Contrastive Consistent Representation Distillation

Shipeng Fu
fushipeng97@gmail.com
Haoran Yang
haoran__yang@outlook.com
Xiaomin Yang†
arielyang@scu.edu.cn

College of Electronics and Information Engineering
Sichuan University
Sichuan, China

Abstract

The combination of knowledge distillation with contrastive learning has great potential to distill structural knowledge. Most of the contrastive-learning-based distillation methods treat the entire training dataset as the memory bank and maintain two memory banks, one for the student and one for the teacher. Besides, the representations in the two memory banks are updated in a momentum manner, leading to representation inconsistency. In this work, we propose **Co**ntrastive **Co**nsistent **R**epresentation **D**istillation (**CoCoRD**) to provide consistent representations for efficient contrastive-learning-based distillation. Instead of momentum-updating the cached representations, CoCoRD updates the encoders in a momentum manner. Specifically, the teacher is equipped with a momentum-updated projection head to generate consistent representations. The teacher representations are cached in a fixed-size queue which serves as the only memory bank in CoCoRD and is significantly smaller than the entire training dataset. Additionally, a slow-moving student, implemented as a momentum-based moving average of the student, is built to facilitate contrastive learning. CoCoRD, which utilizes only one memory bank and much fewer negative keys, provides highly competitive distillation results. On ImageNet, CoCoRD-distilled ResNet50 *outperforms* the teacher ResNet101 by 0.2% top-1 accuracy. Furthermore, in PASCAL VOC and COCO detection, the detectors whose backbones are initialized by CoCoRD-distilled models exhibit considerable performance improvements. Code is available at https://github.com/ShipengFu/CoCoRD

1 Introduction

The remarkable performance of convolutional neural networks (CNNs) in various computer vision tasks, such as image recognition [10, 14] and object detection [9, 20, 21], has triggered interest in employing these powerful models beyond benchmark datasets. However, the cutting-edge performance of CNNs is always accompanied by substantial computational costs and storage consumption. Numerous endeavors have been made to reduce computational overheads and storage burdens. Among those endeavors, Knowledge Distillation, a widely discussed topic, presents a potential solution by training a compact *student* model with knowledge provided by a cumbersome but well-trained *teacher* model.

†Corresponding author

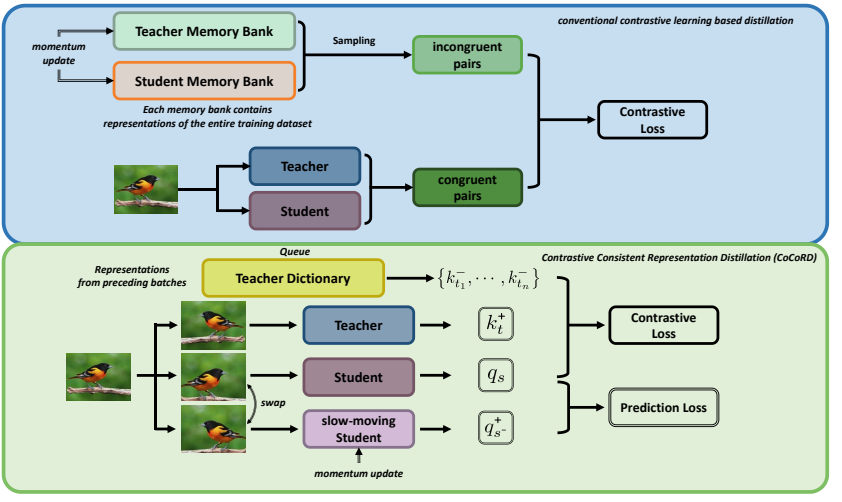


Figure 1: The general pipelines of contrastive learning based distillation methods and CoCoRD. Instead of momentum updating the representations, CoCoRD updates the encoder in a momentum manner. The teacher dictionary, which contains representations from preceding batches, is a queue.

The majority of distillation methods induce the student to imitate the teacher representations [1, 3, 4, 13, 18, 24, 25, 29, 30]. Although representations provide more learning information, the difficulty of defining appropriate metrics to align the student representations to the teacher ones challenges the distillation performance. Besides, failing to capture the dependencies between representation dimensions results in lame performance. To enhance performance, researchers attempt to distill structural knowledge by establishing connections between knowledge distillation and contrastive learning [3, 24].

To efficiently retrieve negative keys for contrastive learning, memory banks cache keys which are updated in a momentum manner, as shown in Fig. 1. The momentum-updated keys in the memory bank will be distinctly different from those not updated in that iteration, making the cached keys inconsistent. Therefore, the student can easily contrast the positive and negative keys, keeping the student from learning good features. The storage size of the memory bank is another concerned factor when applying contrastive-learning-based distillation. As in [3, 24], there are two memory banks and each of them contains representations of all training images, leading to massive GPU memory usage on large-scale datasets.

