Revisiting Knowledge Distillation: An Inheritance and Exploration Framework

Zhen Huang[1, 2]*, Xu Shen[2], Jun Xing[3], Tongliang Liu[4], Xinmei Tian[1]†,
Houqiang Li[1], Bing Deng[2], Jianqiang Huang[2], Xian-Sheng Hua[2]†
[1]University of Science and Technology of China, [2]Alibaba Group
[3]University of Southern California, [4]University of Sydney
hz13@mail.ustc.edu.cn, junxnui@gmail.com, tongliang.liu@sydney.edu.au, {xinmei,lihq}@ustc.edu.cn,
{shenxu.sx,dengbing.db,jianqiang.hjq,xiansheng.hxs}@alibaba-inc.com

Abstract

Knowledge Distillation (KD) is a popular technique to transfer knowledge from a teacher model or ensemble to a student model. Its success is generally attributed to the privileged information on similarities/consistency between the class distributions or intermediate feature representations of the teacher model and the student model. However, directly pushing the student model to mimic the probabilities/features of the teacher model to a large extent limits the student model in learning undiscovered knowledge/features. In this paper, we propose a novel inheritance and exploration knowledge distillation framework (IE-KD), in which a student model is split into two parts - inheritance and exploration. The inheritance part is learned with a similarity loss to transfer the existing learned knowledge from the teacher model to the student model, while the exploration part is encouraged to learn representations different from the inherited ones with a dis-similarity loss. Our IE-KD framework is generic and can be easily combined with existing distillation or mutual learning methods for training deep neural networks. Extensive experiments demonstrate that these two parts can jointly push the student model to learn more diversified and effective representations, and our IE-KD can be a general technique to improve the student network to achieve SOTA performance. Furthermore, by applying our IE-KD to the training of two networks, the performance of both can be improved w.r.t. deep mutual learning.

1. Introduction

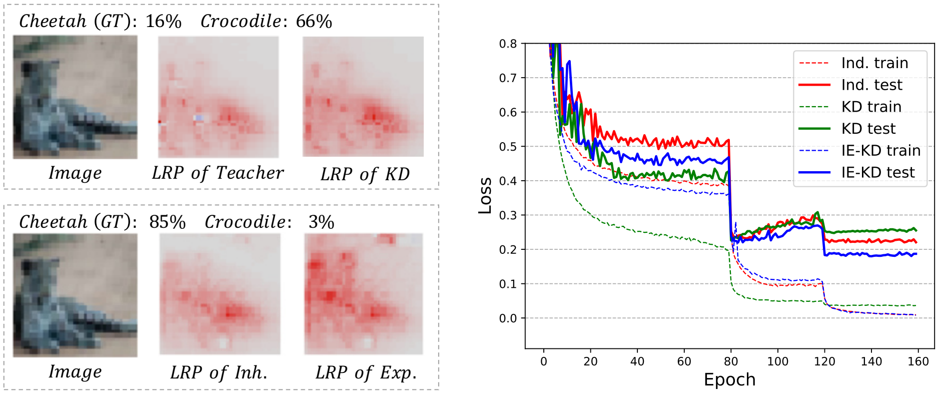
Knowledge distillation is one of the most popular methods for transferring knowledge from one network (teacher) to another (student). It was first proposed by Hinton *et al*. [10] to transfer knowledge from a large teacher network (or ensemble) to a small student network that is easier to deploy. It works by training the student to predict the target classification labels and mimic the class probabilities of the teacher, as these features contain additional information about how the teacher tends to generalize [10]. All recent distillation works follow this philosophy of an additional consistency control between the class probabilities or intermediate representations of the teacher network and the student network. KD [10] and Tf-KD [32] focus on the consistency of output class probabilities. AT [33], AB [13], FT [16], OD [12], FEED [22] and FitNet [24] propose different consistency controls of intermediate features. FSP [31] proposes a consistency control of the intra-similarities among intermediate features. In summary, all recent distillation methods differ in the metric of consistency between the student model and the teacher model.

However, directly pushing the student model to mimic the probabilities/features of the teacher model limits the student model in learning new knowledge/features. As shown in Fig. 1(a), the student model trained with KD learns very similar patterns compared with the well-trained teacher (more results will be shown in supplementary materials). In this case, the “cheetah” misclassified as a “crocodile” by the teacher model is also misclassified by the student model trained by KD. The model attributes most of its prediction to the tail of the “cheetah” which resembles a “crocodile”. As a result, the student network fails to incorporate new relevant patterns on ears and mouth that are quite discriminative between the “cheetah” and “crocodile”. Therefore, we need a mechanism to find more useful features for correct predictions that are omitted by the teacher network.

Intuitively, simply mimicking outputs of the teacher network will narrow the search space for the optimal parameters of the student network and lead to a poor solution from a feature learning view. Furthermore, we find that this phenomenon becomes more evident when transferring knowl-

*This work was done when the author was visiting Alibaba as a research intern.

†Corresponding author.



