Self-boosting for Feature Distillation

Yulong Pei[1] , **Yanyun Qu**[1*] , **Junping Zhang**[2]
[1]Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Fujian, China
[2]Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China
cieusy@qq.com, yyqu@xmu.edu.cn, jpzhang@fudan.edu.cn

Abstract

Knowledge distillation is a simple but effective method for model compression, which obtains a better-performing small network (Student) by learning from a well-trained large network (Teacher). However, when the difference in the model sizes of Student and Teacher is large, the gap in capacity leads to poor performance of Student. Existing methods focus on seeking simplified or more effective knowledge from Teacher to narrow the Teacher-Student gap, while we address this problem by Student's self-boosting. Specifically, we propose a novel distillation method named Self-boosting Feature Distillation (SFD), which eases the Teacher-Student gap by feature integration and self-distillation of Student. Three different modules are designed for feature integration to enhance the discriminability of Student's feature, which leads to improving the order of convergence in theory. Moreover, an easy-to-operate self-distillation strategy is put forward to stabilize the training process and promote the performance of Student, without additional forward propagation or memory consumption. Extensive experiments on multiple benchmarks and networks show that our method is significantly superior to existing methods.

1 Introduction

Knowledge distillation (KD) is a hot topic in deep learning. With the continuous development of portable devices, the demand of cost-efficient and well-behaved deep models is increasing, such as deep object-detection models, deep segmentation models and so on. Knowledge distillation is a useful tool for model compression, which exploits additional information of the well-trained large network (Teacher) to help the small network (Student) to train. It simply and effectively improves the performance of small deep models.

Though KD makes promising results in applications of computer vision, Mirzadeh et al. [Mirzadeh *et al.*, 2020] found that Student cannot imitate Teacher perfectly when the model size of Teacher is considerable large. When the gap of Teacher and Student in model size is large, the performance of Student is far worse than that of Teacher. Thus, Teacher Assistant (TA) is proposed to alleviate the gap between Teacher and Student in [Mirzadeh *et al.*, 2020]. However, TA needs to add an additional network to assist Teacher to guide Student, which is time consuming and high resource consumption.

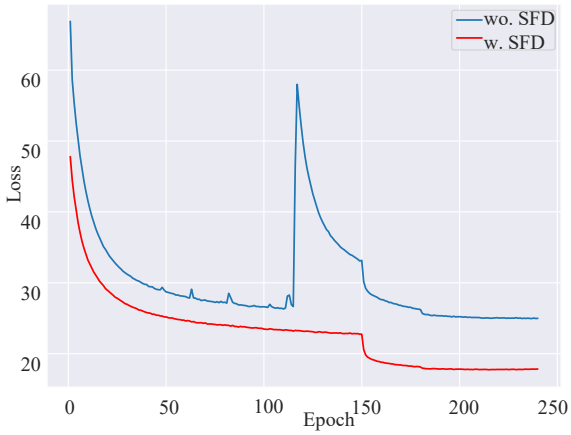


Figure 1: Comparison of training loss curves with and without SFD.

Some latest methods focus on improving Teacher's guidance to Student without the TA network. In [Xu *et al.*, 2020a], the feature from Teacher is normalized in order to appropriately simplify learning goals of Student. In [Yue *et al.*, 2020], an additional feature matching optimization needs to be performed at each iteration. Obviously, such methods may cause part important information from Teacher to be lost. Besides, some methods [Kim *et al.*, 2018] add convolution modules on Teacher, which requires additional pre-training steps.

Different from the abovementioned methods which focus on lowering the capacity of Teacher or exploring novel knowledge, we propose a novel distillation method named Self-boosting Feature Distillation (SFD) which *enhances the ability of Student by self-boosting to bridge the gap of Teacher and Student*. In other words, we aim to improve Student's learning ability by Student's self-boosting, rather than reducing the quality of Teacher's knowledge. SFD contains two aspects: feature boosting and model-parameter boosting. Concretely, as for feature boosting, we adopt fea-

*Corresponding Author.