Motivated by the discussion above, we propose **Co**ntrastive **Co**nsistent **R**epresentation **D**istillation (CoCoRD) as a novel way of distilling consistent representations with one fixed-size memory bank. Specifically, CoCoRD is composed of four major components, as shown in Fig. 2: (1) a fixed-size queue which is referred to as the teacher dictionary, (2) a teacher, (3) a student, and (4) a slow-moving student. The teacher dictionary is regarded as the memory bank, where all the representations serve as the negative keys. The encoded representations of the current batch from the teacher are enqueued. Once the queue is full, the oldest ones are dequeued. By introducing a queue, the size of the memory bank is decoupled from dataset size and batch size, allowing it to be considerably smaller than the dataset size and larger than the commonly-used batch size. The student is followed by a projection head, which maps the student features to a representation space. The teacher projection head is initialized the same as the student one and is a momentum-moving average of the student projection head if the teacher and the student have the same feature dimension; otherwise, the teacher projection head is randomly initialized and not updated. Since the contrast through the teacher dictionary

is to draw distinctions on instance level, the cached teacher representations which share the same class label as the student ones leads to noise in the dictionary. To alleviate the impact of the dictionary noise, a slow-moving student, implemented as a momentum moving average of the student, is proposed to pull together anchor representations and class-positive ones. As shown in Fig. 2, with a momentum-updated projection head, the slow-moving student projects another view of the anchor image to the representation space, which serves as the class-positive key. The main contributions are listed as follows:

- CoCoRD utilizes only one lightweight memory bank, where all the representations are negative keys. We experimentally demonstrate that a miniature teacher dictionary with much fewer negative keys can be sufficient for contrastive learning in knowledge distillation.
- CoCoRD equips the well-trained teacher with a momentum-updated projection head to provide consistent representations. Besides, a slow-moving student provides class-positive representations to alleviate the impact of the potential noise in the teacher dictionary.
- CoCoRD achieves state-of-the-art distillation performance in 12 out of 13 student-teacher combinations. On ImageNet, the CoCoRD-distilled ResNet50 can outperform the teacher ResNet101 by 0.2% top-1 accuracy. Moreover, we initialize the backbones in object detection with CoCoRD-distilled weights and observe considerable performance improvements over the counterparts that the vanilla students initialize.

2 Related Work

2.1 Knowledge Distillation

The core of knowledge distillation lies in the definition of knowledge and the way the knowledge is distilled. Hinton *et al*. [13] propose distilling the softened teacher logits to the student. After the representative work [13], various distillation methods [2, 4, 19, 23, 27] aim to distill more informative knowledge via intermediate features. Among them, Passban *et al*. [19] fuse teacher information to avoid the loss of significant knowledge. Chen *et al*. [2] propose semantic calibration based on the attention for adaptively assigning cross-layer knowledge. Chen *et al*. [4] introduce a novel knowledge review framework in which the knowledge of multiple layers in the teacher can be distilled for supervising one student layer. However, the methods mentioned above are dependent on Euclidean losses. It is challenging to measure the distance appropriately with Euclidean loss in high-dimensional feature space, especially when teachers and students have different feature shapes. The proposed method leverages InfoNCE [26] to avoid regularizing the intermediate features with Euclidean losses, which can largely blur the requirement for significant architecture similarities.

2.2 Contrastive Learning

The main goal of contrastive learning is to learn a representation space where anchor representations stay close to the positive keys and distant from the negative keys. Contrastive learning is a powerful approach in self-supervised learning [5, 12, 28]. To perform effective distillation, CRD [24] combines knowledge distillation with contrastive learning to distill structural knowledge. In addition, WCoRD [3] combines distillation with contrastive learning based on Wasserstein dependency measure [17]. However, the memory banks in CRD and WCoRD contain representations of all the training images, which brings about storage

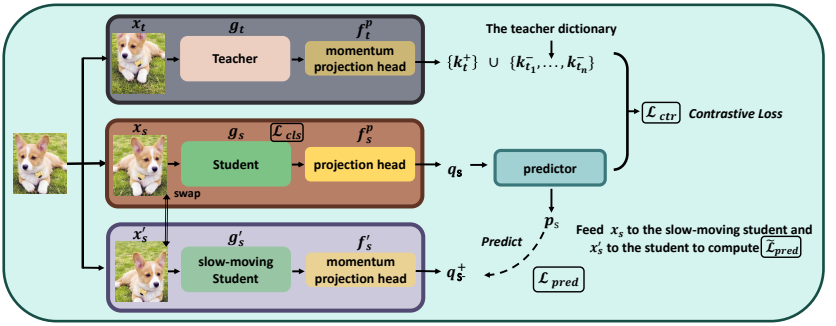


Figure 2: Illustration of the proposed CoCoRD. Note that $q_{s^-}^+$ is detached from the computational graph during the distillation process. $\widetilde{q}_{s^-}^+$, which is obtained by feeding x_s to the slow-moving student, is also detached. The teacher is frozen and the teacher dictionary does not receive gradient.

challenges on large-scale datasets. Besides, momentum updates on representations can also lead to inconsistent representations that negatively affect the distillation performance. Considering contrastive learning as a dictionary lookup task, we implement the memory bank as a fixed-size queue where all cached representations serve as negative keys.