(a) Visualization of learned knowledge

(b) Loss curves

Figure 1. **Left**: Visualization of learned knowledge for classification, including the teacher network (LRP of Teacher), student network trained with KD (LRP of KD), inheritance part of IE-KD (LRP of Inh.) and exploration part of IE-KD (LRP of Exp.). LRP [21] is used to interpret the network by visualizing which pixels contribute how much to the classification. **Right**: Training loss (dotted lines) and testing loss (bold lines) on CIFAR-10 of the student network (ResNet-56) which is trained via independent learning (training-from-scratch), KD and IE-KD (using ResNet-20 as teacher network). For a fair comparison, KD and IE-KD correspond to FT and IE-FT here. Directly pushing the student network to mimic the outputs of the teacher network limits the student network in learning new knowledge. It even leads to a poor solution when the student network is larger than the teacher network (high training and testing loss at the same time).

edge from a small teacher network to a large student network (shown in Fig. 1(b)). According to the observation in [1, 5], small networks often have as sufficient capacity as large networks but represent the features in a more concise manner [24]. Therefore, a large network should not only mimic this compact representation with some of their parameters to reduce the redundancy of itself, but also should free other parameters to explore more different and complementary features to improve its diversity and generalization ability. Based on the aforementioned analyses, in this paper, we propose a novel inheritance and exploration knowledge distillation framework (*IE-KD*), to train a student network by partially following the knowledge from the teacher network and partially exploring for new knowledge that are complementary to the teacher network.

In our IE-KD, the knowledge is transferred by the two principles of consistency and diversity. Consistency ensures that the well learned knowledge encoded in the teacher network is successfully inherited by the student network. Diversity ensures that the student network can explore new features that are complementary to the inherited ones. The motivation of IE-KD comes from the theory of heredity in evolution [7]. Heredity involves inheritance and variation of traits. Evolution results from natural selection acting on diversity in populations, which originally stems from mutations. There are three key factors for evolution: a) inheritance of compact and effective traits from parents encoded by genes, b) new diversified genotypes generated from genetic mutations, and c) natural selection through stressful environments. Motivated by this, we split the student network into two parts: one inherits the compact and effective knowledge encoded by factors from the teacher network via consistency/inheritance loss (similarity), and the other is pushed to generate different features via diversity/exploration loss (dis-similarity). The supervised task (classification/detection) loss plays the role of natural selection, guiding the exploration part to converge to diverse yet effective features.

Another closely related motivation for IE-KD comes from the exploration of actions in Q-learning [20], and the popular AlphaGo [26], where half the actions follow the predictions of the policy network, and the other half are randomly sampled from the remaining action space that ensures adequate exploration of the state space. Besides, [4] proposes a similar form of loss function to attack the heat maps of one white-box DNN, making its attention focus on other regions of the image. Inspired by these insights, we propose our IE-KD framework to improve the training of student network, by exploring the new and undiscovered knowledge apart from the teacher-learned knowledge.

Overall, our IE-KD framework is generic and can be easily combined with existing distillation or mutual learning methods for training deep neural networks. Extensive experiments demonstrate that these two parts can jointly push the student model to learn more diversified and effective representations, and our IE-KD can be a general technique to improve the student network to achieve SOTA performance. Furthermore, by applying our IE-KD to the training of two networks, the performance of both can be improved w.r.t. deep mutual learning.

2. Related Work

In this paper, we focus on knowledge transfer between networks. All related works can be divided into three types: consistency control from a pre-trained teacher network to a student network by distillation, simultaneous learning of network pairs by consistency control, and self-distillation by teacher free regularization.

Consistency Control from a pre-trained teacher network or ensemble to a student network. Various approaches exist to transfer knowledge from a pre-trained large network or ensemble to an untrained small network, *i.e*., knowledge distillation. The transferred knowledge lies in a consistency of output probabilities (KD [10]), intermediate features (AT [33], AB [13], FT [16], OD [12], FEED [22], FitNet [24]), or similarities between intermediate features (FSP [31]). Each method differs in the metric of consistency, including KL divergence between output probabilities (KD [10], BAN [9]), regression with additional parameters between the mapping of intermediate features (FitNet [24]), $L1$ distance between projected factors (FT [16]), $L1$ distance between the pooled attentions (AB [13]), and $L2$ distance between rectified activations (OD [12]). FEED [22] proposed $L1$ distances between the features of an ensemble of teacher networks and the untrained small network. CRD [29] proposed a contrastive-based objective for transferring high-order dependencies in representational space between deep networks.

Simultaneous learning by consistency control among a group of untrained networks. Recently, researchers have proposed to relax the requirements of a pre-trained large network by starting with a pool of untrained networks and learns simultaneously with a consistency control. Deep mutual learning [35] shows that an ensemble of students could learn collaboratively and teach each other throughout the training process by consistency control of output probabilities. More recently, FFT [15], ONE [36] and CL [28] proposed consistency control between an ensemble of sub-network classifiers and each sub-network, where each sub-network mutually teaches one another in an online-knowledge distillation manner.

Teacher-free regularization. In Tf-KD [32], label smoothing regularization was introduced as a virtual teacher model for KD, without any additional peer networks needed. SD [30] proposed to use snapshots from earlier epochs as teacher model. These works still comply to the consistency between student network and referred targets, either manually designed or selected from snapshots.

