Pattern Recognition 145 (2024) 109973

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Transfer easy to hard: Adversarial contrastive feature learning for unsupervised person re-identification

Haoxuanye Ji [a], Le Wang [a,*], Sanping Zhou [a], Wei Tang [b], Nanning Zheng [a], Gang Hua [c]

[a] National Key Laboratory of Human–Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, and Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[b] Department of Computer Science, University of Illinois, Chicago, IL 60607, USA
[c] Wormpex AI Research, Bellevue, WA 98004, USA

ARTICLE INFO

Keywords:
Person re-identification
Unsupervised learning
Hard sample generation

ABSTRACT

Unsupervised Person Re-Identification (Re-ID) is challenging due to the lack of ground-truth labels. Most existing methods address this problem by progressively mining high-confidence pseudo labels to guide the feature learning process. However, how to construct hard-enough samples while maintaining the fidelity of pseudo labels in these samples remains an open issue in the machine learning community. To tackle this challenge, we design a simple yet effective adversarial contrastive feature learning (ACFL) framework, which enhances the discriminative capability of features by introducing more transformed hard samples in the feature learning process. Specifically, it mainly consists of a discriminative feature learning module and a hard sample generation module. The discriminative feature learning module extracts recognizable features of unlabeled training samples to estimate the high-confidence relationship between samples. Then, the hard sample generation module utilizes these high-confidence relationships between samples to transfer all samples into the hard ones via an adversarial learning strategy. Finally, the generated hard samples are further fed into DFL to learn discriminative features for person Re-ID. Extensive experiments on Market-1501, DukeMTMC-reID, and MSMT17 datasets show that our method compares favorably with state-of-the-art methods.

1. Introduction

Given a query image of a target person, person Re-Identification (Re-ID) aims to identify the images of the same person in a large gallery set. It is a challenging task due to the significant appearance variations caused by different poses, light conditions, and background clutter across multiple non-overlapped camera views. Due to the strong feature representation capability of the Deep Neural Network (DNN), man supervised methods [1–3] have achieved impressive performances in the past few years. However, they require a large number of labeled samples in the training process. To alleviate this issue, unsupervised methods have become a popular research topic in both industry and academic communities, because they can avoid the time-consuming and labor-intensive annotation work in practice.

The main challenge of unsupervised person Re-ID lies in learning discriminative features without using ground-truth labels. To address this problem, most methods [4–7] adopt a clustering algorithm to estimate the pseudo labels of unlabeled samples and then use them to supervise the training of the feature learning network. PUL [6] adopts pseudo labels estimated by an iterative clustering method for feature learning. Besides the offline refined pseudo labels, MMT [7] also introduces online-refined soft pseudo labels to learn better features in an alternative training manner. However, it is difficult to determine the number of clusters in the training process. Therefore, the clustering-based methods can hardly obtain high-fidelity pseudo labels, which will in turn degrade the discriminative capability of features.

When the cluster number is set to be small, as shown in Fig. 1(a), it will lead to the under-clustering phenomenon [8], in which the candidate samples of different identities will be put into one cluster. When the cluster number is set to be large, as shown in Fig. 1(b), it will lead to over-clustering phenomenon [8], in which the candidate samples of the same identity will be put into different clusters. The samples that cannot distinguish identity by the current network are usually called hard samples. These samples are more effective than the easy ones in learning discriminative features [9,10]. Therefore, extensive attention has been paid to studying how to construct hard samples and maintain the high confidence of their pseudo labels.

To the best of our knowledge, it is very difficult to mine hard samples to directly guide the feature learning process. On the one hand,

* Corresponding author.
E-mail address: lewang@xjtu.edu.cn (L. Wang).

https://doi.org/10.1016/j.patcog.2023.109973
Received 17 February 2023; Received in revised form 30 July 2023; Accepted 12 September 2023
Available online 16 September 2023

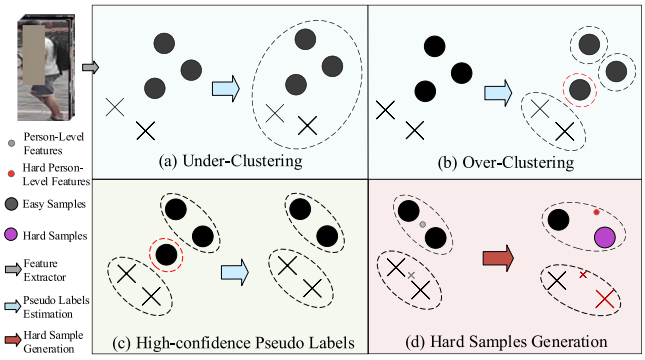

Fig. 1. Illustration of pseudo labels obtained by different iterative clustering methods in the embedding space. Different geometric shapes represent different identities. The small or large number of clusters leads to (a) under-clustering or (b) over-clustering phenomenon, respectively, where the red dotted line denotes outliers. (c) The high-confidence clustering results are used to estimate high-confidence pseudo labels, where the black dotted box means a cluster that includes several samples, and the red dotted box denotes an outlier. (d) The hard person features (red) and hard positive instances (purple) are generated by the hard sample generation module. The "small-size circle" means the prototype of its corresponding cluster, *i.e.*, person-level feature. The "large-size circle" means the feature of a sample extracted by the feature extractor Q, *i.e.*, instance-level feature. Different shapes represent different persons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the initial features are not discriminative enough to mine more valuable hard samples at the early training stage, making it less effective to further enhance the discriminative capability of features by simply feeding easy samples. On the other hand, the discriminative capability of features is very easy to degenerate once the pseudo labels of hard samples are misestimated in the training process. As a result, several research works [11–13] have attempted to use multi-view information to estimate the pseudo labels of hard samples, which achieve significant performance improvement in person Re-ID. To utilize the discriminative information in hard samples, MCN [14] uses the samples with hierarchical confidence pseudo labels to train networks for Re-ID via a co-teaching strategy. However, it is still very difficult to mine the hard samples from the training dataset because not all the hard samples can be handled at the pseudo labels estimation stage.

To obtain the hard samples with high-confidence pseudo labels (less noisy labels), we propose a new technology paradigm that transfers the easy samples (with high-confidence pseudo labels) to hard samples by adversarial learning regime, *i.e.*, "transferring rather than mining". Concretely, it mainly consists of two components, *i.e.*, discriminative feature learning (DFL) and hard sample generation (HSG). DFL first extracts recognizable features from training samples for estimating the high-confidence pseudo labels of (easy) samples. Then, HSG adopts an adversarial learning strategy to transfer these (easy) samples into hard samples for the model training process by the relationships between the query and each pseudo identity. Moreover, the samples with the same pseudo labels have the similarity feature distributions [15,16]. We construct the hard positive instances (instance-level) for each query by adversarial learning strategy based on the feature alignment process of the query and its corresponding positive instances. Finally, these hard samples are incorporated into DFL to further boost the discriminative capability of features in the training process.

These above steps are performed iteratively. Thus, the discriminative capability of features can be consistently enhanced as the iteration continues. Experimental results on Market-1501 [17], DukeMTMC-reID [18], and MSMT17 [19] datasets show that our method has achieved impressive results against other competing methods.

The contributions of this paper are summarized below:

- We design a novel adversarial contrastive feature learning (ACFL) framework for unsupervised person Re-ID, which can generate hard samples with high-confidence pseudo labels to guide the discriminative feature learning process.
- We design a discriminative feature learning (DFL) module to incorporate these hard samples to further enhance the discriminative capability of features.
- We design a novel hard sample generation (HSG) module to generate hard samples for discriminative feature learning, in which an adversarial learning regime is used to generate hard person features (class-level) and hard positive instances (instance-level).

The rest of this paper is organized as follows. In Section 2, we briefly review the related work. Section 3 presents the procedure of the proposed adversarial contrastive feature learning (ACFL) framework for unsupervised person Re-ID. The experiments and discussions are presented in Section 4. Finally, we conclude in Section 5.

2. Related work

We briefly review the related works in supervised, unsupervised domain adaptation, and unsupervised person Re-ID methods.

2.1. Supervised person Re-ID

Supervised person Re-ID methods require labor-intensive annotated images to learn discriminative features. Early methods usually extract a global feature representation per image for image retrieval [20,21]. PersonNet [20] uses a small-scale convolutional filter to capture fine-grained cues. SPReID [21] employs a human-semantic parsing technique to capture pixel-level discriminative clues. Part-level features can boost performance [2,3,22] when the background is cluttered or the pedestrian is occluded. AutoReID [22] utilizes a Re-ID search space with body structure information for Person Re-ID. PGCN [2] constructs sub-graphs based on the relationship between part-level features to highlight the effective body cues.

