Correlation Congruence for Knowledge Distillation

Baoyun Peng*†[1], Xiao Jin*[2], Dongsheng Li[1], Shunfeng Zhou[2]
Yichao Wu[2], Jiaheng Liu[3], Zhaoning Zhang[1], Yu Liu[4]
[1] NUDT [2] SenseTime [3] BUAA [4] CUHK
{pengbaoyun13, dsli, zhangzhaoning}@nudt.edu.cn liujiaheng@buaa.edu.cn
{jinxiao, zhoushunfeng, wuyichao}@sensetime.com yuliu@ee.cuhk.edu.hk

Abstract

Most teacher-student frameworks based on knowledge distillation (KD) depend on a strong congruent constraint on instance level. However, they usually ignore the correlation between multiple instances, which is also valuable for knowledge transfer. In this work, we propose a new framework named correlation congruence for knowledge distillation (CCKD), which transfers not only the instance-level information, but also the correlation between instances. Furthermore, a generalized kernel method based on Taylor series expansion is proposed to better capture the correlation between instances. Empirical experiments and ablation studies on image classification tasks (including CIFAR-100, ImageNet-1K) and metric learning tasks (including ReID and Face Recognition) show that the proposed CCKD substantially outperforms the original KD and achieves state-of-the-art accuracy compared with other SOTA KD-based methods. The CCKD can be easily deployed in the majority of the teacher-student framework such as KD and hint-based learning methods. Our code will be released, hoping to nourish our idea to other domains.

1. Introduction

Over the past few decades, various deep neural network (DNN) models have achieved state-of-the-art performance in many vision tasks [32, 33, 11]. Generally, networks with many parameters and computations perform superior to those with fewer parameters and computations when trained on the same dataset. Nevertheless, it's difficult to deploy such large networks on resource-limited embedded systems. Along with the increasing demands for low cost networks running on embedded systems, there is an urgency for getting smaller network with less computation and memory consumptions, while narrowing the gap of performance between minor network and large network.

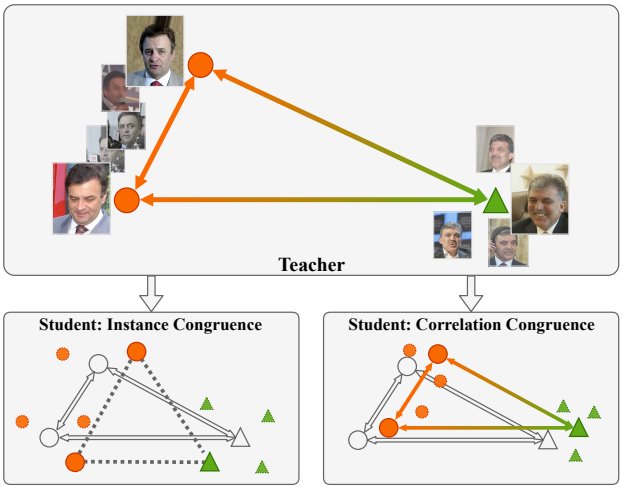


Figure 1: The difference between instance congruence and correlation congruence. When focusing on only instance congruence, the correlation between instances of student may be much different from the teacher's, and the cohesiveness of intra-class would be worser. CCKD solve the problem by adding a correlation congruence when transferring knowledge.

Several techniques have been proposed to address this issue, e.g. parameter pruning and sharing [14, 28], compact convolutional filters [42, 19], low-rank factorization [22, 8] and knowledge distillation [18]. Among these approaches, knowledge distillation has been proved as an effective way to promote the performance of small network by mimicking the behavior of a high-capacity network. It works by adding a strong congruent constraint on outputs of teacher and student for each input instance to encourage the student to mimic teacher's behavior, e.g. minimizing the Kullback–Leibler divergence of predictions [18] or minimizing the euclidean distance of feature representations [25] between teacher and student.

*Equal contribution.
†This work was done when Baoyun Peng was an intern at SenseTime Inc.

However, it's hard for the student to learn a mapping function identical to the teacher's due to the gap (in capacity) between teacher and student. By focusing on only instance congruence, the student would learn a much more different instances correlation from the teacher as shown in figure 1. Usually, the embedding space of teacher possesses the characteristic that intra-class instances cohere together while inter-class instances separate from each other. But its counterpart of student model trained by instance congruence would lack such desired characteristic.

We claim that beyond instance congruence, the correlation between instances is also valuable knowledge for promoting the performance of student. Based on this philosophy, we propose a new distillation framework called Correlation Congruence Knowledge Distillation (CCKD) which focus on not only instance congruence, but also correlation congruence to transfer the correlation knowledge between instances to the student as shown in Figure 1. CCKD can be easily implemented and trained with mini-batch, and only requires the same dimension of embedding space for teacher and student network. To cope with the dismatch of feature representations of teacher student network on image classification tasks, we apply a fully-connected layer with the same dimension for both teacher and student network. We conduct various experiments on four representative tasks and different networks to validate the effectiveness of the proposed approach.

Our contributions in this paper are summarized as follows:

1. We propose a new distillation framework named correlation congruence knowledge distillation (CCKD), which focuses on not only instance congruence but also correlation congruence. To the best of our knowledge, it is the first work to introduce correlation congruence to distillation;

2. We introduce a general kernel-based method to better capture the correlation between instances in a mini-batch. We have evaluated and analyzed the impact of different correlation metrics on different tasks;

3. We explore different sampler strategies for mini-batch training to further improve the correlation knowledge transfer;

4. Extensive empirical experiments and ablation studies show the effectiveness of proposed method in different tasks (CIFAR-100, ImageNet-1K, person re-identification and face recognition) to improve the distillation performance.

2. Related Work

Since this paper focuses on training a small but high performance network based on knowledge distillation, we discuss related works in model compression and acceleration, knowledge distillation in this section. In both areas, there are various approaches have been proposed over the past few years. We summarize them as follows.

Model Compression and Acceleration. Model compression and acceleration aim to create network with few computation and parameters cost meanwhile maintaining high performance. A straight way is to design lightweight but powerful network since the original convolution network has many redundant parameters. For example, depthwise separable convolution is used to replacing standard convolution for building block in [19]. Pointwise group convolution and channel shuffle are proposed to reduce the burden of computation while maintaining high accuracy in [42]. Another way is network pruning which boosts the speed of inference by pruning the neurons or filters with low importance based on certain criteria [14, 28]. In [22, 8], weights were decomposed through low-rank decomposition to save memory cost. Quantization seeks to use low-precision bits to store model's weights or activation outputs [13, 20, 38].

Knowledge Distillation. Transferring knowledge from a large network to a small network is a classical topic and has drawn much attention in recent years. In [18], Hinton *et al*. propose knowledge distillation (KD), in which the student network was trained by the soft output of an ensemble of teacher networks. Comparing to one-hot label, the output from teacher network contains more information about the fine-grained structure among data, consequently helps the student achieve better performance. Since then, there have been works exploring variants of knowledge distillation. In [3], Ba and Caruana show that the performance of a shallower and wider network trained by KD can approximate to deeper ones. Romero *et al*. [29] propose to transfer the knowledge using not only final outputs but also intermediate ones, and add a regressor on intermediate layers to match different size of teacher's and student's outputs. In [41], the authors propose an attention-based method to match the activation-based and gradient-based spatial attention maps. In [40], the flow of solution procedure (FSP) , which is generated by computing the Gram matrix of features across layers, was used for knowledge transfer. To improve the robustness of the student, Sau and Balasubramanian [31] perturbe the logits of teacher as a regularization.

Different from above offline training methods, several works adopts collaboratively training strategy. Deep mutual learning [43] conducts distillation collaboratively for peer student models by learning from each other. Anil *et al*. [1] further extend this idea by online distillation of multiples networks. In their work, networks are trained in parallel and the knowledge is shared by using distillation loss to accelerate the training process.

Besides, there are several works utilizing adversarial method to modeling knowledge transfer between teacher and student [39, 16, 17]. In [39], they adopt generative adversarial networks combined with distillation to learn the loss function to better transfer teacher's knowledge to student. Byeongho *et al*. [17] adopt adversarial method to discover adversaial samples supporting decision boundary.

In this paper, beyond instance knowledge, we take the correlation in embedded space between instances as valuable knowledge to transfer correlation among instances in the embedded space between for knowledge distillation.

3. Correlation Congruence Knowledge Distillation

In this section, we describe the details of proposed method based on correlation congruence for knowledge distillation.