3 Method

The key idea of combining distillation with contrastive learning is straightforward. The proficient teacher can provide consistent keys that are beneficial for contrastive learning. The student can learn powerful features which are close to the positive teacher keys and distant from the negative ones in a representation space.

3.1 Contrastive Learning as Looking up in the Teacher Dictionary

In CoCoRD, the negative keys are encoded by the teacher at previous iterations and cached in a fixed-size queue, which is referred to as the teacher dictionary. The teacher dictionary is initialized with random numbers from the standard normal distribution and we perform L_2 normalization over each initial negative key. Given an input image x, two views of x under random data augmentations form a positive pair (a query and a positive key, which are encoded at every iteration).

We define the input to the student S as the query x_s and the input to the teacher T as the positive sample x_t, as shown in Fig. 2. The outputs at the penultimate layer (before the last fully-connected layer) are projected to a representation space by a projection head. For simplicity of notation, the student nested functions up to the penultimate layer are denoted as $g_s(\cdot)$ and the student projection head is denoted as $f_s^p(\cdot)$. Therefore, the query representations q_s and the positive keys k_t^+ are given by:

$$q_s = f_s^p(g_s(x_s)), \quad k_t^+ = f_t^p(g_t(x_t)), \tag{1}$$

where $g_t(\cdot)$ denotes the teacher nested functions up the penultimate layer and $f_t^p(\cdot)$ is the teacher projection head. $f_s^p(\cdot)$ and $f_t^p(\cdot)$ are two-layer perceptrons. The cached i-th negative key in the queue is denoted as $k_{t_i}^-$ which is produced the same way as k_t^+ but from the preceding batches. The fixed-size teacher dictionary $K=\{k_{t_1}^-, \cdots, k_{t_N}^-\}$ contains N negative keys. The teacher representations of the current batch will be added to the queue after this iteration, while the oldest representations are removed from the queue if the queue is full.

The contrastive loss. The value of the contrastive loss should be small when q_s is close to k_t^+ and distant from $k_{t_i}^-$ in the representation space. To meet this condition, we consider the wildly-used and effective contrastive loss function: InfoNCE [26]:

$$\mathcal{L}_{ctr} = -\log \frac{\exp(q_s \cdot k_t^+/\tau)}{\exp(q_s \cdot k_t^+/\tau) + \sum_{i=1}^{N} \exp(q_s \cdot k_{t_i}^-/\tau)}, \tag{2}$$

where τ is the temperature hyper-parameter. N is the size of the teacher dictionary. $\mathcal{L}_{ctr}$ can be intuitively interpreted as the log loss of a softmax-based (N+1)-way classification task. In our case, we attempt to classify q_s as k_t^+ in the scope of $\{k_t^+\} \cup \{k_{t_1}^-, k_{t_2}^-, \cdots, k_{t_N}^-\}$.

The consistency in the teacher dictionary. The core to learning good features by contrastive learning lies in the challenging negative keys. In CRD [24] and WCoRD [3], the negative keys are momentum updated. The momentum update to the negative keys brings about two main issues: (1) the negative keys were updated only when they were last processed, and (2) the update interval for each negative key can be highly different. The two issues cause inconsistent negative keys that are less challenging. To provide consistent negative keys, we momentum-update the teacher projection head. Specifically, denoting the parameters of $f_t^p(\cdot)$ as w_t and those of $f_s^p(\cdot)$ as w_s, we update w_t as:

$$w_t \leftarrow m_c w_t + (1 - m_c) w_s. \tag{3}$$

$m_c \in [0,1]$ is a momentum coefficient which adjusts the update smoothness. The momentum update of w_t makes $f_t^p(\cdot)$ progress more smoothly than $f_s^p(\cdot)$. The difference between $f_t^p(\cdot)$ at different iterations can be made small. Therefore, the negative keys encoded at different iterations can be consistent. Besides, the current keys are enqueued, while the oldest keys are dequeued. This gradual replacement is beneficial for maintaining the consistency of the queue since the oldest keys are the least consistent with the current ones.

