Distilling Holistic Knowledge with Graph Neural Networks

Sheng Zhou[1,2]*, Yucheng Wang[1]*, Defang Chen[1],Jiawei Chen[3],Xin Wang[4],Can Wang[1], Jiajun Bu[1]†
[1]Zhejiang Provincial Key Laboratory of Service Robot, Zhejiang University
[2]School of Software Technology, Zhejiang University
[3]University of Science and Technology of China [4]Tsinghua University
{zhousheng_zju,wangyuc,defchen}@zju.edu.cn, cjwustc@ustc.edu.cn, xin_wang@tsinghua.edu.cn
{wcan,bjj}@zju.edu.cn

Abstract

Knowledge Distillation (KD) aims at transferring knowledge from a larger well-optimized teacher network to a smaller learnable student network. Existing KD methods have mainly considered two types of knowledge, namely the individual knowledge and the relational knowledge. However, these two types of knowledge are usually modeled independently while the inherent correlations between them are largely ignored. It is critical for sufficient student network learning to integrate both individual knowledge and relational knowledge while reserving their inherent correlation. In this paper, we propose to distill the novel holistic knowledge based on an attributed graph constructed among instances. The holistic knowledge is represented as a unified graph-based embedding by aggregating individual knowledge from relational neighborhood samples with graph neural networks, the student network is learned by distilling the holistic knowledge in a contrastive manner. Extensive experiments and ablation studies are conducted on benchmark datasets, the results demonstrate the effectiveness of the proposed method. The code has been published in `https://github.com/wyc-ruiker/HKD`

1. Introduction

Deep Neural Networks (DNNs) have shown tremendous success in various applications [13, 29, 12, 28, 9, 40]. However, their success heavily relies on extensive computational and storage resources, which are usually unavailable in embedded and mobile systems. To reduce the cost while maintaining satisfactory, knowledge distillation [14] is proposed to transfer knowledge from a larger well-trained *teacher network* to a smaller learnable *student network*, hoping that the transferred knowledge will benefit the student network.

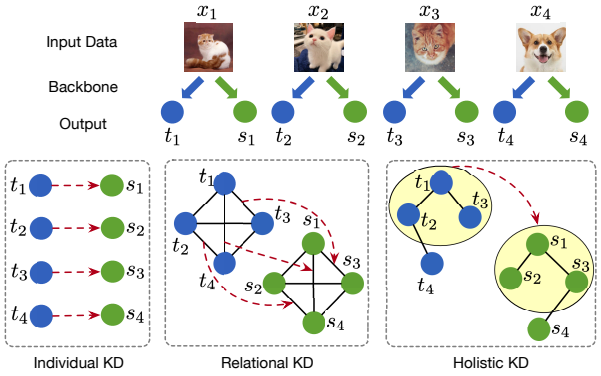

Figure 1. Comparison between Individual/Relational/Holistic Knowledge Distillation. The blue circle indicates the teacher representation, and the green circle indicates the student representation. The red arrow denotes the knowledge transfer from the teacher network to the student network. The yellow area in the holistic KD indicates the unified graph-based representation.

The knowledge distilled from the teacher network has played the central role in knowledge distillation. Among existing knowledge distillation methods, two types of knowledge have been widely studied, namely the *individual knowledge* and the *relational knowledge*. The individual knowledge is extracted from each data instance independently and provides more favorable supervision than the discrete labels, including logits [14], feature representations [31, 24] and feature maps [27, 39, 20], etc. The relational knowledge [25, 21, 23, 19] is extracted from pairs of instances which is invariant to the difference between architectures of the teacher network and the student network.

Despite the success of the above two types of knowledge, existing methods have extracted them independently, ignoring their inherent correlations. However, each type of knowledge that extracted independently will be insufficient for the student network learning, especially when the capability of the teacher network is limited. Intuitively, the

*Equal Contribution
†Corresponding Author

individual knowledge and the relational knowledge can be treated as two views of the same teacher network, which are naturally correlated. The closely related instances tend to have similar individual features and shared patterns, which is critical for more discriminative student network learning. Simultaneously integrating the individual and relational knowledge while reserving their inherent correlation is of primal importance for knowledge distillation.

To resolve the above limitations, we propose the **H**olistic **K**nowledge **D**istillation (HKD) method with graph neural networks. We introduce a novel holistic knowledge which is an integration of both individual knowledge and relational knowledge. Given the feature representations and predictions learned by the teacher and the student network, we first build an attributed graphs for each network, where each node denotes an instance, the node attributes denote the learned feature representation, the edges among instances are constructed by the K-nearest-neighbor (KNN) on the predictions. Inspired by the recent success of Graph Neural Networks (GNNs) [12, 19] in simultaneously modeling network topology and node attributes, we extract the holistic knowledge by aggregating node attributes from the neighborhood samples in the attributed graph, represented as a unified graph-based embedding. Figure 1 illustrates the comparison among the individual, relational and holistic knowledge. We also theoretically prove that existing individual knowledge and relational knowledge are special cases of holistic knowledge under certain conditions.

Given the holistic knowledge represented by graph-based embedding, a naive way of knowledge distillation is directly aligning the embedding of the same instance from the teacher and the student network. However, since the student network usually has lower capability than the teacher network, force aligning the graph-based embedding is too strict for transferring the shared patterns in the neighborhood and holistic knowledge. Instead, HKD aims at maximizing the mutual information between the graph-based representation from the teacher and the student network, which is optimized with InfoNCE estimator[22] in a contrastive manner. The holistic knowledge guides the student network learning in two ways: first, the student should learn similar instance features and relational neighborhood as the teacher network; second, the student should capture similar patterns from the neighborhood instances in the attributed graph. The memory bank technique is also employed to further improve the training efficiency. To conclude, we summarize our contributions as follows:

1. We propose Holistic Knowledge Distillation (HKD), a novel method to efficiently distill holistic knowledge for the student network learning.