3.1. Background and Notations

We refer a well-performed teacher network with parameters $\boldsymbol{W}_t$ as T and a new student network with parameters $\boldsymbol{W}_s$ as S like in [18, 41, 40, 1, 29]. The input dataset of the network is noted as $\chi = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$, and the corresponding ground truth is noted as $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n\}$, n represents the number of samples in dataset. Since deep network can be viewed as a mapping function stacked by multiple non-linear layers, we note $\phi_t(\boldsymbol{x}; \boldsymbol{W}_t)$ and $\phi_s(\boldsymbol{x}; \boldsymbol{W}_s)$ as the mapping functions of teacher and student, $\boldsymbol{x}$ represents the input data. $\boldsymbol{f}_s$ and $\boldsymbol{f}_t$ represent the feature representations of teacher and student. The logits of teacher and student note as $\boldsymbol{z}_t = \phi(\boldsymbol{x}; \boldsymbol{W}_s)$ and $\boldsymbol{z}_s = \phi(\boldsymbol{x}; \boldsymbol{W}_t)$. $\boldsymbol{p}_t = softmax(\boldsymbol{z}_t)$ and $\boldsymbol{p}_s = softmax(\boldsymbol{z}_s)$ represent the final prediction probilities of teacher and student.

3.2. Knowledge Distillation

Overparameterized networks have shown powerful optimization properties to learn the desired mapping function from data [10], of which the output reflects fine-grained structure one-hot labels might ignore. Based on this insight, knowledge distillation was first proposed in [5] for model compression, then Hinton *et al*. [18] popularized it. The idea of knowledge distillation is to let the student mimic the teacher's behavior by adding a strong congruent constraint on predictions [5, 18, 29] using KL divergence

$$L_{\text{KD}} = \frac{1}{n}\sum_{i=1}^{n} \tau^2 KL(\boldsymbol{p}_s^\tau, \boldsymbol{p}_t^\tau), \tag{1}$$

where τ is a relaxation hyperparameter (referred as temperature in [18]) to soften the output of teacher network, $\boldsymbol{p}^\tau = softmax(\frac{\boldsymbol{z}}{\tau})$. In several works [34, 25] the KL divergence is replaced by euclidean distance,

$$L_{\text{mimic}} = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{f}_s - \boldsymbol{f}_t\|_2^2. \tag{2}$$

Regardless of congruent constraint on final predictions [18], feature representations [34] or activations of hidden layer [29], these methods only focus on instance congruence while ignore the correlation between instances. Due to the gap (in capacity) between teacher and student, it's hard for student to learn a identical mapping function from teacher by instance congruence. We argue that the correlation between instances is also vital for classification since it directly reflect how the teacher model the structure of different instances in embedded feature space.

3.3. Correlation Congruence

In this section, we describe correlation congruence knowledge distillation (CCKD) in detail. Different from previous methods, CCKD considers not only the instance level congruence but also correlation congruence between instances. Figure 2 shows the overview of CCKD. CCKD consists of two part: instance congruence (KL divergence on predictions of teacher and student) and correlation congruence (euclidean distance on correlation of teacher and student).

Let $\boldsymbol{F}_t$ and $\boldsymbol{F}_s$ represent the set of feature representations of teacher and student respectively,

$$\begin{aligned} \boldsymbol{F}_t &= \text{matrix}\big(\boldsymbol{f}_1^t, \boldsymbol{f}_2^t, ..., \boldsymbol{f}_n^t\big), \\ \boldsymbol{F}_s &= \text{matrix}\big(\boldsymbol{f}_1^s, \boldsymbol{f}_2^s, ..., \boldsymbol{f}_n^s\big). \end{aligned} \tag{3}$$

The feature $\boldsymbol{f}$ can be seen as a point in the embedded feature space. Without loss of generality, a mapping function is introduced as follow:

$$\psi : \boldsymbol{F} \rightarrow \boldsymbol{C} \in \mathbb{R}^{n\times n}. \tag{4}$$

where $\boldsymbol{C}$ is a correlation matrix. Each element in $\boldsymbol{C}$ represents the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in embedding space, which is defined as

$$\boldsymbol{C}_{ij} = \varphi(\boldsymbol{f}_i, \boldsymbol{f}_j), \quad \boldsymbol{C}_{ij} \in \mathbb{R} \tag{5}$$

The function φ can be any correlation metric, and we will introduce three metric for capturing the correlation between instances in next section. Then, the correlation congruence can be formulated as follow:

$$\begin{aligned} L_{CC} &= \frac{1}{n^2}\|\psi(\boldsymbol{F}_t) - \psi(\boldsymbol{F}_s)\|_2^2 \\ &= \frac{1}{n^2}\sum_{i,j}(\varphi(\boldsymbol{f}_i^s, \boldsymbol{f}_j^s) - \varphi(\boldsymbol{f}_i^t, \boldsymbol{f}_j^t))^2. \end{aligned} \tag{6}$$

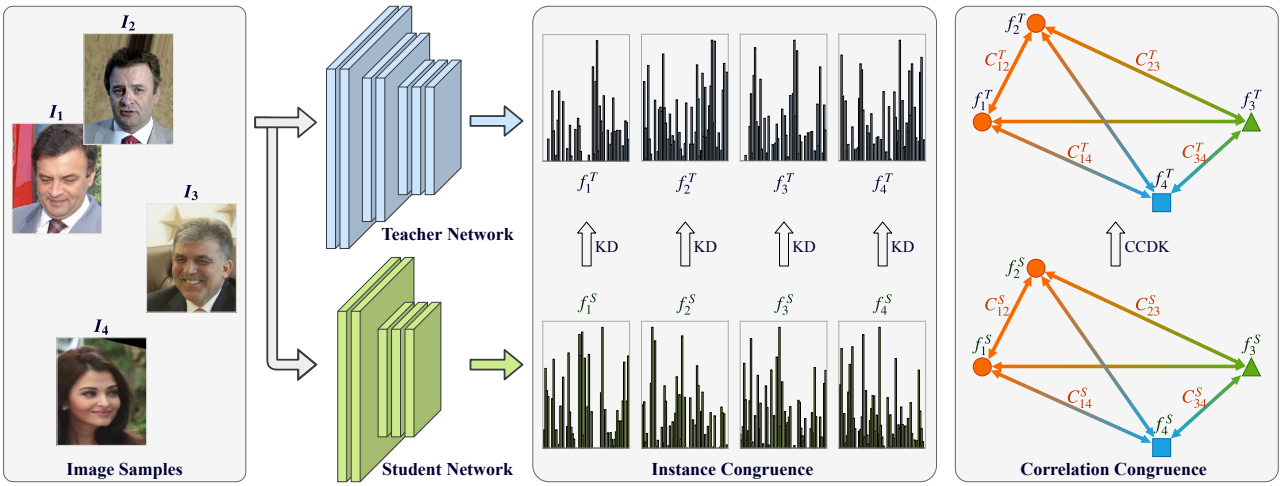


Figure 2: The overall framework of correlation congruence for knowledge distillation (T: teacher; S: student; f_i^T: teacher's output of i_{th} sample; f_i^S: student's output of i_{th} sample; C_i: correlation between i_{th} and j_{th} sample). Original KD focus on only instance congruence between teacher and student network. While CCKD aims to not only instance congruence but also correlation congruence between multiple instances.

Then, the optimization goal of CCKD is to minimize the following loss function:

$$L_{CCKD} = \alpha L_{\text{CE}} + (1-\alpha) L_{\text{KD}} + \beta L_{\text{CC}}, \tag{7}$$

where L_{CE} is the cross-entropy loss, α and β are two hyperparamemters for balancing correlation congruence and instance correlation.

3.4. Generalized kernel-based correlation

Capturing the complex correlations between instances is not easy due to a very high dimension in the embedded space [35]. In this section, we introduce kernel trick to capture the high order correlation between instances in the feature space.

Let $\boldsymbol{x}, \boldsymbol{y} \in \Omega$ represent two instances in feature space, and we introduce different mapping functions $k : \Omega \times \Omega \mapsto \mathbb{R}$ as correlation metric, including:

1. naive MMD: $k(\boldsymbol{x}, \boldsymbol{y}) = \left| \frac{1}{n} \sum_i \boldsymbol{x}_i - \frac{1}{n} \sum_i \boldsymbol{y}_i \right|$;
2. Bilinear Pool: $k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$;
3. Gaussian RBF: $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{2\delta^2})$;

MMD can reflect the distance between mean embeddings. Bilinear Pooling [26] can be seen as a naive 2-th order function, of which the correlation between two instances is computed by element-wise dot product. Gaussian RBF is a common kernel function whose value depends only on the euclidean distance from the origin space.