In this study, we propose a new framework for transferring knowledge from the teacher network to a student network. Beyond the consistency control used in distillation and mutual learning, IE-KD further involves a diversity control. In addition, our IE-KD approach supports similar mutual learning between a group of networks and achieves much better performance.

3. Method

Fig. 2 illustrates the framework of our approach. The features of the student network is divided into two parts. One part (indicated by the orange color) is trained to mimic the compact features of the teacher network using an inheritance loss, and the other part (blue) is encouraged to learn new features different from the teacher network via an exploration loss. The supervised task (classification/detection) loss guides the exploration part to converge to diverse yet effective features. Overall, the student network is trained with the inheritance loss and the exploration loss, together with the conventional supervised target loss.

Since the teacher network is pre-trained, the compact features could be pre-learned as well using auto-encoder, which we will discuss in Sec. 3.1. Then, we will discuss the details of IE-KD in Sec. 3.2, followed by extension to deep mutual learning manner in Sec. 3.3.

3.1. Compact Knowledge Extraction

We denote the features of the teacher as f_T, the features of the inheritance part and exploration part of the student network as f_{inh} and F_{exp}, respectively. The challenge in measuring the similarity/dis-similarity between these features is that they usually have different shapes and sizes. To solve this problem, we embed them into a shared latent feature space of the same dimension via an encoder, and the embedded features are indicated by F_T, F_{inh} and F_{exp}, respectively. We adopt the factor-based embedding module in [16] to extract knowledge from the specific convolutional block of the teacher network.

In particular, an auto-encoder, consisting of several convolutional and deconvolutional layers, is adopted to extract transferable factors F_T from the teacher network. We use three convolution layers and three transposed convolution layers. All six layers use 3×3 kernels, stride of 1, padding of 1, and batch normalization with leaky-ReLU with rate of 0.1 followed by each of the six layers. Only at the second convolution, the number of output feature maps is compressed to the number of factor feature maps. Similarly, the second transposed convolution layer is resized to match the feature maps of the teacher network. The detailed architecture can be found in the supplementary materials. The auto-encoder is trained by the common reconstruction loss:

$$L_{rec} = ||f_T - R(f_T)||^2, \quad (1)$$

where f_T is the feature maps of the teacher network and $R(f_T)$ is the output of the auto-encoder.

3.2. Inheritance and Exploration

The goal of IE-KD is to enhance the features of the student network, f_S, by using the compact features of the

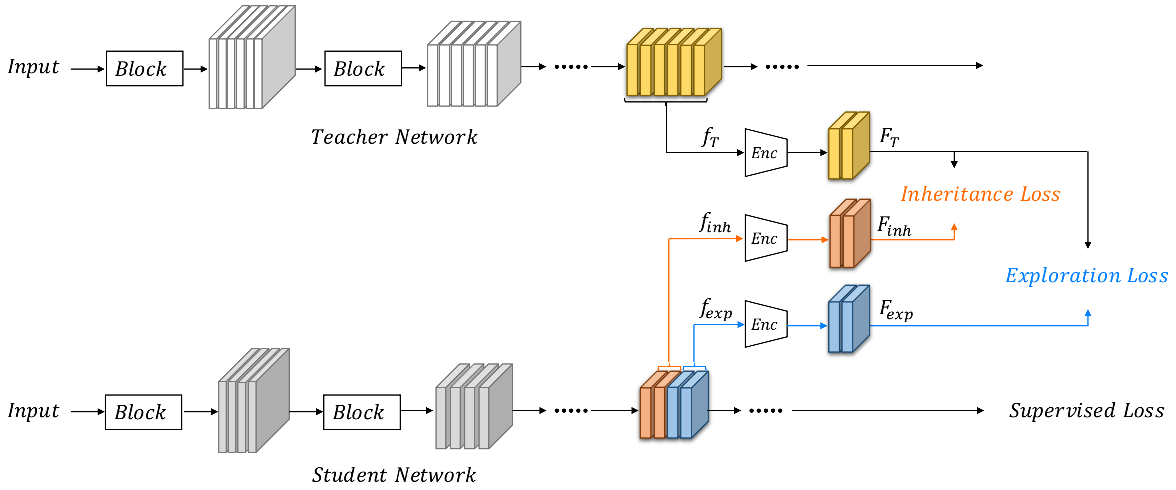


Figure 2. Overview of IE-KD framework. The student network is split into two parts. One part (colored in orange) inherits the compact and effective representations encoded by factors from the teacher network via consistency/inheritance loss (similarity), and the other part (colored in blue) is pushed to generate different features via diversity/exploration loss (dis-similarity). The supervised task (classification/detection) loss guides the exploration part to converge to diverse yet effective features.

teacher network, f_T. Directly pushing the student model to mimic the features of the teacher model limits the student model in learning undiscovered features. Therefore, instead of treating and training f_S as a whole, we randomly split it into two parts, f_{inh} and f_{exp}, and regulate them separately with two counterpart losses, an inheritance loss L_{inh} that pushes f_{inh} to mimic f_T as much as possible, and an exploration loss L_{exp} that allows f_{exp} to learn different or unrelated features to f_T. Similarly, we use two separate encoders to embed f_{inh} and f_{exp} into factors F_{inh} and F_{exp} that have the same dimension as F_T, which relieves the student network from the burden of directly learning the output of the teacher network.