2. The proposed HKD method employs graph neural networks to simultaneously integrate both the individual and relational knowledge into a unified representation, where their inherent relationship can be reserved.

3. We conduct extensive experiments on benchmark datasets to evaluate the performance of HKD and motivation of holistic knowledge, the results demonstrates the effectiveness of the proposed HKD method.

2. Related Work

Knowledge Distillation. Knowledge distillation was first introduced as a neural network compression technique that minimizes the KL-divergence between the output logits of teacher and student networks [1, 14]. Compared with discrete labels, the relative probabilities predicted by the teacher network tend to encode semantic similarities among categories, which are important for the student network learning [14]. Some subsequent works have been proposed to widen its applicability, such as adding regularization on the logits [34, 3], intermediate layers [27, 39, 4, 20] or distillation process [37, 38].

However, the above mentioned methods distill knowledge contained in each instance independently but ignore the relationship among instances, which is critical to achieve a robust and general student model. To make up this shortcoming, relational knowledge distillation [23] is proposed by distilling both instance-wise and relation-wise knowledge. Given particular layer l, GKD [17] build a KNN-based graph on the cosine similarity of the inner representation and the weights represent the strength of proximity between two instances. However, it requires the layer number of the teacher and student network are same, which is not always satisfied. IRG [21] is then proposed by introducing feature space transformation across layers. In MHGD [19], the relation-level knowledge is distilled to a graph using an attention network and optimized by minimizing the KL-divergence between the embedded the teacher and student graphs. Recent works [31, 36] have incorporated contrastive learning and achieve inspiring results. CRD [31] performs contrastive learning by maximizing the mutual information between the teacher and student networks. SSKD [36] performs contrastive learning separately in the teacher and student networks, then the model is optimized by minimizing the loss between the output of self-supervised module from two networks. To clearly show the most critical contribution of our method, we do not utilize the intermediate information and compare with those methods relying on it in the experimental part.

Graph Neural Network. Graph Neural Networks (GNNs) [15, 12] aim at learning node representation by collectively aggregate information from neighborhood instances in graph structure data. The learned representation can model individual features as well as relationship between instances which is critical for data understand-

ing. Profit from this property, GNNs have made remarkable advancements in a great many learning tasks beyond network/graph representation [42, 44, 43], including computer vision [11, 18], natural language processing [26, 2] and recommendations[6, 5] etc. Although success in other domains, to the best of our knowledge, GNNs has not be explored to knowledge distillation and we are the first one to do so.

3. Preliminaries

3.1. Background and Notations

Given a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ from K categories with corresponding labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, where N represents the number of samples in the dataset. We refer a well-optimized deep neural network with fixed parameters $\mathbf{W}^t$ as the teacher network and a relatively shallow neural network with trainable parameters $\mathbf{W}^s$ as the student network [14]. The feature representations learned by the teacher and student networks are denoted as $\mathbf{f}^t \in \mathbb{R}^{d^t}$ and $\mathbf{f}^s \in \mathbb{R}^{d^s}$, which are mainly used in relational knowledge distillation. It is worth noting that d^t and d^s may be different especially when the teacher and the student network architectures are different. The logits predicted by the teacher and student networks are denoted as $\mathbf{z}^t$ and $\mathbf{z}^s$, which are mainly used in individual knowledge distillation.

3.2. Vanilla Knowledge Distillation

The general idea of vanilla knowledge distillation is to distill knowledge from soft targets predicted by the teacher network [14]. The soft targets are produced by Softmax function with temperature scaling:

$$p_i(\mathbf{z}; \tau) = \mathrm{Softmax}(\mathbf{z}; \tau) = \frac{e^{z_i/\tau}}{\sum_{k=1}^{K} e^{z_k/\tau}} \tag{1}$$

where z^i is the corresponding logit of the i-th class and temperature τ is normally set to 1. Using a higher value for τ will produce a softer probability distribution over classes. The student network is then optimized by minimizing the Kullback-Leibler (KL) divergence between soft targets $\mathbf{p}^t$ and $\mathbf{p}^s$ produced by the teacher and the student networks:

$$\mathcal{L}_{KD}(\boldsymbol{p}^s, \boldsymbol{p}^t) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{KL}\left(\boldsymbol{p}^s, \boldsymbol{p}^t\right) \tag{2}$$

In vanilla KD, the student network is also trained with the hard labels and the total loss can be formalized as:

$$\mathcal{L} = \mathcal{L}_{CE}(\boldsymbol{p}^s, \mathbf{y}) + \lambda \mathcal{L}_{KD}(\boldsymbol{p}^s, \boldsymbol{p}^t) \tag{3}$$

where λ is a balancing weight. $\mathcal{L}_{CE}$ is the Cross-Entropy (CE) loss between the hard labels and prediction.

4. Model

As mentioned earlier, holistic knowledge is expected to integrate both individual knowledge and relational knowledge. Inspired by recent success of graph neural networks in simultaneously modeling the network topology and node attributes, we utilize graph neural networks to extract holistic knowledge from the teacher network. In the following subsection, we will elaborate on the details of the proposed holistic knowledge distillation (HKD) method.

4.1. Attributed Context Graph Construction

Given a batch of instances, we first feed them into the teacher network and the student network to get the feature representations $\mathbf{f}^t, \mathbf{f}^s$ as well as prediction $\mathbf{p}^t, \mathbf{p}^s$. Then we build two attributed graphs $\mathbf{G}^t = \{\mathbf{A}^t, \mathbf{F}^t\}$ and $\mathbf{G}^s = \{\mathbf{A}^s, \mathbf{F}^s\}$ for the teacher network and the student network, where $\mathbf{F}^t \in \mathbb{R}^{N \times d^t}, \mathbf{F}^s \in \mathbb{R}^{N \times d^s}$ are the attributes of nodes in the graph, here we directly use the feature representations learned by the teacher and the student networks; $\mathbf{A}^t, \mathbf{A}^s$ are the adjacent matrices of the attributed graphs which are based on the prediction $\mathbf{p}^t, \mathbf{p}^s$ predicted by the teacher and the student networks:

$$\mathbf{A}^t = \phi(\mathbf{p}^t), \quad \mathbf{A}^s = \phi(\mathbf{p}^s) \tag{4}$$

where $\phi(\cdot)$ is the KNN-based graph construction function. Note that the graph $\mathbf{G}^t$ is fixed since the teacher network has been well optimized while the graph $\mathbf{G}^s$ will be updated during training in both node attributes and graph topology.