Comparing to naive MMD and Bilinear Pool, Gaussian RBF is more flexible and powerful in capturing the complex non-linear relationship between instances. Based on Gaussian RBF, the correlation mapping function ϕ can be computed by a kernel function $K : F \times F \to \mathbb{R}^{n \times n}$, where each element can be computed as

$$[k(\boldsymbol{F}, \boldsymbol{F})]_{ij} \approx \sum_{p=0}^{P} \alpha_p (\boldsymbol{F}_{i\cdot} \cdot \boldsymbol{F}_{j\cdot}^\top)^P. \tag{8}$$

which can be approximated by P-order Taylor series. Once specifying the kernel function, then the coefficient α_p is also confirmed. Each element $[k(\boldsymbol{F}, \boldsymbol{F})]_{ij}$ encodes the pairwise correlations between i-th and j-th features in $\boldsymbol{F}$. We take Gaussian RBF kernel function as an example, then

$$\begin{aligned} [k(\boldsymbol{F}, \boldsymbol{F})]_{ij} &= \exp(-\gamma \left\| \boldsymbol{F}_i - \boldsymbol{F}_j \right\|^2) \\ &\approx \sum_{p=0}^{P} \exp(-2\gamma) \frac{(2\gamma)^p}{p!} (\boldsymbol{F}_{i\cdot} \cdot \boldsymbol{F}_{j\cdot}^\top)^p. \end{aligned} \tag{9}$$

where γ is a tunable parameter.

3.5. Strategy for Mini-batch Sampler

Usually, stochastic gradient descent (SGD), which samples batch of training examples uniformly at random from training dataset, is adopted to train the network and then parameters are updated using the sampled batch of examples. The naive random sampler would lead to such a situation that all examples come from different classes. Although it is an unbiased estimation to true gradient of instance congruence, uniformly sampled mini-batch gradient

descent will result in a high biased estimation to gradient of intra-class correlation. To transfer the true correlation information unbiasedly, a proper sampler strategy is important.

To balance the intra-class and inter-class correlation congruence, we propose two strategies for mini-batch sampler: class-uniform random sampler (CUR-sampler) and superclass-uniform random sampler (SUR-sampler). CUR-sampler samples by class and randomly selects fixed k number of examples for each sampled class (eg. each batch consists of 6 class and each class contains $k = 8$ examples, forming a 48 batch size). SUR-sampler is similar to CUR-sampler, but different in that it samples examples by superclass, a more soft form of true class generated by clustering. To get the superclass of training examples, we first extract the feature using teacher model, then use the K-means to cluster. The superclass of example is defined as the cluster it belongs to. Comparing to CUR-sampler, SUR-sampler is more flexible and tolerant for imbalance label since the superclass reflects the coarse structure of instances in embedded space.

3.6. Complexity analysis and implementation details

To cope with the mini-batch training, we compute the correlation in a mini-batch. Formula 9 involves the computation of a large pairwise matrix $b \times b$ (b is the batch size), and each element is approximated by p-order Taylor-series with p times dot product computation between two d dimension vectors. The total computation complexity is $O(pbd^2)$ in a mini-batch, and the extra space consumption is $O(b^2 + d^2)$ for storing the correlation matrix. Compared to huge parameters and computation for training deep neural network, the time and computation consumption for correlation congruence can be ignored. Besides, since the correlation congruent constraint is added on embedding space, it only requires that the feature dimension of student network is the same as teacher. To cope with the mismatch dimension in classification tasks, a fully-connected layer with fixed-length dimension is added for both teacher and student network, which has minor influence on other methods in this paper.

4. Experiments

We evaluate CCKD on multiple tasks, including image classification tasks (CIFAR-100 and ImageNet-1K) and metric learning tasks (including MSMT17 dataset ReID and MegaFace for face recognition), and compare it with closely related works. Extensive experiments and analysis are conducted to delve into the correlation congruence knowledge distillation.

4.1. Experimental Settings

Network Architecture and Implementation Details Given the steady performance and efficiency computation, ResNet [15] and MobileNet [30] network are chosen in this work.

In the main experiments, we set the order $P = 2$, and compute Equation 9 in a mini batch. For the networks in CIFAR-100 and ImageNet-1K, we add a fully-connected layer with 128-d output to form a sharing embedding space for teacher and student. The hyper-parameter α is set to zero, and correlation congruence scale $beta$ is set to 0.003, $\gamma = 0.4$. CUR-sampler is used for all the main experiments with $k = 4$.

On CIFar-100, ImageNet-1K and MSMT17, Original Knowledge distillation (KD) [18] and cross-entropy (CE) are chosen as the baselines. For face recognition, ArcFace loss [7] and $L2$-mimic loss [25, 27] are adopt. We compare CCKD with several state-of-the-art distillation related methods, including attention transfer (AT) [41], deel mutual learning (DML) [43] and conditional adversarial network (Adv) [39]. For attention transfer, we add it for last two blocks as suggested in [41]. For adversarial training, the discriminator consists of FC(128×64) + BN + ReLU + FC (64×2) + Sigmoid activation layers, and we adopt BinaryCrossEntropy loss to train it. All the networks and training procedures are implemented in PyTorch.

4.2. Classification Results on CIFAR-100

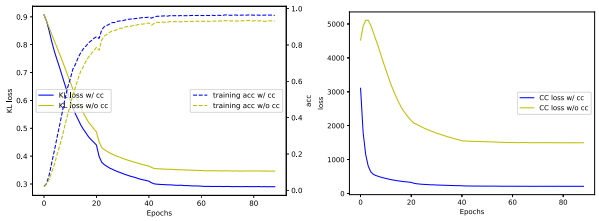
CIFAR-100 [24] consists of colored natural images with 32×32 size. There are 100 classes in CIFAR-100, each class contains 500 images in training set and 100 images in validation set. We use a standard data augmentation scheme (flip/padding/random crop) that is widely used for these dataset, and normalize the input images using the channel means and standard deviations. We set the weight decay of student network to $5e-4$., batch size to 64, and use stochastic gradient descent with momentum. The starting learning rate is set as 0.1, and divided by 10 at 80, 120, 160 epochs, totally 200 epochs. Top-1 and top-5 accuracy are adopted as performance metric.

Table 1 summarizes the results of CIFAR-100. CCKD gets a 72.4% and 70.2% of top-1 accuracy for ResNet-20 and ResNet-14, and substantially surpasses the CE by 4.0% and 3.8%, 1.6% and 1.9% over KD. For the online distillation DML [43], we train target network (ResNet-14 and ResNet-20) collaboratively with ResNet-110, and evaluate performance of target network. Comparing to other SOTA methods, CCKD still significantly outperforms them. All the four distillation related methods significantly surpass the original CE over 2%, which verifies the effectiveness of teacher-student methods.

Figure 3 shows the training loss and validation accuracy of ResNet-20. It can be observed that although KL

Table 1: Validation accuracy results on CIFAR-100. ResNet-110 is as teacher network, ResNet-20 and ResNet-14 as student networks. We keep the same training configuration for all the methods for fair comparison.

method	resnet-20		resnet-14	
	top-1	top-5	top-1	top-5
CE	68.4	91.3	66.4	90.3
KD	70.8	92.4	68.3	90.7
DML	71.2	92.5	69.1	91.2
AT	71.0	92.4	68.6	91.1
Adv	70.5	92.1	68.1	90.6
CCKD	**72.4**	**92.9**	**70.2**	**92.0**



(a) KD loss and top-1 acc (b) CC loss

Figure 3: The curve of training loss and validation accuracy.

divergence loss after convergence is almost the same, the correlation congruence loss for CCKD is much lower than original KD, consequently results in a higher performance.

4.3. Results on ImageNet-1K

ImageNet-1K [6] consists 1.28M training images and 50K testing images in total. We adopt the ResNet-50 [15] as the teacher network, MobileNetV2 with 0.5 width multiplier as the student network. The data augmentation scheme for training images is the same as [15], and apply a center-crop at test time. All the images are normalized using the channel means and standard deviations. We set the weight decay of student network to $5e-4$, batch size to 1,024 (training on 16 TiTAN X, each with 64 batch size), and use stochastic gradient descent with momentum. The starting learning rate is set as 0.4, then divided by 10 at 50, 80, 120 epochs, totally 150 epochs.

For fair comparison, we keep the same configuration for all the methods. Table 2 summarizes the results on ImageNet 1K. CCKD gets a 67.7% Top-1 accuracy, which surpasses the cross-entropy by 3.3%. Compare with original KD[18], CCKD surpasses by 1.0% in top-1 accuracy. AT and DML perform worse than original KD. To our best knowledge, we have not found any works that successfully verify the effectiveness of KD on ImageNet-1K dataset. It has been reported in work [41] that KD struggles to work when the architecture and depth of student network are different from the teacher. But we found that by removing the dropout layer and using a proper temperature (T in [4,8]), the KD can surpass the student over 2.0%.

