Dynamic Gradient Reactivation for Backward Compatible Person Re-identification

Xiao Pan[a], Hao Luo[a], Weihua Chen[b], Fan Wang[b], Hao Li[b], Wei Jiang[a,*], Jianming Zhang[a], Jianyang Gu[a], Peike Li[c]

[a]*Zhejiang University, Hangzhou, China*
[b]*Alibaba Group, Hangzhou, China*
[c]*University of Technology Sydney, Sydney, Australia*

Abstract

We study the backward compatible problem for person re-identification (Re-ID), which aims to constrain the features of an updated new model to be comparable with the existing features from the old model in galleries. Most of the existing works adopt distillation-based methods, which focus on pushing new features to imitate the distribution of the old ones. However, the distillation-based methods are intrinsically sub-optimal since it forces the new feature space to imitate the inferior old feature space. To address this issue, we propose the Ranking-based Backward Compatible Learning (RBCL), which directly optimizes the ranking metric between new features and old features. Different from previous methods, RBCL only pushes the new features to find best-ranking positions in the old feature space instead of strictly alignment, and is in line with the ultimate goal of backward retrieval. However, the sharp sigmoid function used to make the ranking metric differentiable also incurs the gradient vanish issue, therefore stems the ranking refinement during the later period of training. To address this issue, we propose the Dynamic Gradient Reactivation (DGR), which can reactivate the suppressed gradients by adding dynamic computed constant during forward step. To further help targeting the best-ranking positions, we include the Neighbor Context Agents (NCAs) to approximate the entire old feature space during training. Unlike previous works which only test on the in-domain settings, we make the first attempt to introduce the cross-domain settings (including both supervised and unsupervised), which are more meaningful and difficult. The experimental results on all five settings show that the proposed RBCL outperforms previous state-of-the-art methods by large margins under all settings.

Keywords: Person re-identification, Backward compatible training, Deep learning

1. Introduction

Given an query image of a person, person re-identification (Re-ID) retrieves images of the same person from a large gallery of images collected across multiple cameras. In real-world applications, such a gallery is often accumulated over time through pedestrian detection in real-time video streams, with person images represented by learned high-dimensional features. In practice, as the performance of old model becomes worse due to the enlarged gallery, model update is needed. However, it is intractable to recompute the features for all existing images given the sheer size of galleries (billions of images at least), while without recomputing the features, features from different versions of model are in different feature spaces, leading to poor retrieval performance.

To tackle this problem, the ideal solution is to perform **Backward Compatible Training (BCT)** [1] for the new model, which targets at making the new features comparable with the existing old features via adding specially designed losses during the
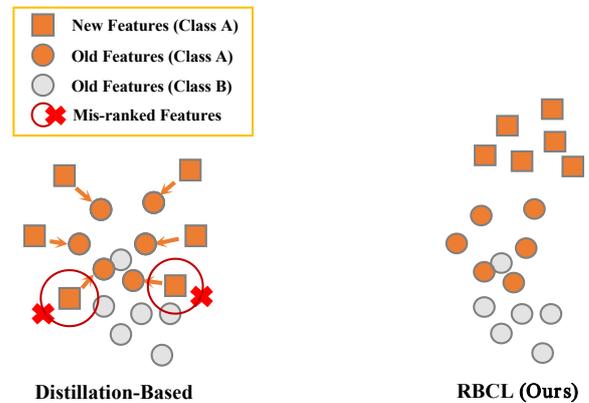

Figure 1: Illustration of the difference between the previous distillation-based methods and RBCL (ours). Squares and circles represent the new features and the old features, respectively. Different colors represent different IDs and the arrows represent the optimization direction. The distillation-based methods lead to some mis-ranked new features after BCT due to the **entangled** old feature clusters. With the help of RBCL, the new features are pushed into proper ranking positions.

training of new model. There has been some works along this direction [1, 2, 3]. For example, the Influence Loss [1] conducts the distillation between the new features and the old ones with the help of the old classifier. RBT [2] first maps new features and old ones into a common feature space and then conduct distillation. R3AN [3] transforms the new features into the old

---

feature spaces with the reconstruction of the face image.

To our best knowledge, the existing methods are mainly distillation-based. Such methods aim to imitate the distribution of the old model that is less discriminative. However, we argue that they are intrinsically sub-optimal for BCT Re-ID due to the following two disadvantages: (1) Forcing the new model to imitate the old model would be harmful to the performance of new model itself, especially for the cross-domain setting where the performance of old model on new domain is extremely poor. (2) Due to the low performance of old model, there may exist *entangled clusters* in the old feature space. The roughly alignment will lead to mis-ranked new features, which will damage the ultimate cross-model retrieval performance.

To address these issues, we propose the **Ranking-based Backward Compatible Lraining (RBCL)**, which directly optimizes the ranking metric between new features and old features. As illustrated in Fig. 1, RBCL aims to push new features into best-ranking positions in the old feature space, instead of aligned strictly with it. In this case, the new feature space will not be damaged too much, and the optimization is consistent with the optimal goal of BCT Re-ID, which is to perform retrieval between new features and old features. However, to make the discrete ranking metric differentiable, the sharp sigmoid function is needed to smooth the indication function, yet also incurs the gradient vanish issue of triplets due to its narrow gradient-effective interval, and therefore stems the refinement of ranking positions during the later period of training. To relieve this issue, we propose the **Dynamic Gradient Reactivation (DGR)**, which can reactivate the suppressed gradients via adding dynamic computed compensate constant during the forward step. With DGR, the gradients for ranking optimization are exponentially enlarged, so that the new features are refined to better-ranking positions. To further target the best-ranking positions, we propose to include the *Neighbor Context Agents (NCAs)* during optimization. NCAs can approximate the entire old feature space with several representative samples, and can provide more triplets for DGR to reactivate.

We evaluate our method under five challenging settings. In contrast to previous works which only test on in-domain settings, **we make the first attempt to study BCT under cross-domain settings**, including supervised and unsupervised for Re-ID. We emphasize that cross-domain Re-ID [4, 5, 6, 7, 8] is important for real-world applications but has not been taken into discussion in previous BCT works due to its challenging nature. The experimental results show that our method can achieve superior performance than the existing methods under both in-domain and cross-domain settings.

The main contributions of this paper can be summarized into the following aspects:

- To the best of our knowledge, we are the first work to study the backward compatible training for person Re-ID under both in-domain and cross-domain settings (including both supervised and unsupervised).

