A New Framework of Collaborative Learning for Adaptive Metric Distillation

Hao Liu, Mang Ye, *Senior Member, IEEE*, Yan Wang, Sanyuan Zhao, Ping Li, *Member, IEEE*, and Jianbing Shen, *Senior Member, IEEE*

***Abstract*— This article presents a new adaptive metric distillation approach that can significantly improve the student networks' backbone features, along with better classification results. Previous knowledge distillation (KD) methods usually focus on transferring the knowledge across the classifier logits or feature structure, ignoring the excessive sample relations in the feature space. We demonstrated that such a design greatly limits performance, especially for the retrieval task. The proposed collaborative adaptive metric distillation (CAMD) has three main advantages: 1) the optimization focuses on optimizing the relationship between key pairs by introducing the hard mining strategy into the distillation framework; 2) it provides an adaptive metric distillation that can explicitly optimize the student feature embeddings by applying the relation in the teacher embeddings as supervision; and 3) it employs a collaborative scheme for effective knowledge aggregation. Extensive experiments demonstrated that our approach sets a new state-of-the-art in both the classification and retrieval tasks, outperforming other cutting-edge distillers under various settings.**

***Index Terms*— Collaborative learning, deep neural networks, knowledge distillation (KD), model compression.**

I. Introduction

DURING the last few years, deep convolution neural networks (CNNs) have achieved many successes in a variety of applications such as computer vision, natural language processing, and reinforcement learning. Although large-scale deep models have achieved overwhelming successes, they often require considerable computational and memory consumption, making it a huge challenge to deploy them on mobile devices with limited resources. There are many efficient deep neural networks training techniques including designing efficient building blocks for deep models [21], [25], [31], [49], network pruning [24], quantization [42], and knowledge distillation (KD) [20], [30]. The focus of this article is KD, which has become an increasingly essential topic as it is applicable to almost all network architectures and can be easily combined with other strategies.

KD transfers the learned knowledge from a teacher model to a student model, that is, it aligns the classification prediction distributions between the two models [20]. Compared with the one-hot labels, student models have been shown to benefit from the richer informative signals contained in the predicted probability distribution. Various distillation variants [1], [22], [30], [37], [48] have been developed to explore what knowledge should be transferred. All of these methods focus on improving the classification results of the student model, but the similarity between feature embeddings is considered first when the task is some variation of image retrieval, like face verification [32] and person re-identification (Re-ID) [43]. Such open-set classification and image retrieval problems are more challenging than the classification problem because they require feature embeddings to preserve semantic similarity between samples. Therefore, the performance of a distillation method should be evaluated in both classification and retrieval metrics. However, when using the student backbone features for image retrieval, the results of previous methods are usually not as satisfactory as their classification performance, the gap between the retrieval results of the distilled student network and the teacher network is much larger than that of the classification results (see Tables I–III). In particular, many methods perform poorly in the person Re-ID [2] task, which requires high representation quality (see Table VIII). To tackle this problem, this article explores an effective distillation method, which can encourage the student network to learn a well-clustered embedding space from the teacher network and significantly enhance its backbone representations quality.

We revisit KD from the metric learning perspective. Deep metric learning aims to map the input data into an embedding space [57], which fits our objective of training a network with good representation quality. Some recent works also make attempts from this aspect. Yu et al. [45] minimize the absolute and relative distance between the teacher and the student. RKD [27] defines the distance and angle relations and distills the structure-wise knowledge into the student network. These carefully designed implicit relations between sample

Manuscript received 20 October 2021; revised 25 April 2022 and 19 August 2022; accepted 24 October 2022. Date of publication 14 February 2023; date of current version 4 June 2024. This work was supported in part by the FDCT Grant SKL-IOTSC(UM)-2021-2023; in part by the MYRG-CRG2022-00013-IOTSC Grant and the SRG2022-00023-IOTSC Grant; in part by the National Natural Science Foundation of China under Grant 61902027, Grant 62176188, and Grant 62066021; and in part by the Key Research and Development Program of Hubei Province under Grant 2021BAA187. *(Corresponding authors: Sanyuan Zhao; Jianbing Shen.)*

Hao Liu, Yan Wang, and Sanyuan Zhao are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: zhaosanyuan@bit.edu.cn).

Mang Ye is with the Hubei Luojia Laboratory and the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: mangye16@gmail.com).

Ping Li is with the Department of Computing and the School of Design, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

Jianbing Shen is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau, China (e-mail: shenjianbingcg@gmail.com).

Digital Object Identifier 10.1109/TNNLS.2022.3226569

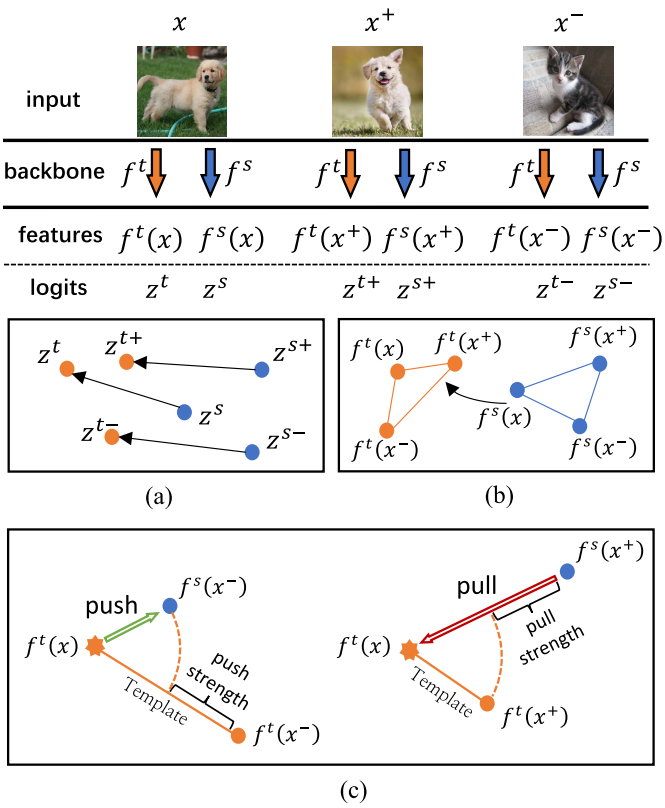


Fig. 1. Illustration of the proposed adaptive metric distillation. The backbone network of the teacher and the student is $f^t(\cdot)$ and $f^s(\cdot)$, respectively. (a) KD: classification logits. Conventional KD [20] focuses on aligning the logits. (b) RKD: feature structure. Representative methods [11], [27], [38], [45] align the feature structures in the representation space. (c) Ours: metric relation. We revisit the KD with metric learning. The relation between the teacher feature is a "template" for each triple input, formulating a metric relation distillation manner. This "template" controls the optimization strength between the student and the teacher's feature distance. In (a) and (b), x, x^-, and x^+ can be any randomly selected samples. In (c) (our method), x is the teacher anchor. x^+ is the hardest positive sample, and x^- is the hardest negative sample.

representations are defined as knowledge, but they are not a guarantee of high-quality student representations. Moreover, their relation knowledge is limited to some randomly selected samples instead of all possible sample combinations. The randomly selected sample combinations are mostly simple and cannot make too much contribution to the distillation. Even if some key combinations are lucky enough to be selected, the domination of easy combinations will overwhelm the contribution of key combinations, which makes their methods tend to be driven to local minima by easy samples [32].

In this work, we introduce an adaptive metric distillation approach, namely collaborative adaptive metric distillation (CAMD). As shown in Fig. 1, we use the student and teacher networks to map data into the same representation space, and the adaptive metric distillation part directly optimizes the explicit distance between data representations of teacher and student networks in the form of a triplet loss [40]. To take full advantage of the metric relations contained in each representation, we consider as many sample combinations as possible instead of only using the randomly selected samples for training. To overcome the computational consumption brought by the excessive sample combinations, we propose a hard mining strategy [26] in the distillation framework, which optimizes the key pairs based on the distance between the teacher and student representations. This not only takes all samples in a batch into account, but also reduces the computational complexity and prevents easy pairs from overwhelming hard pairs. Since hard triplets tend to cause over-fitting in training [32], we reweight each sample pair to highlight the less-optimized pairs by adaptively assigning the weights to different samples according to the teacher templates. In this way, the metric relations in the teacher network can also be fully utilized, and a sample pair will be emphasized if its distance deviates far from the teacher template distance. Finally, we incorporate a collaborative scheme to aggregate the knowledge in the distillation submodules. Our CAMD not only enhances the representation quality, but also improves the results on the original KD classification task. Additionally, our method can also distill knowledge from a Nasty Teacher [58], which is a more challenging network that cannot be distilled by the conventional distillation method. In summary, our work makes the following major contributions.

