Implicit Sample Extension for Unsupervised Person Re-Identification

Xinyu Zhang[1*], Dongdong Li[3,1*], Zhigang Wang[1], Jian Wang[1], Errui Ding[1],
Javen Qinfeng Shi[2], Zhaoxiang Zhang[3,4], Jingdong Wang[1†]

[1]Baidu VIS, China [2]The University of Adelaide, Australia [3]Institute of Automation, CAS & UCAS, China
[4] Centre for Artificial Intelligence and Robotics, HKISI_CAS, China

{zhangxinyu14,wangzhigang05,wangjian33,dingerrui,wangjingdong}@baidu.com
{lidongdong2019,zhaoxiang.zhang}@ia.ac.cn, javen.shi@adelaide.edu.au

Abstract

Most existing unsupervised person re-identification (Re-ID) methods use clustering to generate pseudo labels for model training. Unfortunately, clustering sometimes mixes different true identities together or splits the same identity into two or more sub clusters. Training on these noisy clusters substantially hampers the Re-ID accuracy. Due to the limited samples in each identity, we suppose there may lack some underlying information to well reveal the accurate clusters. To discover these information, we propose an Implicit Sample Extension (ISE) method to generate what we call support samples around the cluster boundaries. Specifically, we generate support samples from actual samples and their neighbouring clusters in the embedding space through a progressive linear interpolation (PLI) strategy. PLI controls the generation with two critical factors, i.e., 1) the direction from the actual sample towards its K-nearest clusters and 2) the degree for mixing up the context information from the K-nearest clusters. Meanwhile, given the support samples, ISE further uses a label-preserving loss to pull them towards their corresponding actual samples, so as to compact each cluster. Consequently, ISE reduces the "sub and mixed" clustering errors, thus improving the Re-ID performance. Extensive experiments demonstrate that the proposed method is effective and achieves state-of-the-art performance for unsupervised person Re-ID. Code is available at: https://github.com/PaddlePaddle/PaddleClas.

1. Introduction

Unsupervised person re-identification (Re-ID) aims to learn person appearance features without annotations. It gains increasing attention and popularity due to its wide
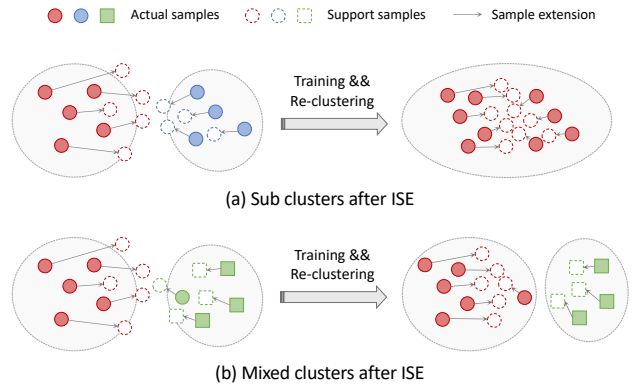
Figure 1. Clustering behaviour changes with our implicit sample extension (ISE). Different shapes represent different ground-truth identities, while different colors stand for different pseudo labels (best viewed in color). Usually, clustering may produce (a) **sub clusters** that samples share the same true identity but are split apart as two or more clusters; and (b) **mixed clusters** that samples with different identities are mixed to the same cluster. Our support samples help to merge the sub clusters as one and split the mixed clusters apart, thus improving the performance.

practical applications in the real world. One kind of works [2, 8, 13, 25, 43, 44, 49] attempt to transfer knowledge from existing labeled data to unlabeled target data, known as unsupervised domain adaptation (UDA). Another kind of works rely on unsupervised learning (USL) to learn representations purely from unlabeled images, which is more data-friendly than UDA. Most USL Re-ID methods [4,9,14] follow an iterative two-stage training procedure: 1) using clustering [12, 22] to generate pseudo labels as the supervision for the next step; 2) training the Re-ID model using the pseudo labels. Here we focus on investigating the USL Re-ID, and follow the two-stage pipeline.

Clustering is critical in the aforementioned USL and has attracted great attention from existing USL methods. For instance, ACT [38] utilizes two models that try to refine pseudo labels for each other. SpCL [14] employs a self-

paced learning to gradually generate more reliable clusters for training. Despite the progresses, these methods still suffer from noisy pseudo labels, especially the *sub* and *mixed* clusters shown in the left part of Figure 1. Based on this observation, we suppose that some underlying information may be lost in the existing data distribution due to the limited samples in each identity. If we can compensate for the missing information, the clustering quality can be improved in the next step. For example, if there exists intermediate variations among two sub clusters with the same identity, they can be correctly merged into one cluster.

To achieve this goal, we propose an implicit sample extension (ISE) method, progressively synthesizing supplementary samples to improve the context representation for each cluster. We name these supplementary samples as *support samples*, since they often patrol around the cluster boundaries. Inspired by [3, 28, 32] that deep features are usually linearized, we propose a progressive linear interpolation operation (PLI) to guide the generation of support samples with two factors, *i.e.*, *direction* and *degree*. Specifically, the direction factor controls that the generation of support samples is from a actual sample to its K-nearest neighbor clusters in the embedding space. Support samples in these directions are more meaningful, since neighbor clusters are more likely to have the above clustering problems. The degree factor decides how much context information from K-nearest clusters should be incorporated by support samples. We increases the degree progressively to fill the cluster gap with farther and farther support samples. This strategy can prevent the training collapse caused by aggressive support samples, especially in the early training stages. Except for PLI, we also propose a label-preserving (LP) contrastive loss to enforce support samples close to their originals to compact each cluster. After the sample generation, we regard support samples as actual ones to let them participate in the model training.

We observe that with the help of support samples, the data distribution is refined, and spurious clustering behaviours, including the sub and mixed clusters, can be alleviated (see experiments for more details). In addition, different from GAN (General Adversarial Network) based methods [11, 51], ISE is parameter-free and performs the sample generation implicitly in the embedding space instead of explicitly in the image pixel space. Therefore, the support samples can be readily utilized in loss functions without any feature extraction procedure. These merits make the proposed ISE a very efficient method. In summary, our contributions are as follows:

- We propose a novel implicit sample extension (ISE) method for USL person Re-ID. The generated support samples from ISE provide complementary information, which can nicely handle the issues of sub and mixed clustering errors;

- We present a novel progressive linear interpolation (PLI) strategy and a label-preserving contrastive loss (LP) for the support sample generation;

- We conduct comprehensive experiments and analyses to show the effectiveness of ISE, and ISE outperforms the current state-of-the-arts by a large margin.