The attributed graph defined above enjoys the following properties: First, compared with fully connected graph among instances built by existing relational knowledge distillation methods, the KNN graph will filter out the most incorrelated sample pairs. This is particular important since only a few samples are correlated in randomly sampled batches and provide sufficient information for the node representation learning. Second, the graph is able to model the inter-class and intra-class information since the edges are constructed based on prediction. The samples from two highly correlated classes will have a high probability to form an edge. Finally, it is straightforward to jointly extract both the individual and relational knowledge from the attributed context graph with graph neural networks.

4.2. Holistic Knowledge Distillation

Inspired by the tremendous success of graph neural networks in simultaneously modeling the network topology and node attributes, we apply Topology Adaptive Graph Convolution Network (TAGCN) [10, 15] on the attributed context graphs $\mathbf{G}^t$ and $\mathbf{G}^s$ to extract the holistic knowledge. We use the graph-based representations $\mathbf{H}^t \in \mathbb{R}^{N \times g^t}$ and $\mathbf{H}^s \in \mathbb{R}^{N \times g^s}$ of the teacher and student networks to denote

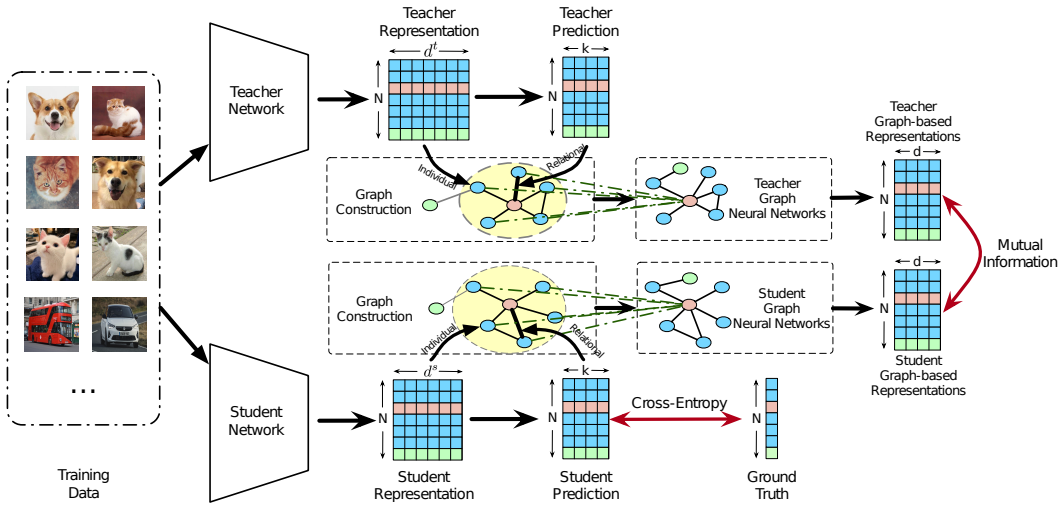

Figure 2. The overall framework of the HKD Method. Three major components are carefully designed: graph construction, graph neural networks, and mutual information estimation to represent, define, and distill the holistic knowledge. The student model is trained under the guidance of ground truth labels and mutual information of the holistic knowledge.

the holistic knowledge, which can be calculated as:

$$\mathbf{H}^t = \sum_{l=0}^{L} \left(\mathbf{D}_t^{-1/2}\mathbf{A}^t\mathbf{D}_t^{-1/2}\right)^l \mathbf{F}^t\mathbf{\Theta}_l^t \tag{5}$$

$$\mathbf{H}^s = \sum_{l=0}^{L} \left(\mathbf{D}_s^{-1/2}\mathbf{A}^s\mathbf{D}_s^{-1/2}\right)^l \mathbf{F}^s\mathbf{\Theta}_l^s \tag{6}$$

where g^t, g^s are the dimension of the graph-based representation, $\mathbf{D}_t = \sum_j \mathbf{A}_{ij}^t$ is the diagonal degree matrix of the teacher network and so is the $\mathbf{D}_s$ matrix, $\mathbf{\Theta}_l^s$ and $\mathbf{\Theta}_l^t$ are the learnable weights to sum the results of l-th hops together and we set $L = 1$ here.

A good student network is expected to distill holistic knowledge from the teacher network by learning similar graph based repesentation $\mathbf{H}^s$ with $\mathbf{H}^t$. There exists several vector-wise metrics for measuring their alignment, including the Cosine Similarity, Euclidean distance, etc. However, these metrics are not suitable for holistic knowledge distillation since the teacher and the student networks usually have different network architectures, there exists a gap between the representation capability. As a result, directly aligning the graph-based representation $\mathbf{H}^s$ and $\mathbf{H}^t$ of the same instance may be over refine. To overcome the limitations, we use the Mutual Information (MI) [32] to measure the amount of holistic knowledge distilled from the teacher network to the student network.

