ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10095744

A CONTRASTIVE KNOWLEDGE TRANSFER FRAMEWORK FOR MODEL COMPRESSION AND TRANSFER LEARNING

Kaiqi Zhao, Yitao Chen, Ming Zhao

Arizona State University

ABSTRACT

Knowledge Transfer (KT) achieves competitive performance and is widely used for image classification tasks in model compression and transfer learning. Existing KT works transfer the information from a large model ("teacher") to train a small model ("student") by minimizing the difference of their conditionally independent output distributions. However, these works overlook the high-dimension *structural* knowledge from the intermediate representations of the teacher, which leads to limited effectiveness, and they are motivated by various heuristic intuitions, which makes it difficult to generalize. This paper proposes a novel Contrastive Knowledge Transfer Framework (CKTF), which enables the transfer of sufficient structural knowledge from the teacher to the student by optimizing multiple *contrastive objectives* across the intermediate representations between them. Also, CKTF provides a generalized agreement to existing KT techniques and increases their performance significantly by deriving them as specific cases of CKTF. The extensive evaluation shows that CKTF consistently outperforms the existing KT works by 0.04% to 11.59% in model compression and by 0.4% to 4.75% in transfer learning on various models and datasets.

Index Terms— knowledge transfer, model compression, transfer learning, contrastive learning

1. INTRODUCTION

Knowledge Transfer (KT) is an important and widely used technique for model compression and cross-domain transfer learning. Deep neural networks (DNNs) are difficult to deploy on resource-constrained devices such as the Internet of Things (IoT) and smart devices [1]. KT can address this challenge by using the original model as the teacher to train a much smaller one as the student for deployment on edge devices. Also, DNNs are difficult to train when there is insufficient labeled data. KT can address this data deficiency by transferring knowledge from a teacher model in the source domain trained with abundant labeled data to the student model in the target domain where labels are unavailable.

Various KT techniques [2–13] have been investigated for different image classification models. Hinton et al. first introduced transferring soft logits (softmax outputs) [2], termed Knowledge Distillation (KD), by minimizing the KL divergence between the teacher's and student's soft logits and the cross-entropy loss with the data labels. Later, other works [3–13] proposed to transfer various forms of intermediate representations, such as FSP matrix [12] and attention [4]. However, these works assume that the output dimensions of intermediate layers are independent, and they let the student replicate the teacher's behavior by minimizing the difference between their probabilistic outputs. We argue that the intermediate representations are interdependent, and this minimization fails to capture the important structural knowledge of the teacher's convolution layers. Also, the various KT works are motivated by different intuitions and lack a commonly agreed theory, which makes it challenging to generalize. Moreover, none of the existing KT works consistently outperform the conventional KD [2].

A recent work, CRD [14] formulated KT as optimizing contrastive objectives, usually used for representation learning [15–18]. Their objective is to maximize a lower bound to the mutual information of the outputs of the penultimate layer (before soft logits) between the teacher and student [14]. However, the low dimensionality of the penultimate layer outputs restricts the amount of transferred information. Particularly in cross-domain transfer learning, the penultimate layer outputs of the teacher and student are irrelevant due to the extraneous data from different domains. Moreover, the effectiveness of the contrastive objective on intermediate representations, which are high-dimension and crucial for guiding gradient updates, is currently unexplored.

To address the aforementioned limitations and improve the performance of KT for model compression and transfer learning, we propose a novel Contrastive Knowledge Transfer Framework (CKTF) to enable the transfer of sufficient structural knowledge from the teacher to the student by optimizing multiple *contrastive objectives* across the intermediate representations between them. CKTF defines *positive representation pairs* as the outputs of the teacher's and student's intermediate modules from the same input sample and *negative representation pairs* as from their modules' outputs given two different data samples, respectively. By optimizing the contrastive objectives constructed across all the modules, CKTF pushes each positive representation pair closer while pushing each negative representation pair farther apart, thereby achieving effective knowledge transfer. Moreover, CKTF can incorporate and improve all the existing KT methods by adding their loss terms to the proposed contrastive loss during optimization.