We would like to emphasize that the specific implementation of the inheritance loss and exploration loss are orthogonal to our IE-KD framework. *All metrics that measures the similarity and dis-similarity of two representations can be easily adopted into our framework for the inheritance part and exploration part, respectively.* Below, we only discuss a simple and effective implementation of L_{inh} and L_{exp}.

Inheritance loss. L_{inh} is designed to inherit the existing knowledge from the teacher model by minimizing the difference between F_{inh} and F_T, and is represented as:

$$\mathcal{L}_{inh} = \left|\left| \frac{F_{inh}}{||F_{inh}||_2} - \frac{F_T}{||F_T||_2} \right|\right|_1. \tag{2}$$

Similar to [16], we apply L_1 normalization to the factors. The L_1 distance acts as a similarity metric in a very simple form. Any other similarity metrics for vectors can be easily adopted as a inheritance loss, such as L_2, cosine distance ($L = 1 - cos(x, y)$), partial L_2 distance [12], *etc*.

Exploration loss. L_{exp} is designed to act oppositely as L_{inh}, learning representations that are different from the inherited ones. Inspired by [4], a straightforward choice is to minimize the negative difference between F_{exp} and F_S:

$$\mathcal{L}_{exp} = -\left|\left| \frac{F_{exp}}{||F_{exp}||_2} - \frac{F_S}{||F_S||_2} \right|\right|_1. \tag{3}$$

We would like to point out that the sign change of the exploration loss is different from pushing F_{exp} to learn a negative teacher factor $-F_T$, which obviously correlates with F_T. *L_{exp} aims to encourage the exploration part to focus on other regions of the image [4], exploring new features that are complementary to the inherited ones.*

Likewise, there exist many different metrics to measure the dis-similarity, such as negative L_2 distance ($L = -||x - y||_2$), orthogonality measure ($L = |cos(x, y)|$), CKA [17], negative partial L_2 distance [12], *etc*.

Training. The teacher network's factor auto-encoder is firstly trained with the reconstruction loss. Then, the factor encoders and backbone network of the student network is trained simultaneously with target loss (classification, detection, *etc*.), inheritance loss and exploration loss:

$$\mathcal{L} = \mathcal{L}_{goal} + \lambda_{inh}\mathcal{L}_{inh} + \lambda_{exp}\mathcal{L}_{exp}, \tag{4}$$

where λ_{inh} and λ_{exp} are the corresponding loss weights, respectively.

3.3. Extension to Deep Mutual Learning

In the above sections, we propose a new framework to improve a student network by transferring knowledge from a teacher network in an inheritance and exploration manner. A straightforward idea is that we can further improve the teacher network via the same process with the better student network. Thus, our IE-KD approach can be extended to a deep mutual learning manner [35] (termed as *IE-DML*), to make both the teacher and student networks benefit from our IE-KD mechanism.

In the original Deep Mutual Learning [35] strategy, two peer networks (Θ_1 and Θ_2) are optimized simultaneously with KL distance to measure the consistency of two network's predictions. In IE-DML, we replace the KL regularization with two IE-KD processes (described in Sec. 3.2) which have opposite directions, *i.e.*, network Θ_1 to Θ_2 and network Θ_2 to Θ_1. For example, in the process of network Θ_1 to network Θ_2, network Θ_2 is trained by regarding network Θ_1 as the teacher network. Additionally, since both networks are trained from scratch, the auto-encoder needs to be trained together with the backbone networks.

The overall loss function $\mathcal{L}_{\Theta_1}$ for network Θ_1 is composed of four components: target loss, reconstruction loss, inheritance loss and exploration loss:

$$\mathcal{L}_{\Theta_1} = \mathcal{L}_{goal1} + \lambda_{rec}\mathcal{L}_{rec1} + \lambda_{inh}\mathcal{L}_{inh1} + \lambda_{exp}\mathcal{L}_{exp1}, \quad (5)$$

where the λ_{rec}, λ_{inh} and λ_{exp} are the corresponding loss weights, respectively. Similarly, the objective loss function $\mathcal{L}_{\Theta_2}$ for network Θ_2 can be computed as:

$$\mathcal{L}_{\Theta_2} = \mathcal{L}_{goal2} + \lambda_{rec}\mathcal{L}_{rec2} + \lambda_{inh}\mathcal{L}_{inh2} + \lambda_{exp}\mathcal{L}_{exp2}. \quad (6)$$

In this way, each model is trained by the inheritance and exploration with the compact knowledge from the other one.

Finally, two networks are updated alternatively following four steps until convergence: a) update the predictions of the teacher and student networks for an input mini-batch; b) compute the stochastic gradient w.r.t. $\mathcal{L}_{\Theta_1}$, and update Θ_1; c) update the predictions of the teacher and student networks for the current mini-batch; d) compute the stochastic gradient w.r.t. $\mathcal{L}_{\Theta_2}$, and update Θ_2.