2. Related Work

Unsupervised Person Re-ID. Unsupervised person Re-ID methods are generally classified into two categories. The first is the unsupervised domain adaptation (UDA) methods, which transfers knowledge from labeled source data to unlabeled target data. These methods still need to annotate partial data. Another one is the purely unsupervised learning (USL) Re-ID without any identity annotation that is more data-friendly. In this paper, we focus on the USL Re-ID. Most recent works focus on pseudo label generation and framework design. BUC [20] presents a bottom-up clustering scheme to gradually merge single sample into bigger clusters. MMCL [30] proposes a multi-label classification regime to construct relationships among individual images. SpCL [14] employs self-paced learning to gradually generate more reliable clusters. Both CycAs [33] and TSSL [36] use tracklet information to predict pseudo labels more correctly. SoftSim [21] directly uses similarity-based soft labels to train the Re-ID model. More recently, ICE [4] leverages inter-instance pairwise similarity to enhance the contrastive learning. Cluster-Contrast [9] stores features and computes contrastive loss in the cluster level to update each cluster consistently. Different from these works, our method emphasises on the issues of sub and mixed clusters, and aims to compensate for some missing information to refine cluster results and improve the performance.

Image Generation. Image generation is a common strategy to enrich limited data. Mixup [42] and cutmix [40] directly fuses two images or image patches as data augmentations. Viewmaker [27] yields diverse views by adding learned perturbations. SPGAN [11] proposes a similarity-preserved GAN for style transfer between two domains. DGNet++ [51] disentangles structures and appearances of person images, and re-fuses them to reconstruct the new ones. Chen *et al*. [5] use 3D meshes to synthesize more views of an person image to learn view-invariant features. Different from these works, our method is feature-level, parameter-free and can be used to alleviate the inaccurate clustering results.

3. Method

In this section, we first set up a general formalization of existing clustering-based USL Re-ID methods, and introduce our baseline. We then elaborate the details of our implicit sample extension (ISE) method, including the progressive linear interpolation strategy (PLI) and the label-

(a) Overall implicit sample extension pipeline (ISE)   (b) Progressive Linear Interpolation (PLI)
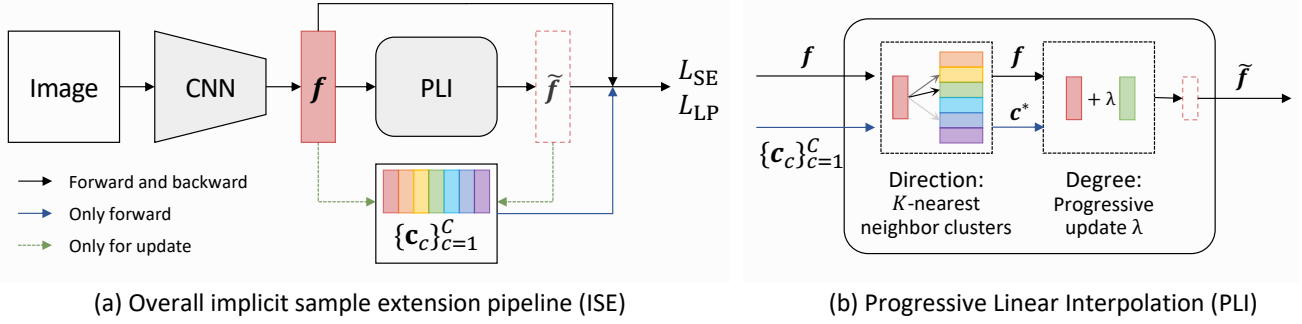
Figure 2. Overview of (a) the overall implicit sample extension pipeline (ISE) and the details of (b) the progressive linear interpolation (PLI) strategy. For a specific sample feature $\boldsymbol{f}$, we apply PLI to generate the support sample $\widetilde{\boldsymbol{f}}$, which are used to optimize the model with the sample extension loss L_{SE} and the label-preserving loss L_{LP}. As for the generation process, PLI finds the K-nearest cluster centroids $\boldsymbol{c}^*$ as the generation *direction*, and adopts the progressively updated λ to control the generation *degree*. More details are in Algorithm 1.

preserving scheme (LP).

3.1. Preliminary

Let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ denote an unlabeled Re-ID dataset, where x_i is the i-th image and N is the total image number. In the USL Re-ID task, we aim to train a deep neural network $f_\theta(\cdot)$ to project an image from the data point space $\mathcal{X}$ into an embedding space $\mathcal{F}$. Given a specific sample x, $f_\theta(x)$ is the extracted feature from the model. Here we omit the model parameters θ and employ $\boldsymbol{f} = f(x)$ as the feature embedding. $\boldsymbol{f} \in \mathbb{R}^d$ is a d-dimensional vector.

Recent clustering-based USL Re-ID methods [4, 8, 9, 13, 14, 30, 31, 43, 50] utilize K-means [22] or DB-SCAN [12] to assign pseudo labels for unlabeled samples. After then, we obtain a "labeled" dataset $\mathcal{X}' = \{(x_1, y_1), (x_2, y_2), ..., (x_{N'}, y_{N'})\}$, where $y_i \in \{1, ..., C\}$ is the pseudo label of the i-th selected image and C is the total cluster number. Here are N' images with valid pseudo labels, while others are ignored. With $\mathcal{X}'$, we can optimize the model under the form of the supervised learning. Items in the same cluster are considered as the same person, serving as positive samples. Others in different clusters are negatives. Based on this, existing methods [9,14,31,50] employ the InfoNCE loss function [23] for the model training. Despite the diverse variants in different approaches, we summarize the loss functions as a general formulation:

$$L_{\mathrm{info}} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{f} \cdot \boldsymbol{m}_+)/\tau)}{\sum_{m=1}^{M} \exp(\mathrm{sim}(\boldsymbol{f} \cdot \boldsymbol{m}_m)/\tau)}, \quad (1)$$

where $\boldsymbol{m}_m$ is the m-th entry of the memory bank $\mathbf{M}$, and $\boldsymbol{m}_+$ shares the same pseudo label with $\boldsymbol{f}$. $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ denotes the cosine similarity between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Some methods [14,50] construct the memory bank $\mathbf{M} \in \mathbb{R}^{d \times N}$ with the instance (*aka.* sample) feature. In this way, M can be the whole sample number N or the mini-batch size B. Other methods [9, 31] alternate

to form the memory bank $\mathbf{M} \in \mathbb{R}^{d \times C}$ with the cluster centroids. Here M is equal to the cluster number C.