Assume that we are given a set of training instances $\mathcal{X}$ with an empirical probability distribution $\mathbb{P}$, after pushing instances through the teacher and the student networks, the graph-based representations will obey the probability distribution $\mathbf{H}^t \sim \mathbb{P}^t$ and $\mathbf{H}^s \sim \mathbb{P}^s$. We wish to train the student network by maximizing the mutual information between the graph-based representations $\mathbf{H}^t$ and $\mathbf{H}^s$:

$$\underset{\mathbf{W}^s,\mathbf{\Theta}^t,\mathbf{\Theta}^s}{\mathcal{L}_{HOL}} = -\mathbf{I}(\mathbf{H}^t, \mathbf{H}^s) \tag{7}$$

where $\mathbf{I}(\cdot)$ denotes the mutual information between two random variables. Inspired by recent success in mutual information estimation, we use the InfoNCE estimator [22] to measure the mutual information, which is defined as:

$$\mathbf{I}(\mathbf{H}^t, \mathbf{H}^s) \geq \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{f(\mathbf{h}_i^t, \mathbf{h}_i^s)}}{\frac{1}{N}\sum_{j=1}^{N} e^{f(\mathbf{h}_i^t, \mathbf{h}_j^s)}}\right] \tag{8}$$

where $f(\cdot)$ is the vector-wise similarity function and we use cosine similarity here, $\mathbf{h}_i^t, \mathbf{h}_i^s$ are the graph-based representations of instance i learned by the teacher network and the student network. The objective of holistic knowledge distillation can be formulated as:

$$\mathcal{L} = \mathcal{L}_{CE} + \beta\mathcal{L}_{HOL} \tag{9}$$

where β is the weight for linear combination.

4.3. Efficient Training

Since the InfoNCE estimator uses all the instances in the dataset as negative samples, computing the holistic knowledge distillation loss with graph neural networks is computational expensive for large scale dataset. To avoid recomputing the representations for each instance during training, the widely used Memory Bank [35] strategy is used for storing them. However, in the HKD method, the attributed context graph $\mathbf{G}^t$ and $\mathbf{G}^s$ are constructed on mini-batch with randomly sampled instances. As a result, the graph-based

representations $\mathbf{H}^t$ and $\mathbf{H}^s$ reflect holistic knowledge in different attributed graphs, which should not be stored in memory bank and serve as negative samples. To overcome this limitation while improve the efficiency of the HKD method, we maintain two memory banks for the teacher network and the student network, where the feature representation $\mathbf{f}^t, \mathbf{f}^s$ are stored and serve as the negative samples for training. The approximate holistic knowledge distillation loss can be formulated as:

$$\begin{aligned}\widetilde{\mathcal{L}}_{HOL} = \sum_{i=1}^{N} log\frac{e^{f(\mathbf{h}_i^t,\mathbf{h}_i^s)}}{e^{f(\mathbf{h}_i^t,\mathbf{h}_i^s)} + \sum_{j=1,j\neq i}^{N} e^{f(\mathbf{h}_i^t,\mathbf{f}_j^s)}} \\ + log\frac{e^{f(\mathbf{h}_i^s,\mathbf{h}_i^t)}}{e^{f(\mathbf{h}_i^s,\mathbf{h}_i^t)} + \sum_{j=1,j\neq i}^{N} e^{f(\mathbf{h}_i^s,\mathbf{f}_j^t)}}\end{aligned} \tag{10}$$

The overall framework of the HKD method is illustrated in Algorithm 1.

Algorithm 1 Holistic Knowledge Distillation.

Input: Training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$; A pre-trained teacher model with parameter $\mathbf{W}^t$; A student model with random initialized parameters $\mathbf{W}^s$;

Output: A well-trained student model;

1: **while** $\mathbf{W}^s$ is not converged **do**
2: Sample a mini-batch $\mathcal{B}$ with size b from $\mathcal{D}$.
3: Forward propagation $\mathcal{B}$ into $\mathbf{W}^t$ and $\mathbf{W}^s$ to obtain feature representation $\mathbf{f}^t, \mathbf{f}^s$ and prediction $\mathbf{p}^t, \mathbf{p}^s$.
4: Construct attributed context graph $\mathbf{G}^t$ and $\mathbf{G}^s$.
5: Extract holistic knowledge with graph neural networks by Equation (5),(6).
6: Calculate the Mutual information between graph-based representation as Equation (10).
7: Update parameters $\mathbf{W}^s$ by backward propagation the gradients of the loss in Equation (9).
8: **end while**

4.4. Analysis with Existing Methods

To further show the generality of HKD, we provide a theoretical analysis that many existing knowledge distillation methods can be viewed as the special cases of our method with certain conditions.

Feature-based KD Methods. Feature-based KD methods are popular which only distill the feature representation learned by the teacher network. Compared with HKD, these methods [33, 14, 39, 24] ignore the relationship among instances, which can be achieved by setting the $L = 0$ or $\mathbf{A} = \text{diag}(N)$ in HKD:

$$\mathbf{H}^t = \mathbf{F}^t\mathbf{\Theta}^t, \quad \mathbf{H}^s = \mathbf{F}^s\mathbf{\Theta}^s \tag{11}$$

where $\text{diag}(\cdot)$ is the diagonal matrix.

Relational KD Methods. The pairwise relationship of instances captured by these methods [23, 25, 33, 31] can be easily reached by setting the feature matrix $\mathbf{H}^t, \mathbf{H}^s \in \mathbb{R}^{N\times N}$ as the simialrity of feature representation $\mathbf{F}^t, \mathbf{F}^s$:

$$\mathbf{H}^t = \varphi(\mathbf{F}^t, \mathbf{F}^t), \quad \mathbf{H}^s = \varphi(\mathbf{F}^s, \mathbf{F}^s) \tag{12}$$

where $\varphi(\cdot)$ is the vector-wise similarity function. For the methods that do not estimate mutual information, they can be viewed as a special formation of Equation (8) without negative samples.

5. Experiment

In this section, we first conduct model compression and representation transferability experiments on benchmark datasets to evaluate the proposed HKD method. Then we conduct several ablation studies on graph construction and graph neural networks to validate their effectiveness. Finally, we provide the experimental analysis on the hyperparameter sensitivity of the HKD method.

5.1. Baselines

Several recently proposed knowledge distillation methods are compared, which can be categorized into two groups. Their main difference is presented in Figure 1.