1) We address a new distillation problem for representation transfer, requiring good performance on both the classification and retrieval tasks, which more comprehensively evaluates the performance of a distillation method.
2) We design an adaptive metric distillation approach. It prevents excessive sample combinations through a hard mining strategy and weights the optimization strength by adaptively utilizing the teacher template supervision.
3) We introduced collaborative learning into our framework and verify that the knowledge diversity of the submodule can improve the quality of representations.
4) We conduct extensive experiments on multiple datasets and tasks. Our method achieved much higher accuracy than existing distillers and can also distill knowledge from undistillable nasty networks.

II. Related Work

A. Knowledge Distillation

KD usually refers to the process of transferring the knowledge learned in a large-scale teacher network to a small-scale student model [6], [46]. The seminal work of Buciluǎ et al. [4] compressed an ensemble of neural networks into a single network by matching output logits. To obtain useful information from the predicted probabilities, Hinton et al. [20] introduced a temperature parameter in the softmax outputs to amplify the impact of small probabilities, which are referred to as "soft targets." They achieved knowledge transfer by aligning the teacher and the student's soft output predictions in the logit space. In addition to transferring the knowledge of the last output layer, intermediate representations were introduced in FitNets [30], where auxiliary linear projection layers were used to extract supervision from the teacher's intermediate representation. Recently, several important works have been proposed to

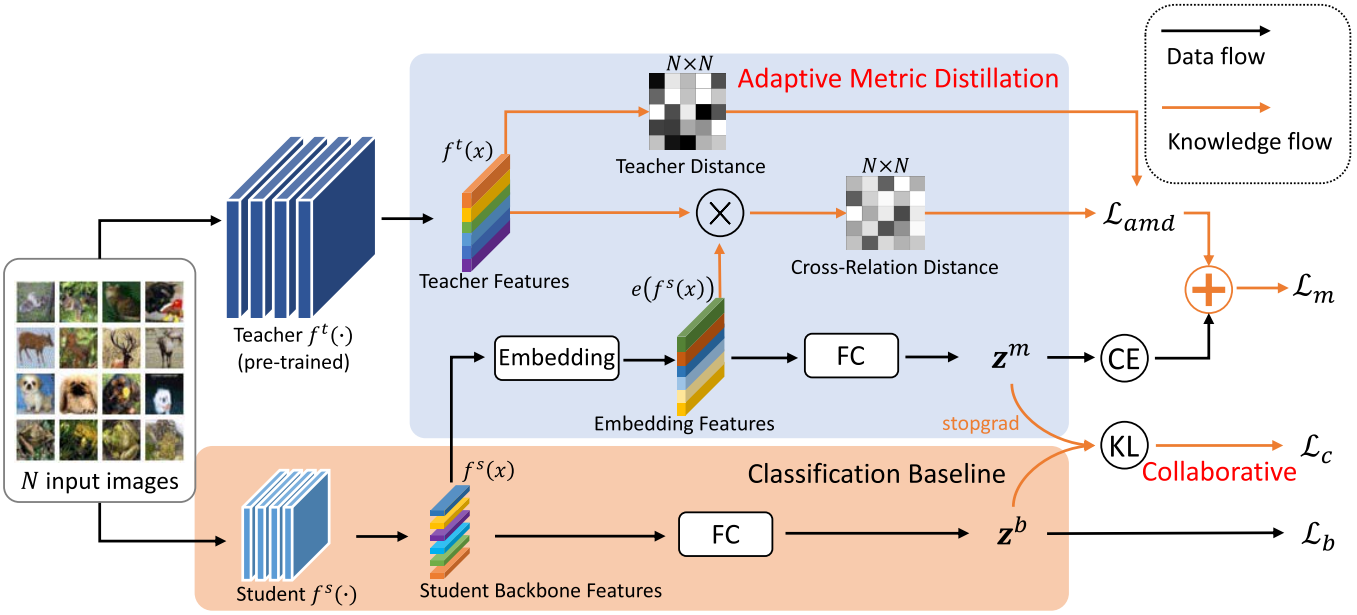


Fig. 2. Framework of the proposed CAMD with a two-branch structure: 1) *Classification baseline* follows a standard multiclass classification flow, and it is kept for testing; 2) *adaptive metric distillation* optimizes the student features in the embedding metric space by distilling the knowledge from the teacher features; and 3) *collaborative module* is integrated to transfer the knowledge into the classification baseline.

transfer attention maps [48], mutual information [1], [37], probability distributions [28], and maximum mean discrepancies [22] from the teacher to student networks. Some methods attempted to distill the relationship between samples. DarkRank [11] formalized distillation as a rank matching between the student and teacher network. SP [38] achieved knowledge transfer by aligning pairwise sample similarity matrices of the teacher and the student. These methods all implicitly distill the sample relations without fully utilizing the representation supervision provided by the teacher features.

B. Metric Learning

Deep metric learning [14] plays an important role in many computer vision applications, such as image retrieval [34], clustering [19], and transfer learning [26]. There are two fundamental approaches when learning from data with class-level or pair-wise labels. The former includes a weight matrix by transforming the embedding features of samples into the class logits vectors. The latter usually operates on the relationships between the embedding features of samples in a batch to optimize their similarity, and it is a suitable choice in many cases such as face verification [32] and person Re-ID [18]. Contrastive loss [16] based on *siamese* architecture and triplet loss [40] based on triplet networks are two fundamental approaches to deep metric learning. Contrastive loss makes the distance between positive pairs closer and the distance between negative pairs larger than some threshold. Triplet loss makes the anchor-positive distances smaller than the anchor-negative distances by a predefined margin. Hard negative mining [59] is widely used to alleviate the excessive sample pairs in deep metric learning. This strategy only focuses on the pairs with the highest loss, learns the most from them, and enhances the training efficiency. In this article, we revisit KD from the perspective of metric learning. Our method can optimize the explicit distance between data representations of the teacher and student networks so that the student network can learn a well-clustered embedding space from the teacher model with better performance.

C. Collaborative Learning

Online distillation methods simultaneously update both the teacher and student models using multiple peer networks in the training process [12], [55] and require a large amount of memory [7]. DML [52] collaboratively transferred knowledge across the peer networks. Anil et al. [3] extended this idea to train distributed networks. Their co-distillation transferred knowledge from the other models after enough burn-in steps. Recently, some works [5], [35], [51], [54] have applied a multibranch structure, sharing the shallow blocks to reduce the training cost. Song and Chai [35] promoted each branch's diversity by scaling the gradient according to the number of branches. In contrast, Chen et al. [5] assembled the knowledge from diverse auxiliary peers into a group leader in an attention-based way.

III. Proposed Approach

Our CAMD distills the knowledge from the metric learning perspective in a collaborative learning manner. We first review the original KD and then present the adaptive metric distillation, which distills and optimizes the student features with hard mining and an adaptive weighting strategy. Besides, we introduce a new collaborative learning scheme to transfer knowledge in submodules, as shown in Fig. 2. In the inference phase, only the baseline classification stream is kept for testing, and our approach does not increase any inference computation, which is more consistent with the distillation target.

A. Background

KD [20] is defined as transferring the generalizability of a large pretrained teacher network $\mathcal{T}$ to a small student network $\mathcal{S}$. Taking the classification task as an example, KD is achieved by treating the predicted class probabilities of $\mathcal{T}$ as "soft targets" to train the student network $\mathcal{S}$. For a multiclassification problem with C classes, denoting the student output logits for sample x_i with label y_i as $\boldsymbol{z}_i^b = \{z_{i,1}^b, z_{i,2}^b, \ldots, z_{i,C}^b\}$, the teacher output logits as $\boldsymbol{z}_i^t = \{z_{i,1}^t, z_{i,2}^t, \ldots, z_{i,C}^t\}$, the training objective of the student network is then represented by

$$\begin{aligned}\mathcal{L}_{\text{kd}} &= \mathcal{L}_b + \lambda \mathcal{L}_{\text{kl}} \\ &= -\sum_{i=1}^{N} \log\left(\frac{\exp z_{i,y_i}^b}{\sum_{c=1}^{C} \exp z_{i,c}^b}\right) \\ &\quad + \lambda \sum_{i=1}^{N}\sum_{c=1}^{C} \tau^2 p\left(c\middle|\boldsymbol{z}_i^m;\tau\right) \log \frac{p\left(c\middle|\boldsymbol{z}_i^m;\tau\right)}{p\left(c\middle|\boldsymbol{z}_i^b;\tau\right)}\end{aligned}$$

where N is the batch size and τ is a temperature parameter to soften the output of networks. z_{i,y_i}^b represents the prediction output of sample x_i being correctly classified as y_i. $\mathcal{L}_b$ is the standard softmax cross-entropy loss and represents the classification baseline. $\mathcal{L}_{\text{kl}}$ is the distillation loss, aligning the soft labels by student network $\mathcal{S}$ against the predictions of the teacher $\mathcal{T}$. λ balances the importance of the distillation part.