- Unlike previous distillation-based methods, we propose the RBCL to fully optimize the ranking metric between new features and old features, which leverages DGR to reactivate the suppressed gradients and includes NCAs to further target the best-ranking positions.

- We perform extensive experiments to demonstrate the effectiveness and superiority of our proposed RBCL.

2. Related Work

2.1. Optimizing Ranking for Re-ID

Multiple works in this community [9, 10, 11, 12] have proved that ranking information is important for Re-ID as a retrieval task. The typical methods for optimizing the ranking for Re-ID are Deep Metric Learning (DML) methods, which intend to optimize the distance metric. Generally, the DML methods can be further divided into pairwise [13, 14] and triplet-wise methods [10, 15, 16, 17, 11, 12]. To better exploit the information when comparing features in a mini-batch, hard example mining is first combined with Triplet loss by [10] and achieves significant improvement. To get richer pairwise information beyond a mini-batch, Wang *et al.* [12] propose the cross-batch memory (XBM) mechanism which memorizes the features of the past iterations.

Another line of works for the ranking optimization is to directly optimize the Average Precision (AP), which belongs to the ranking metric optimization. Average Precision (AP) is an important metric for person Re-ID. The main problem for directly optimizing AP is that the calculation of AP includes a discrete ranking function, which is neither differentiable nor decomposable. Brown *et al.* [18] propose the Smooth-AP which optimizes a smoothed approximation of AP and show its satisfactory performance on large-scale datasets.

The improvement from these ranking-motivated loss functions shows that exploiting the ranking is crucial for the Re-ID task. Our proposed RBCL also takes the ranking into consideration, which has been ignored in previous works for BCT. We follow the ranking metric optimization [18] since the distance metric optimization is sub-optimal when evaluating using a ranking metric.

2.2. Backward Compatible Training

Chen *et al.* [3] first formalize the cross-model face recognition task and propose the R3AN to transfer the features of the query model to the feature space of the gallery model. R3AN is specialized for the face recognition task and requires the two models to be similar, which is not applicable in some real-world model upgrade scenario. Wang *et al.* [2] propose to map the features of the two models into a unified feature space through the proposed Residual Bottleneck Transformation (RBT) blocks. Then, several distillation (imitation)-style losses are applied in the unified feature space. Both R3AN and RBT require extra transformations on features, which brings about extra computation and is intractable in real-world applications. Shen *et al.* [1] bring up the term of Backward Compatible Training (BCT) for the first time, which focus more on the model upgrade scenario
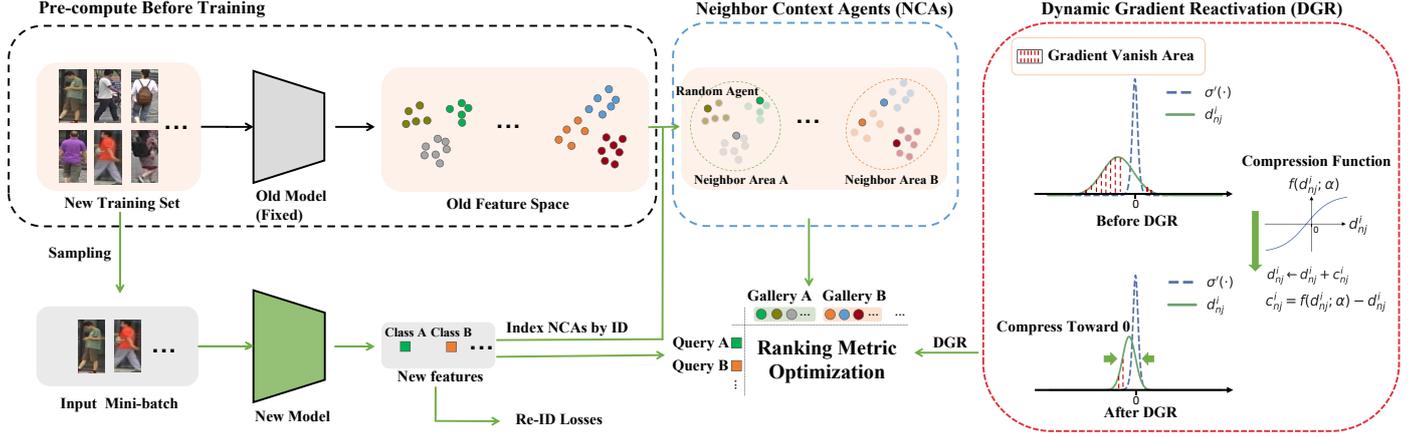
Figure 2: The training pipeline of RBCL. The fixed old features space is pre-computed before iteration. Then, during each iteration, the mini-batch is sent to the new model to get the corresponding new features. After that, the new features are taken as query features and their corresponding NCAs are taken as gallery features for ranking metric optimization. DGR is added during ranking metric optimization to refine the ranking positions. During inference stage, the query features extracted by the new model are retrieved directly with the gallery features from the old model.

compared with the above mentioned two methods. They propose the Influence Loss, which performs distillation between the logits of new features and old ones through the classifier of the old model, and the classification loss of the new model is combined with the Influence Loss when training the new model. This form of combination is similar to Learning Without Forgetting (LWF) [19] which is the typical solution of incremental learning, but differs in that the classifier of the old model is fixed here. Meng *et al.* [20] propose LCE, which aligns the centers between new features and old features, and then restrict more compact intra-class distributions for new features. Although LCE won't distill the new feature distribution to the old one strictly in instance-level, it still aligns these two distributions, which is ineffective for handling the entangled old clusters. Different from the previous methods, our proposed RBCL directly optimizes the ranking metric between new features and old features.

3. Methodology

In this section, we first introduce the backward compatible training in formula and then introduce our proposed Ranking-based Backward Compatible Learning (RBCL).

3.1. Problem Formalization

Assuming that a new model trained without any consideration of BCT is denoted as ϕ_{new}, and the fixed old model is denoted as ϕ_{old}. To improve the backward compatibility, we design a method L and integrate it into the training process of ϕ_{new}. We then name the new model trained with the help of method L as ϕ_{new-L}.

For a test set $\mathcal{D}_{test}$, its evaluation metric (e.g., mAP or Rank-1) is represented as $M(\phi_q, \phi_g; \mathcal{D}_{test})$, where the query features are extracted by ϕ_q, and gallery features are extracted by ϕ_g. Since ϕ_{new} and ϕ_{old} are in different feature spaces, directly evaluating between ϕ_{new} and ϕ_{old}, i.e., $M(\phi_{new}, \phi_{old}; \mathcal{D}_{test})$, gives

unsatisfactory performance. Therefore, our goal is to design a method L, which can make $M(\phi_{new-L}, \phi_{old}; \mathcal{D}_{test}) > M(\phi_{old}, \phi_{old}; \mathcal{D}_{test})$ as much as possible.