Meanwhile, these methods also design various strategies for the memory update. In general, they adopt the momentum update strategy, which is formulated as follows:

$$\boldsymbol{m} \leftarrow \mu \cdot \boldsymbol{m} + (1 - \mu) \cdot \boldsymbol{f}, \quad (2)$$

where $\boldsymbol{f}$ is the encoded feature that has the same identity with the memory entry $\boldsymbol{m}$. Specifically, $\boldsymbol{f}$ is utilized to update its own instance entry in [14, 50], while to update its own cluster entry in [9, 31]. Cluster-Contrast [9] further uses the batch hardest sample for the memory update[1].

In this paper, we follow Cluster-Contrast [9] to establish our *baseline*, in which the memory bank is constructed by the cluster centroids and the hardest/every sample in one mini-batch is used for the memory updating. The detailed training settings and the differences with Cluster-Contrast are described in Sec. 4.2.

3.2. Progressive Linear Interpolation

However, the above clustering-based USL person Re-ID methods still suffer from sub clusters and mixed clusters. We suppose that the underlying missing information in the embedding space causes inaccurate clusters. Our hypothesis lies in that the clustering quality can be improved in the next step by involving richer information as described in Sec.1. Based on this motivation, we present a *progressive linear interpolation* (PLI) method as follows.

Given a sample feature $\boldsymbol{f}$, we generate its corresponding support sample $\widetilde{\boldsymbol{f}}$ by a linear interpolation operation:

$$\widetilde{\boldsymbol{f}} = \boldsymbol{f} + \lambda \Delta \boldsymbol{f}, \quad (3)$$

where $\Delta \boldsymbol{f}$ controls the *direction* and λ controls the *degree*, which are two important factors during the sample extension. From the perspective of problematic clusters, the ideal

---

[1] [9] changes to use every sample in the mini-batch for the momentum updating in the updated version.

support samples are expected to be informative to distribute around the decision boundary. Thus we design the following direction and degree for the sample generation.

Direction. Neighbor clusters are more likely to have mis-classified samples. We thus aim to generate support samples distributed among two similar clusters. Specifically, we first find K-nearest neighbor clusters for $\boldsymbol{f}$ based on the cosine similarities, which are calculated with $\boldsymbol{f}$ and the whole memory bank $\mathbf{M} \in \mathbb{R}^{d \times C}$. Then the directions of support samples are established as vectors from the cluster centroid $\boldsymbol{c}$ of $\boldsymbol{f}$ to the K-nearest ones. Here, we set $K = 1$ as an example and denote the 1-nearest cluster centroid as $\boldsymbol{c}^*$. The direction $\Delta \boldsymbol{f}$ is thus formulated as follows:

$$\Delta \boldsymbol{f} = \frac{1}{2}(\boldsymbol{c}^* - \boldsymbol{c}). \tag{4}$$

In this way, the generated support samples have rich context semantics from both itself and its neighbor clusters, which are informative as the complements. The direction thus controls *what* information is worth compensating for. For $K > 1$, each neighbor cluster centroid decides a direction factor. We generate a respective support sample for each direction, and there are total K support samples for $\boldsymbol{f}$.

Degree. In addition, the degree is also essential in the sample extension. The degree decides *how much* information of neighbor clusters should be incorporated in support samples. Too large λ makes support samples far from their original clusters, misleading the model training. Too small λ inversely generates almost useless support samples since they are similar to the originals. Therefore, we perform a *progressive* update on λ to gradually involve more context information as the training goes on. Specifically, we set λ as a variable of the training iteration, which increases logarithmically over the training proceeds:

$$\lambda = \frac{\lambda_0}{2} \log(\frac{e - 1}{T} \cdot t + 1), \tag{5}$$

where t/T represents the current/total iteration. λ_0 is the base degree. In the early stages, Eq. 5 enforces support samples near to its original sample to avoid introducing noise when features are not good enough. As the training goes on, feature representations become more robust. In that time, farther support samples, involving richer contexts from neighbor clusters, are necessary to include harder underlying information and augment the data distribution.

After the sample extension, support samples can be regarded as actual features for the model optimization. We then modify the InfoNCE loss (Eq. 1) as follows:

$$L_{\text{SE}} = -\log \frac{\exp(\text{sim}(\widehat{\boldsymbol{f}} \cdot \boldsymbol{c})/\tau_1)}{\sum_{c=1}^{C} \exp(\text{sim}(\widehat{\boldsymbol{f}} \cdot \boldsymbol{c}_c)/\tau_1)}, \tag{6}$$

where $\widehat{\boldsymbol{f}} = \{\boldsymbol{f}; \widetilde{\boldsymbol{f}}\}$, both original samples and support samples contribute to the model optimization. $\boldsymbol{c}_c$ is the c-th clus-

---

Algorithm 1: ISE in one epoch

Require: network f_θ; cluster number C; iteration number from T_0 to T_t in one epoch; batch size B; cluster number in a mini-batch C_B.

Input : training dataset with pseudo labels $\mathcal{X}' = \{(x_i, y_i)\}_{i=1}^{N'}$, where $y_i \in \{1, \ldots, C\}$; cluster centroids $\{\boldsymbol{c}_c\}_{c=1}^{C}$.

1 **for** $t = T_0$ **to** T_t **do**
2   Sample a mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}, y_i \in \{c_i\}_{i=1}^{C_B}$;
3   Extract embedding features $\boldsymbol{F} = \{\boldsymbol{f}_i\}_{i=1}^{B}$ on $\mathcal{B}$;
4   Define the generation direction by PLI according to Eq. 4;
5   Adjust the generation degree by PLI according to Eq. 5;
6   Generate support samples $\{\widetilde{\boldsymbol{f}}_i\}_{i=1}^{B}$ according to Eq. 3;
7   Calculate L_{SE} and L_{LP} according to Eq. 6 and Eq. 7;
8   Update cluster centroids $\{\boldsymbol{c}_i\}_{i=1}^{C_B}$;
9   Update the network parameter θ;
10 **end**

Output: updated network f_θ and cluster centroids $\{\boldsymbol{c}_c\}_{c=1}^{C}$.