B. Adaptive Metric Distillation

In original KD [20], the teacher logits provide a richer training signal than the one-hot labels, so the student network can learn knowledge and get a performance improvement. The teacher network has learned a well-clustered embedding space, so can we distill knowledge by reinforcing students to learn the embedding space of the teacher network? When we use the teacher network and the student network to map samples into the embedding space, respectively, the positive sample pairs of the two networks should be close and the negative pairs should be far apart if the distilled student network has learned this clustering relation. Since the teacher network is well trained and does not change during the distillation, our goal is to make the student features close to the positive teacher features and far from the negative teacher features.

For a teacher model $\mathcal{T}$ and a student model $\mathcal{S}$, their backbone networks are $f^t(\cdot)$ and $f^s(\cdot)$, respectively. The annotation label for an image x_i is y_i. In general, the cumbersome teacher network has learned semantic relationships between samples and has a well-clustered embedding space. The goal is to pull the teacher feature $f^t(x_i)$ and the student feature $f^s(x_j)$ closer while pushing the teacher feature $f^t(x_i)$ and the student feature $f^s(x_k)$ apart, where $y_i = y_j$ and $y_i \neq y_k$. Note that i can be equal to j, which means that the features are from the same image in both models. $D(\cdot)$ is a function measuring the embedding distance. A larger D indicates a lower similarity between two images. We use d_{ij}^p and d_{ik}^n to represent $D(f^t(x_i), f^s(x_j))$ and $D(f^t(x_i), f^s(x_k))$, respectively. The learning target is defined as

$$d_{ij}^p < d_{ik}^n \quad \forall i, j, k. \tag{1}$$

1) Online Hard Triplet Selection: After defining the learning target, the way of selecting sample pairs needs to be clarified. If $f^s(x_j)$ and $f^s(x_k)$ are selected randomly for each $f^t(x_i)$, this triplet is likely to be simple and cannot contribute much to the distillation. However, considering all possible combinations will generate excessive triplets to be optimized. Generating and optimizing these triplets is quite time-consuming. The sample selection strategy in the distillation framework has not been explored. To solve this, we propose an online batch hard triplet selection strategy [18], [41] in the distillation framework. In every batch, we calculate the cross-relation feature distance of every image pair. Considering that teacher models are prelearned, we treat the fixed teacher embeddings as the anchors for stable training. For each $f^t(x_i)$, we select the hardest student embeddings

$$d_i^p = \max_{j=1,\ldots,N_p} d_{ij}^p, \quad d_i^n = \min_{k=1,\ldots,N_n} d_{ik}^n \tag{2}$$

where d_i^p/d_i^n represents the distance between the hardest positive/negative samples and the anchor $f^t(x_i)$. Let N_p and N_n be the average number of positive and negative student embeddings in the sampled batch for each anchor, and the number of sample pairs to be optimized in a batch is reduced from NN_pN_n to N.

2) Weighted Soft-Margin Triplet: Although hard mining can significantly reduce the computational cost and focus on optimizing key samples, it is sensitive to noisy data and will lead to over-fitting in some cases [32]. Thus, we distill the metric knowledge by a weighted soft-margin triplet loss [43]

$$\mathcal{L}_{\text{amd}} = \frac{1}{N}\sum_{i=1}^{N} \log\left[1 + \exp\left(\gamma\left(a_i^p d_i^p - a_i^n d_i^n\right)\right)\right] \tag{3}$$

where γ is a scale factor. a_i^p and a_i^n are nonnegative weighting factors. The `softplus` function $\log(1+\exp(\cdot))$ is a smooth approximation of the hinge function, which decays exponentially, resulting in more stable convergence for large-scale scenarios. The challenge of learning good features is how to assign smaller weights to pairs that tend to be outliers and cause over-fitting, and larger weights to difficult key pairs [61]. In *circle loss* [36], the weighting factors are predetermined optimal similarity values and then each pair's similarity is weighted by its deviation from these values, but these values are hyperparameters, making it sensitive to changing network structures and tasks. Moreover, it seems impossible to denote the best weights to different pairs with fixed hyperparameters. So, can we define the weights in a parameter-free manner, such that the distance between teacher and student features is at least as good as the distance in teacher embedding space in some cases such as outliers and difficult samples?

3) Teacher-Guided Adaptive Optimization: The well-trained teacher network provides good guidance to adaptively set the weighting factors in the distillation task. First, we find the hardest positive and negative samples x_j and x_k from the student feature in the batch for every $f^t(x_i)$ and then define a_i^p and a_i^n by the template distances $D(f^t(x_i), f^t(x_j))$ and $D(f^t(x_i), f^t(x_k))$

$$\begin{aligned}a_i^p &= \left[d_i^p - D\left(f^t(x_i), f^t\left(x_j\right)\right)\right]_+ \\ a_i^n &= \left[D\left(f^t(x_i), f^t(x_k)\right) - d_i^n\right]_+\end{aligned} \tag{4}$$

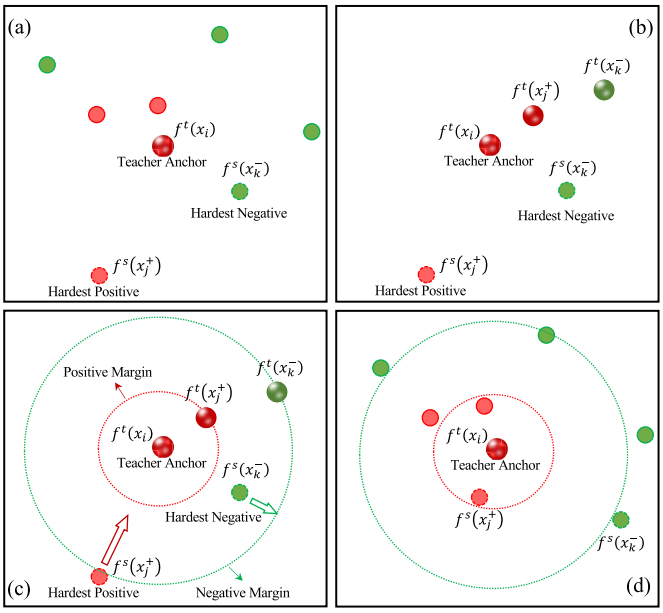


Fig. 3. Illustration of the teacher-guided adaptive optimization. (a) Procedure of online triplet selection. (b) Find the corresponding teacher embeddings. (c) Calculate weights and optimize toward their respective convergence status. (d) Optimized results. The student embeddings (both positive and negative) are optimized using the teacher as the target supervision.

where $[\cdot]_+ = \max(0, \cdot)$. Our design of (4) has some important properties: 1) each sample pair has a corresponding weighting factor. When $d_i^p > D(f^t(x_i), f^t(x_j))$ or $D(f^t(x_i), f^t(x_k)) > d_i^n$, the teacher distances serve as criteria to assign various gradient values to different d_i^p and d_i^n during optimization; 2) we utilize a "cut-off at zero" strategy to stop the optimization process when the student network outperforms the teacher network. This also stabilizes the training process and results in better discriminability; and 3) instead of constantly making positive samples closer and pushing negative samples farther away, d_i^p and d_i^n only need to be better than the teacher's template distance. This allows the distillation progress to be robust to outliers and improves its generalization ability. In addition, this operation promotes the accurate convergence status of the student network [36]. Consider the case of binary classification, in which the decision boundary is achieved at $a_i^p d_i^p - a_i^n d_i^n = 0$, with (4), the decision boundary is achieved as

$$\left(d_i^p - \frac{D\left(f^t(x_i), f^t(x_j)\right)}{2}\right)^2 + \left(d_i^n - \frac{D\left(f^t(x_i), f^t(x_k)\right)}{2}\right)^2 = R^2$$

in which $R^2 = \left(D(f^t(x_i), f^t(x_j))^2 + D(f^t(x_i), f^t(x_k))^2\right)/4$, the decision boundary is the arc of a circle. The center of the circle is at $d_i^p = D(f^t(x_i), f^t(x_j)/2, d_i^n = D(f^t(x_i), f^t(x_k)/2$, and its radius is R. Each sample has a unique decision boundary for the student network to converge. Our method can be regarded as an extension of *circle loss* [36] in the framework of KD, but compared to *circle loss*, our proposed strategy avoids manually fine-tuning the hyperparameters, and different samples have their corresponding weights. Further analysis with extensive experiments in Section IV-D demonstrates that our approach consistently achieves much better results in a more elegant parameter-free way. The illustration of adaptive metric distillation is shown in Fig. 3.