2.2. Unsupervised domain adaptation person Re-ID

Unsupervised Domain Adaptation (UDA) person Re-ID methods are adopted in the unsupervised Re-ID task, which utilizes prior knowledge on a labeled source dataset, and attempts to other unlabeled targets. Early UDA methods mainly exploit examplar-invariance, camera-invariance, and neighborhood-invariance of the target domain properties to improve the generalization ability [23,24]. Other methods train the Re-ID model on a labeled source dataset and then finetune it by mining the potential similarity between the unlabeled target samples [25,26]. MDJL [26] estimates pseudo labels by the correlation between multiple domains. Recently, some methods improve the generalization ability by reducing the gap between source and target domains [27–29]. IDM [29] proposes to build an appropriate intermediate domain to bridge the source and target domains. Isobe et al. [28] adopt a progressive domain adaptation strategy to gradually mitigate the domain gap. MCN [14] utilizes the samples with hierarchical confidence pseudo labels to optimize multiple networks, thus can effectively distract the noisy labels to multiple networks.

Like MCN [14], our ACFL also considers that the easy/hard samples correspond to high/low confidence pseudo labels. Unlike MCN [14], our ACFL only adopts the "easy" samples and transfers them into hard samples rather than using the mined hards. This strategy ensures the transferred hard samples share high-confidence pseudo labels with easy samples, which provide enough discriminative information and effectively alleviate the influence caused by the erroneous pseudo labels.

2.3. Unsupervised person Re-ID

Unlike supervised methods, unsupervised person Re-ID methods remove the requirement for cost-prohibitive annotations. Several

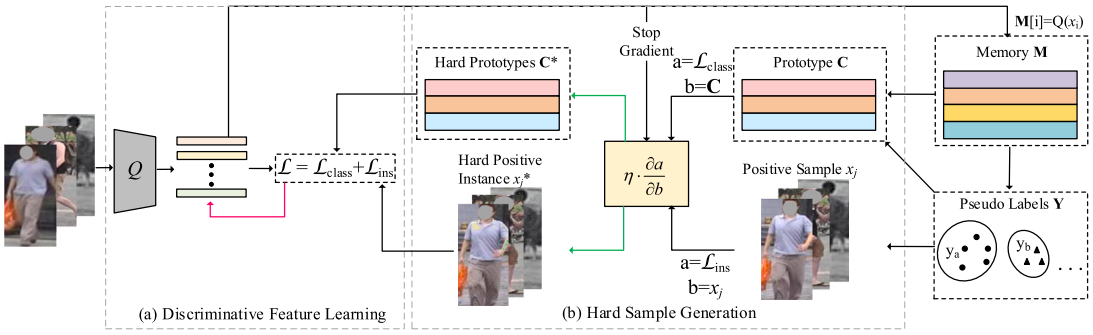

Fig. 2. Overview of our proposed ACFL framework. Before each training epoch, the discriminative feature learning module extracts features of all training images. Then, these extracted features are used to estimate the high-confidence pseudo labels for easy samples. At the training stage, the hard sample generation module generates hard samples (green arrow) based on the relationships between samples. **Y** denotes the estimated pseudo labels. Finally, the discriminative feature learning module uses these generated hard samples to guide the discriminative features learning process (red arrow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

methods learn discriminative features using pseudo labels obtained from clustering methods in the embedding space [5,30]. To address the feature variations caused by camera shift, some methods incorporate ground-truth camera labels to address the appearance variance caused by different camera views [31–33]. They rely on ground-truth camera labels to utilize full ground-truth camera labels for intra-camera and inter-camera learning.

Recently, contrastive learning-based methods have been adopted in unsupervised person Re-ID [5,34,35]. They focus on learning discriminative features by contrasting the difference between each sample pair. SPCL [5] proposes a self-paced strategy to mine reliable clusters for learning better features. PPLR [34] utilizes part-level information to construct contrastive features in the training procedure. ISE [35] utilizes a feature hybrid strategy to incorporate effective information in the contrastive learning process.

Compared with prior methods, our ACFL is built on a novel technical paradigm. It mainly focuses on transferring *easy* samples with high-confidence pseudo labels into *hard* samples to guide the feature learning process, rather than attempting to utilize the original samples with carefully designed high-fidelity pseudo labels. Specially, we design a novel hard sample generation module to generate hard cluster prototypes and positive instances in an adversarial manner, which is more effective for learning discriminative features than simply using the original samples to guide the training process.

3. Methodology

Problem Definition. In unsupervised person Re-ID, we only have an unlabeled dataset $\mathcal{X} = \{x_i\}_{i=1}^{N}$, where x_i denotes the ith training image and N is the number of training images. The goal of the task is to learn a feature extractor Q based on $\mathcal{X}$. In the testing stage, the feature extractor Q takes each query image as input and extracts discriminative features from the query image and a large gallery set. It then calculates the visual similarity between the feature of the query and each image feature in the gallery set. Finally, the similarities are used to retrieve the images containing the same person in the query from the gallery set.

Overview. We introduce the proposed adversarial contrastive feature learning (ACFL) framework for unsupervised person Re-ID. The ACFL framework is illustrated in Fig. 2. First, discriminative feature learning (DFL) extracts the features from all unlabeled training images and estimates the high-confidence pseudo labels of (easy) samples. Then, the hard sample generation (HSG) generates both the hard person-level and instance-level samples by utilizing the relationships between the query and the mined easy samples. Finally, these generated hard samples are feedback into the discriminative feature learning module to boot the training process. After the optimization procedure, the optimized deep neural network (DNN) can extract a feature vector from a given query image to retrieve the images containing the same person from a large gallery set in a nearest neighbor search manner.

3.1. Discriminative feature learning module

The DFL module aims to learn discriminative features for person Re-ID from unlabeled training samples $\mathcal{X} = \{x_i\}_{i=1}^{N}$, where N denotes the number of training samples. It consists of a feature extractor Q and a feature memory **M**. The feature extractor Q extracts the features that can ease the search for the nearest neighbors in the feature space. These features are used to estimate the pseudo labels **Y** for each samples. Q extracts a d-dimensional feature Q(x_i) for each x_i, and we denote its corresponding pseudo labels are $y_i \in \mathbf{Y}$. Afterward, we store all these features before each training epoch in the feature memory **M**, and then update it by replacing each ith stored feature **M**[i] with Q(x_i) after the training step of each x_i, *i.e.*, **M**[i] = Q(x_i).

3.2. Hard sample generation module

Hard training samples can provide more informative clues than easy ones [9,10], and thus they are more effective in learning the discriminative feature in the training process. Nevertheless, the hard samples are very difficult to mine without ground-truth labels. To address the issue, we design a hard sample generation module to transfer the (easy) samples with high-confidence pseudo labels into the hard samples, and then use them to learn discriminative features.

Generation of Hard Person Features. Based on the extracted features, the high-confidence pseudo labels can be estimated by the clustering algorithm. For each query x_i, we can employ the samples with the same high-confidence pseudo labels y_i as its positive candidates. To this end, we average the features of these samples as the person feature of person y_i, which is denoted as $\mathbf{c}_{y_i}$.

Hence, to generate the hard person features, an intuitive solution is to transfer all the training images into its hard version via a basic iterative method [36]. Then, the hard person features are obtained by averaging the features of all hard samples in the same cluster. However, it is time-consuming because this procedure needs to be repeated to update the features of all training images at each iteration.

To address this issue, we convert the person features directly into their hard versions by utilizing the relationships between each query sample and each person. Motivated by the idea of adversarial learning, we generate the hard person features by maximizing the difference between each query feature vector Q(x_i) and person features **C**, where $\mathbf{C} = \{\mathbf{c}_a\}_{a=1}^{N_c^{(t)}}$ stores the feature $\mathbf{c}_a$ of each person a. $\mathbf{c}_a$ is obtained by averaging the features of the samples with the same pseudo person identity a. $N_c^{(t)}$ is the current number of persons. To this end, the hard person features $\mathbf{C}^*$ can be formulated as:

$$\mathbf{C}^* = \underset{\mathbf{C}}{\operatorname{argmax}} \mathcal{L}_{\text{class}}(s(\mathrm{Q}(x_i)), \mathbf{C}), \tag{1}$$

where $\mathcal{L}_{\text{class}}$ denotes a classification loss function, which aims to pull closer $\mathrm{Q}(x_i)$ and the person feature $\mathbf{c}_{y_i}$ and push away $\mathrm{Q}(x_i)$ and other person features. $s(\cdot)$ means the stop-gradient operation.[1]