In this paper, we focus on applying CKTF for image classification models. Compared to the existing KT works, CKTF has several advantages: first, compared to the output-level-only communication in the previous contrastive KT approach (CRD), CKTF allows the student to learn the intermediate

This work is partly supported by National Science Foundation awards CNS-1955593 and OAC-2126291. Our code is at: `https://github.com/kaiqi123/CKTF.git`.

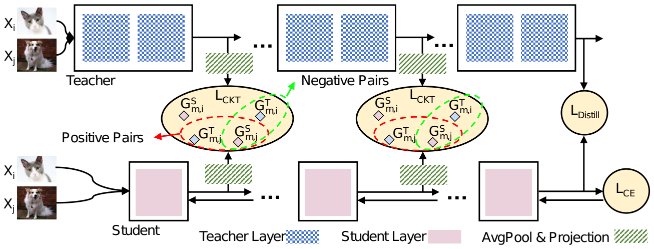


Fig. 1: Workflow of CKTF.

layer state and to capture correlations of high-order output dependencies, leading to faster and better transfer; second, unlike the existing KT works which often perform worse than the conventional KD, CKTF consistently outperforms the conventional KD in all cases; finally, CKTF provides a generalized agreement to existing KT methods and can incorporate existing works to significantly enhance their performance.

Our comprehensive evaluation shows that CKTF outperforms the existing KT works (KD, CRD, and 12 other solutions) significantly. For model compression using CIFAR-100 and Tiny-ImageNet, CKTF yields an accuracy improvement of 0.04% to 11.59% than the existing KT methods, and 0.95% to 4.41% compared to training the student directly using all the data. For transfer learning from Tiny-ImageNet to STL-10, CKTF converges faster than all the baselines and outperforms their accuracy by 0.4% to 4.75%.

2. METHODOLOGY

2.1. Framework Overview

Let $X = \{x_i\}_{i=1}^{B}$ and $Y = \{y_i\}_{i=1}^{B}$ denote a set of inputs with a batch size of B and its ground truth labels Y, respectively. We define a module as a group of convolution layers. The output representations of the modules from the teacher and student can be described as $\{T_m\}_{m=1}^{M}$ and $\{S_m\}_{m=1}^{M}$, respectively, where M denotes the number of modules. Similarly, let T_h and S_h denote the output vectors of the penultimate layer from the teacher and student, respectively.

Figure 1 illustrates the workflow of the proposed Contrastive Knowledge Transfer Framework (CKTF). The optimization objective in CKTF consists of three components: 1) the cross entropy loss with the ground truth labels; 2) the proposed contrastive loss to transfer knowledge from the intermediate representations $\{T_m\}_{m=1}^{M}$ and the penultimate layer T_h of the teacher to $\{S_m\}_{m=1}^{M}$ and S_h of the student, respectively; and 3) the distillation loss from other KT methods. The loss function of CKTF is as follows:

$$\begin{aligned} L = \gamma L_{CE}(Y, S_h) + L_{CKT}(\{T_m\}_{m=1}^{M}, \{S_m\}_{m=1}^{M}, T_h, S_h) \\ + \theta L_{Distill}(T_h, S_h) \end{aligned} \tag{1}$$

where γ equals either 1 or 0 depending on the availability of labels, and θ is a hyper-parameter that controls the weight of the loss term.

The first loss term in Eq. 1 enforces the supervised learning from labels, which is typically implemented as a cross-entropy loss for classification tasks: $L_{CE}(Y, S_h) = \sum_{i=1}^{c}[Y_i log(S_{h,i}) + (1 - Y_i) log(1 - S_{h,i})]$, where c denotes the number of classes of the dataset. The second loss term is the proposed contrastive loss that transfers high-dimension structural knowledge from both the intermediate presentations and the penultimate layer via contrastive learning. It works because, as opposed to just transferring information about conditionally independent output class probabilities, the multiple contrastive objectives constructed in L_{CKT} better transfer all the information in the teacher's representational space (see Section 2.2 for details). The third loss term is used to incorporate existing KT methods into CKTF. For example, for the conventional KD [2], it is defined as the KL-divergence-based loss that minimizes the difference between the teacher's and student's soft logits: $L_{Distill}(T_h, S_h) = KL(softmax(T_h/\rho)||softmax(S_h/\rho))$, where ρ is the temperature. In this way, CKTF can help improve the performance of existing KT methods (see Section 3.1 for evaluation results).