3.2 Representations of one class flock together

As shown in Eq. 2, classifying q_s as k_t^+ in the scope of $\{k_t^+, k_{t_1}^-, k_{t_2}^-, \cdots, k_{t_n}^-\}$ is categorization on instance level. However, $k_{t_i}^-$ which shares the same class label with q_s should be close to q_s in the representation space. To bring q_s closer to its class-positive keys, we introduce a slow-moving student whose nested functions up to the penultimate layer are denoted as $g_s'(\cdot)$. Specifically, the slow-moving student is implemented as a momentum-moving average of the student. The slow-moving student is also accompanied by a projection head $f_s'(\cdot)$, which is also updated in a momentum manner. Denoting the parameters of $g_s(\cdot)$ as θ_s, the parameters of $g_s'(\cdot)$ as θ_s' and those of $f_s'(\cdot)$ as w_s', we update θ_s' and w_s' by:

$$\theta_s' \leftarrow m_r \theta_s' + (1 - m_r)\theta_s \quad w_s' \leftarrow m_r w_s' + (1 - m_r) w_s, \tag{4}$$

where $m_r \in [0,1]$ is another momentum coefficient and w_s denotes the parameters of $f_s^p(\cdot)$. Therefore, the class-positive key q_{s-}^+ can be obtain by:

$$q_{s-}^+ = f_s'(g_s'(x_s')), \qquad \triangleleft \quad \text{the class-positive key} \tag{5}$$

where x_s' is another view of x under the random data augmentations. Instead of directly narrowing down the distance between q_s and q_{s-}^+, q_{s-}^+ is predicted by q_s.Formally, a predictor h_s, implemented as a two-layer perceptron, is proposed to produce the prediction $p_s \triangleq h_s(q_s)$. The loss is defined as the mean squared error between l_2 normalized p_s and q_{s-}^+, as shown

in Eq. 6. Furthermore, we symmetrize the loss by feeding x'_s to the student and x_s to the slow-moving student to compute $\widetilde{\mathcal{L}}_{pred}$. Formally, denoting the representations from x'_s by the student as $\widetilde{q}_s$ and the class-positive keys as $\widetilde{q}^+_{s-}$, we compute $\widetilde{\mathcal{L}}_{pred}$ as shown in Eq. 7.

$$\mathcal{L}_{pred} = \| \frac{p_s}{\|p_s\|_2} - \frac{q^+_{s-}}{\|q^+_{s-}\|_2} \|_2^2. \quad (6) \qquad \widetilde{\mathcal{L}}_{pred} = \| \frac{\widetilde{p}_s}{\|\widetilde{p}_s\|_2} - \frac{\widetilde{q}^+_{s-}}{\|\widetilde{q}^+_{s-}\|_2} \|_2^2. \quad (7)$$

Here $\widetilde{p}_s \triangleq h_s(\widetilde{q}_s)$, $\widetilde{q}^+_{s-} \triangleq f'_s(g'_s(x_s))$ and $\widetilde{q}_s \triangleq f^p_s(g_s(x'_s))$. Note that q^+_{s-} and $\widetilde{q}^+_{s-}$ are detached from the current computational graph during the distillation process. Since the slow-moving student does not receive any gradient, the extra memory consumption caused by introducing an additional student is negligible and does not offset the benefit of keeping one fixed-size dictionary. The memory consumption of the dictionary is provided in the supplements.

3.3 Training the student

With the slow-moving student and the teacher, Eq. 2, Eq. 6 and Eq. 7 aim at assisting the student to effectively learn powerful features through contrastive learning. The student also needs the task-specific loss. Overall, the total loss $\mathcal{L}_{total}$ can be formulated as:

$$\mathcal{L}_{total} = \lambda_{ctr}\mathcal{L}_{ctr} + \lambda_{pred}(\mathcal{L}_{pred} + \widetilde{\mathcal{L}}_{pred}) + \lambda_{cls}\mathcal{L}_{cls}, \quad (8)$$

where λ_{ctr}, λ_{pred} and λ_{cls} are three balancing factors. $\mathcal{L}_{cls} \triangleq \mathcal{H}(y, y_s)$, where $\mathcal{H}(\cdot)$ refers to the standard cross-entropy, y denotes the one-hot label and y_s is the student output.

4 Experiments

The student-teacher pairs are divided into two categories: (1) students share the architecture style with teachers, and (2) the architectures of the students are different from those of the teachers. For both categories, m_c=0.999 and m_r=0.9. More details about CoCoRD and compared methods can be found in the supplemental materials.