3.2. Ranking-based Backward Compatible Learning

The pipeline of our RBCL is illustrated in Fig. 2. Before iteration, the training set of the new model is first sent to the old model to get the fixed old feature space. Then, during each iteration, a mini-batch is sampled from the training set, and then sent to the new model to get the corresponding new features. After that, the new features are taken as the query features and their corresponding NCAs (indexed by IDs from the pre-computed old feature space) are taken as the gallery features. Finally, the metric optimization is conducted between query features and gallery features with the help of DGR, and the new features are sent to Re-ID losses for discriminative learning. During inference, the the query features are extracted by the new model, and then directly retrieved (cosine similarity) with the old gallery features.

3.2.1. Ranking Metric Optimization

Inspired by [18], we use the smoothed mean Average Precision (mAP) metric to optimize the ranking between new features and old features during training. Given a query feature f_i from query feature set F_q, which is extracted by the new model, the gallery feature set F_g extracted by the old model is split into positive set $\mathbb{P}_i$ and negative set $\mathbb{N}_i$, which are formed by all instances of the same class and of different classes, respectively. Then, the smoothed AP for query i becomes:

$$AP_i = \frac{1}{|\mathbb{P}_i|} \sum_{j \in \mathbb{P}_i} \frac{1 + \sum_{p \in \mathbb{P}_i \setminus \{j\}} \sigma(d_{pj}^i)}{1 + \sum_{p \in \mathbb{P}_i \setminus \{j\}} \sigma(d_{pj}^i) + \sum_{n \in \mathbb{N}_i} \sigma(d_{nj}^i)}, \quad (1)$$

where $d_{pj}^i = s_{ip} - s_{ij}$ and $d_{nj}^i = s_{in} - s_{ij}$, in which s_{ij} represents for the cosine similarity between f_i and f_j; and $\sigma(\cdot)$ represents for the sigmoid function:

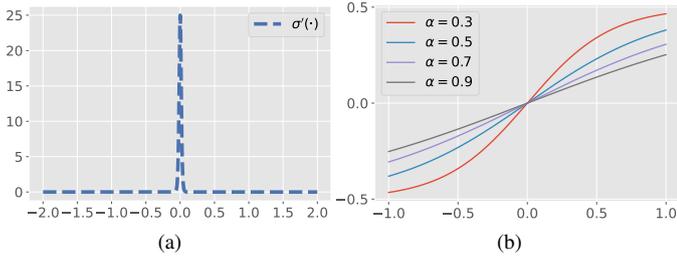$$\sigma = \frac{1}{1 + e^{-x/\tau}}. \quad (2)$$

3

Figure 3: (a) The derivative of $\sigma(\cdot)$ when $\tau = 0.01$. (b) The influence of α to the compression function (Eq. 5).

which is the relaxed form of indication function to make AP differentiable [18]. Then, the final smoothed mean Average Precision (mAP) loss is:

$$\mathcal{L}_m = 1 - \frac{1}{|F_q|} \sum_{i \in |F_q|} AP_i. \tag{3}$$

3.2.2. Dynamic Gradient Reactivation

The main goal of optimizing $\mathcal{L}_m$ is to make AP_i close to 1, thus, although not completely equivalent, the key is to minimize the $\sum_{n \in \mathbb{N}_i} \sigma(d_{nj}^i)$ term in the denominator of Eq. 1. $d_{nj}^i = s_{in} - s_{ij}$ is a triplet composed of an anchor i, a positive sample j, and a negative sample n. Minimizing $\sum_{n \in \mathbb{N}_i} \sigma(d_{nj}^i)$ means to decrease the similarity of negative pairs and increase the similarity of positive pairs, i.e., to optimize the ranking. Therefore, the gradient of each triplet d_{nj}^i plays an important role in the ranking optimization. However, d_{nj}^i is wrapped by the sigmoid function $\sigma'(\cdot)$, whose gradient-effective interval is quite narrow since the derivative decreases exponentially as the distance from 0 becomes larger (see Fig. 3a), especially when τ is small ($\tau = 0.01$ performs best as mentioned in [18]). In this case, only the triplets close to 0 can achieve effective gradients, while for those away from 0 with certain distance, the gradients are extremely small, i.e., vanished.

On the other hand, during the later period of training, we have two observations about the distribution of d_{nj}^i (see the green line in the upper right corner of Fig. 2): (1) Most of the triplets are in proper ranking order after considerable time of optimization, i.e., the majority of d_{nj}^i are smaller than 0 with certain distance, and therefore achieve extremely small gradients. However, we propose that there still exists valuable information among them which can further help refining the ranking positions. Intuitively, the best-ranking positions in the fixed old feature space are more likely to be achieved when more triplets (ranking relations) are considered. (2) There still exist several extremely hard negative samples which are closer to the anchors than the positive samples with certain distance due to the entangled clusters in the old feature space, i.e., several d_{nj}^i are still larger than 0 with certain distance. However, such extremely hard negative samples can hardly be optimized due to the vanished gradients. Therefore, their gradients should also be enhanced. Based on (1) and (2), we conclude that the gradient vanish issue stems the further ranking optimization during the later period of training.

To address this issue, we propose to reactivate these triplets by compressing their distribution toward the gradient-effective interval of $\sigma'(\cdot)$ via adding a reactivation constant c_{nj}^i during the forward step:

$$\begin{aligned} d_{nj}^i &\leftarrow d_{nj}^i + c_{nj}^i, \\ c_{nj}^i &= f(d_{nj}^i; \alpha) - d_{nj}^i, \end{aligned} \tag{4}$$

where $f(d_{nj}^i; \alpha)$ is a compression function, and α controls the degree of compression. Specifically, we choose $f(d_{nj}^i; \alpha)$ as:

$$f(d_{nj}^i; \alpha) = \sigma(d_{nj}^i; \alpha) - 0.5, \tag{5}$$

where α is the anneal of sigmoid function. The influence of α is illustrated in Fig. 3b. The larger the α is, the greater compression is. Note that c_{nj}^i is a constant without gradient, and is customized for each d_{nj}^i, therefore we term it as "dynamic".

As illustrated in the bottom right corner of Fig. 2, the distribution of the triplets (d_{nj}^i) is compressed toward 0 after adding DGR, therefore their gradients are exponentially enhanced in general, which promotes refining the ranking positions.