Note that, in transfer learning, γ (Eq. 1) is set to zero since supervision from labels is not available.

2.2. Contrastive Knowledge Transfer

CKTF constructs the contrastive loss across intermediate representations from multiple modules of the teacher and student. Directly using intermediate representations $\{T_m\}_{m=1}^{M}$ and $\{S_m\}_{m=1}^{M}$ to perform contrastive learning is infeasible, since 1) the dimension between T_m and S_m might be different, and 2) the huge feature dimension of T_m and S_m may cause memory issues or significantly increase the training time. In detail, the dimension of S_m is calculated as $|S_m| = B \times o_m^s \times (k_m^s)^2$, where B, o_m^s and k_m^s denote the batch size, output dimension, and kernel size of the module m of the student. For example, for ResNet-50 on Tiny-ImageNet with a batch size of 32, the feature dimension of one intermediate module can be: $32 \times 1024 \times 16^2 \approx 8.39$ millions, and its teacher may also have a similar level of the feature dimension.

To solve the above problem, CKTF first applies an average pooling over $T_m \in R^{B \times o_m^t \times k_m^t \times k_m^t}$ and $S_m \in R^{B \times o_m^s \times k_m^s \times k_m^s}$, respectively, and it produces the output $\bar{T}_m \in R^{B \times o_m^t \times 1 \times 1}$ and $\bar{S}_m \in R^{B \times o_m^s \times 1 \times 1}$, respectively. Then it uses a reshaping function $h(\cdot)$ that changes the 4-D $\bar{T}_m$ and $\bar{S}_m$ to a 2-D space, yielding $H_m^T \in R^{B * o_m^t}$ and $H_m^S \in R^{B * o_m^s}$, respectively:

$$\begin{aligned} &\bar{S}_m = AvgPool(S_m), \bar{T}_m = AvgPool(T_m) \\ &H_m^S = h(\bar{S}_m), H_m^T = h(\bar{T}_m) \end{aligned} \tag{2}$$

Next, a projection network $g(\cdot)$ takes the presentations $\{H_m^T\}_{m=1}^{M}$ and $\{H_m^S\}_{m=1}^{M}$ as the input, and for the module m, it produces: $G_m^T = g(H_m^T) \in R^{B \times d}$ and $G_m^S = g(H_m^T) \in R^{B \times d}$, respectively, where d denotes the output dimension of the projection network. $g(\cdot)$ used in CKTF is a single linear layer of size $d = 128$ followed by the ℓ_2 normalization. Note that $g(\cdot)$ is discarded after training, so we do not change the model architecture. We will show that the linear projection is better than Multi-Layer Perceptron (MLP) projection used in representation learning [15–18] and discuss the effect of d in Section 3.3.

CKTF constructs the contrastive loss using $\{G_m^T\}_{m=1}^{M}$ and $\{G_m^S\}_{m=1}^{M}$. Given a batch of random samples $X = \{x_i\}_{i=1}^{B}$, we define *positive representation pairs* as $(G_{m,i}^S, G_{m,i}^T)$, which are the outputs of the student's and teacher's module m from the same input sample x_i, and *negative representation pairs* as

Table 1: Top-1 test accuracy (%) on CIFAR-100 and Tiny-ImageNet. Red/black arrows denote the increase/decrease compared to conventional KD.