Datasets. To investigate the performance improvements of students, we employ two benchmarks: (1) CIFAR100 [15] and (2) ImageNet-1K [22]. CIFAR100 has 100 classes and each class has 500 training and 100 validation images. ImageNet-1K, a large-scale dataset, contains 1000 classes and provides 1.28 million training images and 50K validation images. To test the transferability of features that students learn with CoCoRD, we utilize two more datasets: (1) STL-10 [7] and (2) TinyImageNet [6]. We only use the 5K labeled training images and 8K validation images from 10 classes in STL-10.

4.1 Experiments on CIFAR100

We experiment on CIFAR100 with 13 student-teacher pairs in total[1], 7 of which are student-teacher pairs with the same architecture style, and the remaining 6 are student-teacher pairs with different architectures. As observed in Tables 1 and 2, KD [13] provides a strong baseline. CoCoRD can consistently outperform KD and achieve highly competitive performance compared with other state-of-the-art methods. Note that m_c in Formula 3 is set to 1 for the WRN-40-2/WRN-40-1 combination. This means the projection head for WRN-40-2 is not updated. Although the teacher projection head is just randomly initialized and not updated, CoCoRD still achieves the state-of-the-art result. This implies the features provided by the well-trained teacher from the penultimate layer are already distinguishing.

[1] On CIFAR100, λ_{ctr}=1, λ_{cls}=1, λ_{pred}=4. More training details are provided in the supplementary materials

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	resnet56 resnet20	resnet110 resnet20	resnet110 resnet32	resnet32x4 resnet8x4	vgg13 vgg8
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36
KD	74.92	73.54	70.66	70.67	73.08	73.33	72.98
FitNet	73.58 (↓)	72.24 (↓)	69.21 (↓)	68.99 (↓)	71.06 (↓)	73.50 (↑)	71.02 (↓)
AT	74.08 (↓)	72.77 (↓)	70.55 (↓)	70.22 (↓)	72.31 (↓)	73.44 (↑)	71.43 (↓)
SP	73.83 (↓)	72.43 (↓)	69.67 (↓)	70.04 (↓)	72.69 (↓)	72.94 (↓)	72.68 (↓)
CC	73.56 (↓)	72.21 (↓)	69.63 (↓)	69.48 (↓)	71.48 (↓)	72.97 (↓)	70.71 (↓)
VID	74.11 (↓)	73.30 (↓)	70.38 (↓)	70.16 (↓)	72.61 (↓)	73.09 (↓)	71.23 (↓)
RKD	73.35 (↓)	72.22 (↓)	69.61 (↓)	69.25 (↓)	71.82 (↓)	71.90 (↓)	71.48 (↓)
PKT	74.54 (↓)	73.45 (↓)	70.34 (↓)	70.25 (↓)	72.61 (↓)	73.64 (↑)	72.88 (↓)
AB	72.50 (↓)	72.38 (↓)	69.47 (↓)	69.53 (↓)	70.98 (↓)	73.17 (↓)	70.94 (↓)
FT	73.25 (↓)	71.59 (↓)	69.84 (↓)	70.22 (↓)	72.37 (↓)	72.86 (↓)	70.58 (↓)
CRD	**75.48** (↑)	74.14 (↑)	71.16 (↑)	71.46 (↑)	73.48 (↑)	**75.51** (↑)	73.94 (↑)
LCKT	75.22 (↑)	74.11 (↑)	71.14 (↑)	71.23 (↑)	72.32 (↑)	74.65 (↑)	73.50 (↑)
CoCoRD (ours)	**75.48** (↑)	**75.17** (↑)	**71.74** (↑)	**72.11** (↑)	**74.10** (↑)	75.29 (↑)	**73.99** (↑)
CRD+KD	75.64 (↑)	74.38 (↑)	71.63 (↑)	71.56 (↑)	73.75 (↑)	75.46 (↑)	74.29 (↑)
WCoRD	75.88 (↑)	74.73 (↑)	71.56 (↑)	71.57 (↑)	73.81 (↑)	75.95 (↑)	74.55 (↑)
CoCoRD+KD	75.90 (↑)	75.25 (↑)	72.09 (↑)	72.18 (↑)	74.37 (↑)	75.42 (↑)	74.26 (↑)

Table 1: CIFAR100 test *accuracy* (%) of students distilled with different methods when the student has the same architecture style as the teacher. ↑ denotes outperforming KD, and ↓ denotes underperforming. For all the compared methods, we use author-provided or author-verified code from the CRD repository. Results are the averages over 5 runs. The best result among the methods which are *not* combined with another one is shown in **bold**. The best result among the combined methods is underlined.