(1) Individual Knowledge Distillation: This group of methods capture knowledge contained in individual instances, including the logits in vanilla KD [14], the attention map in AT [39], and feature representation in CRD [31] and SSKD [36].

(2) Relational Knowledge Distillation: This group of methods capture pairwise relational knowledge, including PKT [24], RKD[23], CCKD [25], SP[33].

We use the official implementation for these methods and follow the standard experimental settings. For the SSKD method, we remove the data augmentation so that the training samples are consistent with other methods.

5.2. Model Compression

Experimental Setup. Model compression is one of the most fundamental applications of knowledge distillation. The student network is learned by distilling knowledge from a fixed teacher network and ground truth labels. We compare our method with several recent works with different teacher and student network architectures on CIFAR100, TinyImageNet and ImageNet datasets, as shown on Table 1, Table 2 and Table 3 respectively. All results are reported as the mean and variance of classification accuracy with five runs. In order to obtain an intuitive sense about quantitative improvement, we adopt Average Relative Improvement (ARI) as the previous work [31]:

$$\text{ARI} = \frac{1}{M}\sum_{i=1}^{M}\frac{\text{Acc}_{\text{HKD}}^i - \text{Acc}_{\text{BKD}}^i}{\text{Acc}_{\text{BKD}}^i - \text{Acc}_{\text{STU}}^i} \times 100\% \tag{13}$$

Table 1. Test accuracy (%) of the student networks on the CIFAR100 dataset of combining distillation methods with KD.

Teacher Student	ResNet32×4 ResNet8×4	ResNet32×4 ShuffleNetV2	VGG13 MobileNetV2	ResNet50 VGG8	ResNet50 MobileNetV2	ARI (%)
Teacher	79.42	79.42	74.64	79.34	79.34	/
Student	72.79 ± 0.26	72.63 ± 0.71	65.33 ± 0.63	70.56 ± 0.32	65.33 ± 0.63	
KD	73.55 ± 0.20	75.38 ± 0.52	68.08 ± 0.24	73.76 ± 0.09	67.83 ± 0.46	126.48 %
AT+KD	74.80 ± 0.15	76.51 ± 0.16	66.37 ± 0.13	73.91 ± 0.24	66.81 ± 0.11	152.84 %
PKT+KD	74.68 ± 0.07	76.16 ± 0.16	68.08 ± 0.94	74.19 ± 0.27	68.42 ± 0.39	55.63 %
SP+KD	73.99 ± 0.05	76.02 ± 0.34	68.46 ± 0.37	73.50 ± 0.20	68.18 ± 0.57	80.89 %
CC+KD	74.44 ± 0.14	75.81 ± 0.20	68.54 ± 0.21	73.48 ± 0.16	68.92 ± 0.16	58.96 %
RKD+KD	74.18 ± 0.09	75.64 ± 0.24	68.24 ± 0.46	73.81 ± 0.11	68.52 ± 0.14	72.15 %
CRD+KD	75.64 ± 0.25	76.41 ± 0.36	69.82 ± 0.22	74.41 ± 0.31	69.86 ± 0.04	15.32 %
SSKD+KD	75.80 ± 0.58	76.36 ± 0.38	69.12 ± 0.54	74.68 ± 0.22	69.53 ±0.43	18.86 %
HKD	75.63 ± 0.22	76.31 ± 0.30	69.97 ± 0.42	74.86 ± 0.17	69.83 ± 0.15	12.94 %
HKD+KD	**76.13 ± 0.05**	**76.92 ± 0.22**	**70.48 ± 0.25**	**74.88 ± 0.30**	**70.72 ± 0.32**	/

Table 2. Test accuracy (%) of the student networks on the TinyImageNet dataset of combining distillation methods with KD.

Teacher Student	ResNet32×4 ResNet8×4	ResNet32×4 ShuffleNetV2	VGG13 MobileNetV2	ResNet50 VGG8	VGG13 VGG8	ARI (%)
Teacher	57.92	57.92	52.02	55.44	52.02	/
Student	49.91 ± 0.16	50.60 ± 0.23	44.20 ± 0.22	47.00 ± 0.17	47.00 ± 0.17	
KD	52.28 ± 0.07	57.27 ± 0.03	45.39 ± 0.59	51.50 ± 0.36	51.34 ± 0.08	123.18 %
AT+KD	54.79 ± 0.23	57.56 ± 0.38	45.13 ± 0.60	51.42 ± 0.42	51.03 ± 0.28	122.61 %
PKT+KD	54.11 ± 0.18	58.33 ± 0.36	47.73 ± 0.31	51.45 ± 0.28	51.61 ± 0.28	35.51 %
SP+KD	54.22 ± 0.41	58.66 ± 0.25	48.10 ± 0.59	51.70 ± 0.12	51.51 ± 0.32	29.98 %
CC+KD	54.08 ± 0.32	58.20 ± 0.06	47.67 ± 1.14	50.87 ± 0.20	51.07 ± 0.33	44.12 %
RKD+KD	53.78 ± 0.15	57.85 ± 0.24	48.10 ± 0.26	51.01 ± 0.23	50.59 ± 0.32	46.70 %
CRD+KD	55.53 ± 0.41	58.95 ± 0.05	49.12 ± 0.04	52.87 ± 0.30	52.25 ± 0.26	7.88 %
SSKD+KD	55.10 ± 2.05	57.48 ± 0.04	47.02 ± 0.90	52.36 ± 0.36	51.60 ± 0.16	35.51 %
HKD	55.53 ± 0.07	58.83 ± 0.09	49.53 ± 0.32	52.20 ± 0.20	51.97 ± 0.33	10.48 %
HKD+KD	**56.18 ± 0.12**	**59.31 ± 0.01**	**49.57 ± 0.54**	**53.30 ± 0.33**	**52.62 ± 0.03**	/

where M is the number of different architecture combinations and $\text{Acc}^i_{\text{HKD}}, \text{Acc}^i_{\text{BKD}}, \text{Acc}^i_{\text{STU}}$ refer to the accuracy of HKD, baseline knowledge distillation methods and regular trained student network.