DataSet	CIFAR-100							Tiny-ImageNet			
Model											
Teacher	WRN-40-2	WRN-40-2	ResNet-56	ResNet-110	ResNet-110	ResNet-32*4	VGG-13	VGG-19	VGG-16	ResNet-34	ResNet-50
Student	WRN-16-2	WRN-40-1	ResNet-20	ResNet-20	ResNet-32	ResNet-8*4	VGG-8	VGG-8	VGG-11	ResNet-10	ResNet-10
Compression Ratio	3.21	3.96	3.10	6.24	3.67	6.03	2.39	5.01	1.59	4.28	4.78
Baselines											
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64	61.62	61.35	65.38	65.34
Student (w/o KT)	73.26	73.54	69.06	69.06	71.14	72.5	70.36	54.61	58.60	58.01	58.01
Method											
KD [2]	74.92	73.54	70.66	70.67	73.08	73.33	72.98	55.55	62.51	58.92	58.63
FitNet [3]	73.58 (↓)	72.24 (↓)	69.21 (↓)	68.99 (↓)	71.06 (↓)	73.50 (↑)	71.02 (↓)	55.24 (↓)	59.08 (↓)	58.22 (↓)	57.76 (↓)
AT [4]	74.08 (↓)	72.77 (↓)	70.55 (↓)	70.22 (↓)	72.31 (↓)	73.44 (↑)	71.43 (↓)	53.55 (↓)	61.40 (↓)	59.16 (↑)	58.92 (↑)
SP [5]	73.83 (↓)	72.43 (↓)	69.67 (↓)	70.04 (↓)	72.69 (↓)	72.94 (↓)	72.68 (↓)	55.09 (↓)	61.61 (↓)	55.91 (↓)	57.17 (↓)
CC [6]	73.56 (↓)	72.21 (↓)	69.63 (↓)	69.48 (↓)	71.48 (↓)	72.97 (↓)	70.71 (↓)	54.87 (↓)	58.34 (↓)	57.18 (↓)	57.36 (↓)
VID [7]	74.11 (↓)	73.3 (↓)	70.38 (↓)	70.16 (↓)	72.61 (↓)	73.09 (↓)	71.23 (↓)	54.94 (↓)	60.07 (↓)	58.53 (↓)	57.65 (↓)
RKD [8]	73.35 (↓)	72.22 (↓)	69.61 (↓)	69.25 (↓)	71.82 (↓)	71.90 (↓)	71.48 (↓)	54.13 (↓)	59.96 (↓)	57.35 (↓)	57.05 (↓)
PKT [9]	74.54 (↓)	73.45 (↓)	70.34 (↓)	70.25 (↓)	72.61 (↓)	73.64 (↑)	72.88 (↓)	55.35 (↓)	60.46 (↓)	58.41 (↓)	58.66 (↑)
AB [10]	72.50 (↓)	72.38 (↓)	69.47 (↓)	69.53 (↓)	70.98 (↓)	73.17 (↓)	70.94 (↓)	50.31 (↓)	55.65 (↓)	57.22 (↓)	58.05 (↓)
FT [11]	73.25 (↓)	71.59 (↓)	69.84 (↓)	70.22 (↓)	72.37 (↓)	72.86 (↓)	70.58 (↓)	53.65 (↓)	58.84 (↓)	56.22 (↓)	56.48 (↓)
FSP [12]	72.91 (↓)	N/A	69.95 (↓)	70.11 (↓)	71.89 (↓)	72.62 (↓)	70.23 (↓)	N/A	N/A	N/A	N/A
NST [13]	73.68 (↓)	72.24 (↓)	69.60 (↓)	69.53 (↓)	71.96 (↓)	73.30 (↓)	71.53 (↓)	51.08 (↓)	58.47 (↓)	59.23 (↑)	47.83 (↓)
CRD [14]	75.48 (↑)	74.14 (↑)	71.16 (↑)	71.46 (↑)	73.48 (↑)	75.51 (↑)	73.94 (↑)	56.99 (↑)	62.04 (↓)	60.02 (↑)	59.31 (↑)
CKTF	**75.85** (↑)	**74.49** (↑)	**71.20** (↑)	**71.80** (↑)	**73.84** (↑)	**75.74** (↑)	**74.31** (↑)	**57.57** (↑)	**63.01** (↑)	**60.39** (↑)	**59.42** (↑)
CRD+KD [14]	75.64 (↑)	74.38 (↑)	71.63 (↑)	71.56 (↑)	73.75 (↑)	75.46 (↑)	74.29 (↑)	58.09 (↑)	63.66 (↑)	61.99 (↑)	61.26 (↑)
CKTF+KD	**75.89** (↑)	**74.94** (↑)	**71.86** (↑)	**71.66** (↑)	**74.07** (↑)	**75.97** (↑)	**74.55** (↑)	**58.76** (↑)	**63.97** (↑)	**62.31** (↑)	**61.51** (↑)

Table 2: Top-1 test accuracy (%) of KT methods incorporated into CKTF. Numbers inside the parentheses denote the improvement over the original method.