4. Experiments

The efficiency of our IE-KD mechanism is evaluated on both classification and detection tasks. For classification, CIFAR and ImageNet datasets are used. For detection, the PASCAL VOC dataset is used. We compare our proposed IE-KD with independent learning and several state-of-the-art knowledge distillation methods. In independent learning, both the teacher and student networks are independently trained from scratch. In distillation, the student network is trained by transferring knowledge from the pre-trained teacher network via consistency controls. By comparing with distillation, we demonstrate that our IE-KD is a general method to improve the student network to achieve SOTA performance and our exploration loss plays a key role in enhancing the network features.

4.1. Datasets and Settings

CIFAR-10 and CIFAR-100 [18] consist of $50,000$ training and $10,000$ test images drawn from 10 and 100 classes. Networks are trained using SGD with Nesterov momentum. The initial learning rate is set to 0.1, the momentum is set to 0.9, and the mini-batch size is set to 128. The learning rate is divided by 10 at the 80th and 120th epochs. The training process ends at the 160th epoch.

ImageNet consists of 1.2M training images and 50k validation images with $1,000$ classes. We perform large-scale experiments on ImageNet to verify our potential ability to transfer more complex and detailed information. Networks are trained for 100 epochs. The learning rate begins at 0.1 and is multiplied by 0.1 at every 30 epochs.

We apply our method to Single Shot Detector (SSD) [19]. Networks are trained on a mixture of the PASCAL VOC2007 and VOC2012 [8] *trainval* sets, which are widely used in object detection. The backbone network in all models is pre-trained using ImageNet. Networks are trained for 120k iterations with a batch size of 32. Detection performance is evaluated on the VOC 2007 *test* set.

4.2. Implementation Details

We implement all networks in PyTorch [23] and the code will be released later. The ratio of the number of input feature maps to the number of factor feature maps is set to 2. We randomly split representations of the student network into inheritance and exploration parts, since the parameters of the network are randomly initialized and there is no strong correlation among channels before learning. The weights of both inheritance and exploration loss are set to 50 on CIFAR-10, 100 on ImageNet and PASCAL VOC.

Our IE-KD approach is a general framework and can be easily combined with existing distillation methods. In this paper, we combine our IE-KD framework with three SOTA distillation methods, AT [33], FT [16] and OD [12] , and denote them as IE-AT, IE-FT and IE-OD, respectively. In Sec. 3.2, we present the formulation of IE-FT as an instantiation of our approach. For IE-AT, the output factors of the encoders are reduced to spatial attention maps first, then the inheritance loss (Eq.(2)) and exploration loss (Eq.(3)) are applied to the spatial attention maps between the teacher network and the inheritance/exploration part of the student network. For IE-OD, we revise the distance formulations as in OD [12], *i.e.*, Eq.(2) is reformulated as $\mathcal{L}_{inh} = ||(max(F_{inh}, 0) - max(F_S, 0)||_2$ and Eq.(3) becomes $\mathcal{L}_{exp} = -||(max(F_{exp}, 0) - max(F_S, 0)||_2$.

4.3. Results of Image Classification

To control other factors and make a fair comparison, we reproduced the algorithms of other methods based on their codes and papers. Table 1 shows the Top-1 error rate on CIFAR-10 when various architectures, including ResNet [11], Wide ResNet [34] and VGG [27], are used. In the table, the “teacher: baseline” and “student: baseline” columns denote the network architecture and corresponding performance of training from scratch. First, we use ResNet-56 as teacher and ResNet-20 as student, that have same number of channels but different blocks. Then, we test different types

Table 1. KD [10], AT [33], FT [16], OD [12], Tf-KD [32], CRD [29] and IE-KD experiments results by training the student network with pre-trained teacher (error, in %) on CIFAR-10.

Teacher: baseline	Student: baseline	KD	AT	FT	OD	Tf-KD (S)	CRD	IE-AT	IE-FT	IE-OD
ResNet-56: 6.39	ResNet-20: 7.78	7.37	7.13	6.85	6.81	7.41	6.80	6.70	6.57	**6.53**
WRN-40-1: 6.84	ResNet-20: 7.78	7.46	7.14	6.85	6.69	7.51	6.77	6.81	6.57	**6.49**
WRN-46-4: 4.44	VGG-13: 5.99	5.59	5.48	4.84	4.81	5.48	4.81	4.75	4.67	**4.65**
WRN-16-2: 6.27	WRN-16-1: 8.62	8.22	8.01	7.64	7.48	8.10	7.49	7.76	7.38	**7.26**

Table 2. KD [10], AT [33], FT [16], OD [12], Tf-KD [32], CRD [29] and IE-KD experiments results by training the student network with pre-trained teacher (Top-1 and Top-5 error, in %) on ImageNet. The teacher is ResNet-34 and the student is ResNet-18.

	Teacher	Student	KD	AT	FT	OD	Tf-KD (S)	CRD	IE-AT	IE-FT	IE-OD
Top-1	26.73	29.91	29.34	29.30	28.57	28.49	29.58	28.83	28.41	28.27	**28.19**
Top-5	8.57	10.68	10.12	10.00	9.71	9.67	10.06	9.87	9.54	9.39	**9.33**

Table 3. AT [33], FT [16], OD [12], CRD [29] and IE-KD experiments results by training the student network with pre-trained teacher (mAP, in %) on PASCAL VOC2007. The teacher is ResNet-50 and the student is ResNet-18.