Results and Analysis. The basic observation is that our method outperforms the conventional student network and baseline methods on most teacher and student pairs. Even without KD loss, our proposed HKD method still achieves comparable performance. This demonstrates the effectiveness of HKD method in distilling holistic knowledge from the teacher network to guide the student network learning.

We also find that the existing relational knowledge distillation methods can not always outperform the individual knowledge distillation methods. This implies that the noisy signal due to aligning all pairs of relations among instances may hurt the student network learning, motivating our KNN based graph construction for noise filtering. Another interesting observation is that the HKD method is not restricted to the same teacher and student network architecture. More surprisingly, we find that the HKD method sometimes gains slightly more improvement over conventional student network when the teacher and the student networks have different architectures. For example, on the TinyImageNet dataset, when the teacher network is fixed to ResNet32×4 architecture, the student gains 12.56 % improvement with ResNet8×4 architecture. However, 17.21 % improvement over conventional student network is gained when the student uses ShuffleNetV2. When the student network is fixed with VGG8 architecture, 11.95 % improvement is gained when the teacher network uses VGG13 architecture. However, 13.4 % improvement is gained when the teacher network uses ResNet50 architecture. This demonstrates the advantage of utilizing mutual information to measure the teacher and student networks' alignment since it is not re-

Table 3. Test accuracy (%) of the student networks on the ImageNet dataset. The results of competing methods are obtained from [4].

Method	Teacher	Student	KD	FitNet	AT	SP	VID	CRD	HKD
Top-1 Accuracy	73.54	53.78	53.73	51.46	52.83	51.73	53.97	53.76	**54.07**

Table 4. Representation transferability experiments of the student network. The student network is trained on the CIFAR100 dataset and transferred to the TinyImageNet and the STL10 dataset. A linear classifier is evaluated on the frozen representations of the student network.

Dataset	TinyImageNet	STL-10
T:ResNet50	30.79 ± 0.01	70.16 ± 0.07
S:MobileNetV2	23.01 ± 0.05	61.42 ± 0.10
KD	22.92 ± 0.13	61.25 ± 0.09
AT+KD	25.02 ± 0.01	62.05 ± 0.06
PKT+KD	26.04 ± 0.11	63.71 ± 0.05
SP+KD	24.98 ± 0.08	62.25 ± 0.13
CC+KD	25.68 ± 0.03	62.52 ± 0.10
RKD + KD	26.10 ± 0.03	63.26 ± 0.03
CRD + KD	28.98 ± 0.05	65.87 ± 0.10
SSKD + KD	24.24 ± 0.02	61.78 ± 0.02
HKD + KD	$\mathbf{30.55 \pm 0.03}$	$\mathbf{67.28 \pm 0.08}$

stricted to the same network architectures.

5.3. Representation Transferability

Experimental Setup. To evaluate the transferability of representations learned by the student network, we follow the experiment setting of existing works [31, 24, 39] and compare HKD with multiple baseline methods. We first train the student network on the CIFAR100 dataset, and employ it to get representations of each data instance on the TinyImageNet and STL-10 datasets. Then, we froze these representations and evaluate the performance with a randomly initialized linear classifier to measure the student network's transferability.

Results and Analysis. Table 4 shows the experimental results of representation transferability from CIFAR100 dataset to TinyImageNet and STL10 datasets. Among them, HKD achieves better performance on all the transferred datasets, which proves the transferability of representations learned by the HKD method. We also observe that the conventional KD method performs worse than the student network. This indicates that only transferring the logits to the student network will limit the transferability of representations, motivating the HKD to transfer the holistic knowledge in a unified framework.

5.4. Ablation Study

To further show the benefit of distilling holistic knowledge, we design ablation studies on the CIFAR100 dataset. We test both similar and different architectures for the teacher and the student networks.

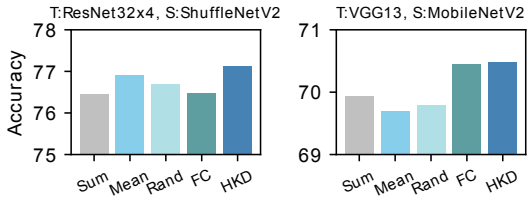

Figure 3. Ablation study on the definition of holistic knowledge for the HKD method on the CIFAR100 dataset.

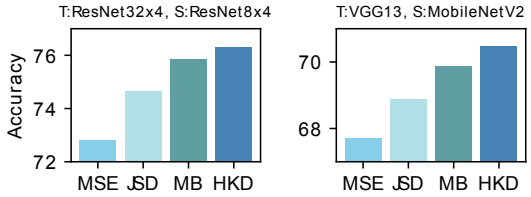

Figure 4. Ablation study on the training strategy for the HKD method on the CIFAR100 dataset.

Graph Construction and Graph Neural Networks. In the HKD method, graph construction and graph neural networks play a critical role in defining holistic knowledge. To explore the impact of different graph construction strategies, we test two graph construction strategies: random graph construction (Rand) and fully connected graph construction (FC). To demonstrate graph neural networks' superiority in combining the graph topology and the instance features, we compare two basic graph-based representation learning strategies: sum-pooling (Sum) and mean-pooling (Mean).

Figure 3 illustrates the ablation study results. We can observe that the HKD method that utilizes K-Nearest-Neighbors and graph neural networks achieves the best performance, demonstrating the effectiveness of the HKD method.

Training Strategy. In the HKD method, we utilize mutual information with a graph-independent memory bank to guide holistic knowledge transfer. To verify the advantage of such a training strategy, we compare with the following strategies: the first one is the Mean Square Error (MSE) to measure the similarity between representations; the second one is the JS-divergence (JSD) with few negative samples in each mini-batch without memory bank; the third one is using memory bank (MB) to store the graph-based representations directly.