3.2.3. Neighbor Context Agents

Although DGR can refine the ranking positions by enhancing gradients, the ranking optimization still lacks the global perspective for targeting the best-ranking positions. Intuitively, the global best-ranking positions can be achieved only when the model sees the entire old feature space. However, the computation overhead will be extremely expensive, especially when the training set is large. Thus, we propose to include Neighbor Context Agents (NCAs) during training to approximate the entire old feature space. As illustrated in the middle of Fig. 2, NCAs are composed of random agents from each class in the neighbor area of current mini-batch. More concretely, we precompute the old features of the training set, and then build an adjacency matrix based on the euclidean distance between cluster centroids. Then, during each iteration, given the samples from the c-th class in the mini-batch, we choose the K classes closest to the c-th class in the old feature space as its neighbor set $\mathcal{N}_c$. For each class in $\mathcal{N}_c \cup \{c\}$, we randomly pick one sample as the agent at each iteration:

$$A_{old}^c = \{Rand(F_{old}^k) | k \in \mathcal{N}_c \cup \{c\}\}, \tag{6}$$

where $Rand(\cdot)$ represents for the operation of randomly selecting one feature, F_{old}^k represents the set of old features for k-th class, and A_{old}^c represents the set of the selected agents for c-th class. Assuming that there are totally B classes in a mini-batch ($\{k_1, k_2, ..., k_B\}$), all the gallery features picked at each iteration are:

$$F_g = A_{old}^{k_1} \cup A_{old}^{k_2} \cup ... \cup A_{old}^{k_B}. \tag{7}$$

After training with abundant iterations, the NCAs will walk through the entire old feature space, which can help targeting the best-ranking positions. Also, NCAs can provide more triplets for DGR to reactivate.

4

		$\mathcal{T}_{old}$	$\mathcal{B}_{old}$	$\mathcal{T}_{new}$	$\mathcal{B}_{new}$	$\mathcal{D}_{test}$	Sup	CD	NS
ID-S-1	M2M	$\mathcal{T}_{M0.1}$	R_{50}	$\mathcal{T}_M$	R_{50}	$\mathcal{D}_M$	✓	✗	✗
	D2D	$\mathcal{T}_{D0.1}$	R_{50}	$\mathcal{T}_D$	R_{50}	$\mathcal{D}_D$			
ID-S-2	M2M	$\mathcal{T}_{M0.1}$	R_{50}	$\mathcal{T}_M$	R_{101ibn}	$\mathcal{D}_M$	✓	✗	✓
	D2D	$\mathcal{T}_{D0.1}$	R_{50}	$\mathcal{T}_D$	R_{101ibn}	$\mathcal{D}_D$			
CD-S-1	M2D	$\mathcal{T}_M$	R_{50}	$\mathcal{T}_D$	R_{50}	$\mathcal{D}_D$	✓	✓	✗
	D2M	$\mathcal{T}_D$	R_{50}	$\mathcal{T}_M$	R_{50}	$\mathcal{D}_M$			
CD-S-2	M2D	$\mathcal{T}_M$	R_{50}	$\mathcal{T}_D$	R_{101ibn}	$\mathcal{D}_D$	✓	✓	✓
	D2M	$\mathcal{T}_D$	R_{50}	$\mathcal{T}_M$	R_{101ibn}	$\mathcal{D}_M$			
CD-US	M2D	$\mathcal{T}_M$	R_{50}	$\mathcal{T}_D^u$	R_{50}	$\mathcal{D}_D$	✗	✓	✗
	D2M	$\mathcal{T}_D$	R_{50}	$\mathcal{T}_M^u$	R_{50}	$\mathcal{D}_M$			

Table 1: The detailed configurations of each setting. "Sup", "CD", and "NS" represent for "Supervised", "Cross Domain", and "New Structure", separately. $\mathcal{T}_{M0.1}$ and $\mathcal{T}_{D0.1}$ represent for 10% of the training set (randomly split by identities). $\mathcal{T}_M^u$ and $\mathcal{T}_D^u$ represent for the unlabelled training set.

3.2.4. Overall Losses

We train $\mathcal{L}_m$ together with the Re-ID losses. Formally, the final loss function for backward compatible training is:

$$\mathcal{L}_{total} = \mathcal{L}_{tri} + \mathcal{L}_{id} + \mathcal{L}_m, \tag{8}$$

where $\mathcal{L}_{tri}$ and $\mathcal{L}_{id}$ represent for the Hard Mining Triplet Loss [10] and ID Loss [21], which are commonly used losses in Re-ID for discriminative learning.

4. Experimental Setting Formalization

As the backward compatible training for person Re-ID is a relatively new topic, we would like to first formalize the experimental setup in this section before describing the details of experimental results.

4.1. Datasets and Backbones

We conduct experiments on Market-1501 [22] and DukeMTMC-reID [23, 24]. Market-1501 contains 12,936 images from 751 identities for training and 19,732 images from 750 identities for testing. The images are captured from 6 cameras. DukeMTMC-reID contains 16,522 images from 702 identities for training and the rest images from 702 identities for testing. ResNet-50 [25] and IBN-ResNet-101 [26] are used as the backbones.

4.2. Setting Types

Notations. Formally, we represent the different backbones as $\mathcal{B} = \{R_{50}, R_{101ibn}\}$, where R_{50} and R_{101ibn} represent for ResNet-50 and IBN-ResNet-101, respectively. The training sets are represented as $\mathcal{T} = \{\mathcal{T}_M, \mathcal{T}_D\}$, where $\mathcal{T}_M$ represents for the training set of Market1501, and $\mathcal{T}_D$ for DukeMTMC-reID. Similarly, their test sets are denoted as $\mathcal{D} = \{\mathcal{D}_M, \mathcal{D}_D\}$. The detailed configurations of all the settings below are illustrated in Table. 1.

4.2.1. In-Domain Settings

We first test the BCT performance under the basic in-domain settings. For each in-domain setting, we test on both Market1501 (M2M) and DukeMTMC-reID (D2D).

In-Domain Supervised Setting 1 (ID-S-1): Under this setting, we randomly split 10% IDs from the original training set for the training of old model, and train the new model with the entire training set.

In-Domain Supervised Setting 2 (ID-S-2): To verify the robustness of the methods to the structure variance, under this setting, we replace the backbone of the new model in ID-S-1 with IBN-ResNet-101.

4.2.2. Cross-Domain Settings

Aside from the basic in-domain settings, we also test our methods under challenging cross-domain settings targeting several real-world scenarios. For each cross-domain setting, we test on two directions, i.e., from Market1501 to DukeMTMC-reID (M2D) and from DukeMTMC-reID to Market1501 (D2M).