To obtain the hard person features $\mathbf{C}^* = \{\mathbf{c}_a^*\}_{a=1}^{N_c^{(t)}}$, where $\mathbf{c}_a^*$ denotes the generated hard person feature of person a, we optimize Eq. (1) by utilizing an adversarial learning regime in the following two steps. First, we initialize the hard person features as $\mathbf{C}^{(0)} = \mathbf{C}$. Second, we employ stochastic gradient descent (SGD) to obtain the optimal solution $\mathbf{C}^*$. In particular, the maximization process in Eq. (1) is equivalent to iteratively minimizing the negative loss value between x_i and $\mathbf{C}$. The intermediate optimized solution at the υth iteration can be formulated as follows:

$$\begin{aligned}\mathbf{c}_a^{(\upsilon)} &= \mathbf{c}_a^{(\upsilon-1)} - \eta \cdot \mathbf{g}_{\text{class}},\\ \mathbf{g}_{\text{class}} &= \frac{\partial}{\partial \mathbf{c}_a^{(\upsilon-1)}}(-\mathcal{L}_{\text{class}}(s(\mathrm{Q}(x_i)), \mathbf{C}^{(\upsilon-1)})),\end{aligned} \tag{2}$$

where $\mathbf{c}_a^{(\upsilon)}$ represents the intermediate optimized results at the υth iteration. $\mathbf{g}_{\text{class}}$ means $-\mathcal{L}_{\text{class}}$ with partial respect to $\mathbf{c}_a^{(\upsilon-1)}$. η is the updating rate of generating hard person features, which is determined by the relationship between x_i and each person a. Specifically, if x_i belongs to the person a ($y_i = a$), we set $\eta = \eta_1$, which denotes the updating rate of generating hard positive person feature; otherwise, $\eta = \eta_2$, which denotes the updating rate of generating hard negative person features. After V iterations, we stop the above optimization process and obtain the optimal solution $\mathbf{C}^{(V)} = \{\mathbf{c}_a^{(V)}\}_{a=1}^{N_c^{(t)}}$. The optimal hard person features can be denoted as $\mathbf{C}^* = \mathbf{C}^{(V)}$ for all clusters.

After the hard person features generation process, the hard training sample pair $(x_i, \mathbf{c}_a^*)$ is generated for each person feature $\mathbf{c}_a$ in $\mathbf{C}$ according to the relationship between the sample x_i and person a. In the following process, we utilize the relationship between instances, *i.e.*, x_i and another sample from the high-confidence positive candidate of x_i have a high-confidence positive relationship. The hard positive instances are generated based on the pairwise relationship.

Generation of Hard Positive Instances. Considering the relationship between samples, we select a positive sample x_j of x_i in its high-confidence positive candidates (*i.e.*, $y_j = y_i$) and utilize (x_i, x_j) to describe the easy positive sample pair. For (x_i, x_j), we need to generate a hard positive instance x_j^* by simply transferring the positive sample x_j of each query sample x_i into its hard version. To this end, we generate the hard positive instance x_j^* by maximizing the difference between x_i and x_j, which is defined as follows:

$$x_j^* = \text{argmax}_{x_j} \mathcal{L}_{\text{ins}}(s(\mathrm{Q}(x_i)), \mathrm{Q}(x_j)), \tag{3}$$

where $\mathcal{L}_{\text{ins}}$ is a metric loss function, which aims to reduce the difference between $\mathrm{Q}(x_i)$ and $\mathrm{Q}(x_j)$.

To obtain the hard positive instance x_j^*, we optimize Eq. (3) by utilizing the adversarial learning regime in the following two steps. First, we initialize the hard positive instance as $x_p^{(0)} = x_p$. Next, we employ the SGD algorithm to minimize the negative loss value between x_i and x_j iteratively. The optimized solution at the zth iteration can be computed as follows:

$$\begin{aligned}x_j^{(z)} &= x_j^{(z-1)} - \eta_3 \cdot \mathbf{g}_{\text{ins}} / \|\mathbf{g}_{\text{ins}}\|_2,\\ \mathbf{g}_{\text{ins}} &= \frac{\partial}{\partial x_j^{(z-1)}}(-\mathcal{L}_{\text{ins}}(s(\mathrm{Q}(x_i)), \mathrm{Q}(x_j^{(z-1)}))),\end{aligned} \tag{4}$$

where $x_j^{(z)}$ represents the optimization results at zth iteration. η_3 is the updating rate of generating hard instances. The partial derivative of $-\mathcal{L}_{\text{ins}}$ with respect to $x_j^{(z-1)}$ is denoted as $\mathbf{g}_{\text{ins}}^{(z-1)}$. In practice, we stop the above optimization process after Z iterations, and the optimized hard positive instance can be obtained as $x_j^* = x_j^{(Z)}$ for each query sample x_i.

[1] We use the stop-gradient operation to avoid the impact of the hard sample generation process on the parameters in Q.

Algorithm 1 Optimization Procedure of ACFL.

Input: Unlabeled training dataset $\mathcal{X} = \{x_i\}_{i=1}^N$, initialized encoder Q, training epoch T.
Output: Optimized encoder Q,

Init.: Feature memory $\mathbf{M} = \{\mathrm{Q}(x_i)\}_{i=1}^N$;
for $t = 1, t \le T, t$ ++ **do**
  Utilize a non-parametric clustering method with $\{\mathrm{Q}(x_i)\}_{i=1}^N$ to estimate high-confidence pseudo labels $\mathbf{Y} = \{y_i\}_{i=1}^N$;
  for each x_i in $\mathcal{X}$ **do**
    # Hard person features generation
    Calculate person features $\mathbf{C} = \{\mathbf{c}_a\}_{a=1}^{N_c^{(t)}}$;
    Generate hard person features $\mathbf{C}^*$ for x_i using Eq. (2) after V iterations;
    # Hard positive instances generation
    Randomly select a positive sample x_j for x_i , where $y_j = y_i$ and $x_j \in \mathcal{X}$;
    Generate hard positive instance x_j^* for x_i using Eq. (4) after Z iterations;
    # Discriminative feature learning
    Optimize Q by minimizing Eq. (5);
    Update $\mathrm{M}^{(t)}[i]$ by $\mathrm{Q}(x_i)$;
  end for
end for
return Optimized Q

Algorithm 2 Generation Hard Person Features.

Input: Query sample x_i, Person features $\mathbf{C}$, Updating rate η_1, η_2.
Output: Hard cluster prototypes $\mathbf{C}^* = \{\mathbf{c}_{y_i}^*, \{\mathbf{c}_a^*\}_{a \neq y_i}\}$ of x_i.

Init.: $\mathbf{C}^{(0)} = \{\mathbf{c}_{y_i}, \{\mathbf{c}_a\}_{a \neq y_i}\}$;
for $\upsilon = 1, \upsilon \le V, \upsilon$ ++ **do**
  Calculate $\partial \mathcal{L}_{\text{class}} / \partial \mathbf{c}_{y_i}^{(\upsilon-1)}$;
  $\mathbf{c}_{y_i}^{(\upsilon)} = \mathbf{c}_{y_i}^{(\upsilon-1)} + \eta_1 * \partial \mathcal{L}_{\text{class}} / \partial \mathbf{c}_{y_i}^{(\upsilon-1)}$;
  for $\mathbf{c}_a^{(\upsilon-1)}$ in $\{\mathbf{c}_a^{(\upsilon-1)}\}_{a \neq y_i}$ **do**
    Calculate $\partial \mathcal{L}_{\text{class}} / \partial \mathbf{c}_a^{(\upsilon-1)}$;
    $\mathbf{c}_a^{(\upsilon)} = \mathbf{c}_a^{(\upsilon-1)} + \eta_2 \cdot \partial \mathcal{L}_{\text{class}} / \partial \mathbf{c}_a^{(\upsilon-1)}$;
  end for
end for
return $\mathbf{C}^*$

It is worth noting that the updating strategies of x_j^* and $\mathbf{C}^*$ are slightly different, *i.e.*, a larger updating rate but fewer updating iterations are taken to optimize x_j than $\mathbf{C}$. The reason is that $\mathbf{g}_{\text{class}}$ is directly used to obtain the hard person features, while $\mathbf{g}_{\text{ins}}$ is normalized to update the hard positive instance x_j^*. As a result, the normalized x_j and $\mathbf{g}_{\text{ins}}$ have the same scale. This can drastically reduce the difficulty of hyper-parameter search by using a large updating rate but fewer updating iterations to obtain the optimal solution x_j^*.

The hard samples generation module generates the hard cluster prototypes $\mathbf{C}^*$ and the hard positive instance x_j^* for each training image x_i. These hard samples share the high-confidence pseudo labels of the original (easy) samples, but they are more effective than their easy versions in the feature learning process. We discuss the effectiveness of these generated hard samples in the ablation study (Section 4.4).

Loss Function. These generated hard sample pairs are fed into the discriminative feature learning module for the optimization procedure, as presented in Algorithm 1. The overall loss function $\mathcal{L}_a$ of our ACFL is formulated as:

$$\mathcal{L}_a = \mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}^*) + \mathcal{L}_{\text{ins}}(\mathrm{Q}(x_i), \mathrm{Q}(x_j^*)). \tag{5}$$

Algorithm 3 Generation Hard Positive Instance.

Input: Query sample x_i, Positive sample x_j, Updating rate η_3.
Output: Hard positive instance x_j^*.

