zhixin, X., Xing, S., Rongrong, J., and Guangming, L. Filter grafting for deep neural networks. arXiv preprint arXiv: 2001.05868, 2020.

Graves, A., Mohamed, A.-r., and Hinton, G. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing, pp. 6645–6649. IEEE, 2013.

He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T.-Y., and Ma, W.-Y. Dual learning for machine translation. In Advances in neural information processing systems, pp. 820–828, 2016.

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

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097–1105, 2012.

Leng, Q., Ye, M., and Tian, Q. A survey of open-world person re-identification. IEEE Transactions on Circuits and Systems for Video Technology, 2019.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710, 2016.

Lopes, R. G., Fenu, S., and Starner, T. Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535, 2017.

Lotter, W., Kreiman, G., and Cox, D. Deep predictive coding networks for video prediction and unsupervised learning. arXiv preprint arXiv:1605.08104, 2016.
Mirzadeh, S.-I., Farajtabar, M., Li, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*, 2019.

Nayak, G. K., Mopuri, K. R., Shaj, V., Babu, R. V., and Chakraborty, A. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114*, 2019.

Parisotto, E., Ba, J. L., and Salakhutdinov, R. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

Phuong, M. and Lampert, C. Towards understanding knowledge distillation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5142–5151, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/phuong19a.html.

Ristani, E., Solera, F., Zou, R., Cucchiara, R., and Tomasi, C. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pp. 17–35. Springer, 2016.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Suau, X., Zappella, L., Palakkode, V., and Apostoloff, N. Principal filter analysis for guided network compression. *arXiv preprint arXiv:1807.10585*, 2018.

Tang, Z., Wang, D., and Zhang, Z. Recurrent neural network training with dark knowledge transfer. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5900–5904. IEEE, 2016.

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

Yang, H., Kang, G., Dong, X., Fu, Y., and Yi, Y. Soft filter pruning for accelerating deep convolutional neural networks. In *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, 2018.

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., and Hoi, S. C. Deep learning for person re-identification: A survey and outlook. *arXiv preprint arXiv:2001.04193*, 2020.

Zhang, X. and LeCun, Y. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015.
Historical ICML Locations and Numbers of Accepted Papers

Ann Arbor, MI
Ithaca, NY
Austin, TX
Evanston, IL
Aberdeen, Scotland
Amherst, MA
New Brunswick, NJ
Tahoe, CA
Bari, Italy
Nashville, TN
Madison, WI
Bled, Slovenia
Stanford, CA
Williamstown, CA
Sydney, Australia
Washington, DC
Banff, Canada
Bonn, Germany
Pittsburgh, PA
Corvalis, OR
Helsinki, Finland (estimated)