	CKTF+FitNet	CKTF+AT	CKTF+SP	CKTF+CC	CKTF+VID	CKTF+RKD	CKTF+PKT	CKTF+AB	CKTF+FT	CKTF+NST
T: ResNet-32×4 S: ResNet-32×4 (CIFAR-100)	73.18 (1.68 ↑)	74.92 (1.48 ↑)	75.30 (2.36 ↑)	75.86 (2.89 ↑)	75.43 (2.34 ↑)	74.92 (3.02 ↑)	75.82 (2.18 ↑)	75.38 (2.21 ↑)	75.39 (2.53 ↑)	75.08 (1.78 ↑)
T: VGG-19 S: VGG-8 (Tiny-ImageNet)	56.19 (0.95 ↑)	55.33 (1.78 ↑)	56.22 (1.13 ↑)	55.99 (1.12 ↑)	56.34 (1.4 ↑)	55.96 (1.83 ↑)	56.82 (1.47 ↑)	52.63 (2.32 ↑)	56.39 (2.74 ↑)	51.97 (0.89↑)

($G^S_{m,i}, G^T_{m,j}$) from their modules' outputs given two different data samples x_i and x_j, respectively.

CKTF aims to push closer each positive pair $G^S_{m,i}$ and $G^T_{m,i}$ for every input x_i, while pushing $G^S_{m,i}$ apart from $\{G^T_{m,j}\}^N_{j=1,j\neq i}$. N is the number of negative representation pairs. CKTF defines the contrastive loss based on the intermediate representations as follows:

$$L_{MCKT}(G^S_m, G^T_m) = -E\left[log \frac{f(G^S_{m,i}, G^T_{m,i})}{\sum_{j=1}^N f(G^S_{m,i}, G^T_{m,j})}\right] \quad (3)$$

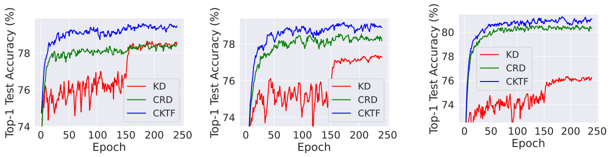
where the function $f(\cdot)$ is similar with that used in [15–18], specifically, $f(G^S_{m,i}, G^T_{m,i}) = \frac{exp(G^S_{m,i} G^T_{m,i}/\tau)}{exp(G^S_{m,i} G^T_{m,i}/\tau)+N/N_d}$. N_d is the number of training samples of the dataset, and τ is a temperature that controls the concentration level. The previous works [15–18] use the function for different domains or objectives, such as self-supervised representation learning [17] and density estimation [16], whereas we are the first to construct multiple contrastive objectives on the intermediate representations of image classification models for knowledge transfer. The minimization of the contrastive loss L_{MCKT} is maximizing the lower bound of the mutual information [15–18] between $\{G^T_m\}^M_{m=1}$ and $\{G^S_m\}^M_{m=1}$.

Similar to L_{MCKT}, CKTF constructs the contrastive objective on the outputs of the penultimate layer as:

$$L_{PCKT}(S_h, T_h) = -E\left[log \frac{f(S_{h,i}, T_{h,i})}{\sum_{j=1}^N f(S_{h,i}, T_{h,j})}\right] \quad (4)$$

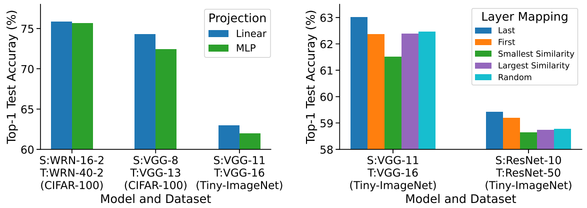
Finally, the proposed contrastive loss (the second loss term in Eq. 1) is defined as the weighted sum of L_{MCKT} and L_{PCKT}:

$$L_{CKT} = \alpha_1 \sum_{m=1}^{M} L_{MCKT}(G^S_m, G^T_m) + \alpha_2 L_{PCKT}(T_h, S_h). \quad (5)$$



(a) T:VGG-19/S:VGG-19 (b) T:VGG-19/S:VGG-8 (c) T:ResNet-18/S:ResNet-18

Fig. 2: Top-1 test accuracy of KD, CRD, and the proposed CKTF on STL-10 when transferring knowledge from Tiny-ImageNet.