Table 2: Validation accuracy results on ImageNet 1K. The teacher network is ResNet-50, student network is MobileNetV2 with 0.5 width multiplier. We keep the same configuration for CE and other four student networks.

method	top-1 accuracy	top-5 accuracy
teacher	75.5	92.7
CE	64.2	85.4
KD	66.7	87.3
DML	65.3	86.1
Adv	66.8	87.3
AT	65.4	86.1
CCKD	**67.7**	**87.7**

4.4. Person Re-Identification on MSMT17

Comparing to closed set classification, open set classification is more dependent on a good metric learning and more realistic scenario. We apply the proposed method to two open-set classification: person re-identification (ReID) and face recognition.

For ReID, we evaluate proposed method on MSMT17 [37]. It contains 180 hours of videos captured by 12 outdoor cameras, 3 indoor cameras under different seasons and time. There are 126,441 bounding boxes of 4,101 identities that are annotated. All the bounding boxes are split to training set (32621 bounding boxes, 1041 identities), query set (11659 bounding boxes, 3060 identities) and gallery set (82161 bounding boxes). There is no intersection of identities between training set and query & gallery set. We train the networks on training set, and perform identification on query and gallery set. Rank-1&5 and mean average precision (mAP) are adopted as performance metric.

ResNet-50 is used as the teacher network and ResNet-18 as student network. The dimension of the feature representation is set to 256. We set the weight decay to $5e-4$, batch size to 40, and use stochastic gradient descent with momentum. The learning rate is set as 0.0003, then divided by 10 at 45, 60 epochs, totally 90 epochs.

Table 3 summarizes the results of MSMT17 with CCKD, as well as the comparison against other SOTA methods. For fair comparison, all the distillation based methods (except DML) are trained without ImageNet-1K pretraining. For DML, both the results with/without ImageNet-1K pretraining are represented. It can be seen that the performance of the CCKD significantly surpasses KD and other SOTA KD-based methods, and promotes the original KD by 3.1% for rank-1 accuracy and 2.4% for mAP. Without the guidance of teacher, the student trained by cross-entropy only achieves 14.2% mAP, which is much lower than 28.3% of KD.

Table 3: Validation accuracy results on MSMT17. The teacher network is ResNet-50, student network is Resnet-18.

method	pretrained?	rank-1	rank-5	mAP
teacher	yes	66.4	79	34.3
CE	no	32.4	49.0	14.2
DML-1	no	34.5	51.5	16.5
DML-2	yes	50.2	66.4	25.3
KD	no	56.8	72.3	28.3
AT	no	57.6	72.5	28.7
Adv	no	56.0	71.6	27.8
CCKD	no	**59.7**	**74.1**	**30.7**

4.5. Face recognition results on Megaface

Similar to ReID, face recognition is a classical metric learning problem. Learning a discriminative embedded space is the key to get a powerful recognition model. Usually, thousands of identities (class) are required for training a well-performed recognition model. Empirical evidence shows that mimicking the feature layer with hint-based L2 Loss can bring great improvement for small network [25, 27]. In this experiment, instead of using KD loss, we adopt the $L2$-mimic loss. MS-Celeb-1M [12] and IMDB-Face [36] are used as training datasets.

We choose MegaFace [23], a very popular benchmark, as testing set to evaluate the proposed method. MegaFace aims at the evaluation of face recognition algorithms at million-scale of distractors (people who are not in the testing set). We adopt 1:N identification protocol in Megaface to evaluate the different methods. Rank-1 identification rate at different number of distractors is used as metric for evaluation. We set weight decay to 5e-4, batch size to 1024, and use stochastic gradient descent with momentum. The learning rate is set as 0.1, and divided by 10 at 50, 80, 100 epochs, 120 epochs in total. ResNet-50 is used as teacher network, and MobileNetV2 with 0.5 width multipler as student network.

Table 4: Results on Megaface. The teacher network is ResNet-50 trained on MsCeleb-1M [12] and IMDb-face [36] using ArcFace [7]. The student network is MobileNetV2 with a width multiplier=0.5. We keep the same training configuration for mimic, mimic with Adv and CCKD.

method	Rank-1 Identification rate at different distractors					
	ds=10^1	ds=10^2	ds=10^3	ds=10^4	ds=10^5	ds=10^6
teacher	99.76	99.66	99.58	99.49	99.23	98.15
student	99.20	96.37	91.49	84.45	75.60	65.91
mimic	99.63	98.73	97.25	94.39	89.60	83.01
mimic+Adv	99.64	98.80	97.43	94.81	90.52	84.13
CCKD	**99.66**	**99.07**	**97.93**	**95.76**	**91.99**	**86.29**

Table 4 shows the results on Megaface. It can be observed that ArcFace loss, which is trained by only using pure one-hot labels, achieves 65.91% Rank-1 identification rate with 1M distractors. When guided by the teacher using $L2$-mimic loss, the student network can achieves 83.01%, promoting by 18.1%. This result shows that even a much small network can get a substantial improvement of performance when designing proper target and optimization goal. By adding the constraints on correlations among instance, CCKD achieves 86.29% Rank-1 identification rate with 1M distractors, which surpasses the mimicking by 3.28% and 2.16% promotion over Adv [39].

4.6. Ablation Studies

Correlation Metrics. To explore the impact of different correlation metrics on CCKD, we evaluate three popular metrics, namely max mean discrepancy (MMD), Bilinear Pool and Gaussian RBF. We approximate the Gaussian RBF by using 2-order Taylor series. MMD reflects the difference between two isntances in mean embeddings. Bilinear Pool evaluate the similarity of instances pair, and we adopt identity matrix as the linear matrix. When the features are normalized to unit length, it is equal to the cosine similarity. Gaussian RBF is a common kernel function whose value depends only on the euclidean distance from the origin space.

Table 5: Results on MSMT17 with different correlation methods, including MMD, Bilinear Pool and Gaussian RBF. The Gaussian RBF achieves the best result.

correlation metric	rank-1	rank-5	mAP
MMD	58.9	73.6	29.4
Bilinear	59.2	73.8	30.2
Gaussian RBF	59.6	74.0	30.4

Table 5 shows the results of MSMT17 with different correlation metrics. Gaussian RBF achieves the better performance comparing to MMD and Bilinear Pool, while MMD performs worst. So in the main experiments, we use the Gaussian RBF approximated by 2-order Taylor series. All the three correlation matrics greatly surpass the original KD, which proves the effectiveness of correlation in knowledge distillation.

Order of Taylor series. To exploit the high order of correlations between instances, we expand the Gaussian RBF by Taylor series to 1, 2, 3 -order respectively.

Table 6 summarizes the results on MSMT17 with approximated Gaussian RBF at different orders. It can be observed that 3-order is better than $1, 2$-order, and 1-order performs worst. Generally speaking, expanding Gaussian RBF to high order can capture more complex correlations, and consequently achieves higher performance in knowledge distillation.

Impact of Different Sampler Strategies. To explore a proper sampler strategy, we evaluate the impacts of dif-

Table 6: Results on MSMT17 with different order ($p = 1, 2, 3$) Taylor series.

Expand order	rank-1	rank-5	mAP
p=1	59.2	73.7	30.1
p=2	59.6	74	30.4
p=3	60.5	74.5	30.7

ferent sampler strategies including uniform random sampler (UR-sampler), class-uniform random sampler (CUR-sampler) and superclass-uniform random sampler (SUR-sampler) on MSMT17 dataset. For SUR-sampler, the k-means is adopted and the number of clusters is set to 1000 to generate superclass. For fair comparasion the batchsize is set to 40 for all three strategies, and we set different $k = 1, 2, 4, 8, 20$ both for CUR-sampler and SUR-sampler.