Figure 4 illustrates the ablation study results. We can observe the HKD method achieves better performance than the compared strategies, demonstrating the effectiveness of us-

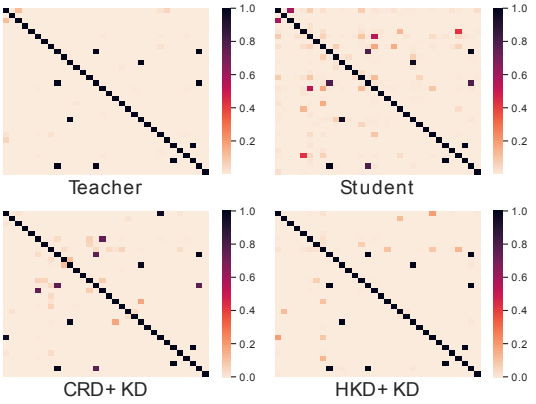

Figure 5. HeatMap visualization of four networks. The color denotes the strength of the similarity between pairs of instances.

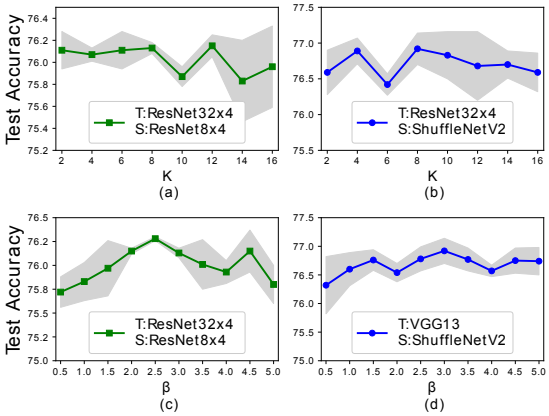

Figure 6. Hyper-parameter tuning of the HKD method. The first row of subfigures denotes the hyperparameter tuning results on the number of neighbors. The second row of subfigures denotes the hyperparameter tuning results on β.

ing the mutual information to measure the alignment, using the InfoNCE to estimate mutual information and memory bank for efficient training.

5.5. Visualization and analysis

To delving into essence beyond results, we perform analysis based on visualization. We first train a network and then randomly select a batch of data with 32 instances. These instances are fed into four networks: the teacher network, the student network, CRD, and HKD. We use cosine similarity to measure the pairwise similarity between the prediction and use different colors to represent the different strength of similarity.

Figure 5 illustrates the experimental result. Each block represents the pairwise cosine similarity between two instances. The darker color denotes higher cosine similarity while the lighter color denotes lower cosine similarity. From this figure, we have the following observations: First, most pairs have superficial similarities among the batch instances. This means most pairs of instances are not similar to each other, which motivates the HKD method of modeling the holistic knowledge instead of studying relationships between all pairs of instances. Second, compare with the student network and the CRD network, our proposed HKD method have a more similar visualization result to the teacher network. This demonstrates the effectiveness of the HKD method in distilling holistic knowledge from the teacher network.

5.6. Hyperparameter Tuning

In this subsection, we tune hyperparameters on the CIFAR100 dataset to test the sensitivity of the HKD method. More specifically, we test the number of neighbors in K-Nearest-Neighbor and the β in the loss function.

Figure 6-(a) and Figure 6-(b) illustrate the impact of the number of nearest neighbors. The basic observation is that HKD is not very sensitive to the number of neighbors in graph construction as the performance varies a little with different numbers of neighbors. We get both the best performance in the two tested teacher network and student network architectures when we select 8 neighborhood instances. When the number of neighbors goes larger than 8, we observe a decrease in performance, which is related to the over smoothing of graph neural networks. Figure 6-(c) and 6-(d) illustrate the impact of β on the HKD method. We can observe that the HKD method slightly varies with different β. This is reasonable as the holistic knowledge is of different importance with different β.

6. Conclusion

This paper proposes a holistic knowledge distillation method (HKD) with graph neural networks. Compared with existing methods, the holistic knowledge integrates the individual and the relational knowledge while reserving their inherent correlations. The graph neural networks (GNNs) are utilized to extract the holistic knowledge by aggregating feature representation from relational neighborhood samples. The student network is trained under the supervision of holistic knowledge in a contrastive manner. Extensive experiments are conducted on benchmark datasets to evaluate the performance and the motivation of HKD, the results demonstrate the effectiveness of the HKD method.

7. Acknowledgements

This work is supported by the National Key Research and Development Program (Grant No. 2018YFB1403202), the National Natural Science Foundation of China (Grant No. 61972349,U1866602) Tsinghua GuoQiang Research Center (Grant 2020GQG1014) and Alibaba-Zhejiang University Joint Institute of Frontier Technologies.

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[2] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

[3] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *AAAI*, pages 3430–3437, 2020.

[4] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. *arXiv preprint arXiv:2012.03236*, 2020.

[5] Jiawei Chen, Yan Feng, Martin Ester, Sheng Zhou, Chun Chen, and Can Wang. Modeling users' exposure with social knowledge influence and consumption influence for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 953–962, 2018.

[6] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. Samwalker: Social recommendation with informative sampling strategy. In *The World Wide Web Conference*, pages 228–239, 2019.

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

[10] Jian Du, Shanghang Zhang, Guanhang Wu, José MF Moura, and Soummya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017.

[11] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Carlos Lassance, Myriam Bontonou, Ghouthi Boukli Hacene, Vincent Gripon, Jian Tang, and Antonio Ortega. Deep geometric knowledge distillation with graphs. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8484–8488. IEEE, 2020.

[18] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1576–1585, 2018.

[19] Seunghyun Lee and Byung Cheol Song. Graph-based knowledge distillation by multi-head attention network. In *BMVC*, page 141, 2019.