1: **Init.**: $x_j^{(0)} = x_j$;
2: **for** $z = 1, z \leq Z, z$ ++ **do**
3: Calculate $\partial \mathcal{L}_{\text{ins}} / \partial x_j^{(z-1)}$;
4: $x_j^{(z)} = x_j^{(z-1)} + \eta * \partial \mathcal{L}_{\text{ins}} / \partial x_j^{(z-1)}$;
5: **end for**
6: **return** x_j^*

3.3. Contrastive learning-based implementation

Considering the success of contrastive learning in unsupervised feature learning [5,35], we incorporate the generated hard person-level and instance-level samples into a contrastive learning-based framework. As a result, the classification loss with hard person features is formulated as:

$$\mathcal{L}_{\text{class}} = -\log \frac{\exp(\mathrm{Q}(x_i) \cdot \mathbf{c}_{y_i}^* / \tau)}{\exp(\mathrm{Q}(x_i) \cdot \mathbf{c}_{y_i}^* / \tau) + \sum_{a \neq y_i} \exp(\mathrm{Q}(x_i) \cdot \mathbf{c}_a^* / \tau)}, \tag{6}$$

where $\mathbf{c}_{y_i}^*$ and $\mathbf{c}_a^*$ denote the generated hard positive and negative person features of x_i, respectively. The generation procedure is presented in Algorithm 2. We will explore the impact of the generated hard positive cluster prototype $\mathbf{c}_{y_i}^*$ and the hard negative cluster prototype $\mathbf{c}_a^* (a \neq y_i)$ in the ablation study (Section 4.4). τ is a temperature hyper-parameter.

Besides, $\mathcal{L}_{\text{ins}}$ is to reduce the difference between $\mathrm{Q}(x_i)$ and $\mathrm{Q}(x_j)$. To align the two features, we first obtain the similarity distribution between x_i and each person, *i.e.*, $v(\mathrm{Q}(x_i)) = \{\exp(\mathrm{Q}(x_i) \cdot \mathbf{c}_a / \tau) / \sum_{\mathbf{c}_a \in \mathbf{C}} \exp(\mathrm{Q}(x_i) \cdot \mathbf{c}_a / \tau)\}_{a=1}^{N_c^{(t)}}$. Considering x_i and x_j have the same identity, $v(\mathrm{Q}(x_i))$ and $v(\mathrm{Q}(x_j))$ are similar. As a result, we utilize the relative entropy loss function as $\mathcal{L}_{\text{ins}}$, which is formulated as:

$$\mathcal{L}_{\text{ins}} = \mathrm{KL\_div}(\mathrm{Q}(x_i), \mathrm{Q}(x_j^*)), \tag{7}$$

where x_j^* denotes the generated hard positive instance. The generation procedure is reported in Algorithm 3. We also explore the impact of the generated positive instance x_j^* in Section 4.4. After the hard samples generation process, we insert $\mathcal{L}_{\text{class}}$ and $\mathcal{L}_{\text{ins}}$ into Eq. (5) to train the discriminative feature learning module.

4. Experiments and discussions

In this section, we evaluate the unsupervised person Re-ID performance of our ACFL method against state-of-the-art methods and carry out detailed ablation studies to isolate the performance contribution of each component.

4.1. Datasets and evaluation protocol

Datasets. We evaluate our method on three standard large-scale person Re-ID datasets, including Market-1501 [17], DukeMTMC-reID [18], MSMT17 [19], and CUHK03 [37].

- Market-1501 dataset consists of 32,668 images of 1,501 identities captured by 6 cameras. Its training set comprises 12,936 images of 751 identities, and the testing set contains 19,732 images of 750 identities.
- DukeMTMC-reID dataset contains 36,411 images of 1812 identities captured by 8 cameras, including 16,522 images of 702 identities for training and the remaining images for testing.
- MSMT17 dataset includes 126,411 images of 4,101 identities captured by 15 cameras. Its training set has 32,621 images of 1,041 identities, and its testing set has 93,820 bounding boxes of 3,060 identities.

Table 1
Experiments on the impact of iteration Z of generating hard positive instances on Market-1501 and DukeMTMC-reID datasets.

Z	Market		DukeMTMC-reID	
	Top-1	mAP	Top-1	mAP
$Z = 1$	93.8	84.0	85.0	73.2
$Z = 2$	94.1	84.9	85.2	73.5
$Z = 3$	**94.7**	**85.8**	85.5	**74.0**
$Z = 4$	94.7	85.6	85.5	73.7
$Z = 5$	94.5	85.5	**85.8**	73.1

- CUHK03 consists of 1,467 identities and 28,192 bounding boxes, where 26,264 images of 1,367 identities are used for training, and 1,928 images of 100 identities are used for testing. We adopt the same protocol in Auto-reid [22] to evaluate our ACFL.

We compare our method with state-of-the-art methods on the Market-1501, DukeMTMC-reID, and MSMT17 datasets, and we conduct ablation studies on the Market-1501 and DukeMTMC-reID datasets.

Evaluation Protocol. Following the common experimental setting in [17–19], we employ Cumulative Matching Characteristic (CMC) scores and mean Average Precision (mAP) to evaluate the performance of the methods. We report Top-1, Top-5, and Top-10 scores to represent the CMC curve, which reflects the retrieval precision; while mAP is calculated as the mean value of average precision across all queries, which reflects the recall.

4.2. Implementation details

We implement our method in PyTorch with two NVIDIA GeForce 1080Ti GPUs. We resize each training image as 256×128, and then utilize random crop, random flip, and random erasing for data augmentation. We adopt ResNet50 [38] with the layers after pooling-5 removed and a batch normalization layer appended as the backbone network Q. The encoder Q is optimized by the Adam optimizer with a learning rate of 0.0005 and batch size of 64. We train the model with 60 epochs in total and the learning rate decreases by a factor of 0.1 every 25 epochs.

The temperature factor τ is set to 0.05 in $\mathcal{L}_{\text{class}}$ and $\mathcal{L}_{\text{ins}}$. We then employ DBSCAN as the non-parametric clustering method with $eps = 0.5$ for both Market-1501 and DukeMTMC-reID and $eps = 0.7$ for MSMT17. In the hard sample generation module, we set $V = 10$ and $Z = 3$ for generating hard person features and hard positive instances, respectively. And we set the updating rates $\{\eta_1, \eta_2, \eta_3\} = \{0.00035, 0.0028, 1.0\}$. At the test stage, we use the encoder Q to extract 2048-dimensional discriminative features for adoption in the person Re-ID task.

4.3. Parameter analysis

We first explore the impacts of the iteration V and Z of our method on Market-1501 and DukeMTMC-reID datasets.

Iteration V of Generating Hard Person Features. In Eq. (2), V is the iteration number of generating hard person features. To explore the impact of the iteration number V on the performance, we draw Top-1 accuracy and mAP curves by varying V from 2 to 14 with a step size of 2, as shown in Fig. 3. It demonstrates that both Top-1 accuracy and mAP increase significantly over the first 10 iterations and then decrease. We speculate that a smaller V will make the generated person features not hard enough, while a larger V will make the person features drift drastically and thus deviate from their original features. Therefore, choosing a suitable V is important for the overall performance, and we set $V = 10$.

Iteration Z of Generating Hard Positive Instances. In Eq. (4), Z is the iteration number of generating hard positive instances. To explore the impact of the iteration number Z on the performance, we report Top-1 accuracy and mAP by varying Z from 1 to 5 with a step size

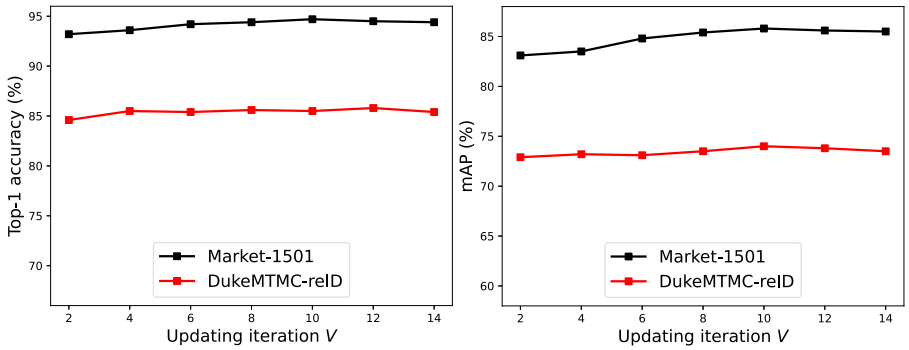

Fig. 3. Experiments on the impact of the iteration V of generating hard cluster prototypes on Market-1501 and DukeMTMC-reID datasets.

Table 2
Ablation study on individual components of our method on Market-1501 and DukeMTMC-reID datasets.