Algorithm 1 Adaptive Metric Distillation on Mini-Batch

Mini-Batch Settings: The batch size N;
Parameters: The scale factor γ;
Input: $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, the teacher and student backbone network f^t and f^s, the embedding block f^e, the learning rate β;
Output: The Updated f^s and f^e;
Step 1: Feed all images $\{\mathbf{x}_i\}_{i=1}^N$ into f^t to obtain the teacher normalized features $\{\bar{f}^t(\mathbf{x}_i)\}_{i=1}^N$, then feedforward all images to f^s and f^e in turn to get the normalized embedding features $\{\bar{f}^e(f^s(\mathbf{x}_i))\}_{i=1}^N$;
Step 2: Iterative loss calculation
  for all $f_i^t \in \{\bar{f}^t(\mathbf{x}_i)\}_{i=1}^N$ **do**
    Get cross-relation distance with $\{\bar{f}^e(f^s(\mathbf{x}_i))\}_{i=1}^N$;
    Mine the hardest positive distance d_i^p using Eq.(2);
    Mine the hardest negative distance d_i^n;
    Get the selected positive and negative indexes;
    Calculate corresponding teacher template distance;
    Calculate a_i^p and a_i^n using Eq.(4);
    Compute the metric distillation loss as Eq.(3);
  end for
  Compute $\mathcal{L}_m$ using Eq.(5);
Step 3: Gradient computation and back-propagation to update the parameters of f^s and f^e:

4) Distance Measure: Considering that the gradient's magnitude for each sample pair is inversely proportional to the embedding norm, a feature embedding with different norms will have varying gradients [50]. The variation is more drastic because d_i^p and d_i^n are distances between the teacher and student features. Moreover, embeddings with different norms also make it extremely challenging to compute the weighting factors a_i^p and a_i^n uniformly. Therefore, we adopt the normalized embedding to compute (3). We observe that the training becomes stable when applying ℓ_2-normalization to the features, and the overall accuracy is also greatly improved. To measure the distance between two normalized embedding features $\bar{f}(x_i)$ and $\bar{f}(x_j)$, we adopt the Euclidean distance, which is the most commonly used distance in deep metric learning [18], [32], [43], where $d_{ij} = D(f(x_i), f(x_j)) = \|\bar{f}(x_i) - \bar{f}(x_j)\|_2$.

To tackle the network structure difference, an embedding block $e(\cdot)$, which consists of a `FC` layer followed by `BN` and `ReLU`, is added to encode $f^s(x)$. It can not only map student backbone features into the teacher representation space, but also bring in stronger generalizability for the learned representation on the testing set because of the nonlinear projection for feature embedding, as verified in [10]. Experimentally, our method is not sensitive to embedding transformations, and using an embedding block with more layers does not further improve the results. We also incorporate a classification layer to guide embedding learning. Specifically, we represent this classifier output logits as $\boldsymbol{z}_i^m = \{z_{i,1}^m, z_{i,2}^m, \ldots, z_{i,C}^m\}$ for sample x_i. The widely used

softmax cross-entropy loss is adopted. With the adaptive metric distillation loss, the overall loss for this branch is represented as

$$\mathcal{L}_m = -\sum_{i=1}^{N} \log\left(\frac{\exp z^m_{i,y_i}}{\sum_{c=1}^{C} \exp z^m_{i,c}}\right) + \mathcal{L}_{\text{amd}}. \tag{5}$$

Our adaptive metric distillation is illustrated in Algorithm 1.

C. Collaborative Module

Although the above process can get a well-optimized backbone network, in the classification baseline, the final FC layer cannot be improved directly. Therefore, we introduce a collaborative module to aggregate the knowledge of the submodule into the baseline as

$$\begin{aligned}\mathcal{L}_c &= \text{KL}\big(z^m_i, z^b_i\big) \\ &= \sum_{i=1}^{N}\sum_{c=1}^{C} \tau^2 p\big(c|z^m_i;\tau\big) \log \frac{p\big(c|z^m_i;\tau\big)}{p\big(c|z^b_i;\tau\big)}.\end{aligned} \tag{6}$$

We align the logit distribution between the adaptive metric distillation module and the classification baseline. This is similar to the well-known KD [20], which aligns the logit distribution between the student network and the teacher network. However, applying the vanilla KD will introduce a performance decline, especially in the retrieval results. We speculate that during back-propagation, a poor classification baseline affects the well-distilled submodules. To this end, we added a stop-gradient operation. Experiments show that this operation can effectively prevent performance degradation and bring a slight improvement by revising (6) as

$$\mathcal{L}_c = \text{KL}\big(\texttt{stopgrad}\big(z^m_i\big), z^b_i\big). \tag{7}$$

The overall loss for CAMD is as

$$\mathcal{L}_{\text{CAMD}} = \mathcal{L}_b + \mathcal{L}_m + \mathcal{L}_c. \tag{8}$$

During the evaluation, only the classification baseline is kept. The extra structures introduced by adaptive metric distillation will all be removed for a fair comparison with other distillers, that is, our method does not increase any inference computation.

IV. Experimental Results

A. Image Classification and Retrieval

This section demonstrates that our CAMD consistently outperforms state-of-the-art methods on both classification and retrieval tasks, even achieving teacher-level performance.

1) Datasets and Experimental Settings: We evaluate our approach on two classification datasets, including CIFAR-100 and TinyImageNet. CIFAR-100 [23] contains 100 classes of 50k training images and 10k test images with image size 32×32. TinyImageNet [13] is a subset of the original ImageNet. It contains 64×64 images from 200 classes, each of which contains 500 training images and 50 validation images. For all the compared methods, we follow the training settings of [37] with the standard data augmentation methods including horizontal flip and random crops and adopt the stochastic gradient descent (SGD) optimizer with weight decay $5e^{-4}$ and momentum 0.9 for training with 240 epochs. The batch size N is 64 and the initial learning rate is 0.05. For MobileNet and ShuffleNet, the initial learning rate is 0.01. The learning rate drops by 0.1 after 150, 180, and 210 epochs. We set γ to 80. The temperature hyperparameter τ is set to 4. We select teacher–student combinations of VGG [33], ResNet [17], Wide-ResNet [47], MobileNet [31], and ShuffleNet [25], [49]. We evaluate the classification accuracy (Top-1) and retrieval accuracy (Ranks 1 and 2), respectively, on the test set. For retrieval, we use the normalized backbone feature $\bar{f}^s(x)$. To slightly improve the performance, we average the feature of an image and its horizontally flipped copy before retrieval.

Tables I and II list the results on CIFAR-100. Table III shows the results on TinyImageNet. The accuracies of the vanilla training of the teacher and student models are presented in the third partition after the header. We list as many results of teacher–student pairs with similar and different architectures as possible to verify the robustness. Table I (see Table II) transfers knowledge across similar (different) architectures. CAMD achieves consistent improvements in all teacher–student pairs. Compared with CRD, our method achieves an average improvement of 0.83% in the classification results and 1%–3% improvements in the image retrieval results. In particular, our Rank 1 accuracy surpasses other methods by a large margin, indicating that CAMD is an expert in transferring representation knowledge. When the original KD [20] is added to the baseline (CAMD + KD), the classification results are further improved.

For computational complexity, our time complexity comes from the operation of finding the maximum and minimum values in a batch. The CRD maintains two huge memory banks of the same size as the training set to calculate the contrast loss, which takes up a lot of storage space, and the CRD has to update the embeddings in the memory banks for each iteration, which is also time-consuming. In addition, the functions for taking min and max are optimized in PyTorch, so these operations do not significantly increase the runtime of CAMD. We test the runtime on one Tesla V100 GPU. When the student is ResNet 8×4 and the teacher is ResNet 32×4 on the CIFAR-100 dataset, the required runtime per epoch for CAMD is 54.42 s, while the time required for CRD is 95.36 s.

2) Ablation Study: We evaluate the effect of each component, as shown in Tables I–III and Fig. 4. AMD(b) means that only $\mathcal{L}_{\text{amd}}$ is integrated, that is, $\mathcal{L}_b + \mathcal{L}_{\text{amd}}$, and $\mathcal{L}_{\text{amd}}$ is calculated by the backbone features. Since there is no embedding block, the student feature dimension must equal the teacher dimension (see Table I). AMD(e) means the results without the collaborative loss $\mathcal{L}_c$, and only $\mathcal{L}_m$ is kept, that is, $\mathcal{L}_b + \mathcal{L}_m$. CAMD refers to our complete method, that is, (8).