---

ter centroid. τ_1 is a temperature hyper-parameter. At the same time, we utilize $\widehat{\boldsymbol{f}}$ to update the corresponding cluster centroid $\boldsymbol{c}$ in Eq. 2. Here $\boldsymbol{m}_c$ stores the cluster centroid $\boldsymbol{c}_c$. In this circumstance, each cluster can tune its representations by considering more context information, other than directly fitting to the inaccurate clustering results.

3.3. Label-Preserving Scheme

The generated support samples are helpful to extend cluster boundaries by including abundant context information. However, these support samples are not expected to change so much that they will become outliers of the original clusters. Except for the generation degree of support samples, we also propose a label-preserving (LP) loss to enforce them to be close to their original samples:

$$L_{\text{LP}} = -\log \frac{\exp(\text{sim}(\boldsymbol{f} \cdot \widetilde{\boldsymbol{f}}^+)/\tau_2)}{\sum_{c=1}^{C_B} \exp(\text{sim}(\boldsymbol{f} \cdot \widetilde{\boldsymbol{f}}_c^-)/\tau_2)}, \tag{7}$$

where C_B is the cluster number of a mini-batch. $\widetilde{\boldsymbol{f}}^+$ is the hardest positive support sample for a given $\boldsymbol{f}$, and $\widetilde{\boldsymbol{f}}_c^-$ is the hardest negative one in the c-th cluster of a mini-batch. τ_2 is a temperature hyper-parameter in Eq. 7.

With the proposed label-preserving loss, support samples are constrained to be similar to their own cluster members, and dissimilar to samples from other clusters.

3.4. Implicit Sample Extension

We name our whole method as implicit sample extension (ISE), in which support samples are generated implicitly in the embedding space and dynamically changed on-the-fly. Overall, we first generate support samples via the progressive linear interpolation method, and then utilize them as actual samples to help the model optimization. Our final loss function is as follows:

$$L = L_{\text{SE}} + \beta L_{\text{LP}}, \tag{8}$$

where β is the loss weight, balancing two loss functions L_{SE} and L_{LP}. The details of ISE are in Algorithm 1.

4. Experiments

4.1. Datasets and Evaluation Protocol

Datasets. We evaluate our proposed method on Market-1501 [45] and MSMT17 [35]. Market-1501 includes 32,668 images of 1,501 identities with 6 camera views. There are 12,936 images of 751 identities for training and 19,732 images of 750 identities for test. MSMT17 contains 126,441 images from 4,101 identities captured by 15 cameras. The training set has 32,621 images of 1,041 identities and the test set has 93,820 images of 3,060 identities.

Evaluation protocol. We use the cumulative match characteristic (CMC) curve [15] and the mean average precision (mAP) [1] as the evaluation metrics. There is no post-processing operations in testing, *e.g.*, reranking [48].

4.2. Implementation Details

The backbone of the Re-ID model is ResNet-50 [16] pretrained on ImageNet [10]. The model modification follows Cluster-Contrast [9]. The size of the input image is 256×128 and the batch size is 256. At the beginning of each epoch, we perform the DBSCAN clustering to generate pseudo labels. The maximum distance between two samples in DBSCAN is set to 0.4 in Market-1501 and 0.7 in MSMT17. We initial the memory bank $\mathbf{M}$ with cluster centroids and re-initialize it at the start of each epoch. The memory update rate μ is empirically set to 0.2 in Market-1501 and 0.1 in MSMT17. The loss weight β is set to 0.1. The temperature factors τ_1 and τ_2 are 0.05 and 0.6, respectively. We adopt Adam [18] optimizer, and the weight decay is 5e-4. The initial learning rate is 3.5e-4, and the base degree λ_0 is 1.0. For Market-1501, the total epoch is 70, and the learning rate is multiplied by 0.1 after every 30 epochs. For MSMT17, we divide the learning rate by 10 after every 20 epochs and the total epoch is 50. Our baseline is based on the Cluster-Contrast [9], which achieves great performance. Differently, we utilize the hardest sample in one mini-batch to update the memory bank on Market-1501, while using all batch samples on MSMT17[2].

4.3. Ablation Study

In this subsection, we thoroughly analyze the effectiveness of each component in our ISE, and also illustrate the impact of ISE on the clustering quality.

---

[2]This setting is based on our re-implementation of Cluster-Contrast [9]. In fact, we find that the baseline performance drops a lot when directly using the hardest sample for memory update on MSMT17. The setting of the hardest sample for memory updating only achieves 19.5%/41.2% mAP/top-1, which largely lower than that of the all batch samples for memory updating (30.1%/58.6%).

Method	No.	Components		Market-1501		MSMT17	
		PLI	LP	mAP	top-1	mAP	top-1
Baseline	No.1	-	-	82.5	92.5	30.1	58.6
ISE	No.2	λ	-	83.9	93.9	33.5	63.9
	No.3	-	✓	83.6	92.7	31.4	59.9
	No.4	λ_0	✓	83.3	93.4	33.8	**64.8**
	No.5	λ	✓	**84.7**	**94.0**	**35.0**	64.7

Table 1. The effectiveness of each component in our implicit sample extension (ISE). ISE includes the progressive linear interpolation (PLI) and the label-preserving scheme (LP). λ_0 is the constant base degree in PLI, and λ is the progressively updated one.
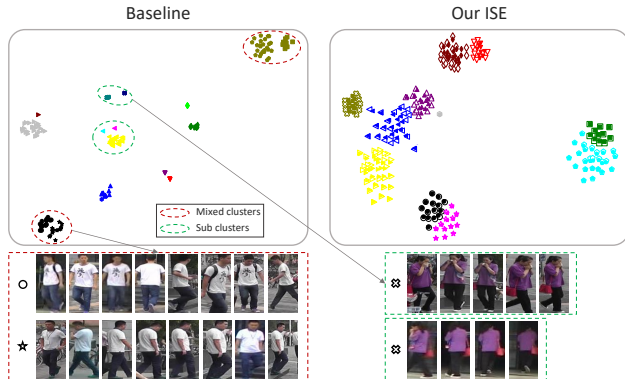

Figure 3. The t-SNE [29] visualization of clustering results of baseline and our ISE on same samples and epochs. Except for real samples, we draw the generated support samples (the hollow shapes), to better show the effectiveness of ISE. Different shapes and colors represent diffident ground-truth identities and pseudo labels, respectively. All inaccurate clusters of these samples are corrected by ISE. (Best viewed in color and magnification.)