Teacher Student	vgg13 MobileNetV2	ResNet50 MobileNetV2	ResNet50 vgg8	resnet32x4 ShuffleNetV1	resnet32x4 ShuffleNetV2	WRN-40-2 ShuffleNetV1
Teacher	74.64	79.34	79.34	79.42	79.42	75.61
Student	64.60	64.60	70.36	70.50	71.82	70.50
KD	67.37	67.35	73.81	74.07	74.45	74.83
FitNet	64.14 (↓)	63.16 (↓)	70.69 (↓)	73.59 (↓)	73.54 (↓)	73.73 (↓)
AT	59.40 (↓)	58.58 (↓)	71.84 (↓)	71.73 (↓)	72.73 (↓)	73.32 (↓)
SP	66.30 (↓)	68.08 (↑)	73.34 (↓)	73.48 (↓)	74.56 (↑)	74.52 (↓)
CC	64.86 (↓)	65.43 (↓)	70.25 (↓)	71.14 (↓)	71.29 (↓)	71.38 (↓)
VID	65.56 (↓)	67.57 (↑)	70.30 (↓)	73.38 (↓)	73.40 (↓)	73.61 (↓)
RKD	64.52 (↓)	64.43 (↓)	71.50 (↓)	72.28 (↓)	73.21 (↓)	72.21 (↓)
PKT	67.13 (↓)	66.52 (↓)	73.01 (↓)	74.10 (↑)	74.69 (↑)	73.89 (↓)
AB	66.06 (↓)	67.20 (↓)	70.65 (↓)	73.55 (↓)	74.31 (↓)	73.34 (↓)
FT	61.78 (↓)	60.99 (↓)	70.29 (↓)	71.75 (↓)	72.50 (↓)	72.03 (↓)
CRD	69.73 (↑)	69.11 (↑)	74.30 (↑)	75.11 (↑)	75.65 (↑)	76.05 (↑)
LCKT	68.21 (↑)	68.81 (↑)	73.21 (↑)	74.62 (↑)	74.70 (↑)	75.08 (↑)
CoCoRD (ours)	**69.86** (↑)	**70.22** (↑)	**74.52** (↑)	**75.99** (↑)	**77.28** (↑)	**76.42** (↑)
CRD+KD	69.94 (↑)	69.54 (↑)	74.58 (↑)	75.12 (↑)	76.05 (↑)	76.27 (↑)
WCoRD	69.47 (↑)	70.45 (↑)	74.86 (↑)	75.40 (↑)	75.96 (↑)	76.32 (↑)
CoCoRD+KD	69.90 (↑)	70.30 (↑)	74.62 (↑)	76.48 (↑)	77.39 (↑)	76.56 (↑)

Table 2: CIFAR100 test *accuracy* (%) of students distilled with different methods when the teachers' architectures are *different* from those of the students. ↑ denotes outperforming KD, and ↓ denotes underperforming. Results are the averages over 5 runs. The best result among the methods which are *not* combined with another one is in **bold**. The best result among the combined methods is underlined.

Based on the discussion above, the teacher projection heads in Table 2 are randomly initialized since the difference in architecture style is very likely to bring about the difference in the input shape. Note that it is because of the projection heads that CoCoRD can achieve distillation under cross-architecture setting. The projection heads can project features at the penultimate layer of different shapes into one representation space.

	Student	KD	AT	CRD	CRD+KD	CoCoRD	Teacher
CIFAR100→STL-10	69.93	70.82	70.39	71.36	71.59	**73.63**	68.31
CIFAR100→TinyImageNet	34.53	33.83	33.80	35.88	36.07	**38.39**	32.38

Table 3: To evaluate the transferability of features, we employ linear probing to perform a 10-way classification on STL10 and 200-way classification on TinyImageNet. For this experiment, we use the combination of teacher WRN-40-2 and student WRN-16-2. Top-1 accuracy (%) is reported. The student baseline and teacher are trained from scratch. Details are in the supplemental materials.

	Teacher	Student	AT	KD	SP	CC	CRD	CRD+KD	ReviewKD	SSKD	WCoRD	CoCoRD
Top-1	26.70	30.24	29.30	29.34	29.38	30.04	28.83	28.62	28.39	28.48	28.51	**28.26**
Top-5	8.58	10.92	10.00	10.12	10.20	10.83	9.87	9.51	9.49	9.33	9.84	**9.30**

Table 4: Top-1 and Top-5 *error* rates (%) of the students ResNet-18 trained with different distillation methods on ImageNet-1K validation set. The lower, The better. The best performance is shown in **bold**.

As shown in Table 2, CoCoRD is highly effective for pairs of different architectures. Even if the teacher projection head is not updated, CoCoRD can consistently achieve the best performance compared to methods that are not combined with another method. Especially, for the resnet-32x4/ShuffleNetV2 pair, CoCoRD presents 77.28% Top-1 accuracy, which is 1.32% higher than WCoRD (75.96%). The observation suggests that CoCoRD can largely blur the requirement for significant similarities between students and teachers.