Module	Market		DukeMTMC-reID	
	Top-1	mAP	Top-1	mAP
Base.	88.3	75.8	78.5	64.3
Base.+distance	90.4	77.1	79.8	65.5
Base.+distance+HPI	91.9	80.2	81.8	68.5
Base.+RE	91.6	78.6	81.1	67.4
Base.+HPF	93.0	82.8	82.5	70.2
Base.+HPF+RE	94.2	84.2	83.7	72.0
Base.+RE+HPI	93.1	82.0	84.5	71.7
Base.+RE+(HPI+HPF)	**94.7**	**85.8**	**85.5**	**74.0**

Table 3
The impact of the hard positive person features $(\mathbf{c}^*_{y_i}, \{\mathbf{c}_a\}_{a\neq y_i})$ and the hard negative person features $(\mathbf{c}_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$ and both of them $(\mathbf{c}^*_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$ on Market-1501 and DukeMTMC-reID.

Prototype	Market		DukeMTMC-reID	
	Top-1	mAP	Top-1	mAP
$(\mathbf{c}^*_{y_i}, \{\mathbf{c}_a\}_{a\neq y_i})$	91.2	78.3	81.5	68.1
$(\mathbf{c}_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$	91.5	79.8	82.1	68.7
$(\mathbf{c}^*_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$	**93.0**	**82.8**	**82.5**	**70.2**

Fig. 4. Illustration of some easy and hard positive images to each query image. In particular, the query image, the easy positive image, and the hard positive image are shown on the left, middle, and right, respectively.

of 1 in Table 1, while $V = 10$. It can be observed that both Top-1 accuracy and mAP are highest when $Z = 3$. Therefore, the hard sample generation module with a suitable Z can generate reliable hard positive instances to train the discriminative feature learning module.

4.4. Ablation study

We conduct a series of ablative experiments to evaluate the contribution of each module in our ACFL. (1) "Base". includes the discriminative feature learning module to generate the discriminative features of each image and DBSCAN to estimate their pseudo labels, and the loss function $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C})$ for the feature learning process. (2) "Base.+distance" directly reduces the difference of each positive feature pair, *i.e.*, we utilize $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}) + \mathcal{L}_{\text{d}}(x_i, x_j)$ to train our ACFL framework, where $\mathcal{L}_{\text{d}}(x_i, x_j) = 2 * (1 - \mathrm{Q}(x_i) \cdot \mathrm{Q}(x_j))$. (3) "Base.+distance+HPI" incorporates the hard positive instance x^*_j generated by $\partial\mathcal{L}_{\text{d}}/\partial x_j$ into $\mathcal{L}_{\text{d}}$, *i.e.*, $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}) + \mathcal{L}_{\text{d}}(x_i, x^*_j)$, to optimize ACFL. (4) "Base.+RE" adopts $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}) + \mathcal{L}_{\text{ins}}(x_i, x_j)$ to train the ACFL framework. (5) "Base.+HPF" only exploits the generated hard person features $\mathbf{C}^*$ in $\mathcal{L}_{\text{class}}$, *i.e.*, $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}^*)$, to learn discriminative features for Re-ID. (6) "Base.+HPF+RE" uses $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}) + \mathcal{L}_{\text{ins}}(x_i, x^*_j)$ as the loss function. (7) "Base.+RE+HPI" utilizes $\mathcal{L}_{\text{class}}(\mathrm{Q}(x_i), \mathbf{C}) + \mathcal{L}_{\text{ins}}(x_i, x^*_j)$ to optimize ACFL. (8) "Base.+RE+(HPI+HPF)" is our full ACFL method, which incorporates KL, HPF, and HPI.

Feature Alignment. We adopt the relative entropy ("RE") loss in Eq. (7) to reduce the difference between a pair of positive samples with the same identity, the experimental results are presented in Table 2 "Base.+RE". In contrast, we also use the euclidean distance to measure the difference between features, which is reported in Table 2 "Base.+distance". The results show that "Base.+distance" improves the Top-1 accuracy and mAP by 2.1% and 1.3% on Market-1501 and 1.3% and 1.2% on DukeMTMC-reID compared to "Base"., while "Base.+RE" improves the Top-1 accuracy and mAP by 3.3% and 2.8% on Market-1501 and 2.6% and 3.1% on DukeMTMC-reID. We observe that "Base.+RE" outperforms "Base.+distance". We speculate the reason is aligning the high-dimension features (2048-dimension) in each positive pair is very hard, while the difficulty of aligning the N^t_c-dimension histogram distributions generated by v is reduced.

Hard Person Features. To verify the effectiveness of these generated hard person features ("HPF"), we report the comparison results in Table 2 "Base.+HPF". After incorporating the hard person features, both the Top-1 accuracy and mAP are increased by 4.7% and 7.0% on Market-1501 and 4.0% and 5.9% on DukeMTMC-reID, respectively. The performance improvements indicate that the hard person features can provide more effective pedestrian information for the feature learning process.

Besides, we also explore the impact of the hard positive person features $(\mathbf{c}^*_{y_i}, \{\mathbf{c}_a\}_{a\neq y_i})$ and the hard negative person features $(\mathbf{c}_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$ in $\mathcal{L}_{\text{class}}$. As presented in Table 3, $(\mathbf{c}^*_{y_i}, \{\mathbf{c}^*_a\}^*_{a\neq y_i})$ outperforms both $(\mathbf{c}^*_{y_i}, \{\mathbf{c}^*_a\}_{a\neq y_i})$ and $(\mathbf{c}_{y_i}, \{\mathbf{c}^*_a\}^*_{a\neq y_i})$. The reason is that more hard person features provide more informative cues during the feature learning process.

Hard Positive Instance. We can obtain hard positive instances by the hard sample generation module. We evaluate the effectiveness of the hard positive instances ("HPI"). The results are recorded in the third and sixth rows in Table 2 "Base.+RE+HPI". Compared with

Table 4
The difference between the transferred hard samples and the hard samples are mined by visual similarity on Market-1501 and DukeMTMC-reID.

Variants	Market-1501		DukeMTMC-reID	
	Top-1	mAP	Top-1	mAP
Easy	91.6	78.6	81.1	67.4
Mining	92.4	81.8	82.7	70.2
ACFL	**94.7**	**85.8**	**85.5**	**74.0**

"Base.+RE", we can observe that both Top-1 accuracy and mAP are improved by 1.5% and 3.4% on Market-1501 and 3.4% and 4.3% on DukeMTMC-reID by utilizing the hard positive instances, respectively. Moreover, we also evaluate the performance of inserting the hard positive instances into $\mathcal{L}_d$, the results are presented in Table 2 "Base.+distance+HPI". The Top-1 accuracy and mAP are increased by 1.5% and 3.1% on Market, and 2.0% and 3.0% on DukeMTMC-reID, compared to "Base.+distance". Compared to the original easy positive samples, the hard positive instances can provide more helpful instance-level information for the training process.

To intuitively show the effectiveness of the generated hard positive instances, we also show some visualization examples of the hard positive instances on the Market-1501 dataset in Fig. 4. For each given easy sample pair, the hard sample generation module can transfer the positive original samples (left) into their hard versions (right). Afterward, these generated hard positive instances and the generated hard person features are jointly used to learn discriminative features for person Re-ID.

Transferred v.s. Mined Hard Samples. To explore the difference between the transferred and mined hard samples, we replace the transferred hard samples with the mined hardest samples via visual similarity in our ACFL training process, called "Mining". The results are presented in Table 4 "Mining". Moreover, we also report the results of only using easy samples, which remove the hard sample generation stage, to optimize Q in Table 4 "Easy". We observe that the results determine the transferred hard samples ("ACFL") outperforms the hard samples mined by visual similarity ("Mining"). The reason is that the transferred hard samples can provide more discriminative information than easy samples in training and have higher confidence pseudo labels than the hard samples mined by visual similarity.

Overall. Our ACFL framework includes the above components (such as RE, HPF, and HPI). We report the performance presented in Table 2 "Base.+RE+(HPI+HPF)". Since "HPI" and "HPF" are generated by the HSG module, "(HPI+HPF)" can be considered as the HSG module. We could observe that Top-1 accuracy and mAP are improved by 6.4% and 10.0% on Market-1501 and 7.0% and 9.7% on DukeMTMC-reID, respectively. In summary, our contributions can improve the performance of person Re-ID significantly.

4.5. Comparison with state-of-the-art methods

We proceed to compare our proposed CFL framework with other state-of-the-art unsupervised person Re-ID methods on three popular benchmarks, *i.e.*, Market-1501 [17], DukeMTMC-reID [18], and MSMT17 [19] datasets, respectively. Table 5 presents the experimental results on them.

Comparison with unsupervised methods. We first compare our ACFL method to state-of-the-art unsupervised methods. Under the unsupervised setting, there are no manual annotations to supervise the feature learning process. Our ACFL mainly focuses on this setting. Table 5 "Unsupervised" presents the experimental results on Market-1501, DukeMTMC-reID and MSMT17 datasets. We can observe that our ACFL is better than all the other unsupervised methods on Market1501 and MSMT17 in all metrics, and higher Top-5, Top-10, and mAP on DukeMTMC-reID. The results of our ACFL on the CUHK03 labeled and detected dataset are presented in Table 8. Our ACFL can achieve better results than supervised methods, even without the complicated labeling process. The reason is that the pseudo labels estimated by our ACFL gradually converge to a reliable and stable state in the training process. The comprehensive comparison results on the four datasets validate the effectiveness of our method.