Cross-Domain Supervised Setting 1 (CD-S-1): In real-world applications, the poor performance of the old model may be caused by the domain bias between the training set and the application scenario (test set). To improve the performance, new training set from the target domain is collected for model upgrade. Moreover, training set for the old model is not accessible any more due to the expensive storage overhead and privacy issue. This setup is challenging due to the domain gap and the poor performance of the old model.

Cross-Domain Supervised Setting 2 (CD-S-2): Similar to ID-S-2, we use IBN-ResNet-101 as the backbone of new model, while ResNet-50 for the old model under this setting.

Cross-Domain Unsupervised Setting (CD-US): In real-world applications, achieving the large-scale labelled training set is expensive, thus unsupervised domain adaption (UDA) is extensively studied [4, 5, 6, 7, 8]. We also prove the effectiveness of our method under this challenging setting. We conduct UDA from the domain of old model to the domain of new model.

5. Experimental Results

In this section, we first complement some experimental details. Then, we conduct primary experiments for all the settings. After that, we compare RBCL with baseline methods in the above mentioned five setting types. Finally, we analyze the effectiveness of RBCL with the ablation study and visualization.

5.1. Experimental Details

For all the supervised learning settings, we conduct experiments based on BoT baseline [28, 29], and all the tricks and hyper-parameters are kept except that we remove the Center Loss [30].

For the unsupervised setting (CD-US), we use the baseline modified based on the open sourced UDA strong baseline [1]. To be consistent with BoT, we remove the non-local block and generalized mean pooling from the default setting and use the basic ResNet-50 as the backbone. We conduct the backward compatible training together with the UDA iteration. We update the labels of old features with the pseudo labels after each time of clustering.

---

[1] https://github.com/zkcys001/UDAStrongBaseline

Method	ID-S-1				ID-S-2				CD-S-1				CD-S-2				CD-US			
	M2M		D2D		M2M		D2D		M2D		D2M		M2D		D2M		M2D		D2M	
	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1
Direct	37.5	61.0	36.5	56.4	37.5	61.0	36.5	56.4	17.9	31.9	22.4	49.2	17.9	31.9	22.4	49.2	28.7	44.5	30.0	58.8
LB	50.0	74.1	43.6	66.2	0.2	0.1	0.2	0.1	32.2	54.5	33.0	57.5	0.2	0.0	0.2	0.1	0.4	0.3	0.2	0.1
UB	85.8	93.8	76.6	86.3	89.1	95.7	81.0	90.6	76.6	86.3	85.8	93.8	81.0	90.6	89.1	95.7	58.7	74.5	67.6	85.0

Table 2: The primary experiments of BCT under all the settings. No additional loss for improving the backward compatibility is used. "Direct", "LB" and "UB" represent for $M(\phi_{old}, \phi_{old})$, $M(\phi_{new}, \phi_{old})$, and $M(\phi_{new}, \phi_{new})$ (defined in Sec. 3.1), separately.

Method	ID-S-1				ID-S-2				CD-S-1				CD-S-2				CD-US			
	M2M		D2D		M2M		D2D		M2D		D2M		M2D		D2M		M2D		D2M	
	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1
baseline	50.0	74.1	43.6	66.2	0.2	0.1	0.2	0.1	32.2	54.5	33.0	57.5	0.2	0.0	0.2	0.1	0.4	0.3	0.2	0.1
L2	51.1	74.8	46.1	67.3	49.8	73.9	45.9	67.6	29.8	47.9	40.1	69.1	23.0	39.5	31.7	58.0	24.3	40.2	22.2	41.0
	(77.0)	(89.5)	(68.9)	(80.7)	(81.4)	(92.0)	(73.9)	(86.1)	(64.2)	(75.7)	(79.6)	(90.6)	(64.6)	(76.9)	(83.0)	(92.4)	(47.1)	(63.0)	(50.1)	(74.0)
Influence [1]	46.7	71.8	41.7	64.8	32.5	53.8	25.9	48.4	32.9	56.8	39.5	67.5	28.5	52.0	35.3	61.7	34.5	53.9	34.5	56.7
	(81.9)	(92.8)	(70.9)	(83.7)	(82.3)	(92.6)	(64.8)	(85.4)	(70.8)	(82.0)	(79.9)	(91.9)	(75.6)	(87.2)	(85.3)	(94.4)	(55.8)	(71.0)	(59.9)	(79.4)
MMD [27]	42.5	74.6	42.9	66.0	49.0	73.1	42.9	66.1	37.7	64.0	42.5	74.6	36.6	64.1	37.4	67.4	16.2	27.4	15.5	27.0
	(59.8)	(78.5)	(36.2)	(52.3)	(57.5)	(74.6)	(37.7)	(55.0)	(46.6)	(60.7)	(61.7)	(79.9)	(48.3)	(64.8)	(59.1)	(77.7)	(38.0)	(52.9)	(35.1)	(59.6)
Triplet [10]	53.1	75.4	44.2	64.8	51.4	73.8	44.8	66.2	40.5	62.1	48.6	70.5	39.8	61.1	46.1	67.1	29.0	47.3	27.5	50.1
	(82.6)	(92.8)	(71.9)	(84.4)	(86.1)	(93.9)	(76.4)	(87.5)	(71.8)	(83.9)	(84.5)	(93.5)	(77.6)	(88.6)	(87.8)	(94.8)	(48.8)	(65.2)	(50.3)	(74.6)
RBCL (ours)	61.3	82.7	53.2	75.0	60.8	82.8	54.0	75.3	51.2	73.5	58.4	81.6	50.6	73.7	58.0	80.2	40.5	63.0	39.5	64.8
	(85.0)	(93.7)	(74.3)	(85.5)	(88.0)	(95.0)	(79.5)	(89.8)	(75.6)	(86.4)	(85.4)	(93.8)	(79.7)	(90.2)	(88.5)	(95.4)	(60.6)	(75.5)	(64.9)	(83.6)
UB	85.8	93.8	76.6	86.3	89.1	95.7	81.0	90.6	76.6	86.3	85.8	93.8	81.0	90.6	89.1	95.7	58.7	74.5	67.6	85.0

Table 3: The experimental results for the comparison between our proposed RBCL and other methods. The gray number in between "()" is the performance of the self-test retrieval. Our RBCL outperforms others in both cross-model and self-test retrieval by large margins. The best and second best performance for cross-model and self-test retrieval are in bold and underlined, respectively.