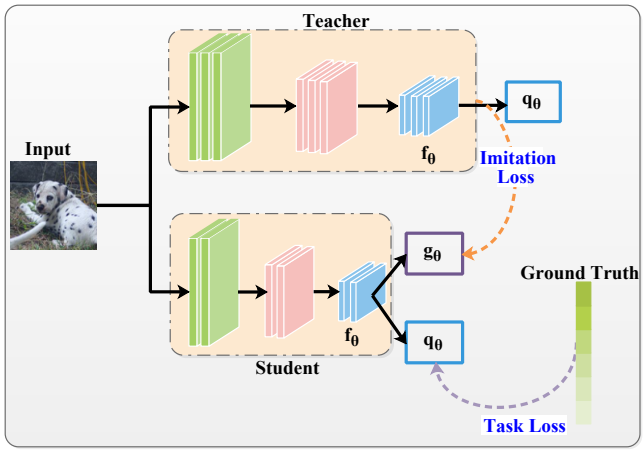


Figure 2: The overall framework of SFD.

ture integration strategy to enhance the discriminability of Student's feature by a carefully designed feature integration module. Student's integrated feature is encouraged to imitate Teacher's original feature, which builds a bridge between Student and Teacher and prompts Student to adaptively pay attention to Teacher's useful information. Note that our feature integration module is only used during the training phase, which is jointly optimized with Student and introduces a little computation. As for model-parameter boosting, we propose an easy-to-operate self-distillation method which does not require additional forward propagation or memory, to stabilize training process and promote Student's behavior. Compared with previous methods, our method does not require additional pre-training steps, while retaining Teacher's information to the greatest extent. Furthermore, SFD can be explained in theory while other methods are only explained empirically. As shown in Figure 1, SFD helps Student to learn more stable and converge faster.

Our contributions can be summarized as follows:

- We propose a novel distillation method called Self-boosting Feature Distillation (SFD), which bridges the gap between Teacher and Student by self-boosting in feature integration and self-distillation of Student. Unlike the existing methods which focus on lowering Teacher's capability, SFD improves the capability of Student.
- We design three feature integration modules to improve the discriminability of Student, in order to reduce the difference between Teacher and Student in model discrimination. Besides, self-distillation is proposed to further promote the convergency of Student, in which only the parameters of the previous model are used, so no additional forward propagation or memory is required.
- Unlike the existing methods which bridge the Teacher-Student gap experimentally, we explain SFD in theory of Richardson extrapolation: the feature integration increases the order of convergence.
- The proposed method is evaluated on multiple benchmarks and networks. Experimental results show that our method greatly enhances the performance of Student.

2 Related Work

2.1 Knowledge Distillation

Most researches focus on exploring diverse Teacher knowledge. In [Komodakis and Zagoruyko, 2017], the difference between the attention maps of Teacher and Student is minimized to optimize Student. In [Park *et al.*, 2019], multiple outputs of Teacher is treated as a structural unit, and Student is encouraged to learn Teacher's structured information. Variational Information Distillation (VID) [Ahn *et al.*, 2019] defines the optimal transfer performance of middle layers as maximizing the mutual information between Teacher and Student. Contrastive Representation Distillation (CRD) [Tian *et al.*, 2020] captures the relevance of instances and higher-order output dependence through the transfer loss based on contrastive learning. We do not mine new Teacher knowledge, but simply utilize the network's features and weights.

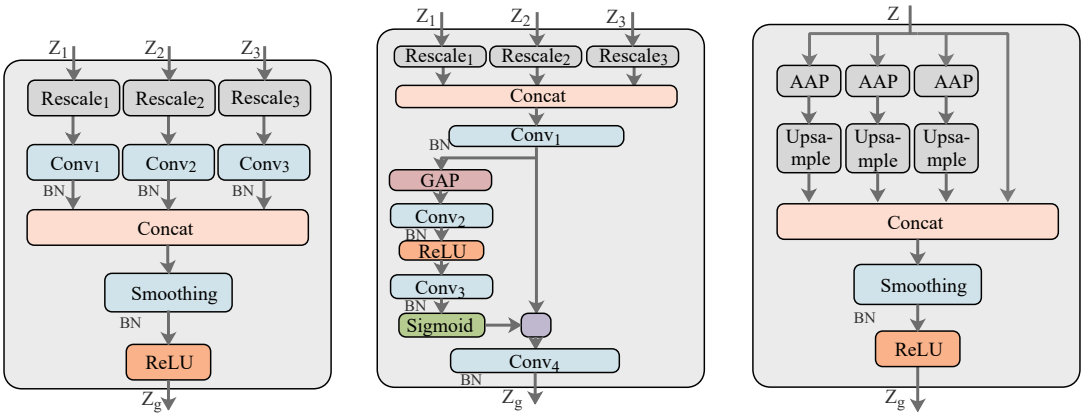
Some methods try to appropriately simplify Teacher's knowledge. In [Kim *et al.*, 2018], Teacher's information is showed to be difficult for Student to understand, so Teacher's middle-layer features are transformed into a simpler representation by a paraphraser. Xu et al. [Xu *et al.*, 2020a] proposed to decompose features into direction and magnitude, and encourage Student to learn the direction of Teacher. Recently, Matching Guided Distillation (MGD) [Yue *et al.*, 2020] argues to pose matching features of Teacher and Student as an assignment problem. These methods may cause missing of Teacher's knowledge to varying degrees, and some even require additional pre-training steps (such as FT [Kim *et al.*, 2018]) to train additional modules. We only do some transformations on Student's feature, so the transformation module can be trained simultaneously with Student, and avoid the missing of Teacher's information to the greatest extent.