3) Effects of $\mathcal{L}_{amd}$, $\mathcal{L}_m$, and $\mathcal{L}_c$: The results of AMD(b) show that our adaptive metric distillation has already outperformed the state-of-the-art methods and can bring a large gain compared to baseline, especially in the retrieval results, verifying the effectiveness of our "adaptive metric distillation" and showing the advantage of the explicit feature alignment using the supervision from the teacher features. The comparison between AMD(e) and AMD(b) verifies the

TABLE I

COMPARISON DISTILLATION RESULTS ON CIFAR-100 DATASET FOR DISTILLING ACROSS SIMILAR TEACHER AND STUDENT ARCHITECTURES. BOTH CLASSIFICATION ACCURACY (%) AND RETRIEVAL ACCURACY (%) ARE REPORTED. AVERAGE OVER THREE RUNS

	WRN (T:40-2, S: 16-2)			*ResNet* (T:110, S: 32)			*ResNet* (T:32 $\times$4, S: 8 $\times$ 4)			*VGG* (T:13, S:8)		
Methods	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2
Teacher	75.61	70.18	77.90	74.31	67.67	76.33	79.42	75.04	81.47	74.64	69.88	77.04
Student	73.29	63.73	73.56	71.14	61.33	72.08	72.80	60.04	70.75	70.75	61.10	70.88
KD [20]	75.30	66.78	75.96	73.34	66.03	74.99	73.44	64.83	74.63	73.41	67.26	75.39
RKD [27]	73.92	63.93	73.50	72.04	63.21	72.80	72.57	56.18	67.76	71.24	61.43	70.92
PKT [28]	74.70	67.64	76.52	72.68	64.04	73.96	74.75	65.64	74.89	73.19	67.43	76.04
VID [1]	74.36	63.70	73.99	73.03	63.63	73.42	73.03	58.91	69.71	71.20	62.80	72.38
SP [38]	74.09	65.80	75.24	73.03	63.96	73.84	73.48	63.45	73.64	72.74	66.76	74.80
CRD [37]	75.44	67.44	76.76	73.52	65.21	74.72	75.27	64.87	74.09	73.99	67.17	75.21
AMD(b)	75.86	69.67	77.94	**73.85**	66.04	75.05	75.30	67.76	76.36	74.01	68.17	75.93
AMD(e)	76.27	70.39	78.57	73.66	**66.99**	**75.92**	75.66	67.80	76.40	74.15	68.53	75.44
CAMD	**76.45**	**70.53**	**78.85**	73.30	66.79	75.48	**76.58**	**68.01**	**76.51**	**74.43**	**68.87**	**76.76**
CAMD+KD	76.52	70.36	78.90	73.62	66.50	75.44	76.60	67.42	76.69	74.46	68.52	76.23

TABLE II

COMPARISON DISTILLATION RESULTS ON THE CIFAR-100 DATASET FOR DISTILLING ACROSS DIFFERENT TEACHER AND STUDENT ARCHITECTURES. BOTH CLASSIFICATION ACCURACY (%) AND RETRIEVAL ACCURACY (%) ARE REPORTED. AVERAGE OVER THREE RUNS

	T:*ResNet* 50, S:*MobileNet V2*			T:*ResNet* 50, S:*VGG* 8			T:*ResNet* 32$\times$4, S:*Shuffle V1*			T:*ResNet* 32$\times$4, S:*Shuffle V2*		
Methods	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2
Teacher	79.34	74.60	81.75	79.34	74.60	81.75	79.42	75.04	81.47	79.42	75.04	81.47
Student	64.60	53.60	64.61	70.75	61.10	70.88	71.61	62.64	72.08	73.37	65.18	74.44
KD [20]	68.62	61.45	70.56	73.89	66.28	74.87	74.78	68.75	76.73	74.87	68.75	77.56
RKD [27]	65.67	54.69	66.78	71.51	59.31	68.99	72.60	64.22	73.44	73.63	66.01	75.31
PKT [28]	67.57	59.04	68.43	73.08	65.46	74.04	74.26	69.22	76.81	75.50	70.19	77.66
VID [1]	65.83	53.75	65.91	70.79	60.84	69.57	74.06	66.38	75.32	74.15	66.47	75.85
SP [38]	67.37	58.72	68.87	73.84	65.88	74.56	75.49	70.24	77.78	76.04	70.89	78.21
CRD [37]	69.22	60.48	69.66	74.50	66.95	75.05	75.40	70.67	77.25	75.72	69.59	78.03
AMD(e)	69.88	62.90	71.77	74.72	67.06	75.39	75.87	71.63	78.69	76.14	70.93	78.46
CAMD	**70.10**	**63.48**	**71.96**	**74.94**	**68.79**	**76.38**	**76.00**	**72.38**	**78.96**	**76.47**	**72.34**	**79.10**
CAMD+KD	70.15	63.44	71.69	74.56	67.53	75.52	76.09	71.64	78.46	76.72	71.33	78.20

TABLE III

COMPARISON RESULTS ON THE TINYIMAGENET DATASET. BOTH CLASSIFICATION ACCURACY (%) AND RETRIEVAL ACCURACY (%) ARE REPORTED. AVERAGE OVER THREE RUNS

	VGG (T:13, S:8)			T:*ResNet* 32$\times$4, S:*Shuffle V1*			T:*ResNet* 50, S:*VGG 8*			T:*WRN*-40-2, S:*Shuffle V1*		
Methods	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2	Top-1	Rank1	Rank2
Teacher	62.65	52.19	61.38	66.00	53.62	63.30	68.40	59.31	66.86	62.77	47.73	58.25
Student	56.94	38.27	48.95	62.51	45.17	55.66	56.94	38.27	48.95	62.51	45.17	55.66
KD [20]	61.72	48.44	58.08	66.39	55.63	64.57	60.86	48.52	58.39	65.08	51.88	61.75
RKD [27]	58.15	38.77	48.56	63.22	47.00	57.34	57.78	36.67	47.65	63.04	47.52	57.64
PKT [28]	58.53	42.50	52.62	63.97	48.29	58.50	58.41	41.60	52.17	63.96	48.37	58.25
VID [1]	57.86	39.29	49.47	63.93	47.03	57.55	57.26	37.77	48.45	64.51	46.48	57.03
SP [38]	59.39	43.63	54.21	65.47	51.02	61.11	59.18	43.24	53.92	65.82	52.43	62.04
CRD [37]	61.48	44.87	55.62	65.62	52.23	62.22	61.25	43.48	53.80	65.40	51.02	60.79
AMD(e)	62.83	48.78	58.50	66.46	57.30	65.79	62.38	49.54	58.89	**65.94**	54.87	64.28
CAMD	**62.97**	**50.46**	**60.17**	**66.80**	**57.55**	**66.04**	**62.74**	**50.22**	**59.90**	65.90	**55.01**	**64.48**
CAMD+KD	63.38	50.04	59.89	67.23	56.82	65.89	62.92	50.47	59.58	66.41	54.16	63.53

effectiveness of the embedding block in our KD framework. Compared to AMD(e), the improvement of our CAMD validates the feasibility of transferring knowledge of the submodule into the baseline by $\mathcal{L}_c$. Although adding the embedding block and $\mathcal{L}_c$ will cause a slight decrease in the classification result when the teacher network is *ResNet* 110 and the student network is *ResNet* 32, the retrieval results are still improved. In conclusion, all these components contribute consistently to the overall performance gain.

4) Impact of the Hyperparameter: We performed the ablation study on the scale factor γ in (3) on both the classification and retrieval tasks. In many softmax loss variants, the scale factor is an important one [36]. We vary γ from 10 to 150. The experimental results are summarized in Fig. 5. It can be seen clearly from Fig. 5 that our approach is quite robust to γ, which can be set to any value between 40 and 100 with only a slight change in result.