Effectiveness of the progressive linear interpolation (PLI). In Table 1, the efficacy of our PLI strategy is revealed when comparing (No.1 *vs.* No.2) and (No.4 *vs.* No.5). With the generated support samples from PLI, the performance largely outperforms the baseline, especially +1.4%/+1.4% mAP/top-1 improvement on Market-1501 and +3.4%/+5.3% mAP/top-1 improvement on MSMT17. It clearly shows that our PLI is beneficial for improving the feature representation. We also verify the effectiveness of the progressive update scheme on λ (*i.e.*, Eq. 5). When comparing No.5 with No.4, this scheme improves the performance by +1.4% and +1.2% mAP on Market-1501 and MSMT17, showing that gradually farther support samples are necessary to improve the model performance.

Effectiveness of the label-preserving scheme (LP). The results of (No.1 *vs.* No.3) and (No.2 *vs.* No.5) demonstrate the efficacy of LP. The No.3 denotes that we apply LP on actual samples. In this way, $\widetilde{\boldsymbol{f}}^+$ is reduced to $\boldsymbol{f}^+$ and $\widetilde{\boldsymbol{f}}_c^-$ is reduced to $\boldsymbol{f}_c^-$ in Eq. 7, representing that positive and negative samples are chosen from the actual samples in a mini-batch. Compared with baseline, the results of No.3 show
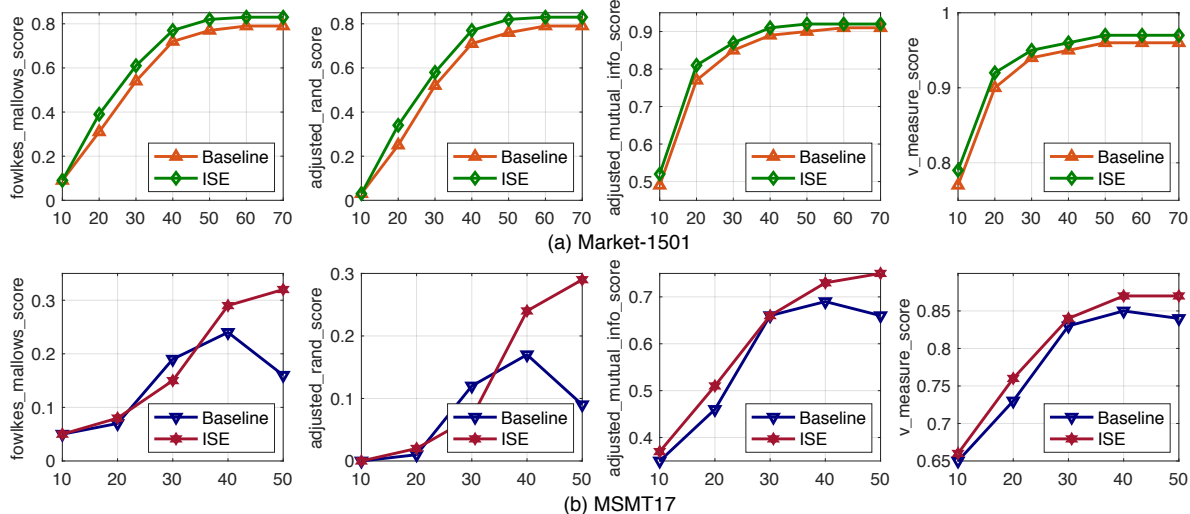
Figure 4. Clustering quality over different epochs from the baseline and our ISE on (a) Market-1501 and (b) MSMT17 datasets.

Method	Type of λ	Market-1501		MSMT17	
		mAP	top-1	mAP	top-1
Baseline	-	82.5	92.5	30.1	58.6
ISE	Constant	83.3	93.4	33.8	**64.8**
	Square	84.0	93.1	32.4	61.0
	Linear	84.6	93.7	33.8	63.0
	Logarithm	**84.7**	**94.0**	**35.0**	64.7

Table 2. Comparison with different types of degree λ in PLI

Method	Direction	Base degree λ_0	Market-1501		MSMT17	
			mAP	top-1	mAP	top-1
Baseline	-	-	82.5	92.5	30.1	58.6
ISE	Random	1.0	83.8	92.7	28.3	56.5
	Farthest		83.8	93.1	27.8	56.2
	Nearest		**84.7**	**94.0**	**35.0**	**64.7**
	Nearest	0.1	83.2	92.4	31.2	60.0
		0.5	83.6	93.1	33.0	62.3
		1.0	**84.7**	**94.0**	**35.0**	**64.7**
		2.0	84.2	93.6	33.0	64.0

Table 3. Direction and degree of sample extension in ISE

that LP leads to +1.1% and +1.3% mAP improvement on Market-1501 and MSMT17, indicating LP can improve the feature representation even on the actual samples. When applying LP on support samples (No.2 vs. No.5), the performance improves from 83.9% to 84.7% mAP on Market-1501 and 33.5% to 35.0% mAP on MSMT17.

When combining PLI and LP together, our ISE achieves +2.2%/+1.5% and +4.9%/+6.1% mAP/top-1 improvements on Market-1501 and MSMT17 compared to the baseline. We believe that our ISE indeed generates informative support samples as complements for the data distribution, which is beneficial to improve the model discrimination.

Clustering quality. We intuitively visualize the behavior of support samples. As shown in Figure 3, the proposed ISE effectively improves the clustering quality, in which sub clusters and mixed clusters are well alleviated. We also evaluate the clustering quality over different training epochs on (a) Market-1501 and (b) MSMT17 datasets in Figure 4. We employ four metrics from [7][3], including fowlkes_mallows_score, adjusted_rand_score, adjusted_mutual_info_score and v_measure_score. The larger score represents the better result in all metrics. It shows that our ISE achieves superior performance than baseline on all metrics. In addition, the clustering quality of our ISE progressively improves along the training process, while it

---

[3] scikit-learn.clustering-performance-evaluation

may get worse in the baseline. We believe that the generated support samples can help refine the data distribution in the embedding space, and thus improve the clustering quality.