Table 7: Results on MSMT17 with different batch sampler strategies. The teacher network is ResNet-50 and the student network is ResNet-18.

sampler	rank-1	rank-5	mAP
UR-sampler	57.2	72.3	28.6
CUR-sampler(k=1)	57.4	72.4	28.8
CUR-sampler(k=2)	58.9	73.6	29.4
CUR-sampler(k=4)	**59.7**	74.1	30.2
CUR-sampler(k=8)	55.7	71.8	29.1
CUR-sampler(k=20)	24.7	40.9	10.7
SUR-sampler(k=1)	56.2	72.2	29.4
SUR-sampler(k=2)	58.3	73.9	29.9
SUR-sampler(k=4)	59.6	**75.0**	**31.1**
SUR-sampler(k=8)	56.2	72.2	29.4
SUR-sampler(k=20)	30.1	47.7	13.7

Table 7 summarizes the results. It can be observed that the sampler strategy have a great impact on performance. Both SUR-sampler and CUR-sampler are sensitive to the value of k, which plays a role of balancing the intra-class and inter-class correlation congruence. When given fixed batch size, a larger k means a smaller number of classes in a mini-batch. Both CUR-sampler and SUR-sampler become worse when $k = 8$ or above. A possible explanation is that small number of classes in a mini-batch results a high bias estimation for true gradient. While the SUR-sampler performs better than CUR-sampler in such bad cases. By selecting proper k (eg. 2 or 4 in our experiments), Both CUR-sampler and SUR-sampler performs better than UR-sampler.

4.7. Analyze

To delving into essence beyond results, we perform analysis based on visualization. We count the cosine similarities of intra-class instances and inter-class instances on MSMT17 since it is a common metric for openset recognition. Figure 4 shows the heatmaps of cosine similarities. The top row shows intra-class instances and the bottom row shows inter-class instances from two different identities. Each cell relates to cosine similarity between corresponding instance pair.

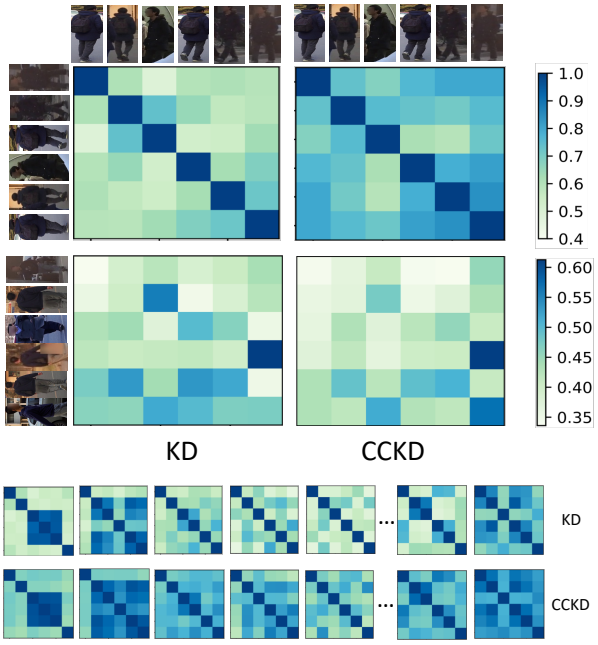


Figure 4: The heatmaps of cosine similarities between instances pairs. The top row shows intra-class similarities and the middle row shows inter-class similarities between two identities. More intra-class heatmap are showed in bottom two rows. (best viewed in color)

It can be observed that, cosine similarity between intra-class instances of CCKD is more larger than KD overall, whichs means a more cohesion of intra-class instances in embedding space. Although there is not much difference between CCKD and KD in inter-class cosine similarity. It seems that CCKD can help the student to learn a more discriminative embedding space. While CCKD by considering the correlation congruence between instances, consequently getting a better performance.

5. Conclusions

In this paper, we propose a new distillation framework named correlation congruence knowledge distillation (CCKD), which considers not only instance information but also correlation information between instances when transferring knowledge. To better capture correlation, a generalized method based on Taylor series expansion of kernel function is proposed. To further improve the CCKD, two new mini-batch sampler strategies are proposed. Extensive

experiments on four representative tasks show that the proposed approach can significantly promote the performance of student network.

A. Generalization Analysis for CCKD

This document analyzes the generalization of CCKD theoretically. We build on the Neural Tangent Kernel (NTK) framework and use stability analysis of optimization dynamics to derive a new geometric condition. Under this condition, CCKD reduces the norm in the RKHS, which lets us prove a formal generalization bound. This shows how CCKD's feature alignment improves generalization over standard KD [18], clarifying the link between feature matching, loss curvature, and model complexity.

A.1. Preliminaries and Notation

We define key concepts and establish the critical link between feature Gram matrices and NTK, laying the groundwork for the core logical chain.

A.1.1 Basic Setup

Definition 1 (Input/Label Space and Models)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^p$ (input space) and $\mathcal{Y} = \{1, \dots, K\}$ (label space). Data is sampled from an unknown distribution $\mathcal{D} \sim \mathcal{X} \times \mathcal{Y}$, with training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}^n$. We study two networks:*

- *A **fixed, fully trained teacher** $h_T : \mathcal{X} \to \mathbb{R}^K$ (with generalization error converged to a stable minimum).*
- *A **trainable student** $h : \mathcal{X} \to \mathbb{R}^K$ (to be optimized via KD/CCKD).*

A.1.2 NTK, RKHS, and Representer Theorem

Definition 2 (Neural Tangent Kernel (NTK) and RKHS)**.** *For a student network parameterized by $\boldsymbol{\Phi}$, the NTK is a kernel function that captures gradient similarity:*

$$\Theta_S(\boldsymbol{x}, \boldsymbol{x}') = tr\left(\nabla_{\boldsymbol{\Phi}} h(\boldsymbol{x})^\top \nabla_{\boldsymbol{\Phi}} h(\boldsymbol{x}')\right)$$

The NTK matrix of the student network on the training set is $\boldsymbol{\Theta}_S = [\Theta_S(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^n \in \mathbb{R}^{n \times n}$.

The kernel Θ_S uniquely defines a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{\Theta_S}$ [21, 2]. For any $h \in \mathcal{H}_{\Theta_S}$, the RKHS norm is defined as $\|h\|_{\mathcal{H}_{\Theta_S}}^2 = \langle h, h \rangle_{\mathcal{H}_{\Theta_S}}$. By the Representer Theorem, any minimizer of a regularized empirical risk in $\mathcal{H}_{\Theta_S}$ can be expressed as $h(\boldsymbol{x}) = \sum_{i=1}^n \boldsymbol{\alpha}_i \Theta_S(\boldsymbol{x}_i, \boldsymbol{x})$ where $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ are coefficient vectors, with $\|h\|_{\mathcal{H}_{\Theta_S}}^2 = \sum_{i,j=1}^n \boldsymbol{\alpha}_i^\top \boldsymbol{\alpha}_j \Theta_S(\boldsymbol{x}_i, \boldsymbol{x}_j) = tr(A^\top \boldsymbol{\Theta}_S A)$ where $A = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n]^T \in \mathbb{R}^{n \times K}$, which is a critical metric for complexity via Rademacher theory.

A.1.3 Feature Gram Matrix and NTK Decomposition

Definition 3 (Feature Gram Matrix and NTK Decomposition)**.** *For networks with a **linear final layer**, any function $h : \mathcal{X} \to \mathbb{R}^K$ can be decomposed as $h(\boldsymbol{x}) = W f(\boldsymbol{x}) + b$, where $f : \mathcal{X} \to \mathbb{R}^d$ is the **pre-final feature extractor** (the output of the network before the final linear layer), $W \in \mathbb{R}^{K \times d}$ is the final layer weight matrix, and $b \in \mathbb{R}^K$ is the bias.*

In the NTK regime with a fixed network architecture, the feature extractor f is implicitly defined by the network's structure and the parameters that realize a given function $h \in \mathcal{H}_{\Theta_S}$. While the mapping $h \mapsto f$ may not be unique in general, we work within a fixed parameterization where the feature Gram matrix is well-defined. For a function h with associated feature extractor f, the feature Gram matrix of the student network on the training set is:

$$\boldsymbol{K}_S(h) = [\langle f(\boldsymbol{x}_i), f(\boldsymbol{x}_j) \rangle]_{i,j=1}^n \in \mathbb{R}^{n \times n}$$

We use $\boldsymbol{K}_S(h)$ for the student and $\boldsymbol{K}_T = \boldsymbol{K}_S(h_T)$ for the teacher (computed on the same training set).

The function $G(h) = \frac{1}{n^2} \|\boldsymbol{K}_S(h) - \boldsymbol{K}_T\|_F^2$ is well-defined on $\mathcal{H}_{\Theta_S}$ under our framework.