B. Image Classification on Large-Scale Dataset

These experiments are conducted on ImageNet [13], which provides 1.28 million images from 1000 classes for training and 50 000 for validation. We apply ResNet-34 as the teacher and ResNet-18 as the student, following the standard 100 epoch training of ImageNet on PyTorch ($\gamma = 30$). AMD(b) also means that the results when only the "adaptive metric distillation" is used directly on the backbone features of the student without the embedding block. As shown in Table IV, AMD(b) performs better than CRD, and CAMD reduces the Top-1 accuracy between the teacher and the

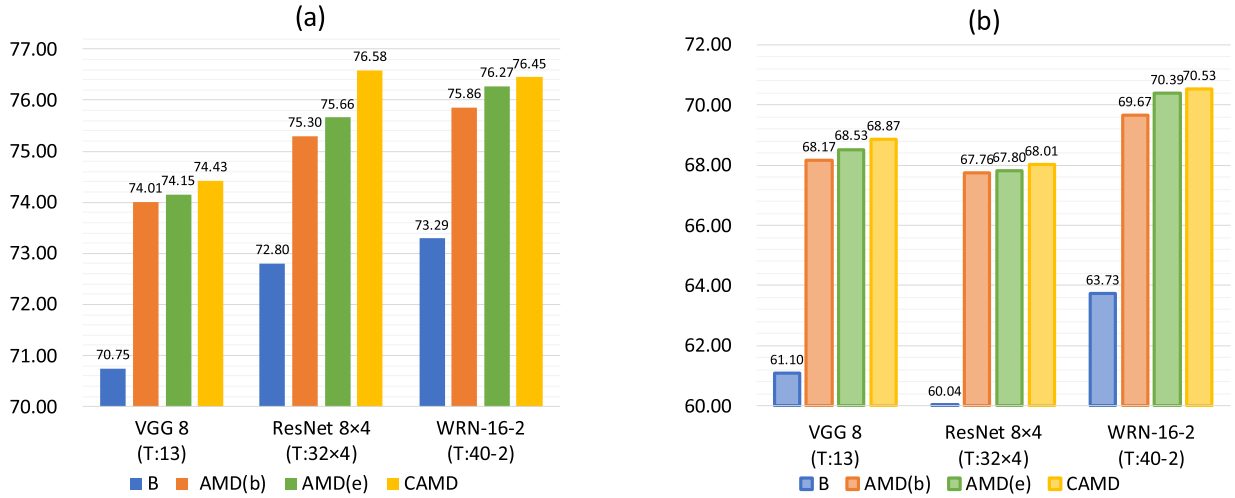


Fig. 4. Evaluation of different components on CIFAR-100. (a) Top-1 classification results. (b) Rank-1 retrieval results. AMD(b) represents the baseline performance when only $\mathcal{L}_{\text{amd}}$ is applied to the student backbone features. AMD(e) represents the performance with $\mathcal{L}_m$. And, CAMD further integrates the collaborative loss $\mathcal{L}_c$.

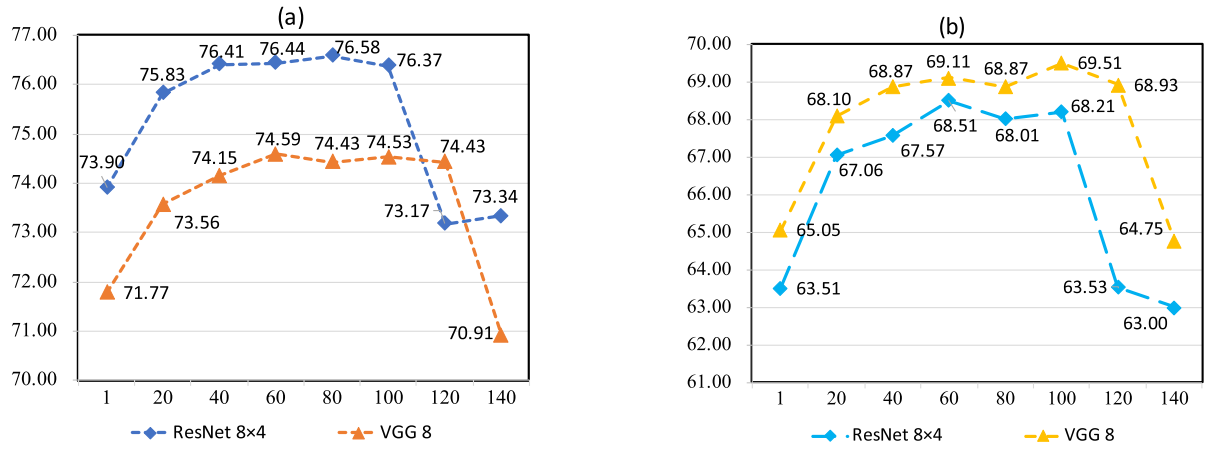


Fig. 5. Sensitivity analysis of our CAMD to the hyperparameter γ. We compared the results of two models on the CIFAR-100. (a) Top-1 classification results. (b) Rank-1 retrieval results.

TABLE IV

TOP-1 AND TOP-5 CLASSIFICATION ACCURACY (%) OF STUDENT NETWORK RESNET-18 SUPERVISED BY TEACHER NETWORK RESNET-34 ON IMAGENET VALIDATION SET

	Teacher	Student	KD [20]	AT [48]	SP [38]	CC [29]	ONE [54]	CRD [37]	WCoRD [8]	Review [9]	AMD(b)	CAMD
Top-1	73.31	69.75	70.66	70.70	70.62	69.96	70.55	71.17	71.49	71.61	71.39	**71.65**
Top-5	91.42	89.07	89.88	90.00	89.80	89.17	89.59	90.13	90.16	90.51	90.41	**90.58**

student by 1.9%. Results on ImageNet demonstrated the scalability of our approach to large-scale benchmarks.

C. Collaborative Module

1) Impact of a Larger Embedding Block: We also evaluated CAMD with a deeper embedding block (two or three layers MLP), as shown in Table V. We find that CAMD did not improve under these conditions. We conjecture that an overly large MLP will compete with the student backbone for the limited knowledge in the teacher network, resulting in the performance degradation of the student network.

2) Comparison With Online Distillation: Online distillers update the teacher and student networks simultaneously and commonly use the multibranch collaborative learning structure, which can improve student performance by increasing the knowledge diversity in different branches [15], [35], [54]. We compare CAMD with some recently proposed online methods, including DML [52], KDCL [15], and network-based OKDDip [5], under the same experimental settings. The results are shown in Table VI. For DML, we use the teacher network and the student network for mutual learning. Our classification and retrieval results outperform the best

TABLE V

PERFORMANCE OF CAMD ON CIFAR-100 WITH DEEPER MLP

	T:*WRN*-40-2 S:*WRN*-16-2		T:*ResNet* 32×4 S:*ResNet* 8×4	
Description	Top-1	Rank1	Top-1	Rank1
CAMD	76.45	70.53	76.58	68.01
CAMD *w* 2-layer MLP	75.34	68.40	76.01	66.01
CAMD *w* 3-layer MLP	75.20	67.09	74.92	65.79

TABLE VI

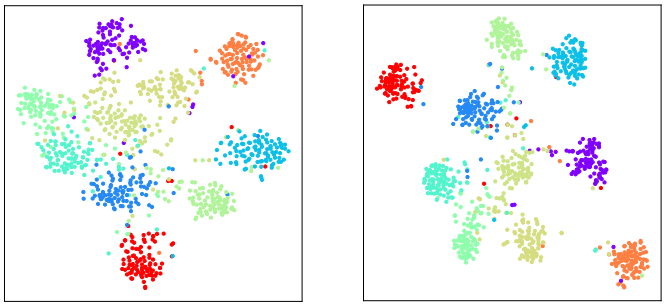
COMPARISON WITH ONLINE DISTILLERS ON THE CIFAR-100 ONLY REPORT THE RESULTS OF THE STUDENT NETWORK

		T:*ResNet* 32×4 S:*ResNet* 8×4		T:*VGG* 13 S:*VGG* 8	
Method	Type	Top-1	Rank1	Top-1	Rank1
Student	Baseline	72.80	60.04	70.75	61.10
DML [52]	Online	74.50	62.01	72.60	63.45
KDCL [15]	Online	74.53	65.46	72.70	67.59
OKDDip [5]	Online	75.91	66.41	74.13	68.69
CAMD	Offline	**76.58**	**68.01**	**74.43**	**68.87**

TABLE VII

ANALYSIS OF DIFFERENT PARTS OF ADAPTIVE METRIC DISTILLATION ON THE CIFAR-100

		T:*ResNet* 32×4 S:*ResNet* 8×4		T:*WRN*-40-2 S:*WRN*-16-2	
Methods	Setting	Top-1	Rank1	Top-1	Rank1
Teacher	Baseline	79.42	75.04	75.61	70.18
Student	Baseline	72.80	60.04	73.29	63.73
(a) Eq.(7) without vs. with stop-gradient operation					
Stop-gradient.	*w/o*	75.74	66.90	75.95	70.02
	w	76.58	68.01	76.45	70.53
(b) Eq.(4) Fixed vs. adaptive weights					
Weight.	Fixed	74.75	66.49	74.92	67.52
	Adapt	76.58	68.01	76.45	70.53
(c) Batch all vs. batch hard					
Sampling Way.	All	76.03	67.84	76.06	68.78
	Hard	76.58	68.01	76.45	70.53



(a) (b)

Fig. 6. *t*-SNE visualization of the features from ten randomly selected classes. (a) Features of vanilla ResNet 8 × 4 on the CIFAR-100 (Vanilla student). (b) Same student network optimized by CAMD (ours).