We speculate that the performance improvement lies in hard training samples that effectively enhance the discriminative capability of features. The hard person features generated by the hard samples generation module can provide more effective person-level information than the original person features. Moreover, the generated hard positive instance with the feature alignment strategy can provide more instance-level information to enhance the discriminative capability of features.

Comparison with UDA methods. We then compare our ACFL method with state-of-the-art UDA methods. The comparison results are shown in Table 5 "UDA". Under the UDA setting, these methods can make full use of the labeled source domain dataset, and thus usually can achieve better results than the unsupervised methods. These UDA methods are evaluated on Market-1501 with DukeMTMC-reID as the source dataset, on DukeMTMC-reID with Market-1501 as the source dataset, and on MSMT17 with Market-1501 as the source domain. Although our ACFL does not use extra labeled training data, it still outperforms the state-of-the-art UDA methods on Market-1501, DukeMTMC-reID, and MSMT17.

4.6. Results for ACFL with camera labels

Our proposed ACFL framework is also compatible with the camera labels. The camera version of ACFL is denoted as "ACFL-C". In this setting, we can provide extra camera labels for model training. To utilize these camera labels, we incorporate the cross-camera proxy contrastive loss $\mathcal{L}_{\text{cross}}$ based on the camera-aware memory proposed in ICE [33] into our ACFL, *i.e.*, $\mathcal{L}_{\text{a}}+\mu * \mathcal{L}_{\text{cross}}$, for the ACFL optimization procedure. We set $\mu = 0.5$ following ICE [33]. The experimental results of ACFL-C are shown in Table 6. Compared with ACFL, the performances are improved on Market-1501, DukeMTMC-reID, and MSMT17, especially on the MSMT17 with 15 cameras. It determines that camera labels can effectively address the feature variations caused by cameras shift, and the efficacy of our ACFL method and the camera labels.

4.7. Results for ACFL with different frameworks

With ViT. Without losing generality, we also evaluate the performance of the transformer-based network in our ACFL. Specifically, we replace ResNet50 [38] with ViT-S [44] as the feature extractor in the discriminative feature learning module, which is called "ACFL-VIT". The experimental results of ACFL-VIT are presented in Table 7. We can observe that "ACFL-VIT" outperforms the previous transformer-based unsupervised methods on Market-1501 and MSMT17, and achieves comparable or better performance than them on DukeMTMC-reID. These experimental results validate the effectiveness of our designed training strategy.

With Student-Teacher Framework. Moreover, we also evaluate our ACFL with the student-teacher framework in [49]. Specifically, we replace the ResNet50 with the student-teacher framework as a feature extractor in the discriminative feature learning module, which is called "ACFL-ST". The experimental results of ACFL-ST are presented in Table 9. We can observe that "ACFL-ST" outperforms the previous unsupervised method [49] on Market-1501 and MSMT17, and achieves higher Top-1, Top-5, and Top-10, but slightly lower mAP than Lan et al. [49] on DukeMTMC-reID. We speculate the reason is that the part-level information is more suitable for DukeMTMC-reID, the part-level information is adopted in [49] but is not used in our ACFL. The experimental results validate the effectiveness of our designed ACFL method.

Table 5
Performance comparison with state-of-the-art unsupervised and unsupervised domain adaptation (UDA) methods on Market-1501, DukeMTMC-reID and MSMT17.

	Method	Yenue	Market-1501				DukeMTMC-reID				MSMT17			
			Top-1	Top-5	Top-10	mAP	Top-1	Top-5	Top-10	mAP	Top-1	Top-5	Top-10	mAP
UDA	OPLG [39]	ICCV'21	91.5	95.9[b]	96.6[b]	80.0	82.2	91.3[b]	93.5[b]	70.1	54.9	69.6[b]	74.6[b]	28.4
	IDM [29]	ICCV'21	93.2	**97.5**	98.1	82.8	83.6	91.5	93.7	70.5	61.3	73.9	78.4	33.5
	TDRL [28]	ICCV'21	93.6	–	–	82.2	82.7	–	–	69.4	61.8	–	–	32.9
	SECRET [40]	AAAI'22	93.3	–	–	83.0	82.0	–	–	69.2	60.0	–	–	31.7
	MCRN [41]	AAAI'22	**93.8**	**97.5**	**98.5**	**83.8**	**84.5**	**91.7**	**93.8**	**71.5**	**64.4**	**75.1**	**79.2**	**32.8**
	MCN-MT [14]	ML'22	84.3	93.6	95.9	64.9	74.7	83.8	86.3	57.8	36.6[c]	47.1[c]	57.6[c]	15.4[c]
	Chen *et.al* [26]	PR'23	80.3	87.4	89.9	59.8	78.6	86.6	88.7	62.8	34.3	44.5	50.6	13.4
Unsupervised	SPCL [5]	NeurIPS'20	88.1	95.1	97.0	73.1	81.2	90.3	92.2	65.3	42.3	55.6	61.2	19.1
	RLCC [42]	CVPR'21	90.8	96.3	97.5	77.7	83.2	91.6	93.8	69.2	56.5	68.4	73.1	27.9
	ICE[a] [33]	ICCV'21	92.0	97.0	98.1	79.5	81.3	90.1	93.0	67.2	59.0	71.7	77.0	29.8
	CC [30]	ACCV'21	93.0	97.0	98.1	82.6	**85.7**	92.0	93.5	72.8	63.3	73.7	77.8	33.3
	MCRN [41]	AAAI'22	92.5	–	–	80.8	83.5	–	–	69.9	63.6	–	–	31.2
	SECRET [40]	AAAI'22	92.6	–	–	81.0	77.9	–	–	63.9	60.4	–	–	31.3
	PPLR [34]	CVPR'22	92.8	97.1	98.1	81.5	82.0[b]	90.5[b]	92.8[b]	69.8[b]	61.1	73.4	77.8	31.4
	ISE [35]	CVPR'22	94.3	98.0	98.8	85.3	85.8[b]	92.1[b]	93.6[b]	73.5[b]	67.6	77.5	81.0	37.0
	ACFL	–	**94.7**	**98.0**	**98.7**	**85.8**	85.5	**92.4**	**94.4**	**74.0**	**67.7**	**78.6**	**82.1**	**39.0**

[a] Means the camera-aware methods' agnostic version.
[b] Denotes the results by the authors' code.
[c] Denotes the results of our implementation.

Table 6
Performance comparison with state-of-the-art camera-aware methods on Market-1501, DukeMTMC-reID, and MSMT17.

Method	Yenue	Market-1501				DukeMTMC-reID				MSMT17			
		Top-1	Top-5	Top-10	mAP	Top-1	Top-5	Top-10	mAP	Top-1	Top-5	Top-10	mAP
IICS [31]	CVPR'21	88.8	95.3	96.9	72.1	76.9	86.1	89.8	59.1	45.7	57.7	62.8	18.6
CAP [32]	AAAI'21	91.4	96.3	97.7	79.2	81.1	89.3	91.8	67.3	67.4	78.0	81.4	36.9
MGH [43]	MM'21	93.2	96.8	98.1	81.7	83.7	92.1	93.7	70.2	70.2	81.2	84.5	40.6
ICE [33]	ICCV'21	94.1	97.7	98.3	83.0	83.3	91.5	94.1	69.9	70.9	81.0	84.5	39.4
PPLR [34]	CVPR'22	94.3	97.8	98.6	84.4	**86.7**[a]	92.1[a]	94.4[a]	74.3[a]	73.3	83.5	86.5	42.2
ACFL-C	–	**95.2**	**98.2**	**98.8**	**87.3**	86.4	**92.6**	**94.5**	**75.8**	**74.2**	**84.1**	**87.6**	**45.4**

[a] Denotes the results by the authors' code.

Table 7
Performance comparison with transformer-based unsupervised methods on Market-1501, DukeMTMC-reID, and MSMT17.

	Method	Venue	Top-1	Top-5	Top-10	mAP
Market-1501	PASS [44]	ECCV'22	94.9	**97.9**	**98.6**	88.5
	ACFL-VIT	–	**95.1**	97.1	**98.6**	**89.1**
DukeMTMC-reID	PASS [44][a]	ECCV'22	87.9	93.2	94.1	76.9
	ACFL-VIT	–	**88.1**	**93.6**	**94.1**	**77.6**
MSMT17	PASS [44]	ECCV'22	67.0	78.2	82.3	41.0
	ACFL-VIT	–	**70.1**	**80.3**	**83.7**	**45.7**

[a] Denotes the performance is evaluated based on the authors' code.

Table 8
Performance comparison with state-of-the-art unsupervised learning (USL) and supervised learning (SL) methods on CUHK03 labeled and detected datasets.