The new model is **initialized** with the parameters of old model **by default** for all the reported results, except when the network structure changes (i.e., ID-S-2 and CD-S-2). We calculate $\mathcal{L}_m$ on the features before classifier with cosine similarity. We add the DGR after the convergence of training loss. The mAP and Rank-1 accuracy are reported as evaluation metrics. Considering the variance between supervised and unsupervised learning, we set $K = 100, \alpha = 0.5$ for supervised settings and $K = 40, \alpha = 0.3$ for the unsupervised setting.

5.2. Primary Experiment

We first conduct primary experiments under all the settings without adding any loss for improving the backward compatibility. The results are illustrated in Table. 2. By observing the table, we summarize the conclusions as follows:

(1) The extended cross-domain settings are more challenging than the in-domain settings. The performance of Direct under the cross-domain settings is significantly lower than that of the in-domain settings. For example, the mAP of Direct in ID-S-1-M2M is 37.5%, whilst only 17.9% for CD-S-1-M2D. This is caused by the domain gap between source domain and target domain, and achieving backward compatibility in such a chaos old feature space is rather challenging.

(2) The unsupervised setting is more difficult than the supervised settings. This can be observed by comparing between the LB of CD-US and CD-S-1. For the supervised setting CD-S-1-M2D, it can achieve 32.2% mAP in LB, whilst the unsupervised setting CD-US-M2D only achieves 0.4% mAP. This decrement may be caused by the unstable pseudo labeling process in UDA iterations. The label space is different after each time of clustering, therefore the current feature space is gradually shifted away from the old feature space.
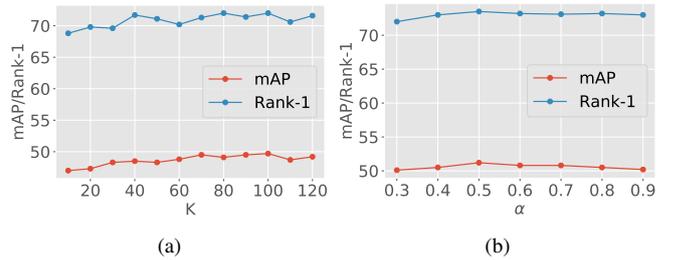

Figure 4: (a) The influence of hyper-parameter K in NCAs under CD-S-1-M2D. (b) The influence of hyper-parameter α in DGR under CD-S-1-M2D.

(3) The structure variation will increase the difficulty of achieving backward compatibility. By comparing the performance between ID-S-1 and ID-S-2, CD-S-1 and CD-S-2, we find that the performance of new model itself (UB) is improved, while the cross-model retrieval performance (LB) is decreased a lot, e.g., the mAP decreases from 50.0% in ID-S-1-M2M to 0.2% in ID-S-2-M2M. Intuitively, different structures have different capacities and parameters, thus the final feature spaces are different.

5.3. Comparison with Baseline Methods

5.3.1. Baseline Methods

The comparison between RBCL and baseline methods under all the settings is illustrated in Table. 3. The performance of cross-model retrieval without additional loss for improving the backward compatibility is reported as the baseline. Aside from the cross-model retrieval performance, we also report the

Method	ID-S-1				ID-S-2				CD-S-1				CD-S-2				CD-US			
	M2M		D2D		M2M		D2D		M2D		D2M		M2D		D2M		M2D		D2M	
	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1	mAP	R-1
baseline	57.0	79.5	50.2	72.0	53.4	77.1	46.7	69.1	44.6	67.0	52.4	76.2	42.8	65.8	50.1	73.5	36.5	57.6	35.0	59.6
+ DGR	57.5	79.9	50.6	72.9	55.2	78.5	48.8	71.3	45.2	67.7	52.8	76.6	44.0	66.8	51.6	75.3	37.8	58.9	36.8	62.7
+ NCAs	60.4	81.9	52.1	73.7	59.4	81.7	52.2	73.6	49.7	72.0	57.6	80.3	49.4	71.6	56.6	79.1	39.6	62.0	38.9	63.4
+ NCAs + DGR (ours)	61.3	82.7	53.2	75.0	60.8	82.8	54.0	75.3	51.2	73.5	58.4	81.6	50.6	73.7	58.0	80.2	40.5	63.0	39.5	64.8

Table 4: The ablation of NCAs and DGR in RBCL. Cross-model retrieval performance is reported.

(a) Distribution of triplets (d_{nj}^i).

(b) Gradient of triplets (d_{nj}^i).

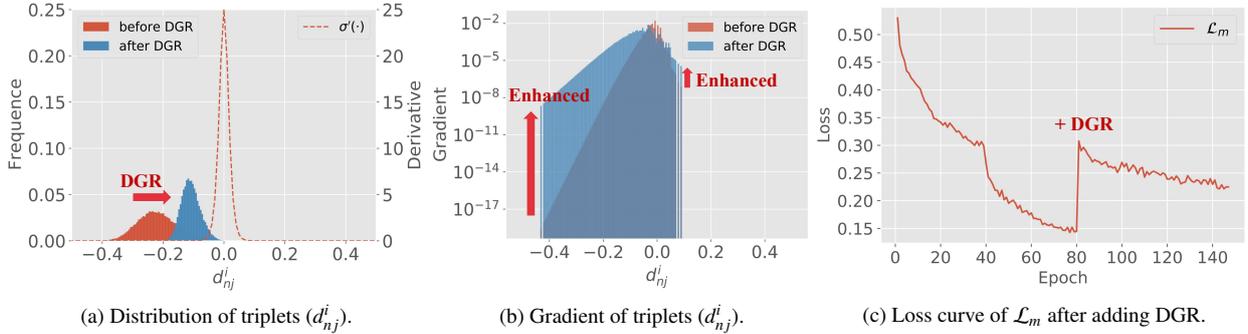(c) Loss curve of $\mathcal{L}_m$ after adding DGR.

Figure 5: The visualization of distribution and gradient change of triplets (d_{nj}^i) in a mini-batch, and the loss curve of $\mathcal{L}_m$ (with NCAs) after adding DGR. We add DGR after the convergence of $\mathcal{L}_m$ under the CD-S-1-M2D setting and α is set as 0.5. (a) The distribution of triplets before (red) and after (blue) adding DGR, and the derivative of sigmoid function when $\tau = 0.01$. (b) The average gradients in each interval before (red) and after (blue) adding DGR. We split the range of d_{nj}^i into several intervals, and then report the average gradients for each interval. Note that the y axis is logarithmic. (c) The loss curve of $\mathcal{L}_m$ under CD-S-1-M2D. DGR is added after epoch 80.

self-test performance of new model to illustrate the superiority of our RBCL.