2.2 Feature Integration

Feature integration is mainly used in object detection and semantic segmentation. Feature Pyramid Networks (FPN) [Lin *et al.*, 2017] achieve a comparable effect with the image pyramid algorithm by accumulating the shallow and deep features; In [Li and Zhou, 2017], resized features from different layers with different resolutions are concatenated, followed by some downsampling blocks, which forms the new feature pyramid; DeepLab [Chen *et al.*, 2018] utilizes dilated convolutions for multi-scale feature extraction to obtain richer feature information. We perform feature integration at the middle layers of Student to generate more discriminative features.

2.3 Self-distillation

Self-distillation is a kind of distillation using the information of Student itself. In [Xu and Liu, 2019], the authors propose a method which transfers knowledge between different distorted versions of the same training data. The actual batch size is twice that of conventional training (there are two versions of an image in each batch), which significantly increases the memory. Snapshot Distillation [Yang *et al.*, 2019] proposes to use the model of a previous time step as Teacher and its output as transferable knowledge. Xu et al. [Xu *et al.*, 2020b] take the average of the parameters from



(a) Layer-wise Integration (b) Attention Integration (c) Receptive-field Integration

Figure 3: The architecture of three different modules based on feature integration.

Student's past K time steps as Teacher, which also needs to calculate Teacher's output. Most methods of self-distillation inevitably increase the forward propagation calculation and memory consumption.

3 Methodology

Figure 2 shows the framework of SFD. There are two streamlines: Teacher and Student. The feature integration module bridges Teacher and Student. In the training stage, Student fits the feature of Teacher via distillation loss. Simultaneously, Student updates its own model parameters by self-distillation. In the testing stage, we only use Student to predict the class without the feature integration modules.

3.1 Feature Distillation

In our method, we perform feature distillation which uses Teacher's feature as the transferable knowledge for Student to learn, rather than using the output distribution of the classes. Considering that features from higher layers are more distinctive, we simply adopt the feature of the last stage of Teacher as knowledge to guide Student's learning.

Feature distillation can be treated as a multi-task learning (learning classification labels and Teacher's feature). There is a certain correlation between these two tasks. Student contains three components in the training phase: an encoder f_θ (CNN backbone), an integrator g_θ (feature integration module) and a predictor q_θ (classifier), as shown in Figure 2. Suppose that images X are fed into Student, and the outputs are formulated as follows:

$$Z = f_\theta(X), Z_g = g_\theta(Z), Z_q = q_\theta(Z). \tag{1}$$

where Z is the backbone's output, Z_g is the result of feature integration by the integrator, and Z_q is the classification result.

In previous feature-distillation methods, g_θ is usually a module consisting of 1×1 convolution layers for channel alignment. Obviously, 1×1 convolution can only make a linear combination of features from different channels, which cannot alleviate the semantic differences between features of Student and Teacher. In order to make full use of the feature integration module to bridge the Teacher-Student gap, we propose three solutions (as shown in Figure 3): Layer-wise Integration, Attention Integration and Receptive-field Integration.

Given a set of intermediate features $\{Z_l\}_{l=1}^3$ in different layers of Student, and the target feature $Y \in \mathbb{R}^{c\times h\times w}$ from Teacher's last stage, the three integration methods can be formulated as follows. Note that we simply choose features of the last three stages if Student has more than three stages.