Methods	Yenue	Setting	CUHK03			
			Labeled		Detected	
			Top-1	mAP	Top-1	mAP
PPLR [34]	CVPR'22	USL	84.7[a]	69.1[a]	70.0[a]	64.3[a]
ISE [35]	CVPR'22		**76.5**[a]	71.8[a]	71.1[a]	67.5[a]
ACFL (Ours)	–		**76.5**	**73.2**	**71.9**	**68.5**
LightMBN [45]	ICIP'21	SL	**87.2**	**85.1**	**84.9**	**82.4**
MPN [46]	TPAMI'20		85.0	81.1	83.4	79.1
Pyramid [47]	CVPR'19		78.9	76.9	78.9	74.8
Auto-reid [22]	ICCV'19		77.9	73.0	73.3	69.3
MGN [48]	ACM MM'18		68.0	67.4	68.0	66.0

[a] Denotes the results by the authors' code.

Table 9
Performance comparison with teacher-student (ST) based unsupervised methods on Market-1501, DukeMTMC-reID, and MSMT17.

Dataset	Method	Yenue	Top-1	Top-5	Top-10	mAP
Market-1501	Lan et al. [49]	TIP'23	94.5	97.8	98.7	85.8
	ACFL-ST	–	**94.6**	**98.1**	**98.8**	**86.1**
DukeMTMC-reID	Lan et al. [49]	TIP'23	86.7	93.0	94.3	**76.2**
	ACFL-ST	–	**86.8**	**93.1**	**94.9**	76.0
MSMT17	Lan et al. [49]	TIP'23	67.9	78.0	81.6	39.5
	ACFL-ST	–	**68.8**	**79.2**	**83.0**	**40.3**

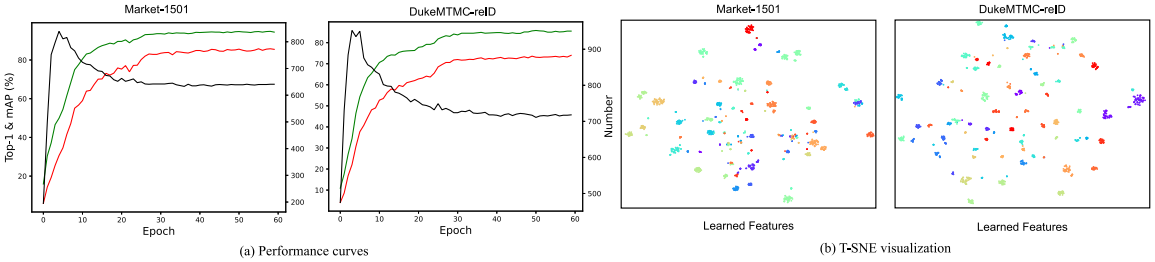

Fig. 5. Qualitative analysis. (a) The Top-1 (green), mAP (red), and the number of clusters (black) after each training epoch. (b) T-SNE visualization of the learned features on a part of Market-1501 (left) and DukeMTMC-reID (right) training sets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. Some retrieval results of our ACFL (top) and the baseline (bottom) on Market-1501 query and gallery sets. The green boxes present the gallery images have the same person identity as the query image, *i.e.*, correct matching, while red boxes denote the gallery images have the different person identities as the query image, *i.e.*, incorrect matching. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.8. Qualitative analysis

Visualization Results. To further analyze the effectiveness of our proposed CFL framework, we illustrate the Top-1 accuracy, mAP, the number of clusters, and outliers of each training epoch in Fig. 5(a). It indicates that continuous and gradual performance improvement follows the reduction of outliers. The performance and the number of clusters converge after optimization.

Moreover, we utilize t-SNE to visualize the learned features on 15 randomly selected identities on Market-1501 and DukeMTMC-reID training sets in Fig. 5(b). The points of the same color indicate these samples are of the same identity. It reveals that images with the same identity are almost always clustered together, while images with different identities tend to be separated. To intuitively observe the Re-ID results, we present the retrieval results between the baseline (bottom) and our ACFL (top) in Fig. 6, where the green boxes means the correct matching between the query and the images in gallery set and the red boxes is the incorrect matching between the query and gallery images. We can observe that our method obtains better retrieval results, especially for confusing samples. The results indicate that our method can learn discriminative features from the diversified unlabeled training images while avoiding the dilemma of determining the cluster number by utilizing the mined samples.

Complexity Analysis. Our ACFL contains a discriminative feature learning module and a hard sample generation module, and the hard sample generation module is leveraged to generate hard samples for supervising the ACFL training stage and excluded at the testing stage. To this end, we remain the feature extractor in the discriminative feature learning module to extract features from person images for Re-ID in the testing stage. Since we adopt ResNet [38] as the feature extractor in ACFL, its flops and parameters are 4.09G and 23.51M, respectively.

The hard sample generation module can be divided into hard person features and hard positive instances. In the hard person features generation stage, its flops are $K \times N_c^{(t)} \times D$, where D denotes the dimension of features, and the learnable parameters are $N_c^{(t)} \times D$. As for the hard positive samples, its flops is $V \times 3 \times H \times W$, where H and W are the height and width of each training sample, 3 is the channel (RGB) of the sample, the corresponding learnable parameters are $3 \times H \times W$.

Memory. We explore the effect of limited memory with different sizes. We denote the limited memory as $\mathbf{M}'$, where the size is $N' \times D$. We set $N' = \mu * N$, and $\mu \leq 1$. $\mathbf{M}'$ can be maintained by using the

Table 10
Experimental results about different sizes of memory on Market-1501 and DukeMTMC-reID.

μ	Dataset	Top-1	Top-5	Top-10	mAP
0.1	Market-1501	85.0	93.2	95.4	67.9
	DukeMTMC-reID	75.3	85.3	88.9	58.4
0.3	Market-1501	91.1	96.6	98.1	80.1
	DukeMTMC-reID	79.9	89.7	92.5	65.1
0.5	Market-1501	92.9	97.5	98.6	83.3
	DukeMTMC-reID	84.2	91.8	94.3	71.8
0.7	Market-1501	93.6	98.0	**98.9**	85.2
	DukeMTMC-reID	85.3	91.4	94.0	72.5
0.9	Market-1501	94.6	**98.0**	98.6	85.6
	DukeMTMC-reID	**85.5**	92.1	94.4	73.9
1.0	Market-1501	**94.7**	**98.0**	98.7	**85.8**
	DukeMTMC-reID	**85.5**	**92.4**	**94.4**	**74.0**

feature queue update strategy in MOCO [50]. Specially, we utilize the current features to replace the oldest features in $\mathbf{M}'$. The only caveat is that the samples in $\mathbf{M}'$ need to cover all persons in $\mathcal{X}$. The results are reported in Table 10. We can observe that the performances are similar when $\mu \leq 0.5$, while performance drops dramatically when $\mu < 0.5$. We speculate that fewer samples in a cluster cannot describe a person better.

5. Conclusion and future work

This paper proposes the adversarial contrastive feature learning (ACFL) framework for unsupervised person Re-ID. It mainly includes a discriminative feature learning module and a hard sample generation module. The ACFL framework aims to address the difficulty of existing methods that obtain hard training samples with high-confidence pseudo labels. Specifically, the discriminative feature learning module extracts features from unlabeled training data to support the feature learning process. Then, the high-confidence pseudo labels are estimated based on these extracted features. The hard sample generation module generates hard person features and hard positive instances based on these pseudo labels via an adversarial learning regime. Finally, these generated hard samples with the high-confidence pseudo labels of their original versions are used for training the discriminative feature learning module. Extensive experiments on Market-1501, DukeMTMC-reID, and MSMT17 datasets demonstrate the effectiveness of the proposed CFL framework for the unsupervised person Re-ID task.

We note that adopting larger V and Z in the adversarial learning regime will make the generated hard samples drift drastically and thus deviate from their original samples, *i.e.* identity confusion phenomenon. So, it must select suitable hyper-parameters to generate reliable hard samples to address the problem. Complex hyper-parameter selection is required in our ACFL. To this end, how to design a progressive adversarial learning strategy to adaptively address the identity confusion phenomenon will be our further focus.

Declaration of competing interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported partly by National Key R&D Program of China under Grant 2021YFB1714700, NSFC under Grants 62088102 and 62106192, Natural Science Foundation of Shaanxi Province, China under Grant 2022JC-41, China Postdoctoral Science Foundation, China under Grant 2020M683490, and Fundamental Research Funds for the Central Universities, China under Grant XTR042021005.