4.4. Parameter Analysis

Progressive update type of degree λ in PLI. As shown in Eq. 5, we let the degree λ increase logarithmically as the training goes on. In addition to the logarithm type, λ can be set as various forms. We explore four types, *i.e.*, constant ($\lambda = \frac{\lambda_0}{2}$), linear ($\lambda = \frac{\lambda_0 \cdot t}{2T}$), square ($\lambda = \frac{\lambda_0 \cdot t^2}{2T^2}$) and logarithm (Eq. (5)). Here we set $\lambda_0 = 1.0$ for all types. In Table 2, ISE consistently surpasses the baseline whatever the update type of λ is. By comparison, the performance of progressive types (*e.g.*, linear, logarithm) are usually better than the constant one. It verifies the assumption that aggressive support sample generation in the early stage may hamper the model optimization. Table 2 also shows that the logarithm type obtains the best results on both Market-1501 and MSMT17. Without specification, we use the logarithm as the default update type for λ in all other experiments.

Direction and degree in PLI. The generation direction and degree are two elements during the sample extension in our PLI. In this subsection, we first verify the necessity of the

Method	K-nearest clusters	Market-1501		MSMT17	
		mAP	top-1	mAP	top-1
Baseline	-	82.5	92.5	30.1	58.6
ISE	1	84.7	**94.0**	**35.0**	**64.7**
	3	84.2	93.3	33.3	63.1
	5	**84.8**	93.9	31.9	61.8
	10	84.2	93.6	30.5	60.2

Table 4. Influences of the K-nearest clusters in our ISE

Method	MS→M		M→MS	
	mAP	top-1	mAP	top-1
Baseline	25.5	53.8	1.5	5.9
ISE	**30.2**	**59.7**	**2.7**	**8.6**

Table 5. Results of the direct domain generalization. "M" and "MS" represents Market-1501 and MSMT17, respectively.

K-nearest direction, and then explore the influences of different base degree λ_0. For the direction, except for selecting K-nearest clusters, we also explore the K-random and K-farthest selection strategies. Here we set $K=1$. The upper block in Table 3 shows that the nearest direction obtains the best results. This is because that support samples distributed between two similar clusters may possibly compensate for some lost yet useful information. By contrast, other two strategies are likely to generate meaningless samples. Therefore, support samples on various directions can only bring limited improvement and even degradation. For the degree, we set $\lambda_0 = \{0.1, 0.5, 1.0, 2.0\}$ in the lower block in Table 3. With the increase of λ_0, the performance gradually improves and then begins to decline. Too large λ_0 results in farther support samples, which may act as outliers and have negative influences on the model training. On the contrary, too small λ_0 is prone to generating useless support samples, which has less positive impact on the performance.

Sample extension to K-nearest clusters. Table 4 shows the influences of different number of K in our method. We set $K = \{1, 3, 5, 10\}$ in this experiment. We can see that as K grows, the performance almost remains the same on Market-1501 but drops a lot on MSMT17. Meanwhile, too large K is also harmful to the performance on Market-1501. We conjecture it may be because the data points are distributed in a manifold, which requires linear interpolation to be within the local Euclidean space near each point. If K is large, the K-th centroid is far from the point-of-interest and even out of the local Euclidean space near the given point. The generated support sample is thus not in the original manifold and does not belong to any cluster. Consequentially, it involves noises and decreases the performance. Thus, we set $K = 1$ in this paper.

4.5. More Discussions

Our ISE *vs.* Mixup [42]. Although ISE and Mixup both generate interpolated samples, the differences are signifi-
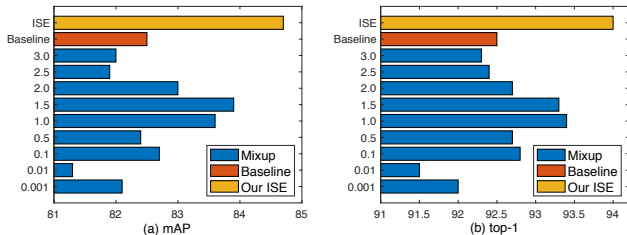

Figure 5. Our ISE *vs.* traditional Mixup [42] on Market-1501. The number of y-axis is the hyper-parameter α, representing the interpolation ratio randomly sampled from $\lambda \sim \text{Beta}(\alpha, \alpha)$.

cant from the following three aspects: 1) The interpolation of Mixup takes place on the raw images or hidden features, while ours is on the output embeddings. 2) The interpolation in Mixup can be between two arbitrary inputs, while ISE chooses the K-nearest centroids as the interpolating direction. 3) The interpolated samples in Mixup are with interpolated labels and do not belong to any existing classes. In contrast, support samples in ISE belong to a single identified class controlled by the generation degree. In Figure 5, we experimentally investigate Mixup and compare it with our ISE. With an appropriate hyper-parameter α, Mixup can improve the performance due to its data augmentation effect on input images. However, the improvement is limited without the controlment on the direction and degree of synthetics. Differently, our ISE can effectively generate informative support samples between two similar clusters in the embedding space to compensate for missing information.

ISE helps domain generalization (DG). We find that several clusters in baseline easily shrink to a small region in the embedding space as shown in Figure 3. Such compact data distribution increases the risk of overfitting. By contrast, our ISE generates support samples that patrol around cluster boundaries and improve clustering behaviours. We think this should help domain generalization as well. Thus we investigate how the baseline and our ISE perform in DG tasks. As shown in Table 5, our ISE significantly outperforms the baseline and obtains consistent superior results. Especially, ISE brings +4.7%/+5.9% mAP/top-1 improvements on the MS→M setting. This demonstrates that support samples indeed help refine the data distribution and improve the generalization ability of the Re-ID model.