Layer-wise Integration Inspired by Feature Pyramid Networks (FPN) [Lin *et al.*, 2017], we perform feature integration at the middle layers of Student. Layer-wise Integration (LI) is different from FPN in operation mechanism: LI only generates the feature with a single scale, and the integrated feature is not applied to subsequent classification. LI can be formulated as:

$$\begin{aligned} Z_l &= Conv_l(Rescale_l(Z_l)), l \in \{1,2,3\}, \\ Z &= Concat(Z_1, Z_2, Z_3), \\ Z_g &= Smoothing(Z). \end{aligned} \tag{2}$$

where $Rescale_l(\cdot)$ is a transform function to rescale features from different stages to the same scale ($h \times w$) as Teacher's feature, which is a simple downsampling (if larger) or upsampling (if smaller) operation. $Conv_l(\cdot)$ implements 1×1 convolution to reduce channels of features, and the numbers of output channels are $\frac{c}{4}$, $\frac{c}{4}$ and $\frac{c}{2}$, respectively. After that, the multi-layer integrated feature Z_g is obtained through concatenating, smoothing and batch normalization operations in sequence. Note that $Smoothing(\cdot)$ is a convolution with kernel size $c \times c \times 3 \times 3$.

Attention Integration LI module treats the features of each channel equally, but in fact they have different degrees of importance. To solve this problem, we design an attention module named Attention Integration (AI) which can be formulated as:

$$\begin{aligned} Z &= Concat(Rescale_l(Z_l)), l \in \{1,2,3\}, \\ Z &= Conv_1(Z) \\ W &= Sigmoid(Conv_3(ReLU(Conv_2(GAP(Z))))) \\ Z &= Conv_4(Z \odot W) \end{aligned} \tag{3}$$

where $Rescale(\cdot)$ resizes features to size of $h \times w$, $GAP(\cdot)$ denotes the global average pooling, and $\odot$ denotes the element-wise multiplication. $\{Conv_i\}_{i=1}^{4}$ represents bottleneck (1×1 convolution) layers, and kernel sizes are $(c_1 + c_2 + c_3) \times c_{mid} \times 1 \times 1$, $c_{mid} \times \frac{c_{mid}}{16} \times 1 \times 1$, $\frac{c_{mid}}{16} \times c_{mid} \times 1 \times 1$ and $c_{mid} \times c \times 1 \times 1$, respectively. Note that $c_{mid} = max(\frac{c_1+c_2+c_3}{16}, 96)$.

Receptive-field Integration Given the feature Z from the last stage of Student, and the target feature $Y \in \mathbb{R}^{c \times h \times w}$ from the last stage of Teacher, Receptive-field Integration (RI) can be expressed as:

$$\begin{aligned} Z_i &= AAP_i(Z), i \in \{1,2,3\} \\ Z_i &= Upsample_i(Z_i), i \in \{1,2,3\} \\ Z &= Concat(Z, Z_1, Z_2, Z_3) \\ Z_g &= Smoothing(Z) \end{aligned} \tag{4}$$

where $AAP_i(\cdot)$ is a transform function consisting of adaptive average pooling (AAP) and 1×1 convolution layers, where the kernel sizes of convolution layers are all $\frac{c}{4} \times \frac{c}{4} \times 1 \times 1$. $Upsample(\cdot)$ is a bilinear interpolation to rescale features to the size $h \times w$, and a rescaling operation is also conducted on Z if its scale is not equal to $h \times w$. Then the integrated feature of different receptive fields can be obtained through concatenation followed by smoothing and batch normalization. Note that the scales of Z_1, Z_2 and Z_3 are $\frac{h}{4} \times \frac{w}{4}$, $\frac{h}{2} \times \frac{w}{2}$ and $\frac{3\times h}{4} \times \frac{3\times w}{4}$, respectively.