Teacher	Student	AT	FT	OD	CRD	IE-AT	IE-FT	IE-OD
76.79	71.61	72.00	72.68	73.08	73.11	73.16	73.32	**73.51**

of residual networks for teacher and student with WRN-40-1 and ResNet-20. To investigate the effect of the absence of shortcut connections, we further use WRN-46-4 as teacher and VGG13 as student. To test the applicability for architectures with the same blocks but different channels, we use WRN-16-2 as teacher and WRN-16-1 as student.

Results of 3 variants of IE-KD (IE-AT, IE-FT and IE-OD) and other distillation methods on CIFAR-10 are presented in Table 1. We have two observations: 1) In all three variants of IE-KD, our inheritance and exploration framework consistently outperforms corresponding consistency-based distillation method (IE-AT vs. AT, IE-FT vs. FT, IE-OD vs. OD) with a significant margin. The inheritance and exploration part can jointly push the student model to learn more effective representations, resulting in a better performance. 2) all variants of IE-KD shows better performances than other latest distillation methods (KD [10], AT [33], FT [16], OD [12], Tf-KD [32] and CRD [29]) consistently, regardless of the type of teacher/student networks. Furthermore, when faced with knowledge transfer from small teacher network to large student network, our IE-KD shows even more improvement than other distillation methods, results are presented in supplementary materials.

For further validation of generalization ability for large-scale image classification task, we compare our IE-KD and other distillation methods on ImageNet [25]. Results are shown in Table 2. Following [16], we set ResNet-34 as a pre-trained teacher network and ResNet-18 as an untrained student network. IE-KD outperforms all other methods again.

These results confirm that our IE-KD is a very general and effective upgrade of existing distillation framework.

4.4. Results of Object Detection

We further verify the effectiveness of IE-KD for detection tasks. We set ResNet-50 as the teacher network and ResNet-18 as the student network. Both networks are pre-trained with ImageNet and fine-tuned on PASCAL VOC 2007. As shown in Table 3, with IE-KD from the teacher network, the mean average precision (mAP) of the student network is increased with a large margin (71.61% to 73.51%). In this scenario, our IE-KD still shows a notable improvement for the student network, showing that our method can be applied to general computer vision tasks.

4.5. Extension to Deep Mutual Learning

As discussed in Sec. 3.3, we can alternatively update the student network by IE-KD and the teacher network also by IE-KD with the improved student network in a deep mutual learning manner [35], termed *IE-DML*. The learning rate strategy is the same as Sec. 4.1. The loss weights for reconstruction, inheritance and exploration loss are set as 0.8, 50 and 50. Table 4 compares IE-DML with DML on CIFAR-100. We experiment on two networks with different depths (ResNet-32 / ResNet-110), different widths (ResNet32 / WRN-28-10), different building blocks (ResNet-32 / MobileNet), and identical architectures (ResNet32 / ResNet32). In all cases, IE-DML shows clearly better improvement than DML and independent learning. This implies that mutual learning of two networks can also benefit significantly from our inheritance and exploration framework. Our IE-KD is also a general upgrade of existing mutual learning methods. These results further confirm that IE-KD is a very general framework for knowledge transfer between any type of networks and training strategies.

Table 4. Comparison of top-1 error (%) on CIFAR-100 between DML [35] and our IE-DML.

Network Types		Independent		DML		IE-DML	
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2
ResNet-32	ResNet-110	31.01	26.91	28.69	25.59	**28.36**	**24.99**
ResNet-32	WRN-28-10	31.01	21.31	29.27	21.04	**28.06**	**20.63**
ResNet-32	MobileNet	31.01	26.35	28.90	23.87	**28.33**	**23.24**
ResNet-32	ResNet-32	68.99	68.99	29.25	28.81	**28.59**	**28.15**

Table 5. Ablation study with different proportions of inheritance and exploration feature channels (top-1 error in %).

Dataset	Teacher	Student	$\eta = 0.0/1.0$	0.3/0.7	0.5/0.5	0.7/0.3	1.0/0.0
CIFAR-10	ResNet-56	ResNet-20	7.56	6.84	**6.53**	6.71	6.86
CIFAR-100	ResNet-110	ResNet-32	28.01	27.23	**25.67**	26.37	27.49
ImageNet	ResNet-34	ResNet-18	30.02	28.90	**28.19**	28.64	29.65

4.6. Ablation Study

Table 6. Ablation study with different metrics and different loss weights on CIFAR-100. The teacher is ResNet-110 and the student is ResNet-32.

(a) metrics

Metric	Error (%)
L_1	**25.67**
L_2	25.76
cos	26.38
partial L_2	26.71
L_1+CKA	26.60
Baseline	31.01

(b) loss weights

λ_{inh}	λ_{exp}	Error (%)
5	5	29.84
50	50	**25.67**
500	500	27.77
50	500	26.69
500	50	26.33
0	0	31.01

In the introduction section, we describe the motivation for a student network to mimic the compact representation of a teacher network, and learn more different and complementary features to improve diversity and generalization. To further analyze the necessity and contribution of inheritance and exploration, We conduct ablation studies to analyze the effects of inheritance and exploration parts.