We compare our proposed RBCL with three distillation-based methods and one . The first one is to optimize the L_2 Loss between new features and the corresponding old ones in a mini-batch. The second one is the Influence Loss proposed in [1]. The third one is the Maximum Mean Discrepancy (MMD) loss proposed in [27] which can make the distribution of new and old features in a mini-batch close to each other. The last one is Hard Mining Triplet Loss [10], which is also a typical ranking-based loss but belongs to the distance metric optimization.

5.3.2. Analysis of Cross-model Performance

The cross-model retrieval performance of our RBCL outperforms other methods by large margins under all the settings.

Compared with the distillation-based losses, in ID-S-1-D2D, RBCL outperforms Influence Loss by 11.7% and 10.2% in mAP and Rank-1, separately. For Influence Loss and MMD Loss, their performance in ID-S-1 is even lower than the baseline, which significantly shows the inferiority of the distillation-based methods. They only push the new features to be close the old features, which will lead to mis-ranked new features, therefore the cross-model retrieval performance is inferior.

Compared with another ranking-based method, in ID-S-2-M2M, RBCL surpasses Triplet Loss by 9.4% and 9.0% in mAP and Rank-1, respectively. In CD-S-1-M2D, RBCL surpasses it by 10.7% and 11.3% in mAP and Rank-1, respectively. The significant improvement shows that optimizing ranking metric is better than distance metric for BCT, and the proposed DGR
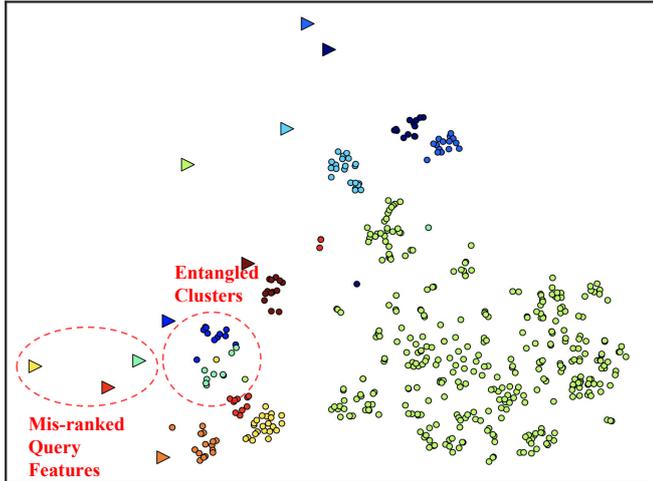
and NCAs can help the new features achieve better ranking positions than the basic Triplet Loss.
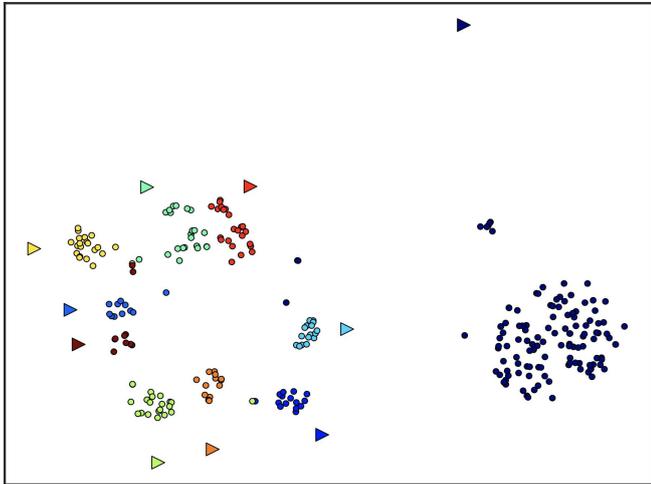
5.3.3. Analysis of Self-test Performance

The self-test performance of RBCL is the closest one to the upper bound. By comparing the self-test performance of RBCL with the upper bound, we find that the backward compatible training process of RBCL has little impact on the performance of new model itself. For example, the Rank-1 in ID-S-1-M2M only decreases from 93.8% to 93.7%, and the mAP and Rank-1 in CD-US-M2D even outperform the upper bound by 1.9% and 1.0%, respectively. This superiority is attributed to that RBCL only adjust the absolute position of each cluster toward a best-ranking position, whilst the relative distribution between clusters are maintained, therefore the self-test performance will not decrease too much.

5.3.4. Robustness to Structure Variation

By comparing the performance between ID-S-1 and ID-S-2, CD-S-1 and CD-S-2, we observe that the performance of RBCL is both high and robust in self-test and cross-model performance. For the cross-model performance, the mAP difference and Rank-1 difference of RBCL between CD-S-1-M2D and CD-S-2-M2D are 0.6% and 0.2%, while for L2 Loss, the differences become 6.8% and 8.4%. As to the self-test performance, the difference between RBCL and upper bound is consistently small. For example, the Rank-1 difference in CD-S-1-D2M is 0.0%. After the change of new model structure in CD-S-2-D2M, the difference merely increases by 0.3%. When the structure changes, the gap between the new feature space

(a) Influence Loss

(b) RBCL (ours)

Figure 6: The visualization of feature distributions for different methods under CD-US-M2D by t-SNE. The features of the first 10 IDs of $\mathcal{D}_D$ are illustrated. Triangles and circles represent for new query features and old gallery features, separately. Different colors denote for different IDs. Best view in color and zoom in.

and the old one becomes larger. Distillation-based methods intend to roughly diminish the gap, whilst RBCL only tries to find better ranking order of samples, therefore the impact on self-test performance is smaller.

5.4. Effectiveness of RBCL

5.4.1. Influence of K and α

For supervised settings, we choose the hyper-parameters under CD-S-1-M2D, then adopt it for all the supervised settings.

For the choice of K (Fig. 4a), we change K sequentially from 10 to 120 with 10 as the step size. In general, the performance increases as K becomes larger (with slight oscillation), which proves the effectiveness of NCAs. We set $K = 100$ for all the supervised settings due to its best performance.

For the ablation of α (Fig. 4b), we add DGR after the convergence of using NCAs ($K = 100$), and then adjust α from 0.3 to 0.9 with 0.1 as the step size. Obviously, $\alpha = 0.5$ gives the best performance because it provides the moderate gradients.

The similar ablation for unsupervised settings is conducted the same way, and we find that $K = 40, \alpha = 0.3$ works well.