4.6. Comparison with State-of-the-art Methods

We compare the proposed ISE with state-of-the-art person Re-ID methods in Table 6. For the USL setting, ISE is superior or competitive to the previous methods. In particular, our method significantly outperforms other previous methods with average pooling. ISE obtains 84.7% mAP and 94.0% top-1 on Market-1501 dataset and 35.0% mAP and 64.7% top-1 on MSMT17 dataset. Considering that Cluster-Contrast [9] utilizes the generalized mean pooling

Method	Reference	Market-1501				MSMT17			
		mAP	top-1	top-5	top-10	mAP	top-1	top-5	top-10
Purely Unsupervised									
BUC [20]	AAAI'19	29.6	61.9	73.5	78.2	-	-	-	-
SSL [21]	CVPR'20	37.8	71.7	83.8	87.4	-	-	-	-
JVTC [19]	ECCV'20	41.8	72.9	84.2	88.7	15.1	39.0	50.9	56.8
MMCL [30]	CVPR'20	45.5	80.3	89.4	92.3	11.2	35.4	44.8	49.8
HCT [41]	CVPR'20	56.4	80.0	91.6	95.2	-	-	-	-
CycAs [34]	ECCV'20	64.8	84.8	-	-	26.7	50.1	-	-
GCL [6]	CVPR'21	66.8	87.3	93.5	95.5	21.3	45.7	58.6	64.5
SpCL [14]	NeurIPS'20	73.1	88.1	95.1	97.0	19.1	42.3	55.6	61.2
IICS [37]	CVPR'21	72.9	89.5	95.2	97.0	26.9	56.4	68.8	73.4
JVTC+* [5]	CVPR'21	75.4	90.5	96.2	97.1	29.7	54.4	68.2	74.2
JNTL-MCSA [39]	CVPR'21	61.7	83.9	92.3	-	15.5	35.2	48.3	-
ICE [4]	ICCV'21	79.5	92.0	97.0	98.1	29.8	59.0	71.7	77.0
CAP† [31]	AAAI'21	79.2	91.4	96.3	97.7	36.9	67.4	78.0	81.4
ICE† [4]	ICCV'21	82.3	93.8	97.6	98.4	38.9	70.2	80.5	84.4
OPLG-HCD [46]	ICCV'21	78.1	91.1	96.4	97.7	26.9	53.7	65.3	70.2
MPRD [17]	ICCV'21	51.1	83.0	91.3	93.6	14.6	37.7	51.3	57.1
Cluster-Contrast [9]	Arxiv'21	<u>82.6</u>	<u>93.0</u>	<u>97.0</u>	<u>98.1</u>	<u>33.3</u>	<u>63.3</u>	<u>73.7</u>	<u>77.8</u>
Cluster-Contrast (Our Baseline)	-	82.5	92.5	96.9	97.9	30.1	58.6	69.6	74.4
Our ISE	-	**84.7**	**94.0**	**97.8**	**98.8**	**35.0**	**64.7**	**75.5**	**79.4**
Our ISE + GeM pooling	-	<u>85.3</u>	<u>94.3</u>	<u>98.0</u>	<u>98.8</u>	<u>37.0</u>	<u>67.6</u>	<u>77.5</u>	<u>81.0</u>
Fully Supervised									
PCB [26]	ECCV'18	81.6	93.8	97.5	98.5	40.4	68.2	-	-
DG-Net [47]	CVPR'19	86.0	94.8	-	-	52.3	77.2	-	-
ICE (w/ ground-truth) [4]	ICCV'21	86.6	95.1	98.3	98.9	50.4	76.4	86.6	90.0
Our ISE (w/ ground-truth)	-	87.8	95.6	98.5	99.2	51.0	76.8	87.1	90.6

Table 6. Comparison of ReID methods on Market-1501 and MSMT17 datasets. The best USL results without camera information are marked with **bold**. The pure **bold** number represents using the average pooling, while the <u>underline</u> number denotes using the GeM [24] pooling. † indicates using the additional camera knowledge.

(GeM) [24] in the final results[4], we also report ISE's results with GeM for a fair comparison. With GeM, our ISE still consistently achieves better results. Note that different from ICE [4] and CAP [31], we do not use the camera information. Under the non-camera setting, our ISE obtains 35.0%/64.7% mAP/top-1 on MSMT17, largely outperforming OPLG-HCD [46], ICE [4] (without cameras) and Cluster-Contrast [9]. In addition, we also transfer our ISE to the supervised setting by using ground-truth identities instead of pseudo labels. From the lower block in Table 6, we can see that our ISE is competitive with well-known supervised methods (*e.g.*, PCB and DG-Net) and surpasses ICE [4] with ground-truths. The results show the effectiveness of ISE on supervised Re-ID.

5. Conclusion and Limitations

Conclusion. In the clustering-based unsupervised person Re-ID methods, when one identity breaks into multiple clusters or different identities get mixed into one cluster, they will confuse the learning and result in unsatisfying performance. To alleviate this issue, we propose an implicit sample extension method to generate support sam-

ples between two neighbouring clusters, and expect to associate the same identity's samples by these support samples. Specifically, we develop a progressive linear interpolation strategy to control the generation of support sample with direction and degree factors. A label-preserving loss is also proposed to constrain support samples close to their own cluster members. Comprehensive experiments show that our method can improve clustering results significantly and achieve new state-of-the-art performance on two popular Re-ID datasets. The proposed method also demonstrates better domain generalization ability than the baseline.

Limitations. Our ISE generates support samples from actual samples to their neighboring cluster centroids in the embedding space. For each sample, the number of the selected nearest cluster centroids is fixed to K. It may not be the optimal solution. If some clusters have been well classified, they do not need to add supplementary samples. In contrast, when a specific cluster is mixed by multiple identities, it is better to generate support samples in multiple K-nearest directions. We will explore a dynamic K in the future work.

---

[4]It is noticing that we carefully check the code and the released training logs of Cluster-Contrast, and find they use GeM pooling in the final results. For a fair comparison, we also report the results of ISE + GeM pooling. Here, we report the results of the first version of Cluster-Contrast [9].

References

[1] Song Bai, Xiang Bai, and Qi Tian. Scalable person re-identification on supervised smoothed manifold. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 5

[2] Zechen Bai, Zhigang Wang, Jian Wang, Di Hu, and Errui Ding. Unsupervised multi-source domain adaptation for person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, page 12914–12923. Computer Vision Foundation / IEEE, 2021. 1

[3] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *Proc. Int. Conf. Mach. Learn.*, pages 552–560. PMLR, 2013. 2

[4] Hao Chen, Benoit Lagadec, and Francois Bremond. Ice: Inter-instance contrastive encoding for unsupervised person re-identification. *Proc. IEEE Int. Conf. Comp. Vis.*, 2021. 1, 2, 3, 8

[5] Hao Chen, Yaohui Wang, Benoit Lagadec, Antitza Dantcheva, and Francois Bremond. Joint generative and contrastive learning for unsupervised person re-identification. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021. 2, 8

[6] Hao Chen, Yaohui Wang, Benoit Lagadec, Antitza Dantcheva, and Francois Bremond. Joint generative and contrastive learning for unsupervised person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2004–2013, 2021. 8