(a) Projection (b) Layer Mapping

Fig. 3: The effect of the (a) projection type and (b) layer mapping.

3. EVALUATION

Models and Datasets. We conducted an extensive evaluation of Wide ResNet Networks (WRN), ResNet, and VGG models on 1) CIFAR-100 [19] that consists of 50K 32×32 RGB images with 100 classes, 2) Tiny-ImageNet [20] containing 100K 64×64 images with 200 classes, and 3) STL-10 [21] that contains 5K 10-category labeled training images, 8K test images, and 100K unlabeled images.

Implementation Details. We implemented CKTF on PyTorch version 1.9.0 and conducted experiments on 4 Nvidia RTX 2080 Ti GPUs. The learning rate is initialized to 5e-2 and decays with a rate of 0.1 at epochs 150, 180, and 210. The total training epochs is 240. Weight decay is set to 5e-4. Nesterov

SGD optimizer is used with a momentum of 0.9. N and τ are set to 16384 and 0.1, respectively. When CKTF is evaluated alone, θ is 0, and $\alpha 1$ and $\alpha 2$ are set to 0.8 and 0.2, respectively. When evaluating the related KT methods incorporated into CKTF, θ is set to 1, which means the third loss term in Eq. 1 is the same as the loss used in their papers, and α_1 and α_2 follow the above settings (0.8 and 0.2).

3.1. Model Compression

We compare CKTF for model compression tasks with three baselines: 1) a large model that is uncompressed and directly trained (teacher), 2) a small model directly trained without KT (student w/o KT), and 3) the same small model trained with various KT methods. For the implementation, we use the public CRD code-base [14] to conduct a fair comparison.

Table 1 presents the Top-1 test accuracy of various teacher and student combinations on CIFAR-100 and Tiny-ImageNet. CFKT significantly outperforms all the existing KT methods in all cases. Specifically, CFKT outperforms: 1) the conventional KD [2] by 0.5% to 2.41% (none of the existing methods consistently outperforms KD), 2) the other KT methods by 0.04% to 11.59%, and 3) the related contrastive learning method CRD [14] (the second best in the results) by 0.04% to 0.97%.

Compared to the student trained directly on the data without KT, CFKT is 0.95% to 4.41% better. We also observe that, compared to the student w/o KT, CFKT performs better on Tiny-ImageNet (better than the student w/o KT by 4.41% to 1.41%) than on CIFAR-100 (better than the student w/o KT by 3.95% to 0.95%). This could be because Tiny-ImageNet is more complicated than CIFAR-100 (with more classes and data), resulting in more complicated intermediate representation, whereas CFKT is good at capturing this complicated high-dimension structural knowledge. Further, CFKT enables the small student to achieve comparable performance to the large teacher with only 16% of its original size. This confirms that CFKT is beneficial to on-device image classification applications that require small, high-performance models.

Results on Incorporating KT Methods. We measure the performance of existing KT methods incorporated into CFKT (following Eq. 1), using ResNet-32×4/ResNet-8×4 and VGG-19/VGG-8 as the teacher/student on CIFAR-100 and Tiny-ImageNet. As shown in Table 2, the Top-1 test accuracy of the existing KT works is significantly improved by 0.89% to 3.02% when incorporated into CFKT. The results demonstrate that CFKT provides a generalized agreement behind knowledge transfer. Another observation is that, when incorporating existing KT methods into CFKT, the improvement on the methods that transfer from the last several layers is higher than the methods that transfer from intermediate representations. For example, PKT+CKTF and SP+CKTF achieve an improvement of 2.18% and 2.36%, compared to PKT and SP, respectively, whereas AT+CKTF and FitNet+CKTF achieve an improvement of 1.48% and 1.68%, compared to AT and FitNet, respectively. This is because methods that transfer from the last several layers lack the teacher's intermediate information, which can be compensated by CFKT after they are incorporated into CFKT. So the improvement is larger. For the methods that transfer knowledge from intermediate representations, the transferred information is partial since they do not explicitly capture correlations or higher-order dependencies in representations. The integration of CKTF though still provides additional intermediate information, the improvement to the final accuracy is smaller than that from the methods completely lacking intermediate information.