3.2 Theoretical Analysis

In this part, we give a theoretical explanation that SFD can help to improve the performance of Student. According to Richardson extrapolation, we can get better numerical results by the linear combination of several numerical results at different inputs due to the increase of the convergence order of a model. Take the 1-D function as an example, according to Taylor expansion, a simple function $a(t)$ w.r.t t is calculated as:

$$\begin{aligned} a(t+\Delta t) &\approx a(t) + \frac{\partial a}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 a}{\partial t^2} \cdot \Delta t^2 \\ a(t-\Delta t) &\approx a(t) - \frac{\partial a}{\partial t}\Delta t + \frac{1}{2}\frac{\partial^2 a}{\partial t^2} \cdot \Delta t^2 \\ \frac{a(t+\Delta t) + a(t-\Delta t)}{2} &\approx a(t) + \frac{1}{2}\frac{\partial^2 a}{\partial t^2} \cdot \Delta t^2 \end{aligned} \tag{5}$$

It can be seen that the linear combination of two results at $a(t+\Delta t)$ and $a(t-\Delta t)$ increase the order of convergence.

We implement Richardson extrapolation on the function matrix of Student. Given Student's function matrix $\mathbf{A}(\cdot)$, then the feature maps (denoted as $\mathbf{A}(\mathbf{t_i})$) from Student's different layers can be regarded as values of the function with different parameters ($\mathbf{t_i}$)). Let $\mathbf{t_0}$ denote Student's optimal parameters to imitate Teacher, the function can be formulated as:

$$\begin{aligned} \mathbf{A}(\mathbf{t_i}) \approx \mathbf{A}(\mathbf{t_0}) + (\mathbf{t_i} - \mathbf{t_0})^\top \frac{\partial \mathbf{A}}{\partial \mathbf{t}} + \\ \frac{1}{2}(\mathbf{t_i} - \mathbf{t_0})^\top \frac{\partial^2 \mathbf{A}}{\partial \mathbf{t}^2}(\mathbf{t_i} - \mathbf{t_0}) \end{aligned} \tag{6}$$

Feature integration provides a trainable operation $\mathbf{B}$, so:

$$\begin{aligned} \sum_i \mathbf{B_i} \odot \mathbf{A}(\mathbf{t_i})) \approx \sum_i \mathbf{B_i} \odot \mathbf{A}(\mathbf{t_0}) + \sum_i \mathbf{B_i} \odot \\ (\mathbf{t_i} - \mathbf{t_0})^\top \frac{\partial \mathbf{A}}{\partial \mathbf{t}} + \frac{1}{2}\sum_i \mathbf{B_i} \odot (\mathbf{t_i} - \mathbf{t_0})^\top \frac{\partial^2 \mathbf{A}}{\partial \mathbf{t}^2}(\mathbf{t_i} - \mathbf{t_0}) \end{aligned} \tag{7}$$

where $\odot$ is the element-wise operation of matrices. Note that with feature integration, the term with one order of derivation can be eliminated by learning suitable $\mathbf{B}$, namely, $\sum_i \mathbf{B_i} \odot (\mathbf{t_i} - \mathbf{t_0})^\top \frac{\partial \mathbf{A}}{\partial \mathbf{t}} = 0$, so the function has a higher order of convergence. Therefore, SFD can help Student learn better.

3.3 Self-distillation

We propose a novel self-distillation method, which updates the parameters only by using those obtained by the previous epoch, without any additional forward propagation or memory consumption.

Given a network with a set of weights $\boldsymbol{\theta}$, and $\boldsymbol{\theta}_t$ denotes the network's parameters of the t-th epoch. After the t-th training epoch, we perform an additional weights updation:

$$\boldsymbol{\theta}_t \leftarrow \tau\boldsymbol{\theta}_t + (1-\tau)\boldsymbol{\theta}_{t-1}, t \geq 2 \tag{8}$$

Considering that the fluctuation range of parameters gets very small during the later stage of training, if Eq. 8 is used to update the parameters, the parameters hardly change. Therefore, we make a little modification:

$$\begin{aligned} \boldsymbol{\theta}_t &\leftarrow \tau\boldsymbol{\theta}_t + (1-\tau)\boldsymbol{\theta}'_{t-1}, t \geq 2 \\ \boldsymbol{\theta}'_{t-1} &\in \{hflip(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}_{t-1}\} \end{aligned} \tag{9}$$

where $hflip(\cdot)$ is a random horizontal flip with probability 0.5. Eq. 8 is named Self Distillation (SD), and Eq. 9 is named Random Self Distillation (RSD). Since the network is expected to be robust to the horizontal flip of the data, it is reasonable to randomly flip parameters of the previous epoch.