*For a function $h \in \mathcal{H}_{\Theta_S}$, we define its **effective kernel matrix** of the student network on the training set as:*

$$\boldsymbol{K}_S^{eff}(h) = [\langle h(\boldsymbol{x}_i), h(\boldsymbol{x}_j) \rangle]_{i,j=1}^n \in \mathbb{R}^{n \times n}$$

where $\langle h(\boldsymbol{x}_i), h(\boldsymbol{x}_j) \rangle = \sum_{k=1}^K h_k(\boldsymbol{x}_i) h_k(\boldsymbol{x}_j)$ is the inner product of output vectors. This matrix captures the correlation structure of function outputs, directly reflecting the model's behavioral characteristics. We use $\boldsymbol{K}_T^{eff} = \boldsymbol{K}_S^{eff}(h_T)$ for the teacher.

Remark 1 (Connection to NTK)**.** *By the Representer Theorem, the effective kernel matrix can be expressed as $\boldsymbol{K}_S^{eff}(h) = \boldsymbol{\Theta}_S G \boldsymbol{\Theta}_S$ where G is a Gram matrix of coefficient vectors that depends on the training method (note that $\boldsymbol{\Theta}_S$ is symmetric). This decomposition is crucial: while $\boldsymbol{\Theta}_S$ is fixed in the NTK regime, $\boldsymbol{K}_S^{eff}(h)$ varies through G, allowing us to discuss "effective kernel alignment" without contradicting the fixed nature of the NTK.*

A.1.4 Learning Objectives

Definition 4 (KD/CCKD Objectives)**.** *Let ℓ_{CE} (cross-entropy), ℓ_{KL} (KL divergence), $\alpha \in [0, 1]$ (distillation weight), $\mu > 0$ (weight decay), and $\lambda > 0$ (feature alignment strength).*

*1. **Standard KD**: Minimizes a mix of classification and*

distillation loss:

$$\mathcal{L}_{KD}(h) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}[(1-\alpha)\ell_{CE}(h(\boldsymbol{x}_i), y_i)}_{\hat{R}_n(h)} + \alpha\ell_{KL}(h(\boldsymbol{x}_i), h_T(\boldsymbol{x}_i))] + \frac{\mu}{2}\|h\|^2_{\mathcal{H}_{\Theta_S}} \tag{10}$$

The optimal KD student is $h^{KD} = \arg\min_{h\in\mathcal{H}_{\Theta_S}} \mathcal{L}_{KD}(h)$.

2. ***CCKD****: Adds a feature alignment term to KD's objective:*

$$\mathcal{L}_{CCKD}(h) = \mathcal{L}_{KD}(h) + \lambda G(h)$$

where $G(h) = \frac{1}{n^2}\|\boldsymbol{K}_S(h) - \boldsymbol{K}_T\|_F^2$ *is the feature alignment loss. The optimal CCKD student is* $h^{CCKD} = \arg\min_{h\in\mathcal{H}_{\Theta_S}} \mathcal{L}_{CCKD}(h)$.

A.2. Assumptions

We group our assumptions by their role: establishing the foundational analysis framework, postulating the core mechanisms that drive our results, and ensuring the technical regularity required for the proofs.

A.2.1 Foundational Framework

Assumption 1 (NTK Regime and Teacher Quality)**.** *We operate within the standard Neural Tangent Kernel (NTK) framework.*

1. ***NTK Regime****: The student's training stays in the NTK regime, where its kernel* Θ_S *is fixed and its function* h *evolves within the corresponding RKHS,* $\mathcal{H}_{\Theta_S}$.

2. ***Teacher Boundedness****: The teacher* h_T *is a well-behaved function with a bounded RKHS norm,* $\|h_T\|_{\mathcal{H}_{\Theta_T}} \leq B_T$.

A.2.2 Core Mechanism Assumptions

These two assumptions form the conceptual core of our argument, each powering one of the main logical paths.

Assumption 2 (Feature-to-Behavior Proxy (Bi-Lipschitz, teacher-relative))**.** *The alignment of feature Gram matrices* relative to the teacher *serves as a valid proxy for the alignment of the model's output behavior. There exist constants* $L, c > 0$ *and a bounded neighborhood* $\mathcal{U}$ *of* h^{KD} *in* $\mathcal{H}_{\Theta_S}$ *such that for all* $h \in \mathcal{U}$*:*

$$c\,\|\boldsymbol{K}_S(h) - \boldsymbol{K}_T\|_F \leq \|\boldsymbol{K}_S^{\mathit{eff}}(h) - \boldsymbol{K}_T^{\mathit{eff}}\|_F \leq L\,\|\boldsymbol{K}_S(h) - \boldsymbol{K}_T\|_F. \tag{11}$$

Moreover, the proxy is ***locally strictly order-preserving*** *relative to the teacher: for any* $h_1, h_2 \in \mathcal{U}$*, if* $\|\boldsymbol{K}_S(h_1) - \boldsymbol{K}_T\|_F < \|\boldsymbol{K}_S(h_2) - \boldsymbol{K}_T\|_F$*, then* $\|\boldsymbol{K}_S^{\mathit{eff}}(h_1) - \boldsymbol{K}_T^{\mathit{eff}}\|_F < \|\boldsymbol{K}_S^{\mathit{eff}}(h_2) - \boldsymbol{K}_T^{\mathit{eff}}\|_F$*. This ensures that strict improvements in feature alignment translate to strict improvements in effective-kernel alignment. This assumption is justified for networks with stable linear final layers (where the weight matrix* W *has bounded condition number).*

Assumption 3 (Geometric Condition for Norm Reduction)**.** *At the KD optimum* h^{KD}*, the introduction of the feature-matching gradient term* $G(h)$ *pushes the solution "towards the origin" in the RKHS. Formally:*

$$\left\langle h^{KD}, \left(H_{\mathcal{L}_{KD}}(h^{KD})\right)^{-1} \nabla_h G(h^{KD}) \right\rangle_{\mathcal{H}_{\Theta_S}} > 0$$

As explained by the optimization intuition, this geometric alignment ensures that minimizing the CCKD objective also reduces model complexity (Lemma 3).

Remark 2 (Geometric Intuition of Assumption 3)**.** *This assumption has a clear geometric interpretation: the natural gradient direction* $(H_{\mathcal{L}_{KD}})^{-1}\nabla_h G$ *for reducing the feature alignment loss is positively correlated with the position vector* h^{KD}*. In the proof of Lemma 3, we show that this implies the derivative of the squared norm is negative, meaning the solution moves towards the origin in the RKHS, reducing its norm. This condition is expected to hold when the teacher's feature structure represents a simpler solution than standard KD produces.*

A.2.3 Technical Regularity Conditions

These are standard assumptions required to ensure the loss landscapes are well-behaved and that our mathematical tools are applicable.

Assumption 4 (Convexity, and Lipschitz Continuity)**.** 1. ***Differentiability and Convexity****: The loss components* $\hat{R}_n(h)$ *and* $G(h)$ *are convex and twice Fréchet differentiable in a neighborhood of* h^{KD}*. This ensures the existence of unique minimizers and allows for perturbation analysis via the implicit function theorem.*

2. ***Lipschitz Continuity****: The empirical risk* $\hat{R}_n(h)$ *is* L_R*-Lipschitz with respect to the model's outputs. This is a standard requirement for applying Rademacher complexity bounds.*

3. ***Feature-map Regularity****: There exists a* C^1 *mapping* $T : h \mapsto f(h)$ *defined on a neighborhood of* h^{KD} *such that* $\boldsymbol{K}_S(h) = [\langle f(h, \boldsymbol{x}_i), f(h, \boldsymbol{x}_j)\rangle]_{i,j}$ *depends* C^1*-smoothly on* h*. Consequently* $G(h)$ *is Fréchet differentiable in that neighborhood; in particular, if* $\boldsymbol{K}_S(h^{KD}) \neq \boldsymbol{K}_T$ *then* $\nabla_h G(h^{KD}) \neq 0$.

A.2.4 Lemma 1: CCKD Ensures Feature Alignment

Lemma 1 (Feature Alignment Guarantee)**.** *For any $\lambda > 0$:*

1. *$G(h^{CCKD}) \leq G(h^{KD})$ (non-strict reduction of feature alignment loss).*
2. *If $\nabla_h G(h^{KD}) \neq 0$, the inequality is **strict** for small $\lambda > 0$ (CCKD strictly improves feature alignment).*

Proof. **Part 1: Non-Strict Inequality**. By definition of minimizers:

$$\mathcal{L}_{\text{KD}}(h^{\text{KD}}) \leq \mathcal{L}_{\text{KD}}(h^{\text{CCKD}}) \tag{12}$$

$$\mathcal{L}_{\text{CCKD}}(h^{\text{CCKD}}) \leq \mathcal{L}_{\text{CCKD}}(h^{\text{KD}}) \tag{13}$$

Expanding (13) gives $\mathcal{L}_{\text{KD}}(h^{\text{CCKD}}) + \lambda G(h^{\text{CCKD}}) \leq \mathcal{L}_{\text{KD}}(h^{\text{KD}}) + \lambda G(h^{\text{KD}})$. Rearranging and using (12):

$$\lambda[G(h^{\text{CCKD}}) - G(h^{\text{KD}})] \leq \mathcal{L}_{\text{KD}}(h^{\text{KD}}) - \mathcal{L}_{\text{KD}}(h^{\text{CCKD}}) \leq 0$$

Since $\lambda > 0$, this yields $G(h^{\text{CCKD}}) \leq G(h^{\text{KD}})$.