Linear probing. Following CRD [24], we employ linear probing to evaluate the transferability of the student features. We freeze the student and train a linear classifier on the global average pooling features. As shown in Table 3, CoCoRD exhibits strong transferability and outperform the second best (CRD+KD) by a large margin (2.04% improvement on STL10 and 2.32% on TinyImageNet). The CoCoRD-distilled student, which has a negligible performance drop on CIFAR100 compared with the teacher, (shown in Table 1), exhibits better transferability than the teacher (5.32% improvement on STL10 and 6.01% TinyImageNet). The linear probing indicates CoCoRD-distilled models have better generalization ability.

4.2 Experiments on ImageNet

To investigate the scalability of CoCoRD to large-scale datasets, we employ ResNet-18 and ResNet-34 as the student-teacher combination to perform experiments on ImageNet-1K. For a fair comparison, we follow the standard PyTorch ImageNet training practice except that we have 100 training epochs like CRD and WCoRD. We use the PyTorch-released ResNet-34 as our teacher. On ImageNet, we set λ_{ctr}=1, λ_{cls}=1, λ_{pred}=4 and only calculate $\mathcal{L}_{pred}$. The results in Table 4 show that the proposed CoCoRD achieves the best performance on the large-scale ImageNet. The relative improvement of CoCoRD over WCoRD [3] on Top-1 error is 14.45%, and the relative improvement over CRD [24] on Top-1 error is 40.43%. Both improvements validate the scalability of the proposed CoCoRD to large-scale datasets.

4.3 Transfer Learning

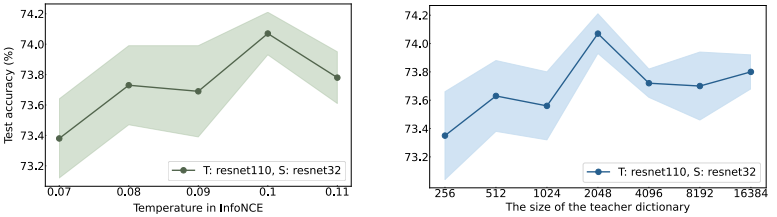
We further validate the feature transferability of CoCoRD-distilled models by transferring the model weights to object detection task, including PASCAL VOC [8] and COCO detection [16]. We fine-tune the pre-trained models in an end-to-end manner on the target datasets. Note that the CoCoRD-distilled ResNet50 outperforms the teacher ResNet101 by 0.2% top-1 accuracy on classification. As shown in Table 5, the CoCoRD-initialized detectors exhibit better performance than the student- or CRD-initialized counterparts. The valid reuse of CoCoRD-distilled weights demonstrates the transferability to object detection.

	Classification	Object Detection					
	ImageNet	PASCAL VOC Detection			CoCo Detection		
	Top-1 accuracy (%)	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}
scratch	-	60.2	33.8	33.1	44.0	26.4	27.8
Student	76.15	81.3	53.5	58.8	59.9	40.0	43.1
CRD	77.06 (+0.91)	81.7 (+0.4)	54.2 (+0.7)	60.0 (+1.2)	60.5 (+0.6)	40.7 (+0.7)	43.9 (+0.8)
CoCoRD	77.57 (+1.42)	**82.0** (+0.7)	**55.0** (+1.5)	**61.1** (+2.3)	**60.9** (+1.0)	**41.0** (+1.0)	**44.5** (+1.4)

Table 5: For PASCAL VOC, Faster R-CNN is fine-tuned on VOC `trainval07+12` and evaluated on `2007test`. For COCO, Mask R-CNN is fine-tuned on COCO `train2017` and evaluated on `val2017`. The Faster/Mask R-CNN models are with the R50-C4 backbones [11]. Numbers in green indicate the performance improvement over the detectors initialized by the vanilla student. Please see the supplementary material for details. ResNet101 is the teacher with 77.37% top-1 accuracy on ImageNet.

Option	A		B		C		D	
Encoder	Contrastive **resnet110**	Cognate **resnet32**	Contrastive resnet32	Cognate resnet32	Contrastive resnet110	Cognate -	Contrastive -	Cognate resnet32
mean (±std)	**74.10** (±0.14)		68.56 (±0.78)		72.92 (±0.23)		fails	

Table 6: CIFAR100 test *accuracy* (%) of resnet32 trained with different encoder pairs. The best performance and the pair are shown in **bold**. The teacher is resnet110. *mean* denotes the average over 5 runs and *std* stands for the standard deviation. Note that the contrastive encoder is pre-trained and the cognate encoder is initialized the same as the student and updated in a momentum manner.



(a) Effects of temperature *(b) Effects of the teacher dictionary size*

Figure 3: Effects of the temperature (τ) in InfoNCE with N=2048, as shown in (a), and the effects of the teacher dictionary size (N) with τ=0.1, as shown in (b).