In fact, our self-distillation strategy conducts an exponential moving average of parameters from previous epochs. On one hand, this method can effectively use the information of the previous epoch model without introducing additional calculations and memory consumption; on the other hand, the exponential moving average can effectively improve the stability of the training process.

LI	AI	RI	SD	RSD	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet50 ShuffleNetV2
					73.26	71.98	71.82
✓					75.79	75.09	77.19
	✓				76.10	75.03	77.30
		✓			76.30	74.79	77.48
		✓	✓		**76.53**	75.01	77.72
		✓		✓	76.51	**75.14**	**78.02**

Table 1: Ablation study of different components on CIFAR-100.

3.4 Distillation Loss

In SFD, the loss function is used to guide feature distillation. Inspired by Overhaul [Heo *et al.*, 2019], we measure the distance between the feature before activation layer of Teacher

Network	baseline (1.0)	0.3	0.5	0.7	0.9	0.995	0.997	0.999	0.995-0.999
resnet32	71.14	71.57	71.64	71.69	71.70	71.77	71.77	71.73	**71.81**
ShuffleNetV2	71.82	72.15	72.49	72.86	72.90	73.02	72.97	73.06	**73.15**

Table 2: Comparison results with different networks in top-1 accuracy (%) with different τs on CIFAR-100.

Methods	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32x4 resnet8x4	vgg13 vgg8
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36
KD [2015]	74.92	73.54	70.66	70.67	73.08	73.33	72.98
AT [2017]	74.08	72.77	70.55	70.22	72.31	73.44	71.43
VID [2019]	74.11	73.30	70.38	70.16	72.61	73.09	71.23
CRD [2020]	75.48	74.14	71.16	71.46	73.48	75.51	73.94
Overhaul [2019]	75.55	74.87	70.27	70.54	72.86	74.30	72.42
MGD [2020]	75.93	74.75	70.43	70.85	72.49	74.22	72.29
Ours	**76.51**	**75.14**	**72.02**	**72.29**	**74.14**	**75.83**	**73.95**

Table 3: Comparison results in top-1 accuracy (%) on CIFAR-100. Teacher and Student have similar architectures.

and the integrated feature before ReLU of Student to conduct feature imitation. The distillation loss is formulated as:

$$L_{distill} = \sum_{i=1}^{c}\sum_{j=1}^{h}\sum_{k=1}^{w}\begin{cases} 0 & s_{ijk} \leq t_{ijk} \leq 0, \\ (t_{ijk} - s_{ijk})^2 & otherwise. \end{cases} \tag{10}$$

where s is Student's integrated feature before ReLU, and $t = max(Y, m)$ in which Y is Teacher's feature before ReLU and $m < 0$ is a margin value computed as expectation over all training samples.

4 Experiments

In this section, we evaluate SFD on several different benchmarks, including classification and fine-grained recognition. CIFAR-100 [Krizhevsky *et al.*, 2009] is a commonly used small dataset for classification, which contains $60,000$ RGB color images within 100 classes ($50,000$ training images and $10,000$ test images) with a resolution of 32×32. CUB-200 [Wah *et al.*, 2011] is a dataset for fine-grained recognition, which consists of $11,788$ images of different birds. ImageNet [Russakovsky *et al.*, 2015] is a large-scale classification benchmark which has around 1.2 million images in $1,000$ classes.

4.1 Experimental Settings

In experiments, we compare our method with several state-of-the-art methods[1]: standard KD [Hinton *et al.*, 2015], AT [Komodakis and Zagoruyko, 2017], VID [Ahn *et al.*, 2019], CRD [Tian *et al.*, 2020], Overhaul [Heo *et al.*, 2019] and MGD [Yue *et al.*, 2020] under multiple network structures to verify the effectiveness of our method. For the fairness of comparison, we use the experimental settings of the compared methods presented publicly by the authors. As for SFD, the weight of distillation loss is 3.