3.2. Transfer Learning

We first train the teacher using the source domain data (Tiny-ImageNet) with ground truth labels. Then, we transfer knowledge from the teacher to the student using the unlabeled data from the target domain (STL-10) with KT methods. Finally, we fine-tune the student (only train its linear classifier) using the training set of STL-10 and evaluate its accuracy on the test set of SLT-10. This is a common practice [14, 22, 23] for evaluating the quality of transfer learning.

We compare CKTF with KD [2] and CRD [14]. The teacher and student can be either the same, e.g., VGG-19/VGG-19, or different, e.g., VGG-19/VGG-8. Figure 2 shows how the Top-1 test accuracy of the student evolves during fine-tuning on STL-10. CKTF converges faster than all the baselines and outperforms them in final accuracy by 0.4% to 4.75%. This result validates that CKTF is advantageous for cross-domain transfer learning, even without labeled data in the target domain.

3.3. Ablation Study

Effect of Projection. Figure 3a compares the Top-1 test accuracy of three student models trained with CKTF, using the linear vs. MLP projection network (discussed in Section 2.2), on CIFAR-10 and Tiny-ImageNet. Linear projection outperforms MLP projection by 0.15% to 1.85%.

Effect of Output Dimension. We analyze the impact of the output dimension d of the linear projection network on four student models by varying the value of d from 16 to 128. We observe that a larger output always leads to better performance, and $d = 128$ is better than others by 0.03% to 2.2% on CIFAR-10 and Tiny-ImageNet.

Effect of Layer Mapping. Figure 3b illustrates the effect of five strategies for mapping the teacher's and student's layers in each module, including mapping the student's last layer with the teacher's 1) first, 2) last, or 3) a randomly chosen convolution layer or mapping between the layer pair whose outputs have the 4) largest or 5) smallest cosine similarity. Last-layer mapping outperforms others by 0.23% to 1.5% on CIFAR-10 and Tiny-ImageNet.

4. CONCLUSIONS

This paper proposes a novel Contrastive Knowledge Transfer Framework (CKTF) for model compression and transfer learning in image classification. Different from previous KT works, CKTF enables the transfer of high-dimension structural knowledge between the teacher and student by optimizing multiple contrastive objectives across the intermediate representations. It also provides a generalized agreement to existing KT methods and increases their accuracy significantly by deriving them as specific cases of CKTF. An extensive evaluation shows that CKTF consistently outperforms the existing KT works by 0.04% to 11.59% in model compression and by 0.4% to 4.75% in transfer learning. In the future, we will investigate the effectiveness of CKTF in ensemble knowledge transfer and large-scale language model compression.

5. REFERENCES

[1] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen, "Edge ai: On-demand accelerating deep neural network inference via edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 447–457, 2019.

[2] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[3] 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.

[4] 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.

[5] Frederick Tung and Greg Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1365–1374.

[6] 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/CVF International Conference on Computer Vision*, 2019, pp. 5007–5016.

[7] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai, "Variational information distillation for knowledge transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9163–9171.

[8] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3967–3976.

[9] Nikolaos Passalis and Anastasios Tefas, "Probabilistic knowledge transfer for deep representation learning," *CoRR, abs/1803.10837*, vol. 1, no. 2, pp. 5, 2018.

[10] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 3779–3787.

[11] Jangho Kim, SeongUk Park, and Nojun Kwak, "Paraphrasing complex network: Network compression via factor transfer," *Advances in neural information processing systems*, vol. 31, 2018.

[12] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.

[13] Zehao Huang and Naiyan Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," *arXiv preprint arXiv:1707.01219*, 2017.

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

[15] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv e-prints*, pp. arXiv–1807, 2018.

[16] Michael Gutmann and Aapo Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[17] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[18] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola, "What makes for good views for contrastive learning?," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6827–6839, 2020.

[19] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[20] Ya Le and Xuan Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 3, 2015.

[21] 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*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[22] Guillaume Alain and Yoshua Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[23] Richard Zhang, Phillip Isola, and Alexei A Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1058–1067.