TABLE VIII

COMPARISON WITH OTHER DISTILLATION METHODS ON THE MARKET-1501 DATASETS. THE TEACHER IS RESNET-50 IN ALL CASES

	ResNet-18			*MobileNetV3*		
Methods	Rank1	Rank5	mAP	Rank1	Rank5	mAP
Teacher	89.01	95.42	71.96	89.01	95.42	71.96
Student	85.80	94.71	64.35	77.73	90.43	48.72
KD [20]	88.77	95.14	71.44	86.43	94.23	65.25
PKT [28]	87.71	95.21	69.74	83.25	93.11	60.46
RKD [27]	86.81	95.16	67.46	81.14	91.86	55.24
CRD [37]	86.63	95.10	67.53	86.49	95.04	67.78
CAMD	**89.10**	**95.40**	**71.91**	**89.31**	**96.02**	**72.25**

competing methods by 0.49% and 0.89% on average. Note that OKDDip needs much more computing resources because they have to train *three* teacher networks to capture peer diversity. Generally speaking, network-based online methods perform better than branch-based ones. Nevertheless, CAMD surpasses online network-based OKDDip by a large margin, which verifies that our collaborative module can improve the performance of the student network by increasing knowledge diversity more effectively.

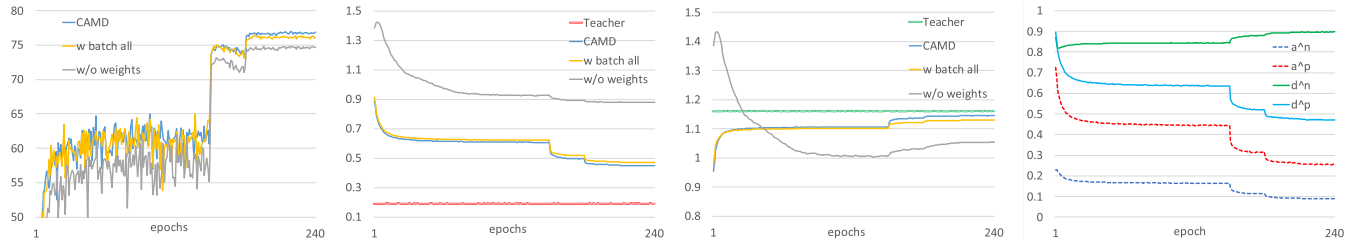
D. Analysis and Discussions

1) Without Versus With Stop-Gradient Operation: In Section III-C, we emphasized the importance of the stop-gradient operation. Here, we list results after removing it, and the results are shown in Table VII(a). Compared with AMD(e) in Table I, the classification result is reduced by 0.12% on average, and the retrieval result has dropped by 0.64% on average. The stop-gradient operation can effectively prevent the classification baseline from affecting submodule and bring a slight improvement.

2) Using Fixed Versus Adaptive Weights: As discussed in § III-B, we use a teacher-guided strategy instead of manually designed weighting factors. Here, a_i^p and a_i^n are replaced by a fixed constant 1, that is, we remove all adaptive weighting factors and regard them as manually designed hyperparameters as [36]. This also means that we treat all samples equally and directly pull positive samples closer and push the negative samples farther. As shown in Table VII(c), both the classification and retrieval performance are much lower than when CAMD employs adaptive weights. This demonstrates that merely increasing the training model's parameters and simply zooming in and out the distance between samples are not the main reasons for the effectiveness of CAMD. In fact, using teacher guidance to generate weighting factors means that teacher's supervision of student features is strengthened.

3) Batch All Versus Batch Hard: The sampling way also plays an important role. We use the *Batch Hard* strategy as (2) when forming the triplets for $\mathcal{L}_{\text{amd}}$. Another common sampling way is *Batch All* [18], which uses all possible combinations of triplets in a batch. When N is 64, the number of triples increases from 64 to thousands, and generating these triples is time-consuming. This will greatly increase the training time (about three times). We compare the results in Table VII(d). Surprisingly, *Batch Hard* consistently outperforms *Batch All*. A common conjecture is that the excessive nonhard triplets will wash out the few useful terms in hard triplets [18]. Most importantly, it also verifies that only a few key triples hidden in the thousands of all possible triples make contributions to the whole distillation process. Our hard mining strategy not only has the advantage of *Batch All*, that is, considers all possible combinations in a batch, but also saves training time.

4) Visualization: In order to intuitively show the improvement brought by our method, we randomly select ten classes in the CIFAR-100 test set and visualize the features of all the samples by *t*-SNE. Fig. 6(a) is their backbone features of the vanilla student network ResNet 8 × 4, while Fig. 6(b) shows those optimized by our CAMD under the supervision of ResNet 32 × 4. Obviously, Fig. 6(b) has a much shorter intraclass distance and a longer interclass distance. We also plot the test accuracy and average distance changes between positive and negative pairs with the training epoch. The student network is ResNet 8 × 4 distilled by ResNet 32 × 4. Fig. 7(a)–(c) presents a comparison on "CAMD versus *Batch All* sampling way" and "CAMD versus fixed weights."



(a) (b) (c) (d)

Fig. 7. More results. (a) Test accuracy changes. (b) Average distance changes between positive pairs. (c) Average distance changes between negative pairs. (d) Changes of variables in (3).

TABLE IX

RESULTS OF DISTILLATION FROM THE NASTY TEACHER ON CIFAR-100 DATASET. BOTH CLASSIFICATION ACCURACY (%) AND RETRIEVAL ACCURACY (%) ARE REPORTED

Nasty Teacher	Nasty Teacher performance		Method	Students performance					
				ShuffleV2		ResNet-18		Teacher Self	
	Top-1	Rank1		Top-1	Rank1	Top-1	Rank1	Top-1	Rank1
Baseline	-	-	-	71.75	60.75	77.62	72.71	-	-
ResNet-18	77.05	71.04	KD	65.02(-6.73)	53.82(-6.93)	74.89(-2.73)	66.50(-6.21)	74.89(-2.73)	66.50(-6.21)
	[77.62]	[72.71]	CAMD	74.18(+2.43)	68.46(+7.71)	78.60(+0.98)	74.34(+1.63)	78.60(+0.98)	74.34(+1.63)
ResNet-50	77.16	70.27	KD	62.96(-8.79)	51.43(-9.32)	72.12(-5.50)	64.51(-8.20)	74.84(-3.26)	66.38(-7.21)
	[78.10]	[73.59]	CAMD	72.82(+1.07)	66.95(+6.20)	77.75(+0.13)	73.28(+0.57)	79.01(+0.91)	74.66(+1.07)
ResNeXt-29	80.42	71.36	KD	59.68(-12.07)	49.16(-11.59)	67.92(-9.70)	59.97(-12.74)	74.58(-6.79)	61.69(-14.79)
	[81.37]	[76.48]	CAMD	70.77(-0.98)	63.02(+2.27)	76.58(-1.04)	72.08(-0.63)	80.85(-0.52)	76.01(-0.47)

The training strategy and all hyperparameters are kept unchanged. The red and green lines in Fig. 7(b) and (c) are the average distances between the teacher representations of positive and negative samples. For positive pairs, the smaller the distance, the better, and vice versa for negative pairs. The results obtained by *Batch All* are close to our results, but it requires three times as much training time as our method. Fig. 7(d) presents some key variable changes during training, the dashed lines are the changes of a^p and a^n in (3) and the solid lines are the changes of d^p and d^n in (3). All values are averaged over an epoch. Since d^p and d^n are from hard examples, d^p is larger than that in (b) and d^n is smaller than that in (c).

E. Person Re-ID

We further evaluate our method on the challenging person Re-ID datasets. For a person-of-interest query, the purpose is to find the exact position of this person appearing in other different camera views [43].

1) Datasets and Experimental Settings: We evaluated our CAMD on the Market1501 [53]. It has 12 936 training images with 751 identities, and 19 732 testing images with 750 identities. 3368 images from another 750 identities are used as a probe set, while the remaining images are used as the gallery. Here, the testing and training sets do not share common categories. All the networks are initialized with weights pretrained on ImageNet. For training, all images are resized to 256×128. We use SGD with a momentum of 0.9 for optimization. The initial learning rate is set to 0.005. We train the model over 60 epochs and decrease the learning rate by a factor of 0.1 after 40 epochs. The batch size is 32. Since person Re-ID is a fine-grained recognition task, we set γ to 1 for stable convergence. Specifically, we select ResNet-50 as the teacher network. Two different student networks are evaluated: ResNet-18 [17] and MobileNetV3 [21]. For all competing methods, we adopted their original hyperparameter settings, but made some necessary modifications. For example, the number of negative samples K for NCE in CRD, whose default value is larger than the whole Re-ID dataset. Perhaps, the insufficient negative sample is the main reason for their poor results on Re-ID tasks. The results in Table VIII demonstrated that CAMD consistently outperforms other distillation counterparts. The person Re-ID task requires better generalizability for the features. The student network's distilled accuracy is even higher than the complicated teacher network in most cases.