5.4.2. Ablation of DGR and NCAs

The ablation of DGR and NCAs is illustrated in Table. 4. The ranking metric optimization between new features and the corresponding old features without DGR and NCAs is used as the baseline. We have the following important observations:

(1) DGR can boost the performance of both baseline method and the baseline+NCAs method under all the challenging settings. For example, for the unsupervised setting CD-US-D2M, the Rank-1 increments are 3.1% and 2.4%, respectively. DGR solves the gradient vanish issue, which can refine the ranking positions and optimize the extremely hard negative samples, thus the performance is improved.

(2) With the help of NCAs, the performance is largely boosted. After adding the NCAs to the baseline, for the in-domain setting ID-S-2-M2M, the mAP is increased by 6.0%. For the cross-domain settings CD-S-2-M2D and CD-S-2-D2M, the mAP are increased by 6.6% and 6.5%, separately. This strongly proves the effectiveness of our proposed NCAs, which successfully approximates the entire ole features space and brings the optimization with global perspective.

(3) The combination of NCAs and DGR gives the best performance. For example, the mAP and Rank-1 achieve increments of 7.8% and 7.9% in CD-S-2-M2D, separately. For the challenging unsupervised setting CD-US-M2D, the mAP and Rank-1 are also increased by 4.0% and 5.4%, separately.

In summary, both NCAs and DGR can boost the performance for all the challenging settings.

5.4.3. Visualization of DGR

We visualize the distribution and gradient changes in a mini-batch in Fig. 5a and Fig. 5b. After adding DGR, the distribution is closer to the gradient-effective interval of sigmoid function (Fig. 5a), therefore the gradients of the majority of triplets are exponentially enhanced (Fig. 5b). We also illustrate the loss curve of $\mathcal{L}_m$ during the training of CD-S-1-M2D in Fig. 5c. We add DGR after epoch 80. Obviously, the loss is enlarged after adding DGR, which means more ranking information is mined.

5.4.4. Visualization of Feature Space

To intuitively illustrate the principle difference between our RBCL and the existing distillation-based methods, we visualize the feature distributions in Fig. 6. We choose the Influence Loss as the representation of the distillation-based methods. We can easily observe that: (1) The old feature clusters are highly entangled with each other, which is due to the poor representation ability of the old model. (2) For the Influence Loss, there exist quite a lot of mis-ranked new query features, while for our RBCL, most of the new query features are in proper-ranking positions.

6. Conclusion

In this paper, we propose the effective RBCL for BCT task in person Re-ID. Different from the existing distillation-based methods, our proposed RBCL optimizes the ranking metric between new features and old features, which is superior in principle. To relieve the gradient vanish issue incurred by the sharp sigmoid function, we propose the DGR, which can reactivate the suppressed gradients. To further help targeting the global best-ranking positions, we propose to include NCAs during training, which can approximate the entire old feature space. RBCL outperforms other methods by a large margin under both in-domain and challenging cross-domain settings. In the future, we will study this problem on more retrieval tasks.

Acknowledgment

References

[1] Y. Shen, Y. Xiong, W. Xia, S. Soatto, Towards backward-compatible representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6368–6377.

[2] C.-Y. Wang, Y.-L. Chang, S.-T. Yang, D. Chen, S.-H. Lai, Unified representation learning for cross model compatibility, arXiv preprint arXiv:2008.04821 (2020).

[3] K. Chen, Y. Wu, H. Qin, D. Liang, X. Liu, J. Yan, R3 adversarial network for cross model face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9868–9876.

[4] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, T. S. Huang, Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6112–6121.

[5] Y. Ge, D. Chen, H. Li, Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification, arXiv preprint arXiv:2001.01526 (2020).

[6] Y. Ge, D. Chen, F. Zhu, R. Zhao, H. Li, Self-paced contrastive learning with hybrid memory for domain adaptive object re-id, arXiv preprint arXiv:2006.02713 (2020).

[7] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, X. Wang, Unsupervised domain adaptive re-identification: Theory and practice, Pattern Recognition 102 (2020) 107173.

[8] Y. Zhai, S. Lu, Q. Ye, X. Shan, J. Chen, R. Ji, Y. Tian, Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9021–9030.

[9] M. Ye, C. Liang, Y. Yu, Z. Wang, Q. Leng, C. Xiao, J. Chen, R. Hu, Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing, IEEE Transactions on Multimedia 18 (12) (2016) 2553–2566.

[10] A. Hermans, L. Beyer, B. Leibe, In defense of the triplet loss for person re-identification, arXiv preprint arXiv:1703.07737 (2017).

[11] X. Wang, Y. Hua, E. Kodirov, G. Hu, R. Garnier, N. M. Robertson, Ranked list loss for deep metric learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5207–5216.

[12] X. Wang, H. Zhang, W. Huang, M. R. Scott, Cross-batch memory for embedding learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6388–6397.

[13] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4004–4012.

[14] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[15] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 1857–1865.

[16] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4004–4012.

[17] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, S. Singh, No fuss distance metric learning using proxies, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 360–368.

[18] A. Brown, W. Xie, V. Kalogeiton, A. Zisserman, Smooth-ap: Smoothing the path towards large-scale image retrieval, arXiv preprint arXiv:2007.12163 (2020).

[19] Z. Li, D. Hoiem, Learning without forgetting, IEEE transactions on pattern analysis and machine intelligence 40 (12) (2017) 2935–2947.

[20] Q. Meng, C. Zhang, X. Xu, F. Zhou, Learning compatible embeddings, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9939–9948.

[21] Z. Zheng, L. Zheng, Y. Yang, A discriminatively learned cnn embedding for person reidentification, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 14 (1) (2017) 1–20.

[22] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1116–1124.

[23] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: European Conference on Computer Vision, Springer, 2016, pp. 17–35.

[24] Z. Zheng, L. Zheng, Y. Yang, Unlabeled samples generated by gan improve the person re-identification baseline in vitro, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3754–3762.

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[26] X. Pan, P. Luo, J. Shi, X. Tang, Two at once: Enhancing learning and generalization capacities via ibn-net, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 464–479.

[27] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, 2015, pp. 97–105.
URL http://jmlr.org/proceedings/papers/v37/long15.html

[28] H. Luo, Y. Gu, X. Liao, S. Lai, W. Jiang, Bag of tricks and a strong baseline for deep person re-identification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[29] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, J. Gu, A strong baseline and batch normalization neck for deep person re-identification, IEEE Transactions on Multimedia (2019) 1–1doi:10.1109/TMM.2019.2958756.

[30] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European conference on computer vision, Springer, 2016, pp. 499–515.