4.4 Ablation Study

4.4.1 Study of encoder combinations

To investigate how the representation quality affects the distillation performance, we utilize different models to provide those representations. The model that generates dictionary-cached keys is referred to as *contrastive* encoder. The model that produces the class-positive keys is referred to as *cognate* encoder. Results are reported in Table 6. Comparing options A (the default option) and B, we can find that leveraging the pre-trained teacher to provide quality keys for contrastive learning is more beneficial for the distillation. Besides, removing the cognate encoder and setting λ_{pred} to zero (option C) lead to poor performance, suggesting the cognate encoder alleviates the adverse impact of the potential noise. If we remove the contrastive encoder and still use the dictionary with cognate encoder (option D), the distillation process fails. The results in Table 6 support the effectiveness of each encoder in CoCoRD.

4.4.2 Study of hyper-parameters

The temperature τ. τ in Eq. 2 varies from 0.07 to 0.11. As shown in Figure 3(a), CoCoRD is sensitive to τ. As suggested in CRD [24], we set τ to 0.1 on CIFAR100, while τ is set to 0.07 on ImageNet. We suggest tuning the value of τ based on the classification difficulty.

The size of the teacher dictionary. The number of negative keys is determined by the teacher dictionary size N. As shown in Figure 3(b), extremely small teacher dictionary provides insufficient negative keys, leading to sub-optimal performance. However, the extremely large teacher dictionary introduces noise, which adversely affects the performance. Based on our experiments, N=2048 should suffice on CIFAR100 while N=65536 on ImageNet. Note that the teacher dictionary in CoCoRD is significantly smaller than the memory banks in CRD [24] and WCoRD [3], which is more economic for large-scale datasets.

The balancing factors. We conduct experiments on CIFAR100 to investigate the effects of the three balancing factors λ_{ctr}, λ_{cls} and λ_{pred}. resnet32-resnet110 is used as the student-teacher combination. For experiments on balancing factors, we set τ=0.1, N=2048, m_c=0.999 and m_r=0.9. "✗" denotes the balance factor is set to 0 and "✓" means the balance factor is set to the value provided in the second row. Details on simple grid search for each balancing factor can be found in the supplementary material. As shown in Table 7, all components in CoCoRD are essential for achieving high distillation performance. When λ_{ctr} is set to 0, there is a serious performance drop, which indicates contrasting student representations with negative keys is essential in improving the student performance. Moreover, comparing the result of λ_{pred}=0 with the result of λ_{pred}=4, we can see the slow-moving student can reduce the negative effect of the potential noise in the teacher dictionary.

λ_{cls}	λ_{ctr}	λ_{pred}	*mean* (*std*)
1	1	4	**74.10** (±**0.14**)
✗	✓	✓	*fails*
✓	✗	✓	71.81 (±0.42)
✓	✓	✗	72.92 (±0.23)

Table 7: The effects of the three balancing factors. CIFAR100 test *accuracy* (%) is reported. The best performance is shown in **bold**. Average over 5 runs. More details can be found in the supplementary material.

5 Conclusion

In this paper, we propose a contrastive-learning-based knowledge distillation method named Contrastive Consistent Representation Distillation. We build only one fixed-size queue to cache consistent teacher representations. Besides, to alleviate the adverse impact of the potential noise in the queue, we employ a slow-moving student, implemented as a momentum-based moving average of the student, to provide class-positive keys. CoCoRD does not employ the entire dataset as the memory bank, which is economic for large-scale datasets. Extensive experiments demonstrate that CoCoRD, which utilizes fewer negative keys, can boost the performance of the students on diverse image classification datasets. Additionally, the models distilled by CoCoRD on ImageNet classification can efficiently improve object detection performance on PASCAL VOC and COCO datasets.

Acknowledgement

The research in our paper is sponsored by National Press and Publication Administration of China (No. UHD-ZD-202306).

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, pages 9163–9171, 2019.

[2] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *AAAI*, volume 35, pages 7028–7036, 2021.

[3] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation. In *CVPR*, pages 16296–16305, 2021.

[4] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, pages 5008–5017, 2021.

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

[6] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

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

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[9] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

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

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[13] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

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

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[17] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[19] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. *arXiv preprint arXiv:2012.14022*, 2020.

[20] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[23] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Robust knowledge transfer via hybrid forward on the teacher-student model. In *AAAI*, pages 2558–2566, 2021.

[24] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[25] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, pages 1365–1374, 2019.

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

[27] Xinglu Wang and Yingming Li. Harmonized dense knowledge distillation training for multi-exit architectures. In *AAAI*, volume 35, pages 10–218, 2021.

[28] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

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

[30] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.