[20] Xiaojie Li, Jianlong Wu, Hongyu Fang, Yue Liao, Fei Wang, and Chen Qian. Local correlation consistency for knowledge distillation. In *European Conference on Computer Vision*, pages 18–33. Springer, 2020.

[21] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[23] 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.

[24] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[25] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5007–5016, 2019.

[26] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. *arXiv preprint arXiv:1804.08049*, 2018.

[27] 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.

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[29] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

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

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

[32] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[33] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019.

[34] Tiancheng Wen, Shenqi Lai, and Xueming Qian. Preparing lessons: Improve knowledge distillation with better supervision. *arXiv preprint arXiv:1911.07471*, 2019.

[35] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[36] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. *arXiv preprint arXiv:2006.07114*, 2020.

[37] Chenglin Yang, Lingxi Xie, Siyuan Qiao, and Alan L Yuille. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5628–5635, 2019.

[38] 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.

[39] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[40] Mengni Zhang, Can Wang, Zhi Yu, Chao Shen, and Jiajun Bu. Active learning for web accessibility evaluation. In *Proceedings of the 14th International Web for All Conference*, pages 1–9, 2017.

[41] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[42] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

[43] Sheng Zhou, Xin Wang, Jiajun Bu, Martin Ester, Pinggang Yu, Jiawei Chen, Qihao Shi, and Can Wang. Dge: Deep generative network embedding based on commonality and individuality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6949–6956, 2020.

[44] Sheng Zhou, Hongxia Yang, Xin Wang, Jiajun Bu, Martin Ester, Pinggang Yu, Jianwei Zhang, and Can Wang. Prre: Personalized relation ranking embedding for attributed networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 823–832, 2018.

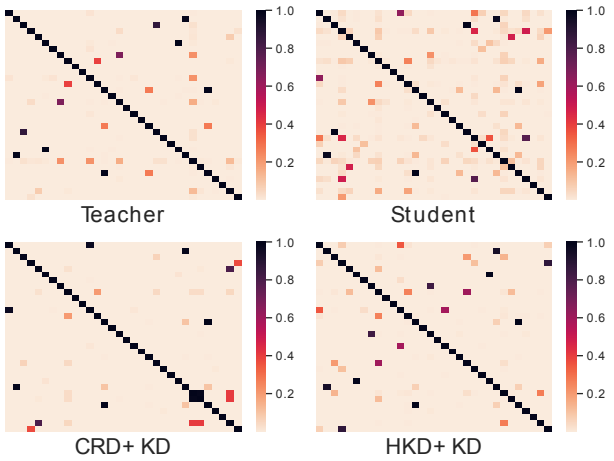

Figure 7. Visualization results on the CIFAR-100 datasets.

8. Supplemental Material

8.1. Datasets and Architectures

We conduct experiments on several benchmark datasets, including CIFAR100 [16], STL-10 [7], TinyImageNet and ImageNet [8]. Four architectures are used for the teacher and student networks, namely ResNet [13], VGG [30], ShuffleNet [41], MobileNet [28].

8.2. Implementation Details

The CIFAR100 dataset consists of 50,000 images of size 32×32 with 500 images per class and 10,000 test images. The TinyImageNet dataset is a subset of ImageNet, consisting of 100,000 images of size 64×64 from 200 classes. STL-10 consists of 5000 labeled training images from 10 classes and 100,000 unlabeled images, and a test set of 8,000 images. To keep our cross-modal transfer experiment's consistency, we down-sample each image to size 32×32. We normalized all images by channel means and standard deviations.

Following the same experimental settings of existing works [31, 23, 25], we use the SGD optimizer with momentum for all networks. For MobileNetV2 and ShuffleNet, we use a learning rate of 0.01. For the rest of the networks, the learning rate is initialized with 0.05. All the learning rates are decayed by 0.1 every 30 epochs after the first 150 epochs until the last 240 epoch. We implement the networks and training procedures in Pytorch.

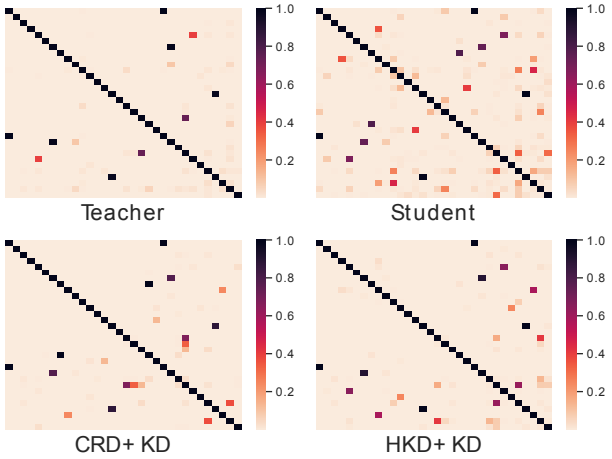

Figure 8. Visualization results on the CIFAR-100 datasets.

8.3. Additional Visualization Results

We further provide more visualization results on the CIFAR-100 datasets, which is illustrated in Fig 7 and Fig 8. We observe the similar results that the proposed HKD method have similar topolopy structure with the teahcer network, which demonstrates the effectiveness of the proposed HKD method.

8.4. Additional Parameter Analysis

We further provide the parameter analysis on the number of layers in Graph Neural Networks. More specifically, we test the one layer(L=1) and two layer(L=2) graph neural networks. We do not set higher number of layers to avoid the over-smoothing problem.

Tabel 5 illustrates the experimental results on the CIFAR100 dataset, from which we can observe that the HKD method is not sensitive to the number of layers.

Table 5. Parameter analysis on number of layers L

Teacher	ResNet32x4	VGG13	ResNet50
Student	ResNet8x4	MobileNetV2	VGG8
L=1	76.13 $\pm$ 0.05	70.48$\pm$0.25	74.85$\pm$0.26
L=2	76.05$\pm$ 0.11	70.28$\pm$0.07	74.82$\pm$0.24