For all networks, we use stochastic gradient descent (SGD) optimizer with momentum 0.9 and weight decay 5×10^{-4}. On CIFAR-100, models are trained for 240 epochs with an initial learning rate of 0.05 and divided by 10 at epoch 150, 180 and 210, and standard data augmentation schemes (padding 4 pixels, random cropping, random horizontal flipping) are carried out. On ImageNet and CUB-200, the number of total epochs is 100 and 120 respectively, the learning rate is dropped by 0.1 per 30 epochs, and we perform random cropping and horizontal flipping as data augmentation. Following [Tian *et al.*, 2020], we use the model denotation: WRN-d-w is a Wide ResNet with depth d and width factor w; resnet$d$ is a CIFAR-style resnet with depth d and basic blocks; resnet$d$xw is a w times wider network; ResNet$d$ represents an ImageNet-style ResNet with depth d and bottleneck blocks.

[1]We used a reference implementation: https://github.com/HobbitLong/RepDistiller.git

4.2 Ablation Studies

Ablation studies are performed on CIFAR-100 to explore the influence of different solutions of feature integration and hyperparameters on the performance of models.

Different Components As shown in Table 1, all of the three feature integration methods we propose can greatly improve the performance of Student, whether Teacher and Student are homogeneous or heterogeneous. Considering the effects and complexity of the three feature integration solutions, we adopt RI as the final solution. Besides, both self-distillation methods are useful for improving Student's performance, and random self-distillation (RSD) behaves slightly better.

Hyperparameters of Self-distillation Different values of τ (in Eq. 9) have much influence on the performance of self-distillation. We select WRN-16-2 and ShufflenetV2 for experiments separately, and comparison of model accuracy with different τ is shown in Table 2. Obviously, τ with different values can help to improve the performance of models to some extent, and models reach the highest accuracy when τ is linear growth from 0.995 to 0.999 with current epochs.

4.3 Comparison with SOTAs

CIFAR-100 We conduct experiments including homogeneous and heterogeneous Teacher-Student combination. Compared with previous methods, our method provides consistent gains in all Teacher-Student frameworks and is significantly better than other methods. For example, in Table 3, when the Teacher-Student combination is resnet110

Methods	vgg13 MobileNetV2	ResNet50 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleNetV1	resnet32x4 ShuffleNetV2	WRN-40-2 ShuffleNetV1
Teacher	74.64	79.34	79.34	79.42	79.42	75.61
Student	64.60	64.60	70.36	70.50	71.82	70.50
KD [2015]	67.37	67.35	73.81	74.07	74.45	74.83
AT [2017]	59.40	58.58	71.84	71.73	72.73	73.32
VID [2019]	65.56	67.57	70.30	73.38	73.40	73.61
CRD [2020]	69.73	69.11	74.30	75.11	75.65	76.05
Overhaul [2019]	66.83	68.86	74.57	77.19	72.82	76.14
MGD [2020]	67.54	68.71	74.52	77.04	74.05	76.28
Ours	**70.23**	**70.91**	**74.96**	**77.90**	**77.94**	**77.31**

Table 4: Comparison results in top-1 accuracy (%) on CIFAR-100. Teacher and Student are heterogeneous.

	T: ResNet34	S: ResNet18	AT [2017]	KD [2015]	Online KD [2018]	CRD [2020]	CRD+KD [2020]	Ours
Top-1	26.70	30.25	29.30	29.34	29.45	28.83	28.62	**28.19**
Top-5	8.58	10.93	10.00	10.12	10.41	9.87	9.51	**9.49**

Table 5: Comparison results in top-1 and top-5 error rates (%) on the ImageNet validation set.

Methods	T: ResNet50 S: MobileNetV2	T: ResNet50 S: ShuffleNetV2
Teacher	79.82 / 93.79	79.82 / 93.79
Student	75.39 / 92.44	68.61 / 89.10
KD [2015]	76.48 / 93.56	71.69 / 90.33
AT [2017]	76.86 / 93.03	71.42 / 90.71
AB [2019]	76.92 / 93.46	71.78 / 90.52
Overhaul [2019]	78.31 / 94.36	72.58 / 91.96
MGD [2020]	79.36 / 94.32	74.05 / 92.54
Ours	**81.27 / 95.48**	**77.39 / 93.37**

Table 6: Comparison results in top-1 and top-5 accuracies (%) with SFD and other methods on CUB-200.