Part 2: Strict Inequality for Small λ. By Assumptions 4 and 3, the implicit function theorem [9] guarantees that the minimizer $h(\lambda)$ of $\mathcal{L}_{\text{CCKD}}$ is continuously differentiable near $\lambda = 0$ with $h(0) = h^{\text{KD}}$. Define $\psi(\lambda) = G(h(\lambda))$.

The optimality condition $\nabla_h \mathcal{L}_{\text{KD}}(h(\lambda)) + \lambda \nabla_h G(h(\lambda)) = 0$ holds for all λ near 0. Differentiating with respect to λ and evaluating at $\lambda = 0$:

$$H_{\mathcal{L}_{\text{KD}}}(h^{\text{KD}})h'(0) + \nabla_h G(h^{\text{KD}}) = 0$$

Therefore $h'(0) = -\left(H_{\mathcal{L}_{\text{KD}}}(h^{\text{KD}})\right)^{-1} \nabla_h G(h^{\text{KD}})$. By the chain rule:

$$\begin{aligned}\psi'(0) &= \left\langle \nabla_h G(h^{\text{KD}}), h'(0) \right\rangle_{\mathcal{H}_{\Theta_S}} \\ &= -\left\langle \nabla_h G(h^{\text{KD}}), \left(H_{\mathcal{L}_{\text{KD}}}(h^{\text{KD}})\right)^{-1} \nabla_h G(h^{\text{KD}}) \right\rangle_{\mathcal{H}_{\Theta_S}}\end{aligned} \tag{14}$$

Since $H^{-1}_{\mathcal{L}_{\text{KD}}}$ is positive definite and $\nabla_h G(h^{\text{KD}}) \neq 0$, we have $\psi'(0) < 0$. By continuity, $\psi(\lambda) < \psi(0)$ for small $\lambda > 0$, proving the strict inequality. □

A.2.5 Lemma 2: CCKD Enables Effective Kernel Alignment

Lemma 2 (Effective Kernel Alignment via Feature Control)**.** *Under Assumptions 1 and 2, CCKD achieves strictly better alignment of the student's effective kernel matrix to the teacher's, compared to KD:*

$$\|\boldsymbol{K}_S^{eff}(h^{CCKD}) - \boldsymbol{K}_T^{eff}\|_F < \|\boldsymbol{K}_S^{eff}(h^{KD}) - \boldsymbol{K}_T^{eff}\|_F$$

Proof. By Lemma 1 (Part 2), for small $\lambda > 0$ we have the strict feature alignment improvement relative to the teacher:

$$\|\boldsymbol{K}_{\text{S}}(h^{\text{CCKD}}) - \boldsymbol{K}_{\text{T}}\|_F < \|\boldsymbol{K}_{\text{S}}(h^{\text{KD}}) - \boldsymbol{K}_{\text{T}}\|_F.$$

Since h^{CCKD} and h^{KD} are both in a neighborhood of h^{KD} (by continuity of $h(\lambda)$ near $\lambda = 0$), they lie in the bounded neighborhood $\mathcal{U}$ specified in Assumption 2. Applying Assumption 2's locally strictly order-preserving property (teacher-relative) immediately yields

$$\|\boldsymbol{K}_{\text{S}}^{\text{eff}}(h^{\text{CCKD}}) - \boldsymbol{K}_{\text{T}}^{\text{eff}}\|_F < \|\boldsymbol{K}_{\text{S}}^{\text{eff}}(h^{\text{KD}}) - \boldsymbol{K}_{\text{T}}^{\text{eff}}\|_F.$$

□

A.2.6 Lemma 3: CCKD Reduces RKHS Norm

Lemma 3 (RKHS Norm Reduction)**.** *Under all Assumptions (1–4), there exists $\lambda_0 > 0$ such that for all $0 < \lambda < \lambda_0$:*

$$\|h^{CCKD}\|_{\mathcal{H}_{\Theta_S}} < \|h^{KD}\|_{\mathcal{H}_{\Theta_S}}$$

Proof. Let $h(\lambda)$ be the minimizer of $\mathcal{L}_{\text{CCKD}}$. The Fréchet differentiability and convexity from Assumption 4 together with the invertibility from Assumption 3 allow us to apply the implicit function theorem [9]. This guarantees that $h(\lambda)$ is a unique, continuously differentiable function of λ near $\lambda = 0$.

We analyze the squared norm $\phi(\lambda) = \|h(\lambda)\|^2_{\mathcal{H}_{\Theta_S}}$. Its derivative is $\phi'(\lambda) = 2\langle h(\lambda), h'(\lambda)\rangle_{\mathcal{H}_{\Theta_S}}$.

From the proof of Lemma 1, we have the expression for the initial direction:

$$h'(0) = -\left(H_{\mathcal{L}_{\text{KD}}}(h^{\text{KD}})\right)^{-1} \nabla_h G(h^{\text{KD}})$$

Evaluating the derivative of the norm at $\lambda = 0$:

$$\begin{aligned}\phi'(0) &= 2\langle h(0), h'(0)\rangle_{\mathcal{H}_{\Theta_S}} \\ &= -2\left\langle h^{\text{KD}}, \left(H_{\mathcal{L}_{\text{KD}}}(h^{\text{KD}})\right)^{-1} \nabla_h G(h^{\text{KD}}) \right\rangle_{\mathcal{H}_{\Theta_S}}\end{aligned} \tag{15}$$

By Assumption 3, the inner product term is strictly positive. Therefore, $\phi'(0) < 0$. Since $\phi(\lambda)$ is continuous (by the differentiability of $h(\lambda)$), this negative derivative implies that for a small $\lambda_0 > 0$, we have $\phi(\lambda) < \phi(0)$ for all $0 < \lambda < \lambda_0$. This proves that the RKHS norm is reduced. □

A.2.7 Theorem 1: CCKD Reduces Rademacher Complexity

Theorem 3 (Rademacher Complexity Bound for CCKD)**.** *Under all Assumptions (1–4), the Rademacher complexity of the function class containing the CCKD solution is smaller than that for the KD solution:*

$$\hat{\mathfrak{R}}_n(\mathcal{F}_{\|h^{CCKD}\|}) < \hat{\mathfrak{R}}_n(\mathcal{F}_{\|h^{KD}\|})$$

where $\mathcal{F}_B = \{h \in \mathcal{H}_{\Theta_s} : \|h\|_{\mathcal{H}_{\Theta_s}} \leq B\}$ is the RKHS ball of radius B.

Proof. The empirical Rademacher complexity of an RKHS ball $\mathcal{F}_B$ is bounded by $\hat{\mathfrak{R}}_n(\mathcal{F}_B) \leq \frac{CB}{\sqrt{n}}$, where C is a constant depending on the Lipschitz constant L_R (from Assumption 4) and properties of the kernel Θ_S [4]. Since complexity is monotonic with the radius B, and Lemma 3 proves $\|h^{\text{CCKD}}\|_{\mathcal{H}_{\Theta_S}} < \|h^{\text{KD}}\|_{\mathcal{H}_{\Theta_S}}$, the result follows directly. □

A.2.8 Theorem 2: CCKD Has a Tighter Generalization Bound

Theorem 4 (Comparative Generalization Bound)**.** *Under all Assumptions (1–4), with probability at least $1-\delta$:*

$$R(h^{CCKD}) \leq \hat{R}_n(h^{CCKD}) + \frac{2C\|h^{CCKD}\|_{\mathcal{H}_{\Theta_S}}}{\sqrt{n}} + 2L_R\sqrt{\frac{\ln(2/\delta)}{2n}}$$

This upper bound on the true risk $R(h^{CCKD})$ is tighter than the corresponding bound for KD. The improvement comes from two sources:

1. *A smaller empirical risk term: $\hat{R}_n(h^{CCKD}) \leq \hat{R}_n(h^{KD}) - \lambda\Delta_G - \frac{\mu}{2}\Delta_{\|\cdot\|^2}$, where $\Delta_G = G(h^{KD}) - G(h^{CCKD}) \geq 0$ and $\Delta_{\|\cdot\|^2} = \|h^{KD}\|^2_{\mathcal{H}_{\Theta_S}} - \|h^{CCKD}\|^2_{\mathcal{H}_{\Theta_S}} \geq 0$ are the non-negative gains from feature alignment and norm reduction.*

2. *A smaller complexity term: The term $\frac{2C\|h^{CCKD}\|_{\mathcal{H}_{\Theta_S}}}{\sqrt{n}}$ is smaller for CCKD due to the norm reduction shown in Lemma 3.*

Proof. The standard Rademacher complexity bound states that for any h, with probability at least $1-\delta$:

$$R(h) \leq \hat{R}_n(h) + 2\hat{\mathfrak{R}}_n(\mathcal{F}_{\|h\|}) + 2L_R\sqrt{\frac{\ln(2/\delta)}{2n}}$$

The Lipschitz constant L_R is guaranteed by Assumption 4.