[7] David Cournapeau and Google members. scikit-learn. https://scikit-learn.org/stable/index.html, 2007. 6

[8] Yongxing Dai, Jun Liu, Yifan Sun, Zekun Tong, Chi Zhang, and Ling-Yu Duan. Idm: An intermediate domain module for domain adaptive person re-id. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 11864–11874, 2021. 1, 3

[9] Zuozhuo Dai, Guangyuan Wang, Weihao Yuan, Siyu Zhu, and Ping Tan. Cluster contrast for unsupervised person re-identification. *arXiv preprint arXiv:2103.11568*, 2021. 1, 2, 3, 5, 7, 8

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255. Ieee, 2009. 5

[11] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 994–1003, 2018. 2

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 1, 3

[13] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S Huang. Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 6112–6121, 2019. 1, 3

[14] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, and hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In *Proc. Advances in Neural Inf. Process. Syst.*, volume 33, pages 11309–11321. Curran Associates, Inc., 2020. 1, 2, 3, 8

[15] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE Int. Workshop on Performance Evaluation for Tracking and Surveillance*, volume 3, pages 1–7. Citeseer, 2007. 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016. 5

[17] Haoxuanye Ji, Le Wang, Sanping Zhou, Wei Tang, Nanning Zheng, and Gang Hua. Meta pairwise relationship distillation for unsupervised person re-identification. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 3661–3670, 2021. 8

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Int. Conf. Learn. Represent.*, 2014. 5

[19] Jianing Li and Shiliang Zhang. Joint visual and temporal consistency for unsupervised domain adaptive person re-identification. In *Proc. Eur. Conf. Comp. Vis.*, pages 483–499. Springer, 2020. 8

[20] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *Proc. AAAI Conf. Artificial Intell.*, volume 33, pages 8738–8745, 2019. 2, 8

[21] Yutian Lin, Lingxi Xie, Yu Wu, Chenggang Yan, and Qi Tian. Unsupervised person re-identification via softened similarity learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3390–3399, 2020. 2, 8

[22] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. 1, 3

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[24] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1655–1668, 2018. 8

[25] Liangchen Song, Cheng Wang, Lefei Zhang, Bo Du, Qian Zhang, Chang Huang, and Xinggang Wang. Unsupervised domain adaptive re-identification: Theory and practice. *Pattern Recogn.*, 102:107173, 2020. 1

[26] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proc. Eur. Conf. Comp. Vis.*, 2018. 8

[27] Alex Tamkin, Mike Wu, and Noah D. Goodman. Viewmaker networks: Learning views for unsupervised representation learning. In *Proc. Int. Conf. Learn. Representations*, 2021. 2

[28] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7064–7073, 2017. 2

[29] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, 2014. 5

[30] Dongkai Wang and Shiliang Zhang. Unsupervised person re-identification via multi-label classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10981–10990, 2020. 2, 3, 8

[31] Menglin Wang, Baisheng Lai, Jianqiang Huang, Xiaojin Gong, and Xian-Sheng Hua. Camera-aware proxies for unsupervised person re-identification. *Proc. AAAI Conf. Artificial Intell.*, 2021. 3, 8

[32] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. Regularizing deep networks with semantic data augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 2

[33] Zhongdao Wang, Jingwei Zhang, Liang Zheng, Yixuan Liu, Yifan Sun, Yali Li, and Shengjin Wang. Cycas: Self-supervised cycle association for learning re-identifiable descriptions. In *Proc. Eur. Conf. Comp. Vis.*, volume 12356, pages 72–88, 2020. 2

[34] Zhongdao Wang, Jingwei Zhang, Liang Zheng, Yixuan Liu, Yifan Sun, Yali Li, and Shengjin Wang. Cycas: Self-supervised cycle association for learning re-identifiable descriptions. *Proc. Eur. Conf. Comp. Vis.*, 2020. 8

[35] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 79–88, 2018. 5

[36] Guile Wu, Xiatian Zhu, and Shaogang Gong. Tracklet self-supervised learning for unsupervised person re-identification. In *Proc. AAAI Conf. Artificial Intell.*, pages 12362–12369, 2020. 2

[37] Shiyu Xuan and Shiliang Zhang. Intra-inter camera similarity for unsupervised person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 11926–11935, 2021. 8

[38] Fengxiang Yang, Ke Li, Zhun Zhong, Zhiming Luo, Xing Sun, Hao Cheng, Xiaowei Guo, Feiyue Huang, Rongrong Ji, and Shaozi Li. Asymmetric co-teaching for unsupervised cross-domain person re-identification. In *Proc. AAAI Conf. Artificial Intell.*, pages 12597–12604, 2020. 1

[39] Fengxiang Yang, Zhun Zhong, Zhiming Luo, Yuanzheng Cai, Yaojin Lin, Shaozi Li, and Nicu Sebe. Joint noise-tolerant learning and meta camera shift adaptation for unsupervised person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4855–4864, 2021. 8

[40] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 6022–6031. 2

[41] Kaiwei Zeng, Munan Ning, Yaohua Wang, and Yang Guo. Hierarchical clustering with hard-batch triplet loss for person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 13657–13665, 2020. 8

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Int. Conf. Learn. Represent.*, 2017. 2, 7

[43] Xinyu Zhang, Jiewei Cao, Chunhua Shen, and Mingyu You. Self-training with progressive augmentation for unsupervised cross-domain person re-identification. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 8222–8231, 2019. 1, 3

[44] Xinyu Zhang, Dong Gong, Jiewei Cao, and Chunhua Shen. Memorizing comprehensively to learn adaptively: Unsupervised cross-domain person re-id with multi-level memory. *arXiv preprint arXiv:2001.04123*, 2020. 1

[45] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1116–1124, 2015. 5

[46] Yi Zheng, Shixiang Tang, Guolong Teng, Yixiao Ge, Kaijian Liu, Jing Qin, Donglian Qi, and Dapeng Chen. Online pseudo label generation by hierarchical cluster dynamics for adaptive person re-identification. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 8371–8381, 2021. 8

[47] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. 8

[48] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1318–1327, 2017. 5

[49] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 598–607, 2019. 1

[50] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Learning to adapt invariance in memory for person re-identification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 3

[51] Yang Zou, Xiaodong Yang, Zhiding Yu, BVK Kumar, and Jan Kautz. Joint disentangling and adaptation for cross-domain person re-identification. *arXiv preprint arXiv:2007.10315*, 2020. 2