Methods	T: ResNet50 / S: ShuffleNetV2			
	Acc (%)	L_{cls}	KL div	CKA sim
Overhaul [2019]	76.42	1.3419	0.6354	0.8764
CRD [2020]	76.02	1.3084	0.6814	0.8781
Ours	**78.02**	**1.2365**	**0.6099**	**0.8856**

Table 7: Comparison in top-1 accuracy, classification loss, KL divergence and similarity on CIFAR-100 test set.

and resnet20, our method achieves the gain in accuracy by 0.83% compared with CRD (ranked second). In Table 4, when the Teacher-Student combination is resnet32x4 and ShuffleNetV2, the accuracy of our method is 2.29% higher than CRD (ranked second).

ImageNet On ImageNet, we use ResNet34 as Teacher and ResNet18 as Student. The results are shown in Table 5. It can be seen that SFD achieves the best performance with the gain of 0.43% against the second place CRD+KD. Obviously, our method also works well in large-scale classification scenarios.

CUB-200 As shown in Table 6, for all configurations, our method achieves significant accuracy gains over other methods used for comparison. When the Teacher-Student combination is ResNet50 and ShuffleNetV2, the top-1 accuracy of SFD is 3.34% higher than MGD (ranked second). Besides, when Teacher is ResNet50 and Student is MobileNetV2, the accuracy of Student even surpasses Teacher with the gain of 1.45%. This may be because Student is sufficient to learn from Teacher, and SFD allows Student to better balance the relationship between self-study and imitation.

4.4 Analysis

It is recognized that integrated features are more discriminative. While our method does not use the feature integration module in the inference phase, but directly adopt the original feature extracted by Student for classification. To verify that Student learns under SFD framework, we visualize Student's (ShuffleNetV2, distilled by a well-trained ResNet50) training loss curves with the number of epochs. As shown in Figure 1, under the action of SFD, the training process is more stable and Student converges faster.

Besides, we make some comparison of several items on the test set: top-1 accuracy, classification loss, feature distance, and CKA similarity [Kornblith *et al.*, 2019] between Student and Teacher. As shown in Table 7, our method achieves the best accuracy (1.60% higher than Overhaul), and has the lowest classification loss on the test set. Moreover, the KL divergence between outputs of Student and Teacher is the smallest, and the Teacher-Student similarity is the highest. Above all, our method makes Student imitate Teacher better.

5 Conclusion

In this paper, we tackle the Teacher-Student gap problem from a new perspective: self-boosting of Student rather than the previous methods lowering the level of Teacher. We propose the self-boosting feature distillation (SFD) method. To improve the learning ability of Student, self-boosting is conducted on Student, which contains two aspects: feature integration of its own feature and self-distillation on the parameters of Student, so Student adaptively learns from Teacher. SFD achieves state-of-the-art performance on multiple datasets with different Teacher-Student architectures. Theoretical analysis shows our method can improve the order of convergence. Extensive experiments shows that our method is significantly superior to other methods.

Acknowledgements

This work is supported by the National Key Research and Development Program of China No.2020AAA0108301, the National Natural Science Foundation of China under Grant 61876161, Shanghai Municipal Science and Technology Major Project (Grant No.2018SHZDZX01) and ZJLab.

References

[Ahn *et al.*, 2019] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[Chen *et al.*, 2018] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018.

[Heo *et al.*, 2019] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1921–1930, 2019.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems*, 2015.

[Kim *et al.*, 2018] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, pages 2760–2769, 2018.

[Komodakis and Zagoruyko, 2017] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017.

[Kornblith *et al.*, 2019] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4), 2009.

[Li and Zhou, 2017] Zuoxin Li and Fuqiang Zhou. FSSD: Feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017.

[Lin *et al.*, 2017] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[Mirzadeh *et al.*, 2020] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

[Park *et al.*, 2019] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[Tian *et al.*, 2020] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[Wah *et al.*, 2011] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. *California Institute of Technology*, 2011.

[Xu and Liu, 2019] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5565–5572, 2019.

[Xu *et al.*, 2020a] Kunran Xu, Lai Rui, Yishi Li, and Lin Gu. Feature normalized knowledge distillation for image classification. In *Proceedings of the European Conference on Computer Vision*, volume 1, 2020.

[Xu *et al.*, 2020b] Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. Improving BERT fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*, 2020.

[Yang *et al.*, 2019] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

[Yue *et al.*, 2020] Kaiyu Yue, Jiangfan Deng, and Feng Zhou. Matching guided distillation. *arXiv preprint arXiv:2008.09958*, 2020.