F. Distillation From Nasty Teachers

Recently, Ma et al. [58] proposed that nasty teachers prevent knowledge from being transferred to a student. The classification performance of a nasty teacher is similar to that of a normal teacher, but the performance of the distilled student by a nasty teacher is even worse than the baseline.

1) Datasets and Experimental Settings: To evaluate the efficiency of our CAMD, we perform the experiments on CIFAR-100 [23] with ResNet-18, ResNet-50 [17], ResNeXt29 [60], and ShuffleV2 [25] models. We use three networks including ResNet-18, ResNet-50, and ResNeXt-29 as teacher networks. And two common lightweight networks including ShuffleNetV2 and ResNet-18 are employed as

student networks. We follow the experiments and hyperparameter settings in [58] to get a nasty teacher. We initialize the learning rate as 0.1 and optimize all the networks by an SGD optimizer with momentum 0.9 and weight decay $5e^{-4}$. The networks are trained by 200 epochs with the learning rate decayed by a factor of 5 at the 60th, 120th, and 160th epochs. γ is set to 80. "Teacher Self" means the student architectures and the teacher are set to be identical. "Baseline" means the student network trained by a standard softmax cross-entropy loss following the same training settings. We also report the classification accuracy (Top-1) and retrieval accuracy (Rank 1), respectively, on the test set. Table IX shows the student performance when distilled from a nasty teacher. The numbers in the bracket represent the performance improvement of the student network compared to their "Baseline." The results in the square bracket are the normal counterparts of the nasty teachers, and they are the "Baseline" of the teacher network. The retrieval performance of the nasty teachers drops more than their classification performance compared with their normal counterparts. By the original KD [20], no student network can improve their performance by distilling from nasty teachers, and the toxic knowledge in the nasty teachers makes the student performance drop dramatically. In contrast, the CAMD students overcome the toxic knowledge and outperform their normal counterparts with better accuracy.

V. Conclusion

In this article, we distill the teacher supervision at metric levels with our newly proposed CAMD framework, and the results of the distilled student are evaluated in both classification and retrieval tasks. The adaptive metric distillation optimizes the student features by applying relation in the teacher features as supervision, in which we also introduce a hard mining strategy. Besides, a collaborative scheme is designed to enrich the diversity and fully aggregate the knowledge in the submodule. We conducted sufficient experiments on multiple classification datasets and a large-scale benchmark to prove the effectiveness of our approach. Extensive experiments have shown that our proposed approach could also work well in more challenging person Re-ID and nasty teacher distillation scenarios.

References

[1] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9163–9171.

[2] M. Ye, J. Shen, D. J. Crandall, L. Shao, and J. Luo, "Dynamic dual-attentive aggregation learning for visible-infrared person re-identification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 229–247.

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

[4] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. ACM SIGKDD*, 2006, pp. 535–541.

[5] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, Apr. 2020, pp. 3430–3437.

[6] J. Shen, Y. Liu, X. Dong, X. Lu, F. S. Khan, and S. Hoi, "Distilled Siamese networks for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8896–8909, Dec. 2022.

[7] H. Chen et al., "Data-free learning of student networks," in *Proc. IEEE Conf. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3514–3522.

[8] L. Chen, D. Wang, Z. Gan, J. Liu, R. Henao, and L. Carin, "Wasserstein contrastive representation distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16296–16305.

[9] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5008–5017.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*.

[11] Y. Chen, N. Wang, and Z. Zhang, "DarkRank: Accelerating deep metric learning via cross sample similarities transfer," 2017, *arXiv:1707.01220*.

[12] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level online adversarial knowledge distillation," 2020, *arXiv:2002.01775*.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[14] M. Ye, J. Shen, X. Zhang, P. C. Yuen, and S.-F. Chang, "Augmentation invariant and instance spreading feature for softmax embedding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 924–939, Feb. 2022.

[15] Q. Guo et al., "Online knowledge distillation via collaborative learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11020–11029.

[16] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 1735–1742.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[18] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[19] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE ICASSP*, Mar. 2016, pp. 31–35.

[20] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[21] A. Howard et al., "Searching for mobilenetv3," in *Proc. IEEE Conf. Int. Conf. Comput. Vis. (ICCV)*, Jun. 2019, pp. 1314–1324.

[22] Z. Huang and N. Wang, "Like what you like: Knowledge distill via neuron selectivity transfer," 2017, *arXiv:1707.01219*.

[23] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[24] J. H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Conf. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5058–5066.

[25] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 116–131.

[26] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4004–4012.

[27] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3967–3976.

[28] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 268–284.

[29] B. Peng et al., "Correlation congruence for knowledge distillation," in *Proc. IEEE Conf. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5007–5016.

[30] A. Romero, N. Ballas, S. Ebrahimi Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[32] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

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

[34] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1857–1865.

[35] G. Song and W. Chai, "Collaborative learning for deep neural networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1832–1841.
[36] Y. Sun et al., "Circle loss: A unified perspective of pair similarity optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6398–6407.
[37] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–19.
[38] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.
[39] A. Vaswani et al., "Attention is all you need," 2017, *arXiv:1706.03762*.
[40] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2006, pp. 1473–1480.
[41] M. Ye, C. Chen, J. Shen, and L. Shao, "Dynamic tri-level relation mining with attentive graph for visible infrared re-identification," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 386–398, 2021.
[42] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4820–4828.
[43] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," 2020, *arXiv:2001.04193*.
[44] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4133–4141.
[45] L. Yu, V. O. Yazici, X. Liu, J. van de Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2907–2916.
[46] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3903–3911.
[47] S. Zagoruyko and N. Komodakis, "Wide residual networks," 2016, *arXiv:1605.07146*.
[48] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.
[49] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6848–6856.
[50] X. Zhang, F. X. Yu, S. Karaman, W. Zhang, and S.-F. Chang, "Heated-up softmax embedding," 2018, *arXiv:1809.04157*.
[51] M. Ye, H. Li, B. Du, J. Shen, L. Shao, and S. C. H. Hoi, "Collaborative refining for person re-identification with label noise," *IEEE Trans. Image Process.*, vol. 31, pp. 379–391, 2021.
[52] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4320–4328.
[53] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Conf. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2015, pp. 1116–1124.
[54] X. Lan, X. Zhu, and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 7517–7527.
[55] M. Ye, X. Lan, Q. Leng, and J. Shen, "Cross-modality person re-identification via modality-aware collaborative ensemble learning," *IEEE Trans. Image Process.*, vol. 29, pp. 9387–9399, 2020.
[56] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 118–126.
[57] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 681–699.
[58] H. Ma, T. Chen, T.-K. Hu, C. You, X. Xie, and Z. Wang, "Undistillable: Making a nasty teacher that cannot teach students," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–12.
[59] B. Harwood, V. K. B. G., G. Carneiro, I. Reid, and T. Drummond, "Smart mining for deep metric learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2821–2829.
[60] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
[61] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6036–6046.

Hao Liu is currently pursuing the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His current research interests include collaborative learning and person re-identification.

Mang Ye (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic information from Wuhan University, Wuhan, China, in 2013 and 2016, respectively, and the Ph.D. degree in computer science from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2019.

He is currently the Research Scientist of the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His current research interests include multimedia content analysis and retrieval, computer vision, and pattern recognition.

Yan Wang is currently pursuing the master's degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China.

His current research interests include knowledge distillation and 3-D vision.

Sanyuan Zhao received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China, in 2012.

She has been working as a Lecturer with the School of Computer Science and Technology, Beijing Institute of Technology, since 2012. Her research areas include computer vision, deep learning, and virtual reality.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently the Research Assistant Professor of The Hong Kong Polytechnic University, Hong Kong. He has published more than 140 top-tier scholarly research articles and has excellent research projects reported worldwide by *ACM TechNews*. His current research interests include image/video stylization, artistic rendering and synthesis, and creative media.

Jianbing Shen (Senior Member, IEEE) was an Adjunct Professor at the School of Computer Science, Beijing Institute of Technology. He is currently a Full Professor with the State Key Laboratory of the Internet of Things for Smart City (SKLIOTSC), Department of Computer and Information Science, University of Macau, Macau, and he also with the Head of Centre for Artificial Intelligence and Robotics, University of Macau. Before that, he was acting as the Lead Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates, and the Chief Scientist with the Research Institute of Autonomous Driving at NavInfo Technology Company Ltd. He has published more than 200 top journal and conference papers, and his Google Scholar Citations are about 19,600 times with H-index 72. He was rewarded as the Highly Cited Researcher by the Web of Science in 2020, 2021, and 2022, and also the most cited Chinese Researchers by the Elsevier Scopus in 2020 and 2021. His research interests include computer vision, self driving cars, deep learning, smart city, and intelligent systems. He was an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, and other journals.