First, we establish the empirical risk relationship. By the optimality of h^{CCKD} for $\mathcal{L}_{\text{CCKD}}$:

$$\mathcal{L}_{\text{KD}}(h^{\text{CCKD}}) + \lambda G(h^{\text{CCKD}}) \leq \mathcal{L}_{\text{KD}}(h^{\text{KD}}) + \lambda G(h^{\text{KD}})$$

Expanding the definition of $\mathcal{L}_{\text{KD}}$ and rearranging gives:

$$\hat{R}_n(h^{\text{CCKD}}) \leq \hat{R}_n(h^{\text{KD}}) - \lambda(G(h^{\text{KD}}) - G(h^{\text{CCKD}})) - \frac{\mu}{2}(\|h^{\text{KD}}\|^2_{\mathcal{H}_{\Theta_S}} - \|h^{\text{CCKD}}\|^2_{\mathcal{H}_{\Theta_S}}) \tag{16}$$

Since both gain terms are non-negative (by Lemma 1 and Lemma 3), we have $\hat{R}_n(h^{\text{CCKD}}) \leq \hat{R}_n(h^{\text{KD}})$.

Second, we analyze the complexity term. Using the bound from Theorem 3, we have:

$$2\hat{\mathfrak{R}}_n(\mathcal{F}_{\|h^{\text{CCKD}}\|}) \leq \frac{2C\|h^{\text{CCKD}}\|_{\mathcal{H}_{\Theta_S}}}{\sqrt{n}} < \frac{2C\|h^{\text{KD}}\|_{\mathcal{H}_{\Theta_S}}}{\sqrt{n}}$$

When plugging these two improvements into the general bound for $R(h^{\text{CCKD}})$, it becomes evident that the resulting bound is tighter than the one for $R(h^{\text{KD}})$. □

A.2.9 Theorem 3: Existence of a Valid λ Range

Theorem 5 (Valid λ Range)**.** *For a sufficiently large sample size n, there exists a non-empty range of λ values for which the generalization advantage of CCKD is non-trivial.*

Proof. (Proof sketch) The total gain in the generalization bound is roughly $\Delta_{\text{Gain}} \approx \lambda\Delta_G + \frac{\mu}{2}\Delta_{\|\cdot\|^2} + \frac{2C}{\sqrt{n}}(\|h^{\text{KD}}\|_{\mathcal{H}_{\Theta_S}} - \|h^{\text{CCKD}}\|_{\mathcal{H}_{\Theta_S}})$. From the proofs of Lemma 1 and Lemma 3, all gain terms (Δ_G, $\Delta_{\|\cdot\|^2}$, and the norm difference) are of order $O(\lambda)$. Thus, $\Delta_{\text{Gain}} = O(\lambda)$. For this gain to be meaningful, it must be larger than the statistical noise term, which is of order $O(n^{-1/2})$. This requires $\lambda \gg O(n^{-1/2})$. Simultaneously, our perturbation analysis in Lemma 3 requires λ to be smaller than some constant λ_0. Therefore, for large enough n, a non-empty range $\lambda \in (c_1 n^{-1/2}, \lambda_0)$ exists where CCKD provides a guaranteed theoretical improvement. The existence of λ_0 is guaranteed by the implicit function theorem under Assumptions 3 and 4. □

A.3. Limitations

The analysis is constrained by its assumptions, which define its practical applicability:

1. **NTK Regime**: Results apply only to networks operating in the NTK regime, where the kernel remains approximately fixed during training. This typically requires wide networks (width $\gg$ depth). In practice, smaller student networks used in knowledge distillation may not fully satisfy this condition, though the analysis may still provide qualitative insights.

2. **Linear Final Layer**: The feature-output relationship (Assumption 2) and effective kernel alignment (Lemma 2) require a linear final layer. Many modern architectures satisfy this, but the assumption may be restrictive for networks with non-linear output layers.

3. **Bi-Lipschitz Feature-Output Relationship**: Assumption 2 assumes a bi-Lipschitz relationship between feature Gram matrices and effective kernel matrices, quantified by constants $c, L > 0$. This assumption connects feature-level alignment to output-level behavioral alignment. While theoretically justified

for networks with stable linear final layers (bounded weight matrix condition number), its practical validity depends on final layer stability. The constants c, L may vary across different network architectures and training conditions, requiring empirical validation in practice.

4. **Small λ Range**: Valid λ is restricted to $\lambda \in (\frac{c_1}{\sqrt{n}}, \lambda_0)$ (Theorem 3). For large n, the lower bound becomes very small, while the upper bound λ_0 depends on network architecture and training dynamics. This may limit hyperparameter tuning flexibility in practice, though empirical validation can guide λ selection.

5. **Assumption Dependencies**: The results rely on Assumptions 2 and 3, whose practical validity depends on network architecture and training conditions. Further empirical or theoretical validation of these assumptions would strengthen the analysis.

6. **Loose Bounds**: Rademacher complexity bounds provide qualitative insights about generalization but may not quantitatively predict real-world generalization error due to potentially large constants and asymptotic nature.

References

[1] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018. 2, 3

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950. 9

[3] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014. 2

[4] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002. 12

[5] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 535–541, New York, NY, USA, 2006. ACM. 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 6

[7] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. 2018. 5, 7

[8] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in neural information processing systems*, pages 1269–1277, 2014. 1, 2

[9] A. L. Dontchev and R. T. Rockafellar. *Implicit Functions and Solution Mappings: A View from Variational Analysis*. Springer Science & Business Media, 2009. 11

[10] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018. 3

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[12] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. pages 87–102, 2016. 7

[13] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2

[14] S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015. 1, 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6

[16] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Improving knowledge distillation with supporting adversarial samples. *arXiv preprint arXiv:1805.05532*, 2018. 3

[17] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge distillation with adversarial samples supporting decisionboundary. *arXiv preprint arXiv:1805.05532*, 2018. 3

[18] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2, 3, 5, 6, 9

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 1, 2

[20] I. Hubara, M. Courbariaux, D. Soudry, E. Y. Ran, and Y. Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18, 2016. 2

[21] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580, 2018. 9

[22] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014. 1, 2

[23] I. Kemelmachershlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 7

[24] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[25] Q. Li, S. Jin, and J. Yan. Mimicking very efficient network for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7341–7349. IEEE, 2017. 1, 3, 5, 7

[26] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015. 4

[27] P. Luo, Z. Zhu, Z. Liu, X. Wang, X. Tang, et al. Face model compression by distilling knowledge from neurons. In *AAAI*, pages 3560–3566, 2016. 5, 7

[28] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz. Pruning convolutional neural networks for resource efficient inference. 2016. 1, 2

[29] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 2, 3

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 5

[31] B. B. Sau and V. N. Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. *arXiv preprint arXiv:1610.09650*, 2016. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[34] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, R. Caruana, A. Mohamed, M. Philipose, and M. Richardson. Do deep convolutional nets really need to be deep and convolutional? *Nature*, 521, 2016. 3

[35] G. Ver Steeg and A. Galstyan. Discovering structure in high-dimensional data through correlation explanation. In *Advances in Neural Information Processing Systems*, pages 577–585, 2014. 4

[36] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy. The devil of face recognition is in the noise. *arXiv preprint arXiv:1807.11649*, 2018. 7

[37] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018. 6

[38] J. Wu, L. Cong, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *Computer Vision and Pattern Recognition*, pages 4820–4828, 2016. 2

[39] Z. Xu, Y.-C. Hsu, and J. Huang. Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. 2018. 3, 5, 7

[40] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 2, 3

[41] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. 2016. 2, 3, 5, 6

[42] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. 2017. 1, 2

[43] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018. 2, 5