References

[1] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, Z. Wang, ABD-Net: Attentive but diverse Person re-identification, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2019, pp. 8350–8360.
[2] Z. Zhang, H. Zhang, S. Liu, Y. Xie, T.S. Durrani, Part-guided graph convolution networks for Person re-identification, Pattern Recognit. 120 (2021) 108155.
[3] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, Improving Person re-identification by attribute and identity learning, Pattern Recognit. 95 (2019) 151–161.
[4] Y. Lin, X. Dong, L. Zheng, Y. Yan, Y. Yang, A bottom-up clustering approach to unsupervised Person re-identification, in: Proc. AAAI. Conf. Artif. Intell., AAAI, 2019, pp. 8738–8745.
[5] Y. Ge, F. Zhu, D. Chen, R. Zhao, H. Li, Self-paced contrastive learning with hybrid memory for domain adaptive object Re-ID, in: Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2020, pp. 11309–11321.
[6] H. Fan, L. Zheng, C. Yan, Y. Yang, Unsupervised Person re-identification: Clustering and fine-tuning, ACM Trans. Multimedia Comput. Commun. Appl. 14 (4) (2018) 1–18.
[7] Y. Ge, D. Chen, H. Li, Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on Person re-identification, in: Proc. Int. Conf. Learn. Represent., ICLR, 2020.
[8] G. Wang, K. Wang, G. Wang, P.H. Torr, L. Lin, Solving inefficiency of self-supervised representation learning, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 9505–9515.
[9] S. Florian, K. Dmitry, P. James, FaceNet: A unified embedding for face recognition and clustering, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR, 2015, pp. 815–823.
[10] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, in: Proc. Int. Conf. Learn. Represent., ICLR, 2021.
[11] X. Xin, J. Wang, R. Xie, S. Zhou, W. Huang, N. Zheng, Semi-supervised Person re-identification using multi-view clustering, Pattern Recognit. 88 (2019) 285–297.
[12] Q. Yin, G. Ding, S. Gong, Z. Tang, et al., Multi-view label prediction for unsupervised learning Person re-identification, IEEE Signal Process. Lett. 28 (2021) 1390–1394.
[13] F. Ma, D. Meng, X. Dong, Y. Yang, Self-paced multi-view co-training, J. Mach. Learn. Res. 21 (57) (2020) 1–38.
[14] S. Xiang, Y. Fu, M. Guan, T. Liu, Learning from self-discrepancy via multiple co-teaching for cross-domain Person re-identification, Mach. Learn. 112 (2023) 1923–1940.
[15] W.Y. Lin, S.y. Liu, J.H. Lai, Y. Matsushita, Dimensionality's blessing: Clustering images by underlying distribution, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2018, pp. 5784–5793.
[16] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, V. Balntas, SOSNet: Second order similarity regularization for local descriptor learning, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2019, pp. 11016–11025.
[17] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable Person re-identification: A benchmark, in: Proc. IEEE Int. Conf. Comput. Vis., ICCV, 2015, pp. 1116–1124.
[18] E. Ristani, F. Solera, R. Zou, R. Cucchiara, C. Tomasi, Performance measures and a data set for multi-target, multi-camera tracking, in: Proc. Eur. Conf. Comput. Vis., ECCV, 2016, pp. 17–35.
[19] L. Wei, S. Zhang, W. Gao, Q. Tian, Person transfer GAN to bridge domain gap for Person re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2018, pp. 79–88.
[20] L. Wu, C. Shen, A. van den Hengel, PersonNet: Person re-identification with deep convolutional neural networks, 2016, arXiv:1601.07255.
[21] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, X. Xue, Multi-scale deep learning architectures for Person re-identification, in: Proc. IEEE Int. Conf. Comput. Vis., ICCV, 2017, pp. 5409–5418.
[22] R. Quan, X. Dong, Y. Wu, L. Zhu, Y. Yang, Auto-ReID: Searching for a part-aware convnet for Person re-identification, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2019, pp. 3749–3758.
[23] Z. Zhong, L. Zheng, S. Li, Y. Yang, Generalizing a Person retrieval model hetero-and homogeneously, in: Proc. Eur. Conf. Comput. Vis., ECCV, 2018, pp. 172–188.
[24] Z. Zhong, L. Zheng, Z. Luo, S. Li, Y. Yang, Invariance matters: Exemplar memory for domain adaptive Person re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2019, pp. 598–607.

[25] L. Song, C. Wang, L. Zhang, B. Du, Q. Zhang, C. Huang, X. Wang, Unsupervised domain adaptive re-identification: Theory and practice, Pattern Recognit. 102 (2020) 107173.
[26] F. Chen, N. Wang, J. Tang, P. Yan, J. Yu, Unsupervised Person re-identification via multi-domain joint learning, Pattern Recognit. 138 (2023) 109369.
[27] H. Li, Z. Kuang, Z. Yu, J. Luo, Structure alignment of attributes and visual features for cross-dataset Person re-identification, Pattern Recognit. 106 (2020) 107414.
[28] T. Isobe, D. Li, L. Tian, W. Chen, Y. Shan, S. Wang, Towards discriminative representation learning for unsupervised Person re-identification, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 8526–8536.
[29] Y. Dai, J. Liu, Y. Sun, Z. Tong, C. Zhang, L.-Y. Duan, IDM: An intermediate domain module for domain adaptive Person Re-ID, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 11844–11854.
[30] Z. Dai, G. Wang, W. Yuan, X. Liu, S. Zhu, P. Tan, Cluster contrast for unsupervised Person re-identification, in: Proc. Asian Int. Conf. Comput. Vis. ACCV, 2022, pp. 1142–1160.
[31] S. Xuan, S. Zhang, Intra-inter camera similarity for unsupervised Person re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2021, pp. 11926–11935.
[32] M. Wang, B. Lai, J. Huang, X. Gong, X.-S. Hua, Camera-aware proxies for unsupervised Person re-identification, in: Proc. AAAI. Conf. Artif. Intell., AAAI, 2021, pp. 2764–2772.
[33] H. Chen, B. Lagadec, F. Bremond, ICE: Inter-instance contrastive encoding for unsupervised Person re-identification, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 14960–14969.
[34] Y. Cho, W.J. Kim, S. Hong, S.-E. Yoon, Part-based pseudo label refinement for unsupervised Person re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2022, pp. 7369–7378.
[35] X. Zhang, D. Li, Z. Wang, J. Wang, E. Ding, J.Q. Shi, Z. Zhang, J. Wang, Implicit sample extension for unsupervised Person re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2022, pp. 7369–7378.
[36] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, in: Proc. Int. Conf. Learn. Represent., ICLR, 2017.
[37] W. Li, R. Zhao, T. Xiao, X. Wang, DeepReID: Deep filter pairing neural network for Person re-identification, 2014, pp. 152–159.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., CVPR, 2016, pp. 770–778.
[39] Y. Zheng, S. Tang, G. Teng, Y. Ge, K. Liu, J. Qin, D. Qi, D. Chen, Online pseudo label generation by hierarchical cluster dynamics for adaptive Person re-identification, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., ICCV, 2021, pp. 8371–8381.
[40] T. He, L. Shen, Z. Guo, Y. Guo, G. Ding, SECRET: Self-consistent pseudo label refinement for unsupervised domain adaptive Person re-identification, in: Proc. AAAI. Conf. Artif. Intell., AAAI, 2022, pp. 879–887.
[41] Y. Wu, T. Huang, H. Yao, C. Zhang, Y. Shao, C. Han, C. Gao, N. Sang, Multi-centroid representation network for domain adaptive Person Re-ID, in: Proc. AAAI. Conf. Artif. Intell., AAAI, 2022, pp. 2750–2758.
[42] X. Zhang, Y. Ge, Y. Qiao, H. Li, Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2021, pp. 3436–3445.
[43] Y. Wu, X. Wu, X. Li, J. Tian, MGH: Metadata guided hypergraph modeling for unsupervised Person re-identification, in: Proc. ACM Int. Conf. Multimedia, ACM MM, 2021, pp. 1571–1580.
[44] K. Zhu, H. Guo, T. Yan, Y. Zhu, J. Wang, M. Tang, PASS: Part-aware self-supervised pre-training for Person re-identification, in: Proc. Eur. Conf. Comput. Vis., ECCV, 2022, pp. 198–214.
[45] F. Herzog, X. Ji, T. Teepe, S. Hörmann, J. Gilg, G. Rigoll, Lightweight multi-branch network for Person re-identification, in: Proc. IEEE Int. Conf. Image Process., ICIP, 2021, pp. 1129–1133.
[46] C. Ding, K. Wang, P. Wang, D. Tao, Multi-task learning with coarse priors for robust part-aware Person re-identification, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1474–1488.
[47] F. Zheng, C. Deng, X. Sun, X. Jiang, X. Guo, Z. Yu, F. Huang, R. Ji, Pyramidal Person re-identification via multi-loss dynamic training, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2019, pp. 8514–8522.
[48] G. Wang, Y. Yuan, X. Chen, J. Li, X. Zhou, Learning discriminative features with multiple granularities for Person re-identification, in: Proc. ACM Int. Conf. Multimedia, ACM MM, 2018, pp. 274–282.
[49] L. Lan, X. Teng, J. Zhang, X. Zhang, D. Tao, Learning to purification for unsupervised Person re-identification, IEEE Trans. Image Process. (2023).
[50] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., CVPR, 2020, pp. 9726–9735.