Inheritance vs. Exploration. In Table 5, we show the results of using different proportions of feature channels for inheritance and exploration, tested on different datasets and network architectures. The settings of "0.0/1.0" and "1.0/0.0" correspond to using either exploration or inheritance only. The result shows that both inheritance and exploration are important, and even division achieves the optimal performance.

Metric. In Table 6(a), we show the results of using different similarity metrics for the inheritance and exploration loss. For simplicity, we use "L_2" to represent L_2 distance as similarity metric and its negative counterpart as dis-similarity metric. The results show that no matter what metric is used our IE-KD consistently improves the performance of the student network.

Loss weights. Table 6(b) shows the results of using different weights for the inheritance and exploration loss. The results show that $\lambda_{inh} = \lambda_{exp} = 50$ achieves the optimal performance.

4.7. How Does IE-KD Work?

Inheritance. We use layer-wise relevance propagation (LRP) [21] to interpret the network by visualizing which pixels contribute how much to the classification [2]. Fig. 3 shows different images and their LRP heat maps from independent learning, KD and IE-KD. We find that LRP heat maps of the inheritance channels in IE-KD resemble the heat maps of the teacher network, which indicates that the inheritance part mimics the compact features of the teacher network well.

Furthermore, we use the number of active neurons [6] to analyze the redundancy of internal representations of networks. For an intermediate representation, the number of active neurons is the number of directions to which classification loss function $c(x)$ is sensitive. The more active neurons, the less redundancy the representation contains [6]. The numbers of active neurons in the teacher network (ResNet-110) and inheritance channels of student network (ResNet-32) are 46 and 45, indicating that the inheritance channels contain approximately the same number of active neurons as the teacher network. Moreover, the total number of neurons in inheritance channels is only half of that in the teacher network, which means that the knowledge is represented more compactly by the inheritance component.

Exploration. First, we demonstrate that the exploration part can help discover more *discriminative* input patterns via some concrete examples. In Fig. 1(a), the "cheetah" is misclassified as a "crocodile" by the student model, as the model attributes most of its prediction to the tail of the "cheetah" that resembles a "crocodile". The exploration part of IE-KD model discovers new relevant patterns on ears and mouth that are quite discriminative between the "cheetah" and "crocodile", and helps predict it correctly. Fig. 3(a) shows another example, where the independently trained student is confused by the leaf part of a "pear" and misclassifies this image as a "butterfly". The exploration part

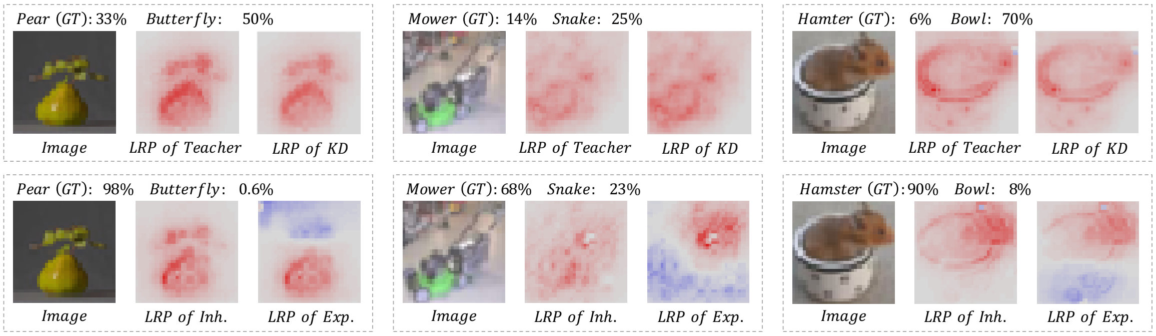
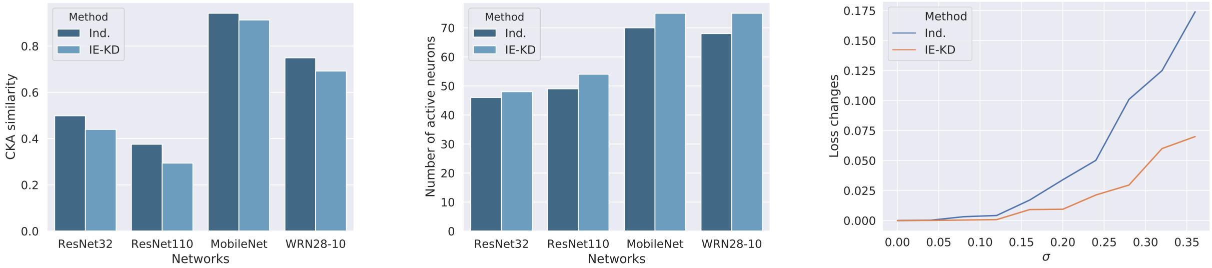


Figure 3. Analysis on how exploration works. "GT" denotes the ground truth class of the image.



(a) CKA

(b) Active neurons

(c) Loss changes

Figure 4. Comparison of CKA similarity, numbers of active neurons, and loss changes when Gaussian noise is added.

finds negative relevance of the leaf part (indicated by blue) that pushes the model to focus more on the pear and less on the leaf. Similar result is shown in Fig. 3(b) and Fig. 3(c). More results are provided in the supplementary materials.

Second, we measure the similarity between features from the inheritance and exploration channels, when student is trained independently or via IE-KD. Centered Kernel Alignment (CKA) is introduced in [17] as a similarity index to measure the similarity between two representations. A larger CKA denotes a higher similarity between two sets of representations For a fair comparison, the independent model and IE-KD model are initialized with the same random seed. As shown in Fig. 4(a), the CKA values are smaller for IE-KD in the four networks on CIFAR-100. This indicates more diverse features in the IE-KD model.

Third, we calculate the number of active neurons for IE-KD and independent student networks. Fig. 4(b) demonstrates that IE-KD student network has more active neurons than the independent student network. As proved in [6], this means that the features are more efficient.

Generalization. Similar to [35, 3, 14], we analyze the sharpness of the converged minima of the independent and IE-KD models in Fig. 4(c). Usually, sharp minima leads to poorer generalization, while flat minima has better generalization ability[3, 14]. The experiments are conducted on CIFAR-100 using MobileNet. The converged training loss of both models is approximately the same, 0.131 for IE-KD model and 0.126 for independent model, which means the depths of the two minima are close. As we increase the scale of Gaussian noise added to model parameters, the training loss of the independent model increases faster than that of the IE-KD model. This suggests that the IE-KD model has found a much flatter minimum, and it also provides another explanation for the lower generalization loss of the IE-KD model in Fig. 1(b).

5. Conclusion

We propose a novel framework for neural network distillation, using the technique of inheritance and exploration. Our IE-KD framework is generic and can be easily combined with existing distillation or mutual learning methods. Through experiments, we examine the performance of the proposed method using various networks in various tasks, and prove that the proposed method substantially outperforms the state-of-the-arts of knowledge distillation. We believe it can shed light on more future works, such as designing different forms of losses, or applying it to other tasks, *i.e.*, reinforcement learning.

Acknowledgements

This work was supported in part by Alibaba Innovative Research (AIR) program, Major Scientific Research Project of Zhejiang Lab (No. 2019DB0ZX01), NSFC No. 61872329, the National Key R&D Program of China under contract No. 2017YFB1002203 and the Fundamental Research Funds for the Central Universities under contract WK3490000005.

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NeurIPS*, pages 2654–2662, 2013. 2

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015. 7

[3] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019. 8

[4] Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 4

[5] Bucila Cristian, Caruana Rich, and Niculescu-Mizil Alexandru. Model compression. *KDD*, pages 535–541, 2006. 2

[6] Chunfeng Cui, Kaiqi Zhang, Talgat Daulbaev, Julia Gusak, Ivan Oseledets, and Zheng Zhang. Active subspace of neural networks: Structural analysis and universal attacks. *arXiv preprint arXiv:1910.13025*, 2019. 7, 8

[7] Charles Darwin. *On the Origin of Species by Means of Natural Selection*. Murray, 1859. 2

[8] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 5

[9] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018. 3

[10] E. Hinton Geoffrey, Vinyals Oriol, and Dean Jeffrey. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 1, 3, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[12] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. *ICCV*, pages 1921–1930, 2019. 1, 3, 4, 5, 6

[13] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. *AAAI*, pages 3779–3787, 2019. 1, 3

[14] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017. 8

[15] Jangho Kim, Minsung Hyun, Inseop Chung, and Nojun Kwak. Feature fusion for online mutual knowledge distillation. *ArXiv*, abs/1904.09058, 2019. 3

[16] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *ArXiv*, abs/1802.04977, 2018. 1, 3, 4, 5, 6

[17] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*, 2019. 4, 8

[18] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 8, 2010. 5

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 5

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. 2

[21] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 193–209. Springer, 2019. 2, 7

[22] SeongUk Park and Nojun Kwak. Feed: Feature-level ensemble for knowledge distillation. *arXiv preprint arXiv:1909.10754*, 2019. 1, 3

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019. 5

[24] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 1, 2, 3

[25] 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. 6

[26] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, pages 484–489, 2016. 2

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[28] Guocong Song and Wei Chai. Collaborative learning for deep neural networks. In *NeurIPS*, pages 1832–1841, 2018. 3

[29] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 3, 6

[30] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *CVPR*, pages 2859–2868, 2019. 3

[31] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *CVPR*, pages 4133–4141, 2017. 1, 3

[32] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisit knowledge distillation: a teacher-free framework. *arXiv preprint arXiv:1909.11723*, 2019. 1, 3, 6

[33] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ArXiv*, abs/1612.03928, 2016. 1, 3, 5, 6

[34] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5

[35] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. *CVPR*, pages 4320–4328, 2017. 3, 4, 5, 6, 7, 8

[36] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, pages 7517–7527, 2018. 3